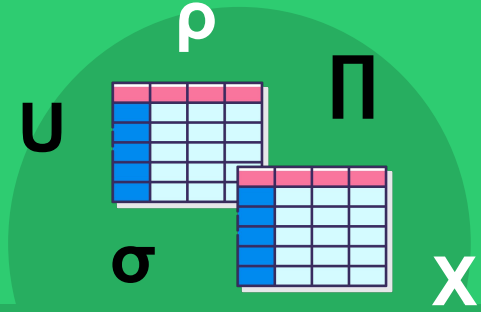
UNIVERSITAS INDONESIA

FAKULTAS ILMU KOMPUTER

13

Relational Algebra
(part 2)

BASIS DATA



Acknowledgements

This slide is a modification to supplementary slide of "Database System", 6th edition, Elmasri/Navathe, 2011: **Chapter 8 The Relational Algebra and Relational Calculus** used in "Basis Data" course in academic years 2018/2019 in the Faculty of Computer Science, Universitas Indonesia and Stanford Database (CS145) Lecture 16 slides

Outline

1. Introduction to Relational Algebra (RA)
2. Unary Operations
3. Binary Operations
4. Query Tree
5. Additional RA Operations

Query Tree

Example:
For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

$$\pi_{\text{Pnumber, Dnum, Lname, Address, Bdate}}(((\sigma_{\text{Plocation='Stafford'}}(\text{PROJECT})) \bowtie_{\text{Dnum=Dnumber}}(\text{DEPARTMENT})) \bowtie_{\text{Mgr\_ssn=Ssn}}(\text{EMPLOYEE}))$$

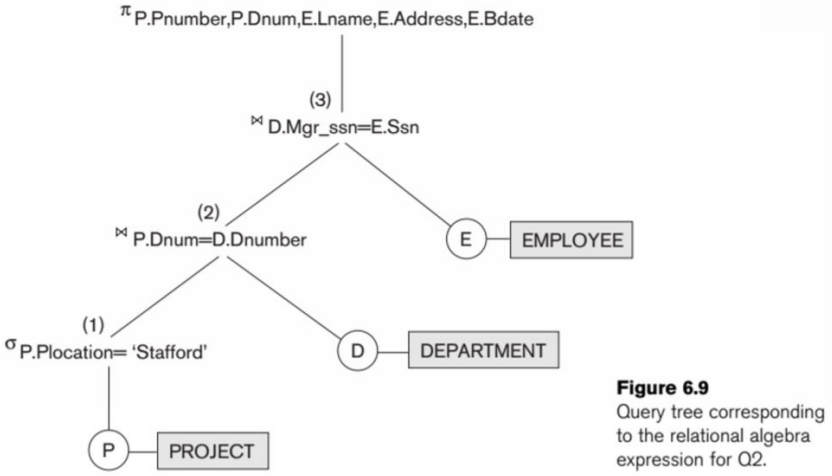
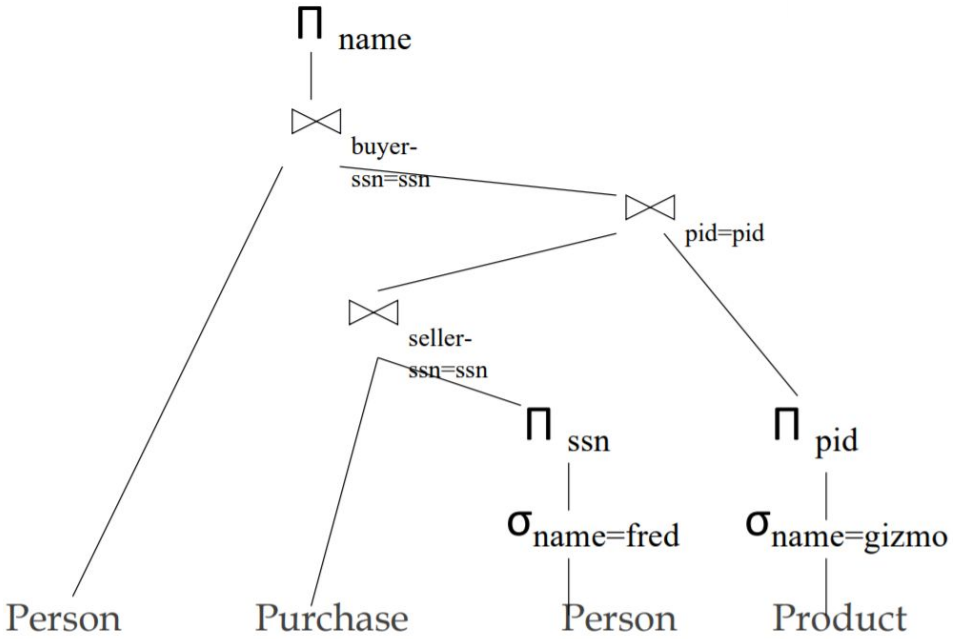


Figure 6.9
Query tree corresponding to the relational algebra expression for Q2.

Query Tree (Cntd.)



