Database Systems and Web (15B11CI312)

Database Systems and Web

Lecture 16: Relational Algebra

Contents to be covered

- Relational Query Language
- Role of Relational Algebra
- Relational Algebra Operators
- Sql vs Relational algebra

Relational Query Languages

- Languages for describing queries on a relational database

- Structured Query Language (SQL)
  - Predominant application level query language
  - Declarative

  - Relational Algebra
  - Intermediate language used within DBMS
  - Procedural

What is an “Algebra”

Mathematical system consisting of:

- *Operands* --- variables or values from which new values can be constructed.
- *Operators* --- symbols denoting procedures that construct new values from given values.

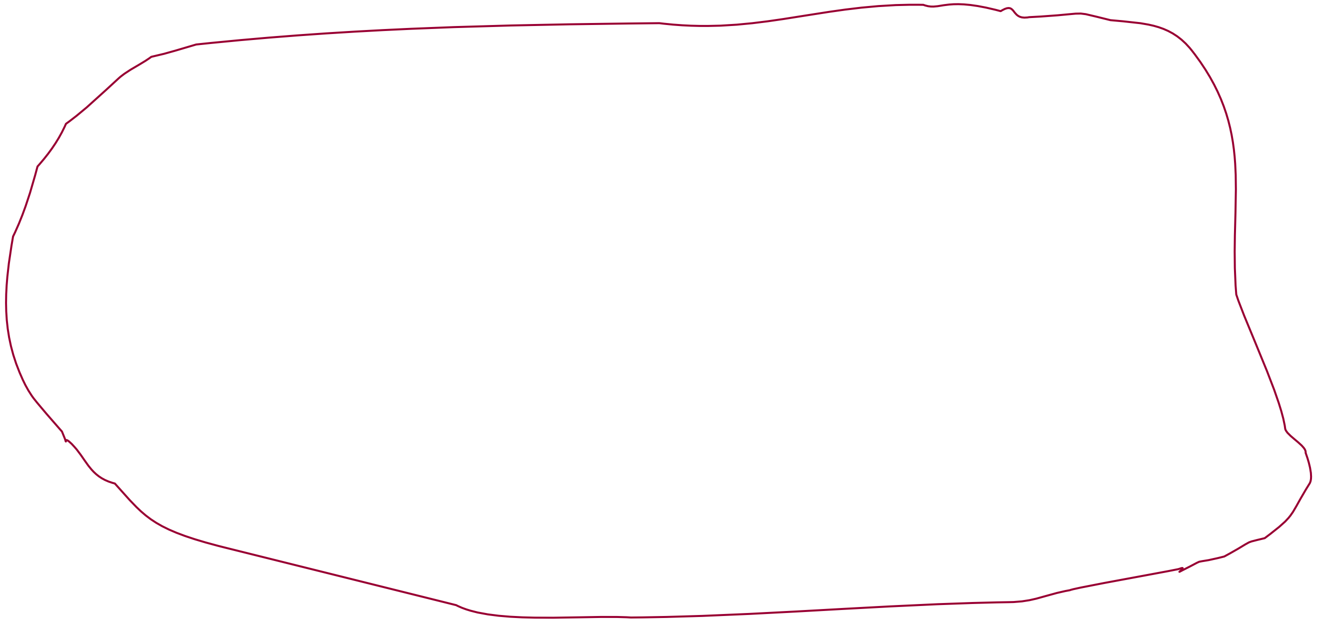
What is Relational Algebra?

An algebra whose *operands are relations or variables that represent relations*.

Operators are designed to do the most common things that we need to do with relations in a database.

- The result is an algebra that can be used as a *query language* for relations.

The Role of Relational Algebra in a DBMS



Relational Algebra and SQL

6 primitive operators:

- union,
- difference,
- product,
- projection,
- selection and
- renaming

Also: derived operators (operators in arithmetic, such as square(x) = x * x). Examples include intersection and join.

