Lecture SQL02
Relational Algebra

Relational Algebra

UAH
CPE 353

Relational Algebra - 1

- ***Operators*** are applied to one or more ***Operands***
  - ***Operands*** are **relations**
  - ***Operators*** produce results that are also **relations**
    - May be **unary** or **binary** operators
    - Since the results of an operation is also a relation, the output of one operator may be used as the input to another operator

Relational Algebra - 2

State	Area	Population
AL	52,419	4,661,900
CA	163,696	36,756,666
GA	59,425	9,685,744
NC	53,819	9,222,414
TX	268,820	24,326,974

stateareapop

citypop

City	State	Population
Birmingham	AL	228,798
Huntsville	AL	176,645
Atlanta	GA	537,958
Charlotte	NC	687,456
Greensboro	NC	250,642
Huntsville	TX	35,078

Relational Algebra - 3

- Five basic relational algebra operations
  - Selection σ
  - Projection Π
  - Union U
  - Difference -
  - Cartesian Product X
- This set of operations is “complete” in that all other operations may be expressed as combinations of these five basic operations

Selection Operation - 1

- $\sigma_{condition}(R)$
  - Selects tuples from relation ***R*** that satisfy specified criteria
  - A ***condition*** may include operators

    { =, ≠, <, ≤, >, ≥}

Selection Operation - 2

citypop

City	State	Population
Birmingham	AL	228,798
Huntsville	AL	176,645
Atlanta	GA	537,958
Charlotte	NC	687,456
Greensboro	NC	250,642
Huntsville	TX	35,078

$\sigma_{state = NC}(citypop)$

City	State	Population
Charlotte	NC	687,456
Greensboro	NC	250,642

$\sigma_{city = Atlanta}(citypop)$

City	State	Population
Atlanta	GA	537,958

Projection Operation - 1

- $\Pi_{attribute1,...,attributek}(R)$
  - Deletes attributes not specifically listed
  - **Must also remove duplicate tuples to maintain the *set property* of a relation**
    - Note: Duplicate removal may not be implemented by a DBMS due to its high cost

Projection Operation - 2

citypop

City	State	Population
Birmingham	AL	228,798
Huntsville	AL	176,645
Atlanta	GA	537,958
Charlotte	NC	687,456
Greensboro	NC	250,642
Huntsville	TX	35,078

$\Pi_{city,\ population}(citypop)$

City	Population
Birmingham	228,798
Huntsville	176,645
Atlanta	537,958
Charlotte	687,456
Greensboro	250,642
Huntsville	35,078

Rename Operation - 1

- $\rho_A(B)$
  - Renames relation B to A without modifying the relation

- $\rho_{NewAttributeName \leftarrow OldAttributeName}(B)$
  - Renames specified attribute(s)

Rename Operation - 2

$$\rho_{TotalResidents \leftarrow Population}(citypop)$$

citypop

City	State	Total Residents
Birmingham	AL	228,798
Huntsville	AL	176,645
Atlanta	GA	537,958
Charlotte	NC	687,456
Greensboro	NC	250,642
Huntsville	TX	35,078

Set Operations - 1

- ***Union, Intersection, and Difference***
  - Traditional set operations that are defined only for relations defined for the same attributes and types

Set Operations - 2

govtemployees

UID	Last Name	First Name
4232	Roosevelt	Franklin
12408	Franklin	Benjamin
31023	Washington	George
007	Lincoln	Abraham
8938	Roosevelt	Theodore

presidents

UID	Last Name	First Name
4232	Roosevelt	Franklin
31023	Washington	George
007	Lincoln	Abraham
8938	Roosevelt	Theodore

postalworkers

UID	Last Name	First Name
12408	Franklin	Benjamin

Union Operation - 1

- ***R(X)* U *S(X)***
  - Results in a relation that contains tuples on attributes X from either R or S or both R and S.

Union Operation - 2

presidents U postalworkers =

UID	Last Name	First Name
4232	Roosevelt	Franklin
12408	Franklin	Benjamin
31023	Washington	George
007	Lincoln	Abraham
8938	Roosevelt	Theodore

Intersection Operation - 1

- ***R(X)*** ∩ ***S(X)***
  - Results in a relation that contains only tuples on attributes X from relations R and S that are common to both R and S.

Intersection Operation - 2

presidents ∩ postalworkers = NULL

govtemployees ∩ postalworkers =

UID	Last Name	First Name
12408	Franklin	Benjamin

Difference Operation - 1

- ***R(X) - S(X)***
  - Results in a relation that contains only tuples on attributes X that belong to relation R but do not also belong to relation S.

Difference Operation - 2

govtemployees - presidents =

UID	Last Name	First Name
12408	Franklin	Benjamin

govtemployees - postalworkers =

UID	Last Name	First Name
4232	Roosevelt	Franklin
31023	Washington	George
007	Lincoln	Abraham
8938	Roosevelt	Theodore

Cartesian Product - 1

- ***R X S***
  - Given arity k_1 tuple of relation R and an arity k_2 tuple of relation S, the Cartesian Product is the pairing of all possible tuples in R with all possible tuples of S
  - Note: In the case of duplicate attribute names, renaming or projection may be required.

