Lois Delcambre
Neena Maldikar
Suki Kangas

cs.pdx.edu/~lmd/cs386

Relational DBs

1970 Edgar Codd

mathematical defn
of a relation
and a query language

relational model

attribute names

schema for the relation

Student(id, name, age, major)

one row
one tuple

(101, 'David', 21, 'CS')
(102, 'Mary', 20, 'EE')
(103, 'David', 30, 'CS')

set

domains for the attributes

id - integer
name - character varying (50)
age - 3digit integer
major - character (15)

{101,
102,
103, ... }

id-domain × name-domain × age-domain × major-domain

current instance of a relation - the Student relation - is a subset of the cross product of the domains.

relational algebra

select $\sigma_{age<20}$ Student

query

note: relations are sets

Student(id, name, age, major)

project $\pi_{major, name}$ Student

query

('CS', 'David')

('EE', 'Mary')

$\sigma_{age \leq 30}\left(\pi_{major, age}(Student)\right)$

$\sigma_{age < 40}\left(\pi_{major, age}(Student)\right)$

major	age
'CS'	21
'EE'	20

major	age
'CS'	21
'EE'	20
'CS'	30

cross product $\times$

Student $\times$ Student

$(\pi_{id}\ \text{Student}) \times (\pi_{id}\ \text{student})$

$(\pi_{id}\ \text{Student})$: (101), (102), (103)

$(\pi_{id}\ \text{student})$: (101), (102), (103)

answer

101	101
101	102
101	103
102	101
102	102
102	103
103	101
103	102
103	103

$\cap$ $\cup$ $-$

union intersection set difference

For the query to be well-formed,
the two relations must
have the same number
of attributes/columns and corresponding
attributes must be defined
on the same domain.
Union-compatible.

employee(emp_id, name, gender, job)

position(id, title)