Relational Algebra

Basic operations:

- *Selection* om relation (horizontal).
- *Projectior* s from relation(vertical).
- *Cross-prod* mbine two relations.
- *Set-difference* (on 1 but not in relation 2.
- *Union* elation 1 and in relation 2
- *Rename* (ρ) rena ibute(s) and relation

Additional operati

- Intersection, *join*, division, renaming.

Project Operator

- Produces table containing subset of columns of argument table

$$\pi_{\textit{attribute list}}(\textit{relation})$$

- Example:

Person

Id	*Name*	*Address*	*Hobby*
1123	akash	123 Main	stamps
1123	akash	123 Main	coins
5556	Maya	7 Lake Dr	hiking
9876	vikas	5 Pine St	stamps

$\pi_{\textit{Name,Hobby}}$**(Person)**

Name	*Hobby*
akash	stamps
akash	coins
Maya	hiking
vikas	stamps

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

- Example:

Person

Id	*Name*	*Address*	*Hobby*
1123	akash	123 Main	stamps
1123	akash	123 Main	coins
5556	Maya	7 Lake Dr	hiking
9876	vikas	5 Pine St	stamps

$\pi_{Name,Address}$(Person)

Name	*Address*
akash	123 Main
Maya	7 Lake Dr
vikas	5 Pine St

Result is a table (no duplicates); can have fewer tuples than the original

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Selection (σ)

Selects rows that satisfy *selection condition*.

Result is a relation.

Schema of result is same as that of the input relation.

Do we need to do duplicate elimination?

$$name,rating^{(\sigma}\ rating>8^{(S2))}$$

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Expressions

$$\pi_{\mathit{Id,\ Name}}\ (\sigma_{\mathit{Hobby}=\text{’stamps’}\ \text{ OR }\ \mathit{Hobby}=\text{’coins’}}\ (\text{Person})\ )$$

Id	*Name*	*Address*	*Hobby*
1123	akash	123 Main	stamps
1123	akash	123 Main	coins
5556	Maya	7 Lake Dr	hiking
9876	vikas	5 Pine St	stamps

Person

Id	*Name*
1123	akash
9876	vikas

Result

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Set Operators

- Relation is a set of tuples, so set operations should apply: $\cap$, $\cup$, $-$ (set difference)
- Result of combining two relations with a set operator is a relation => all its elements must be tuples having same structure
- Hence, scope of set operations limited to *union compatible relations*

Union Compatible Relations

- Two relations are *union compatible* if
  - Both have same number of columns
  - Names of attributes are the same in both
  - Attributes with the same name in both relations have the same domain

- Union compatible relations can be combined using *union*, *intersection*, and *set difference*

Example

Tables:

Person (*SSN, Name, Address, Hobby*)

Professor (*Id, Name, Office, Phone*)

are not union compatible.

But

π_{Name} (Person) and π_{Name} (Professor)

are union compatible so

π_{Name} (Person) - π_{Name} (Professor)

makes sense.

