CS422 Principles of Database Systems
Relational Algebra

Chengyu Sun
California State University, Los Angeles

Adapted from Jeffrey Ullman's lecture notes at http://www-db.stanford.edu/~ullman/dscb.html

Overview

- Operators
  - Core and Extended
- Set vs. Bag
- Express constraints in relational algebra

Relational Algebra

- Basis for SQL
- Operands
  - Relations
- Operators
  - *Core*: Set operators, SPJ, rename
  - *Extended*: duplicate elimination, aggregation, grouping, sorting, outer-join

Selection

- $\sigma_c(R)$ or $SELECT_c(R)$
  - Choose *rows* of R that satisfies condition C
  - C is a boolean expression of *constants* and *R's attributes*

Selection Example

Sells

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

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

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

Projection

- $\pi_L(R)$ or $PROJ_L(R)$
  - Choose *columns* of R
  - L is a list of R's attributes
  - Eliminate duplicates in the results (*set semantics*)

Projection Example

Sells

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

$\pi_{beer,price}$(Sells)

beer	price
Bud	2.50
Miller	2.75
Miller	3.00

Product

- R_1 X R_2
  - Concatenate *each* tuple from R_1 with *each* tuple from R_2
  - Also called Cross Product or Cartesian Product

Product Example

R_1

A	B
1	2
3	4

R_2

A	C
5	6
7	8
9	10

R_1 X R_2

R_1.A	B	R_2.A	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

- $R_1 \triangleright\triangleleft_c R_2$ or R_1 $JOIN_c$ R_2
  - σ_c(R1 X R2)

Theta Join Example

Bars

name	address
Joe's	Maple St.
Sue's	River Rd.

Sells

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

Bars $\triangleright\triangleleft_{Bars.name = Sells.bar}$ Sells

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

Natural Join

- $R_1 \triangleright\triangleleft R_2$ or R_1 JOIN R_2
  - Implies equality condition of the *attributes with the same name*
  - Only one column from each pair of equated attributes is kept in results

Natural Join Example

Bars

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

Sells

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

Bars $\triangleright\triangleleft$ Sells

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

Rename

- $\rho_{S(A1,A2,\ldots,An)}(R)$ or $\text{RENAME}_{S(A1,A2,\ldots,An)}(R)$
  - Rename a relation and its attributes
  - Rename a relation only: $\rho_S(R)$

Rename Example

Bars

name	address
Joe's	Maple St.
Sue's	River Rd.

$\rho_{\text{Bars(bar,address)}}(\text{Bars})$

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

Set Operators

- Union: $\cup$
- Intersection: $\cap$
- Difference: $-$

- Two relations must have the same schemas
  - Same number of attributes
  - Same attribute names
  - Same attribute types

Set Operator Examples

A	B
1	2
2	3

$\cup$

A	B
1	2
3	4

$\Rightarrow$

A	B
1	2
2	3
3	4

A	B
1	2
2	3

$\cap$

A	B
1	2
3	4

$\Rightarrow$

A	B
1	2

A	B
1	2
2	3

$-$

A	B
1	2
3	4

$\Rightarrow$

A	B
2	3

Expressions (Queries)

- Find the name of the bars that are either on Maple St. or sells Bud for less than 3 dollars.

$\pi_{\text{bar}}\sigma_{\text{address="Maple St." OR (beer="Bud" AND price < 3.0)}}(\rho_{\text{Bars(bar,address)}}(\text{Bars}) \triangleright\triangleleft \text{Sells})$

- *Relational Operator Precedence*
  - Unary operators
  - Binary operators
  - Set operators
    - Intersection
    - Union, Difference

Linear Notation for Expressions

R(bar, address) := Bars
S := R $\triangleright\triangleleft$ Sells
T := $\sigma_{\text{address="Maple St." OR (beer="Bud" AND price < 3.0)}}$(S)
Ans := π_{bar}

Extended Algebra

- Eliminate duplicates
- Sort tuples
- Extended projection
- Grouping and aggregation
- Outerjoin

Duplicate Elimination

- δ(R) or DELTA(R)

R

A	B
1	2
1	2
2	3
2	3
3	4
2	3

δ(R)

A	B
1	2
2	3
3	4

Sorting

- τ_L(R) or TAU_L(R)
  - L is a list of R's attributes

R

A	B	C
1	4	2
1	3	3
1	2	3
2	3	3
3	2	4
1	2	3

$\tau_{A,C}$(R)