7. a. $\pi_{emp\_id, name, gender, job}(employee)$

$\sigma_{emp\_id = emp\_id}(employee)$

---

b. $\sigma_{job = 7}(Employee)$

---

d. $\pi_{name, gender, job}(employee)$

---

e. $(\pi_{emp\_id}(employee)) \times (\pi_{emp\_id}(employee))$

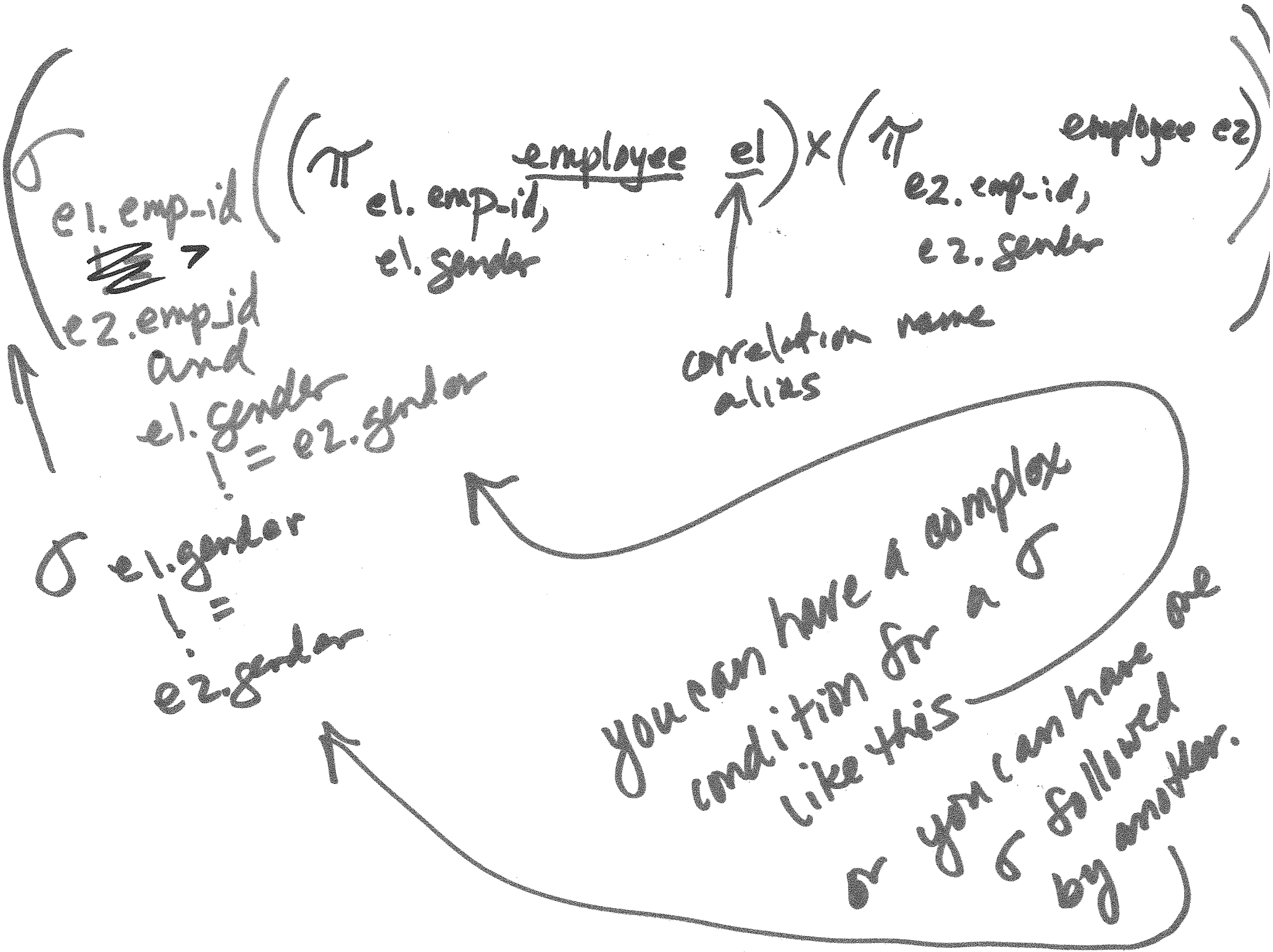
Rename operator

(e)

$$\pi_{EMP1.ID,\ EMP2.ID}\Big(\big(\rho_{EMP1}^{employee\ EMP}\big) \times \underbrace{\big(\rho_{EMP2}^{employee\ EMP}\big)}_{\text{rename}}\Big)$$

$$\longrightarrow$$

$$\pi_{ID1,ID2}\Big(\rho_{EMP(ID1)}(\pi_{ID}\ Employee) \times \rho_{EMP(ID2)}(\pi_{ID}\ Employee)\Big)$$


$\sigma_{e1.emp\_id = e2.emp\_id \text{ and } e1.gender != e2.gender}((\pi_{e1.emp\_id, e1.gender}(employee\ e1)) \times (\pi_{e2.emp\_id, e2.gender}(employee\ e2)))$
correlation name aliases
$\sigma_{e1.gender != e2.gender}$
you can have a complex condition for a σ like this
or you can have one σ followed by another.

SQL - standard language
- queries and many more kinds of statements

tables - default - bags
(they can have duplicates)

dbclass.cs.pdx.edu

a site where you each have a DB

employee(emp_id, name, gender, job)
position(id, title)

$\pi_{=}(\sigma_{\text{job=id}}(\text{employee} \times \text{position}))$

SELECT * ...
FROM employee, position
WHERE job=id

```
CREATE TABLE widget
    ( sku integer primary key,
      name character(15),
      price integer)
INSERT INTO widget
    VALUES (104, 'small motor', 20),
           (105, 'med. motor', 30)
ALTER TABLE widget RENAME TO newname
ALTER TABLE widget ADD COLUMN weight integer
```

Query 1:

INTERSECT

Query 2:

(SELECT ...
FROM ...
WHERE ...)

UNION

(SELECT ...
FROM ...
WHERE ...)

Query 3:

EXCEPT