Union

Union A ∪ B

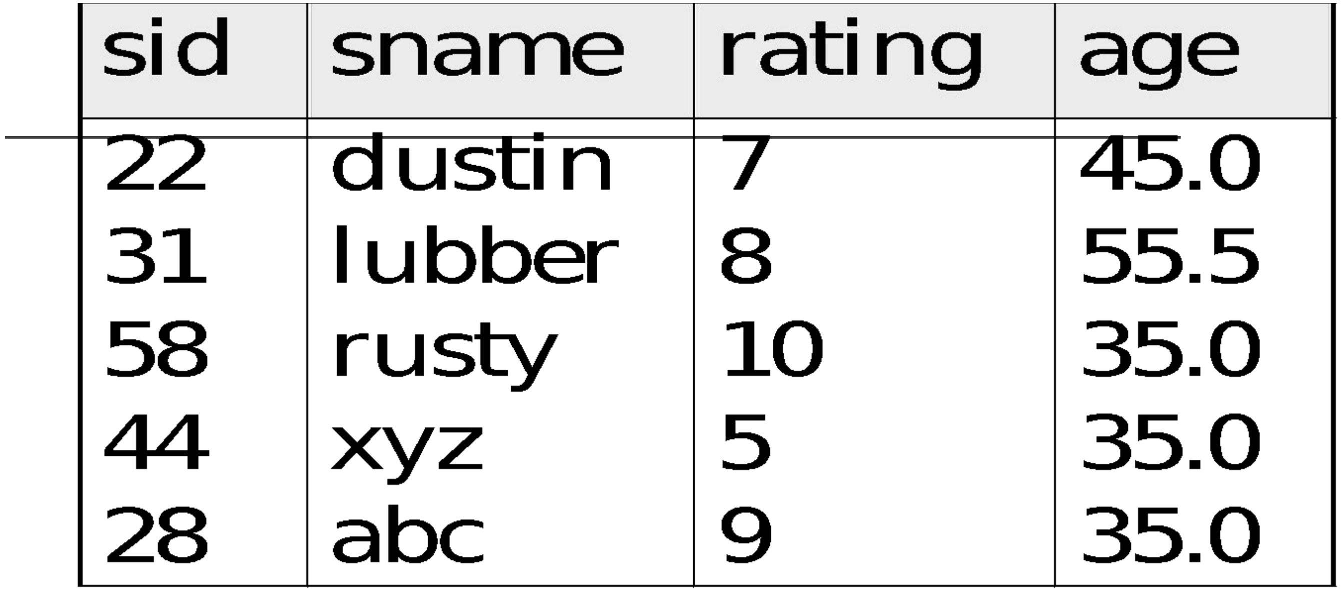
Use SQL keyword UNION. Tables must be compatible ... have the same attributes (column headings).

```
 (SELECT artist FROM Pop_albums
WHERE artist LIKE 'U%')
UNION
 (SELECT artist FROM Band_members
WHERE member = 'Grohl');
```

Result is a one column table containing three entries: *Foo Fighters*, *U2* and *Underworld*.

Union

S1

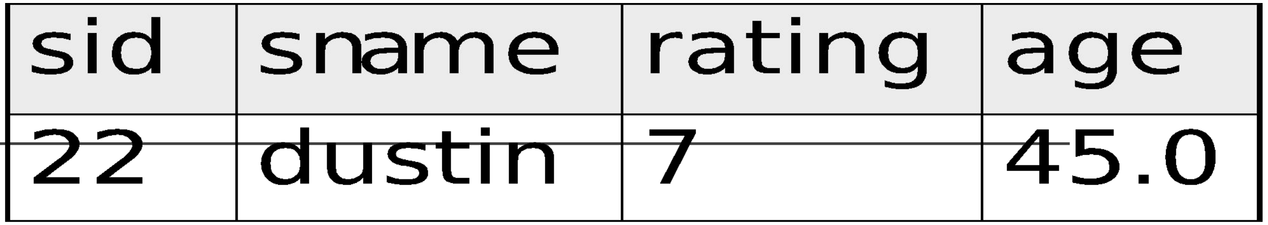


sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	xyz	5	35.0
28	abc	9	35.0

$S1 \cup S2$

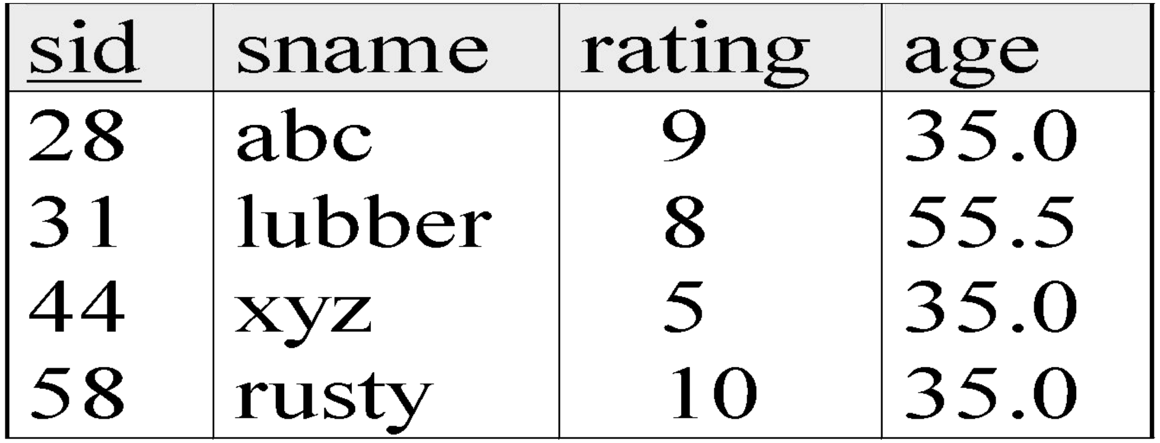
Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Set Difference