Cartesian Product - 2

R

A	B	C
a	4	2
b	1	3
c	5	6

S

D	E
0	false
1	true

R X S =

A	B	C	D	E
a	4	2	0	false
a	4	2	1	true
b	1	3	0	false
b	1	3	1	true
c	5	6	0	false
c	5	6	1	true

Joins

- Combination of a ***Cartesian Product*** and a ***Selection*** operation
- Several variations
  - **Theta Join**
  - **Equi-Join**
  - **Natural Join**

Theta Join - 1

- $R \bowtie_{condition} S$
  - Given arity k_1 tuple of relation R and an arity k_2 tuple of relation S, the Cartesian Product is the pairing of all possible tuples in R with all possible tuples of S such that the specified condition holds true
  - Note: In the case of duplicate attribute names, renaming or projection may be required.

Theta Join - 2

R

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

S

D	E
2	4
5	6

What is $R \bowtie_{C < D} S$?

Theta Join - 3

$R \bowtie_{C < D} S = \sigma_{C < D} (R \times S)$

R X S =

A	B	C	D	E
1	4	2	2	4
1	4	2	5	6
3	1	3	2	4
3	1	3	5	6
2	5	6	2	4
2	5	6	5	6

$\sigma_{C < D} (R \times S)$ =

A	B	C	D	E
1	4	2	5	6
3	1	3	5	6

Equi-Join - 1

- $R \bowtie_{equalitycondition} S$
  - Given arity k_1 tuple of relation R and an arity k_2 tuple of relation S, the Cartesian Product is the pairing of all possible tuples in R with all possible tuples of S such that the specified equality condition holds true
  - ***Special case of Theta-join with only equalities***
  - Note: In the case of duplicate attribute names, renaming or projection may be required

Natural Join - 1

- $R \bowtie S$
  - Compute **R X S**
  - For each attribute $\mathbf{A_i}$ common to both **R** and **S**, select all tuples that agree in $\mathbf{R.A_i}$ and $\mathbf{S.A_i}$
  - Eliminate duplicate columns from **S**
  - In other words,

$$\Pi_{list\_of\_nonduplicate\_columns}(\sigma_{R.Ai = S.Ai \text{ for all } i} (R \text{ X } S))$$

Natural Join - 2

R

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

S

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

What is R ⋈ S?

$$R \bowtie S = \Pi_{R.A,R.B,R.C,S.D}(\sigma_{R.A = S.A \text{ AND } R.C = S.C} (R \times S))$$

Natural Join - 3

(Step #1) Compute R X S

R.A	R.B	R.C	S.A	S.C	S.D
1	4	2	3	3	5
1	4	2	5	6	1
1	4	2	2	6	4
1	4	2	1	2	3
3	1	3	3	3	5
3	1	3	5	6	1
3	1	3	2	6	4
3	1	3	1	2	3
2	5	6	3	3	5
2	5	6	5	6	1
2	5	6	2	6	4
2	5	6	1	2	3

Natural Join - 4

(Step #2) Keep all tuples such that
R.A = S.A AND R.C = S.C

R.A	R.B	R.C	S.A	S.C	S.D
1	4	2	1	2	3
3	1	3	3	3	5
2	5	6	2	6	4

(Step #3) Remove Duplicate Columns

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

$$R \bowtie S = \Pi_{R.A,R.B,R.C,S.D}(\sigma_{R.A = S.A \text{ AND } R.C = S.C} (R \times S))$$

Division - 1

- ***R / S*** "Such That"
  - **R / S = { x | ∀ y ∈ S : ∃<x,y> R}**

"For Each" "There Exists"

- Like a Join, Division can be implemented using the five basic operations

Division - 2

R

A	B
Homer	pants
Homer	shirt
Marge	pants
Marge	shirt
Marge	shoes
Lisa	shirt
Lisa	shoes

S

B
pants
shirt

R / S

A
Homer
Marge

T

A
Lisa
Marge

R / T

B
shirt
shoes

Laws and Theorems - 1

From these five basic operations { Π, X, -, σ, $\cup$ }
one can derive the following operations { /, $\bowtie_{Condition}$, $\bowtie$, $\cap$ }.

The operations { X, $\cup$, $\bowtie$, $\cap$ } are commutative and associative.

$R \bowtie S = S \bowtie R$ **(Commutative Property)**

$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$ **(Associative Property)**

Laws and Theorems - 2

$\sigma_{x \text{ AND } y}(R) = \sigma_x(\sigma_y(R))$ (Splitting Property)

$\sigma_x(\sigma_y(R)) = \sigma_y(\sigma_x(R))$ (Commutative Property)

Tuple Relational Calculus

Relational algebra is a ***procedural query language***

TRC is a ***nonprocedural query language***

A query as expressed in Tuple Relational Calculus is

{ t | P(t) }

where **t** is a tuple variable and **P(t)** is as expression that describes **t**

The result is a relation that consists of the set of tuples for which **P(t)** is true