Outline

Additional Relational Algebra Operations

Aggregate & Grouping

Outer Join

Recursive Closure Relation

Outer Union

Aggregate & Grouping

Notation: $_{\text{<grouping attributes>}}\mathcal{F}_{\text{<function list>}}(R)$

Example:
Retrieve each department number, the number of employee in the department, and their average salary

RA:

$$\rho_{R(Dno,\ No\_of\_employees,\ Average\_sal)}\ (_{Dno}\mathcal{F}_{COUNT\ Ssn,\ AVERAGE\ Salary}(EMPLOYEE))$$

Renaming relation and attributes

Group by Dno

SQL:

```
SELECT
Dno,
COUNT(SSN) AS No_of_employees,
AVG(Salary) AS Average_Sal
FROM EMPLOYEE
GROUP BY Dno;
```

R

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

Aggregate & Grouping (Cntd.)

If no grouping are specified, then the functions are applied to all tuples in the relation

Example:
Retrieve the number of employees and their average salary

$$\rho_{R(\text{ No\_of\_employees, Average\_sal})}\,(\Im_{\text{ COUNT Ssn, AVERAGE Salary}}\,(\text{EMPLOYEE}))$$

Count_ssn	Average_salary
8	35125

Recursive Closure

Applied to a **recursive relationship** between tuples of the same type.

Example:

Retrieve all supervisees (at all levels) of an employee whose name is ‘James Borg’.

We utilize a looping mechanism

- (Level 1) Retrieve all direct supervisees of an employee whose name is ‘James Borg’.
- (Level 2) Retrieve all supervisees of some employee who is directly supervised by ‘James Borg’.
- ….
- (Level n) …

Recursive Closure (Cntd.)

Level 1

Retrieve all direct supervisees of an employee whose name is ‘James Borg’.

$$BORG\_SSN \leftarrow \pi_{Ssn}(\sigma_{Fname='James' \text{ AND } Lname='Borg'}(EMPLOYEE))$$

$$SUPERVISION(Ssn1, Ssn2) \leftarrow \pi_{Ssn,Super\_ssn}(EMPLOYEE)$$

$$RESULT1(Ssn) \leftarrow \pi_{Ssn1}(SUPERVISION \bowtie_{Ssn2=Ssn} BORG\_SSN)$$

SUPERVISION

(Borg’s Ssn is 888665555)

(Ssn)	(Super_ssn)
Ssn1	Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	null

RESULT1

Ssn
333445555
987654321

(Supervised by Borg)

Recursive Closure (Cntd.)

Level 2

Retrieve all supervisees of some employee who is directly supervised by 'James Borg'.

$$RESULT2(Ssn) \leftarrow \pi_{Ssn1}(SUPERVISION \bowtie_{Ssn2=Ssn} RESULT1)$$

To get both set of employees supervised at level 1 and 2 by James Borg:

$$RESULT \leftarrow RESULT2 \cup RESULT1$$

SUPERVISION

(Borg's Ssn is 888665555)

(Ssn) (Super_ssn)

Ssn1	Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	null

RESULT1

Ssn
333445555
987654321

(Supervised by Borg)

RESULT2

Ssn
123456789
999887777
666884444
453453453
987987987

(Supervised by Borg's subordinates)

RESULT

Ssn
123456789
999887777
666884444
453453453
987987987
333445555
987654321

(RESULT1 ∪ RESULT2)

Figure 8.11
A two-level recursive query.

Outer JOIN

Notation: **R1** ⟕ **R2** (left)

or

Notation: **R1** ⟖ **R2** (right)

or

Notation: **R1** ⟗ **R2** (full)

Example:
Left outer join

$$TEMP \leftarrow (EMPLOYEE \overset{}{\bowtie}_{Ssn=Mgr\_ssn} DEPARTMENT)$$
$$RESULT \leftarrow \pi_{Fname,\ Minit,\ Lname,\ Dname}(TEMP)$$

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

Outer UNION

Take the union of tuples from two relations if the relations are **not union compatible.**

Takes **UNION** of tuples in two relations R(X,Y) and S(X,Z) that are partially compatible, meaning that only some of their attributes, say X, are union compatible.

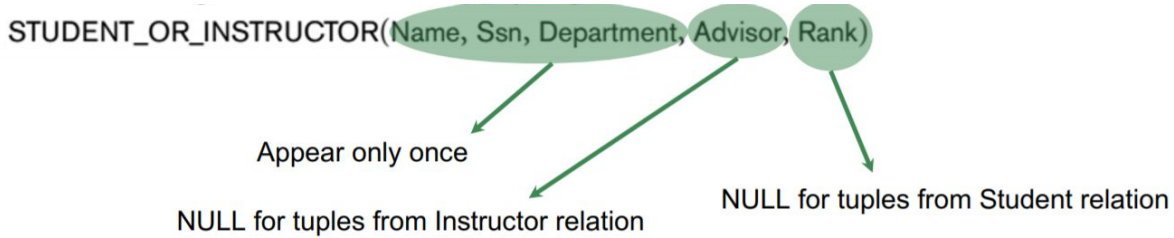
The result relation T(X,Y,Z), the attributes that **are union compatible are represented only once** in the result, and those which are not union compatible from either relation are also kept in the result.