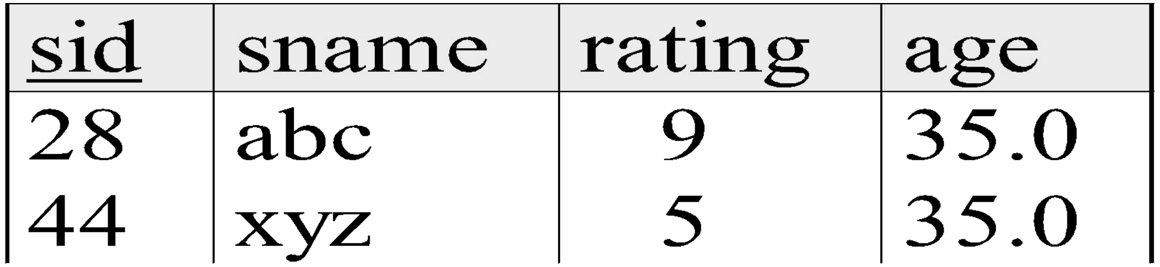
sid	sname	rating	age
22	dustin	7	45.0

S1



sid	sname	rating	age
28	abc	9	35.0
31	lubber	8	55.5
44	xyz	5	35.0
58	rusty	10	35.0

S2



sid	sname	rating	age
28	abc	9	35.0
44	xyz	5	35.0

S2 – S1

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Intersection

Intersection takes two input relations, which must be *union-compatible*.

$$R \cap S = R - (R - S)$$

Intersection

Intersection A ∩ B

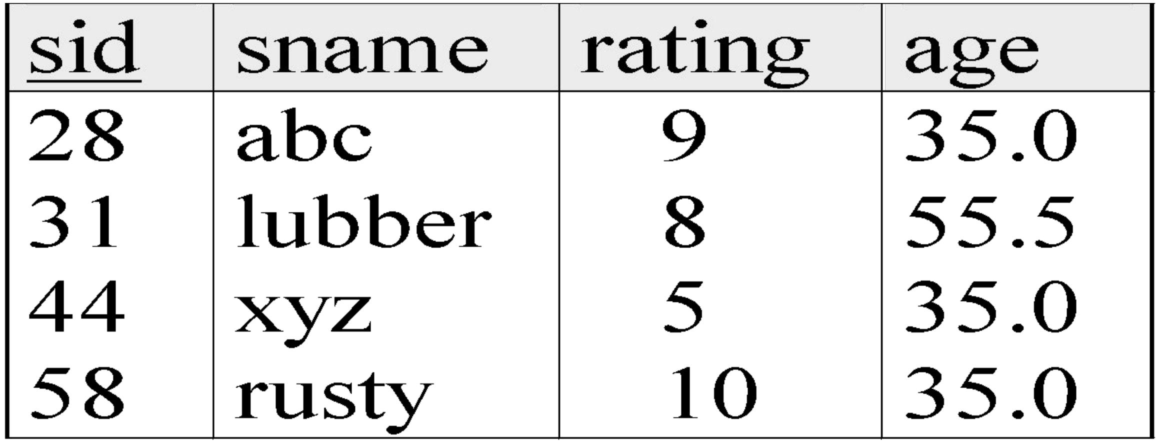
Use SQL keyword `INTERSECT`. Tables must be compatible.

