

Outer Join

Emp

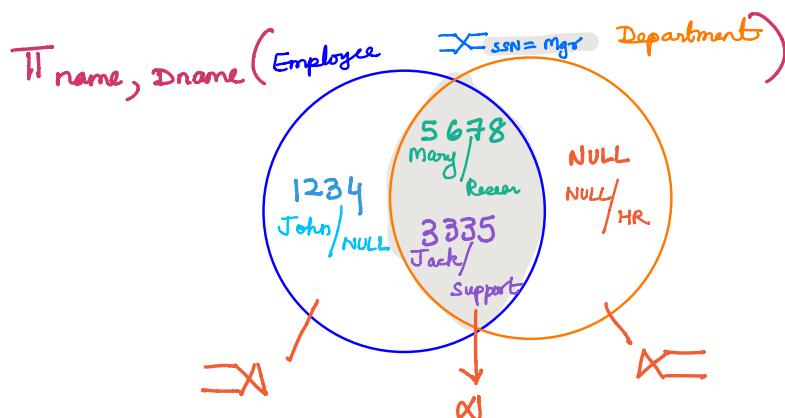
EName	SSN
John	1234
Mary	5678
Jack	3335

Department

Dname	MgrSSN
Research	5678
Support	3335
HR	NULL

E  $\times$  D

Name	SSN	MgrSSN	Dname
John	1234	5678	Research
John	1234	3335	Support
John	1234	NULL	HR
Mary	5678	5678	Research
Mary	5678	3335	Support
Mary	5678	NULL	HR
Jack	3335	5678	Research
Jack	3335	3335	Support
Jack	3335	NULL	HR



Query: List emp name & dept managed, NULL otherwise.

$E \bowtie D$   
 $ssn = mgr$

Fname	Dname
Mary	Research
Jack	Support
John	NULL

Query: List dept name along with manager, NULL otherwise.

$E \bowtie D$   
 $ssn = mgr$

Dname	MgrName
Research	Mary
Support	Jack
HR	NULL

Ename	SSN	Dname
Mary	5678	Research
John	1234	NULL
Jack	3335	Support
NULL	NULL	HR

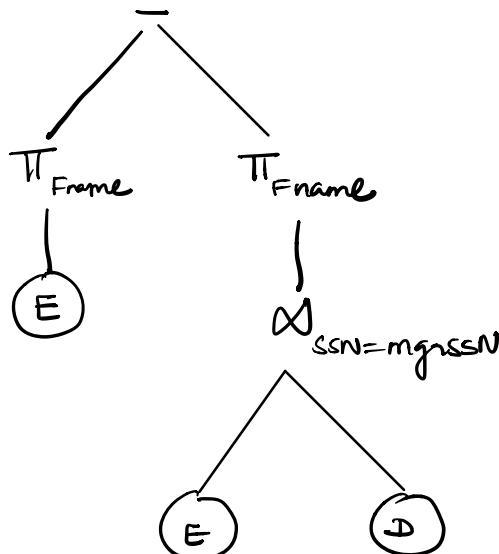
$\Pi_{Ename, SSN, Dname} (Emp \setminus \underset{SSN=mg\&SSN}{Dept})$

Left  $L \setminus R$   $\equiv$  keeps all tuples from L

$L \Delta R$   $\equiv$  keeps all tuples from R

$L \times R$   $\equiv$  keeps all tuples from L & R.

Query: List names of employees who are not managers.



$\Pi_{Frame}(E) - \Pi_{Frame}(E \Delta \underset{SSN=mg\&SSN}{D})$

$\Pi_{Frame} \setminus \underset{dname=NULL}{(E \Delta D)}$

dname IS NULL

Query: List Fname of emp who do not work on any project.

$$\Pi_{\text{Fname}} \sigma_{\text{Pro IS NULL}} (E \bowtie W)_{\text{ssn} = \text{essn}}$$

Query: List Fname of emp who are not supervisors.

$$\Pi_{\text{E-Fname}} \sigma_{\text{S.SSN IS NULL}} (E \bowtie L_s(E))_{\text{E.SSN} = \text{S.supervisor}}$$

Chapter 6      6.1, 6.3    New  
                  4.1, 4.3    Old

SQL : Structured Query Language

\* declarative language : what you want, not how to  
do it.  
↓  
Query Optimizers

\* based on RA

\* interactive : command line, embed in programs, web based

\* SQL

- DDL : CREATE TABLE, DROP TABLE
- DML : select, insert, delete, update
- other commands : indexes, constraints

\* SQL  $\equiv$  multiset : 2 or more identical rows

DML : SELECT

output attributes :  $\Pi$

SELECT  $A_1, A_2, \dots, A_n$   
 FROM  $R_1, R_2, \dots, R_m$       input tables:  
 WHERE conditions;       $\sigma, \alpha$

$\Pi_{A_1, A_2, \dots, A_n} (\sigma_{\alpha \text{ conditions}} (R_1 \times R_2 \times \dots \times R_m))$

Query: List birthday of emp named 'John'.

$\Pi_{Bdate} \sigma_{Fname='John'} (Emp);$

Select	Bdate	$\Pi$
From	Emp	$R_1$
Where	$Fname = 'John'$	$\sigma$

Query: List names of emp who work in 'Research'

$\Pi_{Fname} \sigma_{dname='Research'} (E \times D)$   
 $\sigma_{dnumber=Dno}$

Select	Fname	$\Pi$
From	Emp, Dept	$X$

where Dname = 'Research' AND  
Dnumbers = Dno;  
OR

Query: For every project in 'Stafford', list  
Pnumber, Dnum, Mgr-name

Select Pnumber, Dnum, Fname  
From Emp, Project, Department  
Where Plocation = 'Stafford' and  
Dept  $\Delta$  Project  $\rightarrow$  Dnum = Dnumber and  
 $\uparrow$  mgr-ssn = ssn;  
Emp  $\Delta$  Dept

Query: List names of dependents along with dependents' Bdate & Emp's Bdate.

Select Dependent-name, Dependent-Bdate, emp-Bdate  
From Emp, Dependent  
Where SSN = ESSN  
 $\Delta$

Ambiguous attribute name: put table name first

Query: List name of emp and name of supervisor  
Select E-Fname, S-Fname  
From Emp AS E, Emp AS S

where  $E \cdot \text{Superssn} = S \cdot \text{ssn}$

From Emp E, Emp S

Query: List SSN of all employees.  $\Pi_{\text{ssn}}(\text{Emp})$

Select SSN

From Emp; unspecified Where

Select SSN, Dname

From Emp, Department;  $\text{Emp} \times \text{Department}$

Select \* ← all attributes

from Emp, Department

Select DISTINCT Salary  
from Emp;

removes  
duplicate tuples

basalt

Query: List all employees in dep 5 who earn  
 $\geq 30000$  and  $\leq 50000$ .

Select \*

from Emp

where (Salary BETWEEN 30000 and 50000)  
and dno = 5;

