

Relational Algebra

Basic Operations
Algebra of Bags

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.

Core Relational Algebra

- ◆ Union, intersection, and difference.
 - ◆ Usual set operations, but *both operands must have the same relation schema.*
- ◆ Selection: picking certain rows.
- ◆ Projection: picking certain columns.
- ◆ Products and joins: compositions of relations.
- ◆ Renaming of relations and attributes.

Selection

◆ $R1 := \sigma_C(R2)$

- ◆ C is a condition (as in “if” statements) that refers to attributes of $R2$.
- ◆ $R1$ is all those tuples of $R2$ that satisfy C .

Example: Selection

Relation Sells:

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Miller	3.00

JoeMenu := $\sigma_{\text{bar}=\text{"Joe's"}}(\text{Sells})$:

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75

Projection

◆ $R1 := \pi_L(R2)$

- ◆ L is a list of attributes from the schema of $R2$.
- ◆ $R1$ is constructed by looking at each tuple of $R2$, extracting the attributes on list L , in the order specified, and creating from those components a tuple for $R1$.
- ◆ Eliminate duplicate tuples, if any.

Example: Projection

Relation Sells:

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Miller	3.00

Prices := $\pi_{\text{beer, price}}(\text{Sells})$:

beer	price
Bud	2.50
Miller	2.75
Miller	3.00

Product

◆ $R3 := R1 \times R2$

- ◆ Pair each tuple $t1$ of $R1$ with each tuple $t2$ of $R2$.
- ◆ Concatenation $t1t2$ is a tuple of $R3$.
- ◆ Schema of $R3$ is the attributes of $R1$ and then $R2$, in order.
- ◆ But beware attribute A of the same name in $R1$ and $R2$: use $R1.A$ and $R2.A$.

Example: $R3 := R1 \times R2$

R1(

A,	B)
1	2
3	4

R2(

B,	C)
5	6
7	8
9	10

R3(

A,	R1.B,	R2.B,	C)
1	2	5	6
1	2	7	8
1	2	9	10
3	4	5	6
3	4	7	8
3	4	9	10

Theta-Join

- ◆ $R3 := R1 \bowtie_C R2$
 - ◆ Take the product $R1 \times R2$.
 - ◆ Then apply σ_C to the result.
- ◆ As for σ , C can be any boolean-valued condition.
 - ◆ Historic versions of this operator allowed only $A \theta B$, where θ is $=$, $<$, etc.; hence the name “theta-join.”

Example: Theta Join

Sells(

bar,	beer,	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

)

Bars(

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

)

BarInfo := Sells $\bowtie_{\text{Sells.bar} = \text{Bars.name}}$ Bars

BarInfo(

bar,	beer,	price,	name,	addr
Joe's	Bud	2.50	Joe's	Maple St.
Joe's	Miller	2.75	Joe's	Maple St.
Sue's	Bud	2.50	Sue's	River Rd.
Sue's	Coors	3.00	Sue's	River Rd.

)

Natural Join

- ◆ A useful join variant (*natural* join) connects two relations by:
 - ◆ Equating attributes of the same name, and
 - ◆ Projecting out one copy of each pair of equated attributes.
- ◆ Denoted $R3 := R1 \bowtie R2$.

Example: Natural Join

Sells(bar, beer, price)

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

Bars(bar, addr)

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

BarInfo := Sells \bowtie Bars

Note: Bars.name has become Bars.bar to make the natural join “work.”

BarInfo(bar, beer, price, addr)

bar	beer	price	addr
Joe's	Bud	2.50	Maple St.
Joe's	Miller	2.75	Maple St.
Sue's	Bud	2.50	River Rd.
Sue's	Coors	3.00	River Rd.

Renaming

- ◆ The ρ operator gives a new schema to a relation.
- ◆ $R1 := \rho_{R1(A1, \dots, An)}(R2)$ makes R1 be a relation with attributes $A1, \dots, An$ and the same tuples as R2.
- ◆ Simplified notation: $R1(A1, \dots, An) := R2$.

Example: Renaming

Bars(

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

)

$R(\text{bar}, \text{addr}) := \text{Bars}$

R(

bar,	addr
Joe's	Maple St.
Sue's	River Rd.

)