

Lecture 25



Advanced Relational Algebra



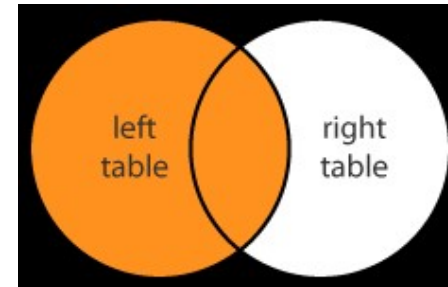
OUTER Join Operation



- In NATURAL JOIN tuples without a *matching* (or *related*) tuple are eliminated from the join result. Tuples with null in the join attributes are also eliminated. **This amounts to loss of information.**
- A set of operations, called Outer Joins, can be used when we want to keep all the tuples in R, or all those in S, or all those in both relations in the result of the join, irrespective of whether or not they have matching tuples in the other relation.



Left Outer Join (\bowtie)



- The **Left Outer Join** operation keeps every tuple in the **first or left** relation R in R \bowtie S.
- *If no matching tuple is found in S, then the attributes of S in the join result are filled or “padded” with null values.*
- **If R has ‘n’ tuples and S has ‘m’, then**

Cardinality: min = n

max = n * m



Left Outer Join Example

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

Left Outer Join Example

- A list of all employee names as well as the name of the departments they manage if they happen to manage a department; if they do not manage one, indicate a NULL value.

Query: $TEMP \leftarrow (EMPLOYEE \underset{Ssn=Mgr_ssn}{\Join} DEPARTMENT)$

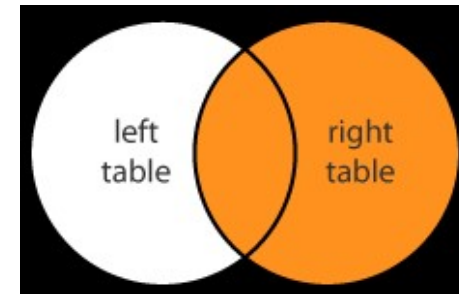
$RESULT \leftarrow \pi_{Fname, Minit, Lname, Dname}^{(TEMP)}$

Output:

RESULT

Fname	Minit	Lname	Dname
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

Right Outer Join ()



- **Right Outer Join**, keeps every tuple in the *second* or right relation S in the result of R S.



- *If no matching tuple is found in R, then the attributes of R in the join result are filled or “padded” with null values.*
- **If R has ‘n’ tuples and S has ‘m’, then**

Cardinality: min = m

max = n * m



RIGHT Outer Join Example

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Query:

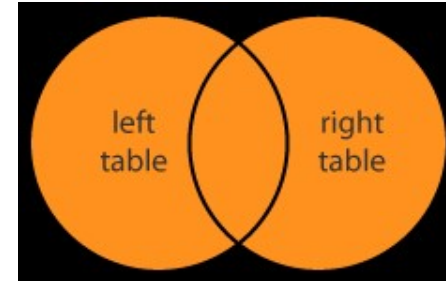
$TEMP \leftarrow (DEPARTMENT \bowtie DEPT_LOCATIONS)$

$RESULT \leftarrow \pi_{Dname, Dnumber, Dlocation} (TEMP)$

Output:

Dname	Dnumber	Dlocation
Headquarters	1	Houston
Administration	4	Stafford
Research	5	Bellaire
Research	5	Sugarland
Research	5	Houston

Full Outer Join (\bowtie)



- Full Outer Join, denoted by \bowtie keeps all tuples in both the left and the right relations when no matching tuples are found, padding them with null values as needed.

$$\bowtie R \quad S = (R \bowtie S) \cup (R \text{ } \text{ } S)$$

- If R has 'n' tuples and S has 'm', then

Cardinality: min = max (n, m)

$$\text{max} = n * m$$

FULL Outer Join Example

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

FULL Outer Join Example

- Query:

$$\text{TEMP} \leftarrow (\text{EMPLOYEE} \bowtie_{\text{Ssn}=\text{Mgr_ssn}} \text{DEPARTMENT})$$

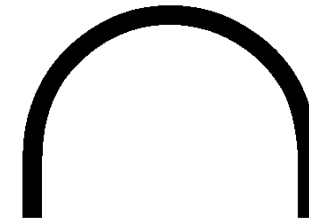
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Minit, Lname, Dname}}(\text{TEMP})$$

- Output:

RESULT

Fname	Minit	Lname	Dname
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

Same as output of Left Outer Join as all tuples of DEPARTMENT had corresponding matching tuples in EMPLOYEE



Intersection Operation

- **INTERSECTION OPERATION**

The result of this operation, denoted by $R \cap S$, is a relation that includes all tuples that are in both R and S. The two operands must be "type compatible"

Example:

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

PROJECT

<u>Pname</u>	<u>Pnumber</u>	<u>Plocation</u>	<u>Dnum</u>
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Intersection Operation Example

- Query:

$$\text{TEMP} \leftarrow \pi_{\text{Dnum}, \text{Plocation}} (\text{PROJECT})$$

$$\text{RESULT} \leftarrow \text{DEPT_LOCATIONS} \cap \text{TEMP}$$

Output:

TEMP

Dnum	Plocation
5	Bellaire
5	Sugarland
5	Houston
4	Stafford
1	Houston
4	Stafford

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Properties of Relational Algebra Operations

- Both union and intersection are *commutative operations*; that is

$$\mathbf{R \cup S = S \cup R, \text{ and } R \cap S = S \cap R}$$

- Both union and intersection can be treated as n-ary operations applicable to any number of relations as both are *associative operations*; that is

$$\mathbf{R \cup (S \cup T) = (R \cup S) \cup T, \text{ and } (R \cap S) \cap T = R \cap (S \cap T)}$$

- The minus operation is *not commutative*; that is, in general

$$\mathbf{R - S \neq S - R}$$

Division Operation

- The division operation is applied to two relations $R(Z) \div S(X)$, where $X \subseteq Z$.