Outer UNION (Cntd.)

Example:

- ➔ Student(Name, Ssn, Department, Advisor)
- ➔ Instructor(Name, Ssn, Department, Rank)

The resulting relation from applying Outer Union:



Exercise

1. For each department, retrieve the department name and the average salary of all employees working in that department.

2. Retrieve the average salary of all female employees.

3. Show the result of following operation:

a. $T1 \bowtie_{T1.P = T2.A} T2$

b. $T1 \bowtie_{T1.Q = T2.B} T2$

c. $T1 \bowtie_{T1.P = T2.A} T2$

d. $T1 \bowtie_{T1.Q = T2.B} T2$

e. $T1 \cup T2$

f. $T1 \bowtie_{(T1.P = T2.A \text{ AND } T1.R = T2.C)} T2$

TABLE T1

P	Q	R
10	a	5
15	b	8
25	a	6

TABLE T2

A	B	C
10	b	6
25	c	3
10	b	5

Exercise

4. For each project, list the project name and the total hours per week (by all employees) spent on that project.

5. List the name of the employees and their total working hours of all the same projects also done by the James Borg.

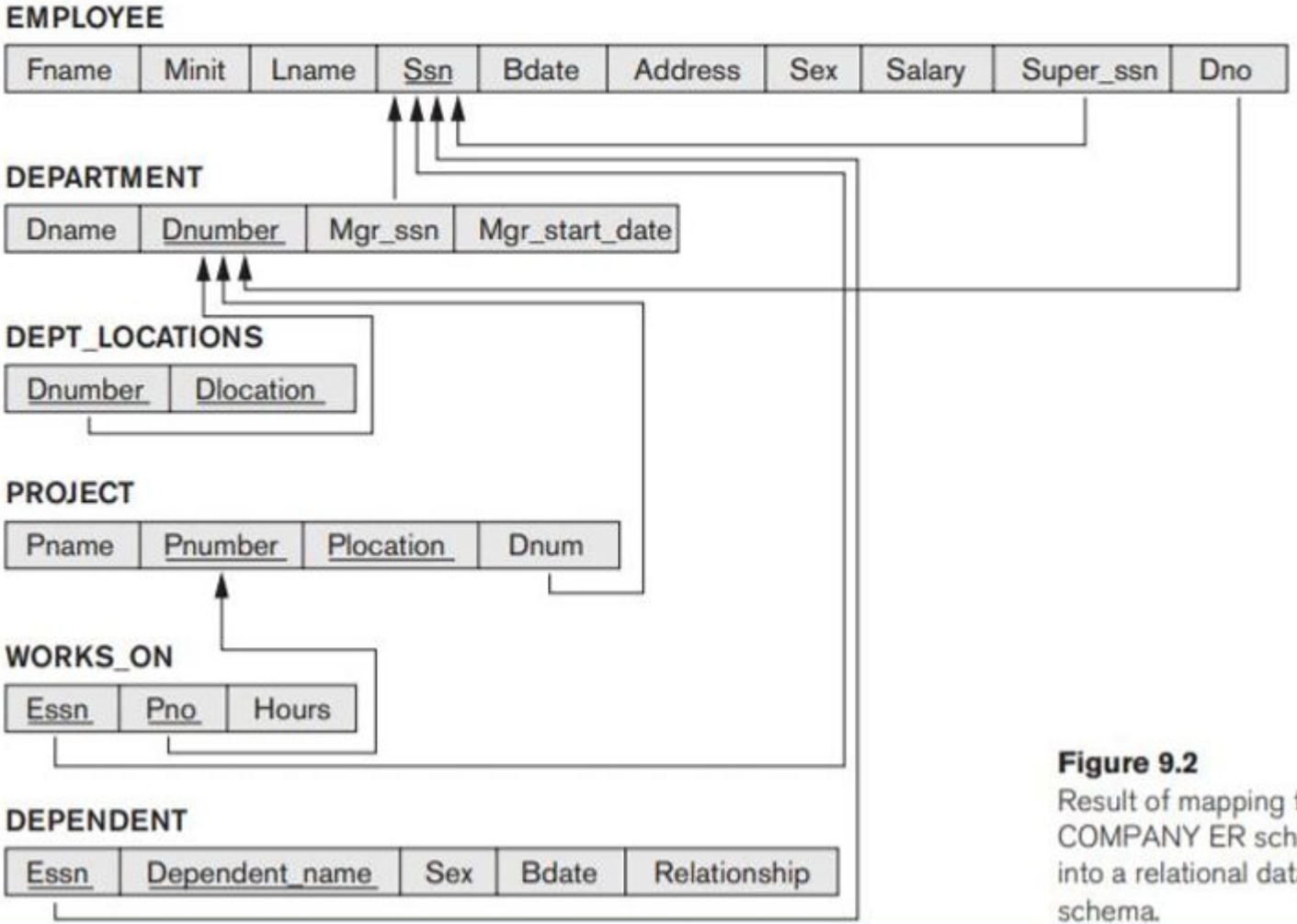


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Q&A