Problem Session 1

Products and Joins

Outline

- Announcements
- Synopsis of Last Week
- Products and Joins – formal expressions, examples
- Products and Joins – sample problems
- Q&A

Announcements

- Make sure you're on the CS145 Coursework site by today
- Gradiance - one due Mon 10/8, two due Wed 10/10
- Challenge Problems #1 due Mon 10/8
  - Correction: #2 – can use union, intersection, difference as well
  - Submit directly through Coursework (text in Assignments preferred, though .doc or .pdf in Drop Box acceptable)
- Always email your questions to the staff list!

Synopsis of Last Week

- Data model: general conceptual way of structuring data
  - Relational model = tables (e.g., SQL)
  - Semistructured model = trees/graphs (e.g., XML)
- Schema: structure of a particular relation or database under a certain data model
- XML: language for semistructured model. DTD describes the structure.
- Relational Algebra: algebra operating on relations. Prelude to SQL.
- SQL: select-from-where

Products and Joins

- Product: R1 X R2

  = {(t1, t2) : t1 in R1 and t2 in R2}
- Theta Join: $R1 \bowtie_{C} R2 = \sigma_{C}(R1 \text{ X } R2)$
- Natural Join: $R1 \bowtie R2$

  $= \pi_{schema(R1)\ SETUNION\ schema(R2)}$

  $(R1 \bowtie_{R1.A=R2.A \text{ and } R1.B=R2.B \text{ and} \ldots} R2)$

Example: Product

R1(

A,	B)
1	2
3	4

R2(

B,	C)
5	6
4	8
2	10

R3 := R1 X R2

R3(

A,	R1.B,	R2.B,	C)
1	2	5	6
1	2	4	8
1	2	2	10
3	4	5	6
3	4	4	8
3	4	2	10

Example: Theta Join

R1(

A,	B
1	2
3	4

)

R2(

B,	C
5	6
4	8
2	10

)

$$R3 := R1 \bowtie_{R1.B=R2.B} R2$$

R3(

A,	R1.B,	R2.B,	C
~~1~~	~~2~~	~~5~~	~~6~~
~~1~~	~~2~~	~~4~~	~~8~~
1	2	2	10
~~3~~	~~4~~	~~5~~	~~6~~
3	4	4	8
~~3~~	~~4~~	~~2~~	~~10~~

)

Example: Natural Join

R1(

A,	B)
1	2
3	4

R2(

B,	C)
5	6
4	8
2	10

$R3 := R1 \bowtie R2$

R3(

A,	~~R1~~.B,	~~R2.B,~~	C)
~~1~~	~~2~~	~~5~~	~~6~~
~~1~~	~~2~~	~~4~~	~~8~~
1	2	~~2~~	10
~~3~~	~~4~~	~~5~~	~~6~~
3	4	~~4~~	8
~~3~~	~~4~~	~~2~~	~~10~~

Sample Problem #1

Drinkers(name, addr, phone)

Likes(drinker, beer)

Find names and addresses of all drinkers who like Bud.

Sample Problem #1

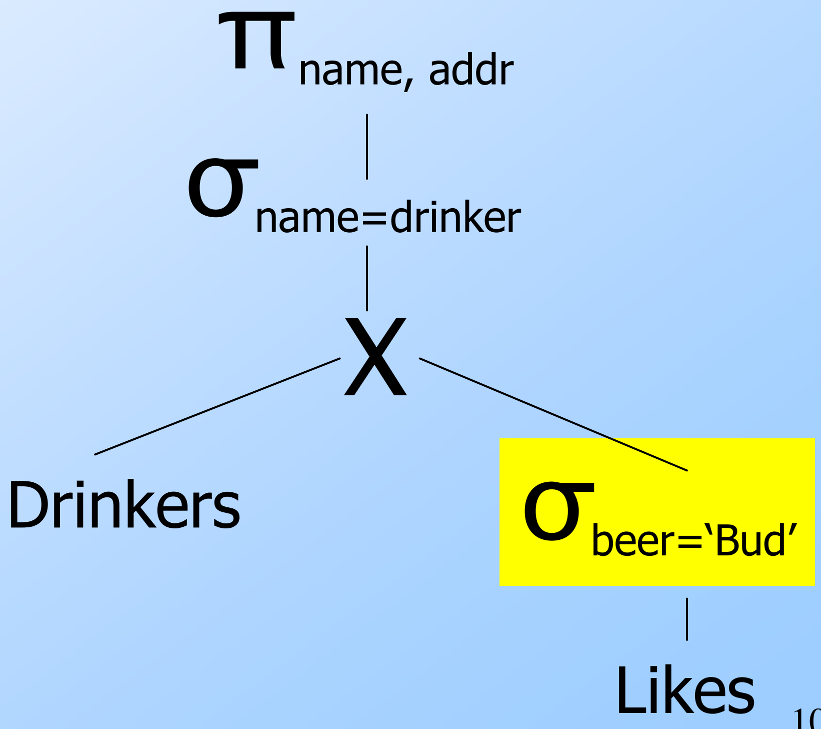
Drinkers(name, addr, phone)

Likes(drinker, beer)

Find names and addresses of all drinkers who like Bud.