Let $Y = Z - X$ (and hence $Z = X \cup Y$); that is, let Y be the set of attributes of R that are not attributes of S .

- The result of DIVISION is a relation $T(Y)$ that includes a tuple t if tuples t_R appear in R with $t_R[Y] = t$, and with

$t_R[X] = t_s$ for every tuple t_s in S

- For a tuple t to appear in the result T of the DIVISION, the values in t must appear in R in combination with every tuple in S .

Division Operation Example

PROJECT

Pname	Pnumber	Plocation	Dnumber
ProductX	1	Bellaire	5
ProductY	2	Stafford	5
ProductZ	3	Houston	5
Newbenefits	10	Stafford	4
ProductZ	3	Houston	1
ProductZ	3	Sugarland	4

DEPARTMENT

Dnumber
1
4
5

Retrieve the Project Number that are being executed in *all* the departments.

Query:

$$(\pi_{\text{Pnumber, Dnumber}}(\text{PROJECT})) \div \pi_{\text{Dnumber}}(\text{DEPARTMENT})$$

Output:

Pnumber
3

Division Operation as a Derived Operation

Sequence of π , \times , $-$ operations = \div operation

- Step 1: Project numbers *not executed in all* departments

$$\pi_{Pnumber} ((\pi_{Pname, Pnumber, Plocation} (PROJECT) \times \pi_{Dnumber} (DEPARTMENT)) - PROJECT)$$

Pname	Pnumber	Plocation	Dnumber
ProductX	1	Bellaire	5
ProductX	1	Bellaire	1
ProductX	1	Bellaire	4
ProductY	2	Stafford	5
ProductY	2	Stafford	1
ProductY	2	Stafford	4
ProductZ	3	Houston	5
ProductZ	3	Houston	1
ProductZ	3	Sugarland	4
Newbenefits	10	Stafford	5
Newbenefits	10	Stafford	1
Newbenefits	10	Stafford	4

Pname	Pnumber	Plocation	Dnumber
ProductX	1	Bellaire	5
ProductY	2	Stafford	5
ProductZ	3	Houston	5
Newbenefits	10	Stafford	4
ProductZ	3	Houston	1
ProductZ	3	Sugarland	4

- **Output of Step 1:**

Pnumber
1
2
10

- **Step 2: Project Number that are being executed in *all* the departments i.e. Project numbers that are being executed in some departments – Project numbers not executed in all departments**

$$\pi_{Pnumber}(PROJECT) - \text{Step 1}$$

i.e.

$$\pi_{Pnumber}(PROJECT) - \pi_{Pnumber}((\pi_{Pname, Pnumber, Plocation}(PROJECT) \times \pi_{Dnumber}(DEPARTMENT)) - PROJECT)$$

- **Output of Step 2:**

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

Division Operation Example

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

Division Operation Example

- Retrieve the names of employees who work on **all** the projects that 'John Smith' works on.

Query:

$$\text{JS_PROJECTS} \leftarrow \pi_{\text{Pno}} ((\sigma_{\text{Fname} = \text{'John'} \text{ AND } \text{Lname} = \text{'Smith'}} (\text{EMPLOYEE})) \bowtie_{\text{Ssn} = \text{Essn}} \text{WORKS_ON}))$$

$$\text{TEMP} \leftarrow (\pi_{\text{Essn}, \text{Pno}} (\text{WORKS_ON})) \div \text{JS_PROJECTS}$$

$$\text{RESULT} \leftarrow \pi_{\text{Fname}, \text{Lname}} (\text{TEMP} \bowtie_{\text{Essn} = \text{Ssn}} \text{EMPLOYEE})$$

Output: JS_PROJECTS

Pno
1
2






TEMP

Essn
123456789
453453453

RESULT

Fname	Lname
John	Smith
Joyce	English

Relational Algebra Operators

Operation	Purpose	Notation
LEFT OUTER JOIN	Produces every tuple in the first or left relation R1 in R1 R2. 	R1  R2
RIGHT OUTER JOIN	Produces every tuple in the second or right relation R1 in R1 R2. 	R1  R2
FULL OUTER JOIN	Produces all tuples in both the left and right relations when no matching tuples are found, padding them with null values as needed.	R1 
INTERSECTION	Produces a relation that includes all the tuples in both R1 and R2; R1 and R2 must be union compatible.	$R1 \cap R2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R1(Z)$ that appear in R1 in combination with every tuple from $R2(Y)$, where $Z = X \cup Y$.	$R1(Z) \div R2(Y)$

Thanks!!