Query :

```
(SELECT artist FROM Pop_albums)
INTERSECT
(SELECT artist FROM Band_members);
```

Intersection

S1



sid	sname	rating	age
28	abc	9	35.0
31	lubber	8	55.5
44	xyz	5	35.0
58	rusty	10	35.0

S2

$$S1 \cap S2$$

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Renaming(ρ)

The RENAME operator gives a new schema to a relation.

R1 := RENAME$_{R1(A1,\ldots,An)}$(R2) makes R1 be a relation with attributes A1,…,A*n* and the same tuples as R2.

Simplified notation: R1(A1,…,A*n*) := R2.

Example

Bars

name,	addr
Joe’s	Maple St.
Sue’s	River Rd.

R(bar, addr) := Bars

R

bar	addr
Joe’s	Maple St.
Sue’s	River Rd.

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

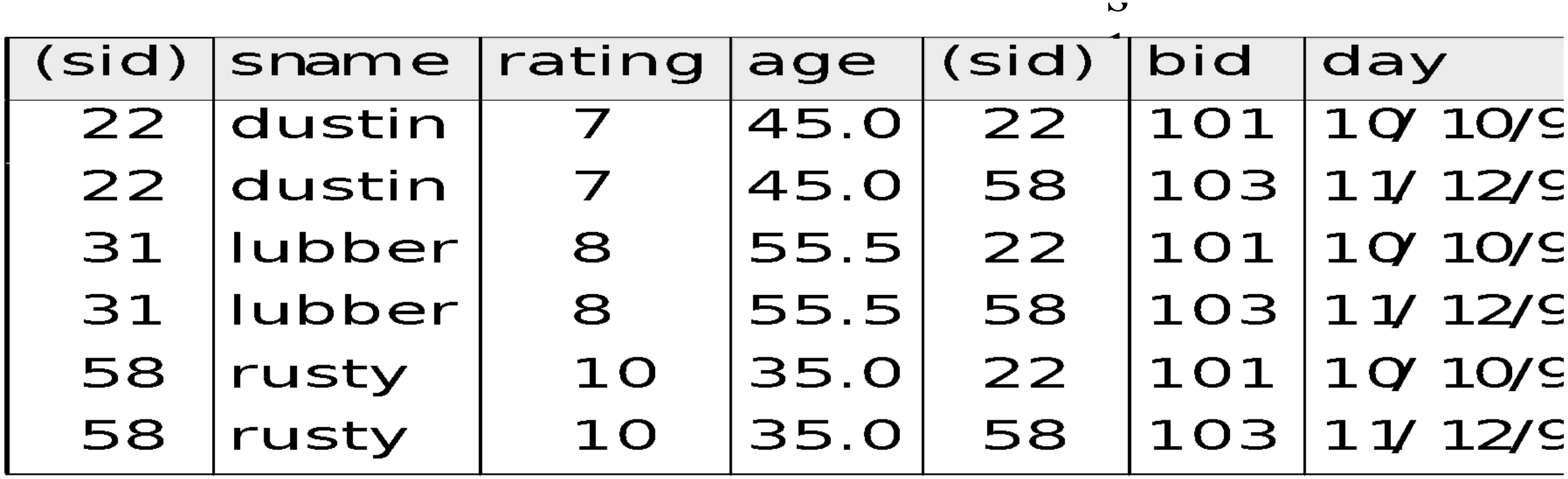
Cross-Product

- S1 x R1: Each row of S1 paired with each row of R1.
- Q: How many rows in the result?
- *Result schema* has one field per field of S1 and R1, with field names `inherited' if possible.
  - *May have a naming conflict*: Both S1 and R1 have a field with the same name.
  - In this case, can use the *renaming operator*:

Cross Product Example

R1

R1 X S1 =



(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/ 10/9
22	dustin	7	45.0	58	103	11/ 12/9
31	lubber	8	55.5	22	101	10/ 10/9
31	lubber	8	55.5	58	103	11/ 12/9
58	rusty	10	35.0	22	101	10/ 10/9
58	rusty	10	35.0	58	103	11/ 12/9

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

References

These Slides were prepared using following resources:

Books:

- A First Course in Database Systems, by J. Ullman and J. Widom
- Fundamentals of Database Systems, by R. Elmasri and S. Navathe