- Method 1: filter, then concatenate
- Question: how would you write this in SQL?
- One Possible Answer:

```
SELECT name, addr
FROM Drinkers, (
        SELECT *
        FROM Likes
        WHERE beer = ‘Bud’)
WHERE name = drinker;
```



$\pi_{\text{name, addr}}$

$\sigma_{\text{name=drinker}}$

X

Drinkers

$\sigma_{\text{beer=‘Bud’}}$

Likes

Sample Problem #1

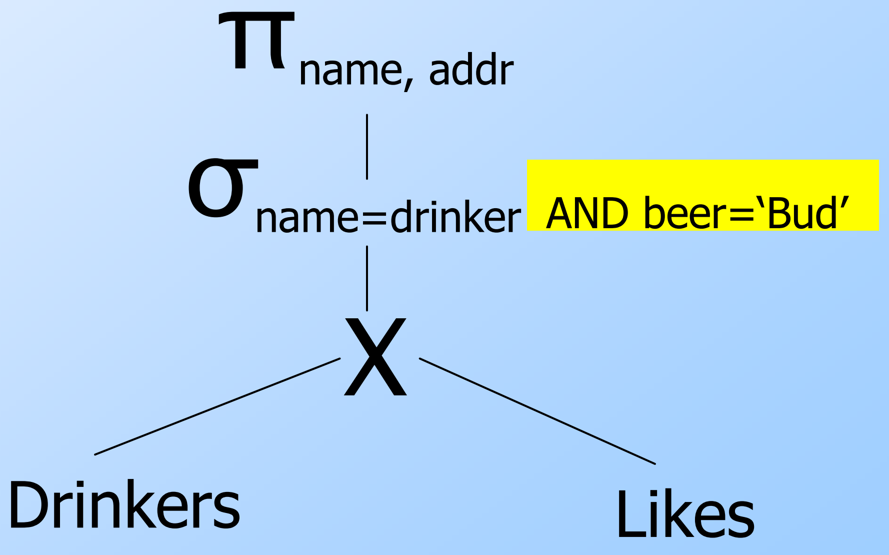
Drinkers(name, addr, phone)

Likes(drinker, beer)

Find names and addresses of all drinkers who like Bud.

- Method 2: concatenate, then filter
- Question: how would you write this in SQL?
- One Possible Answer:

```
SELECT name, addr
FROM Drinkers, Likes
WHERE name = drinker
      AND beer = ‘Bud’;
```


$\pi_{\text{name, addr}}$
$\sigma_{\text{name=drinker AND beer=`Bud'}}$
X
Drinkers
Likes

Sample Problem #2

Drinkers(name, addr, phone)

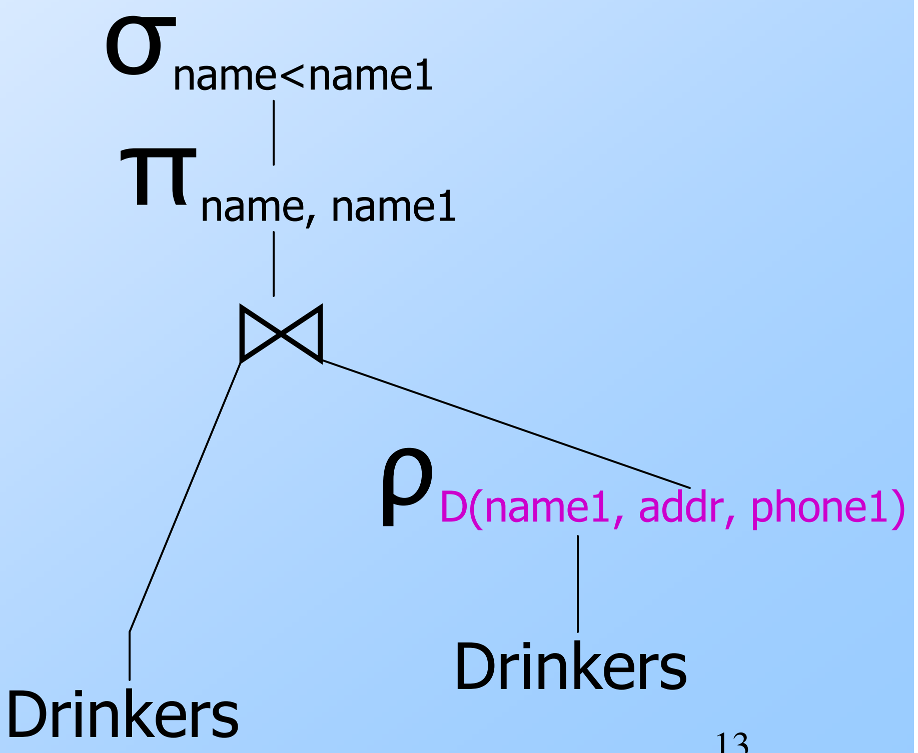
Find names of all pairs of drinkers who live at the same address.

Sample Problem #2

Drinkers(name, addr, phone)

Find names of all pairs of drinkers who live at the same address.

- Comparing drinkers with other drinkers, so reuse and rename
- Natural join contains tuples (name, name1, addr, phone, phone1) such that both drinkers live at this address
- Select condition ensures no duplicates
- Exercise: SQL translation?



$\sigma_{name<name1}$

$\pi_{name,\ name1}$

$\bowtie$

$\rho_{D(name1,\ addr,\ phone1)}$

Drinkers

Drinkers

Q&A

- What's the difference between the relational and semistructured models?
  - See first lecture. Basically, relational = rigid tables and semistructured = flexible graphs.
- The semistructured model is supposed to be flexible, but can't you represent everything in relational algebra if you plan far enough ahead?
  - Well, in reality there are a lot of things that you can't possibly plan ahead for, so you need the semistructured model for that.