A	B	C
1	4	2
1	3	3
1	2	3
1	2	3
2	3	3
3	2	4

Extended Projection

- π_L(R) or $PROJ_L$(R)
  - *Allow arithmetic expressions of R's attributes in L*

R

A	B
1	2
3	4

$\pi_{A,\ A+B,\ A\ \ A1}$(R)

A	A+B	A1
1	3	1
3	7	3

Aggregation Operators

- Aggregation operators apply to *column(s)* of a relation and produces a single result
  - SUM, AVG, COUNT, MIN, MAX

R

A	B
1	2
3	4

SUM(A) = 4
AVG(A) = 2
COUNT(A) = 2
MIN(A) = 1
MAX(A) = 3

Grouping

- $\gamma_L(R)$ or $GAMMA_L(R)$
  - L is a list of elements that are
    - Individual attributes (*grouping attributes*)
    - AGG(A) where AGG is one of the aggregation operators and A is an attributes

Grouping Example

R

A	B	C
1	4	2
1	3	3
1	2	2
2	3	3
3	2	4
2	2	3

⇨

A	B	C
1	4	2
1	2	2
1	3	3
2	3	3
2	2	3
3	2	4

⇨ $\gamma_{A,SUM(B)\ S,C}(R)$

A	S	C
1	6	2
1	3	3
2	5	3
3	2	4

Outer Joins

- $\triangleright\circ\triangleleft$ or OUTERJOIN (*full outer join*)
- $\triangleright\circ\triangleleft_L$ or LEFTJOIN (*left outer join*)
- $\triangleright\circ\triangleleft_R$ or RIGHTJOIN (*right outer join*)

Dangling Tuples

- When two relations join, a tuple is said to be *dangling* if it does not match any tuple in the other relation.

R $\triangleright\triangleleft$ S

A	B
1	2
3	4

B	C
2	3
5	6

dangling tuples

Outer Join Examples

R $\triangleright\circ\triangleleft$ S

A	B	C
1	2	3
3	4	NULL
NULL	5	6

R $\triangleright\circ\triangleleft_L$ S

A	B	C
1	2	3
3	4	NULL

R $\triangleright\circ\triangleleft_R$ S

A	B	C
1	2	3
NULL	5	6

Summary of Operators

$\cup$	UNION
$\cap$	INTERSECTION
$-$	DIFFERENCE
σ_c	$SELECT_c$
π_L	$PROJ_L$
X	PRODUCT
$\triangleright\triangleleft$	JOIN
$\triangleright\triangleleft_c$	$JOIN_c$
ρ	RENAME

δ	DELTA
τ	TAU
γ	GAMMA
π_L	$PROJ_L$
$\triangleright\circ\triangleleft$	OUTERJOIN
$\triangleright\circ\triangleleft_L$	LEFTJOIN
$\triangleright\circ\triangleleft_R$	RIGHTJOIN
SUM, AVG, COUNT,MIN,MAX	

Set vs. Bag

- Bag (or multiset) allows *duplicates* while set does not
- SQL is a bag language
  - A relation may contain duplicate tuples
  - Only eliminate duplicate tuples when the query explicitly asks for it
  - Certain operations like projection are much more efficient on bags then sets.

Relational Operators on Bags

- Most relational operators work the same for both sets and bags – *just keep the duplicates*
- Union, Intersection, Difference

  R = {1, 1, 2, 3, 3, 3, 4 }, S = {1, 2, 2, 3, 5}

  $R \cup S$ = ??
  $R \cap S$ = ??
  $R - S$ = ??

Bag Laws != Set Laws

- Some laws hold for both sets and bags
  - E.g. $R \cup S = S \cup R$
- Some laws do not
  - E.g. $S \cup S = S$

Constraints on Relations

- Constraints
  - not null, unique, primary key, references (foreign key) …
- Express constraints using relational algebra
  - $R = \varnothing$
  - $R \subseteq S$

Referential Integrity

- Senators(senator_id, senator_name)
- Bills(bill_id, bill_name)
- Votes(senator_id, bill_id, vote)

Functional Dependency

- In Senators, {senator_id} {senator_name}
- In Bills, {bill_id} {bill_name}
- In Votes, {senator_id,bill_id} {vote}

Other Constraints

- NOT NULL
- Unique
- Enumeration
  - E.g. vote can only be *Yes*, *No*, or *Abstain*