Session 4

Chapter 2 - Relational Model of data - Part 3

Objectives

1. Understand why do we need Algebraic Query Language
2. Understand how Algebraic Query Language work

Contents

2.4. An Algebraic Query Language

2.4.1. Why do we need a special Query Language?

- One should ask why we need a new kind of programming languages for databases?
- The surprising answer is that RA is useful because it is less powerful than C or Java. By limiting what we can do in RA, we get two huge rewards:
  - Ease of programming
  - The ability of the compiler to produce highly optimized code.

2.4.2. What is an Algebra?

- An algebra, in general, consist of operators and atomic operands.
- RA is another example of an algebra: its atomic operands are:
  - Variables = relations
  - Constants = finite relations

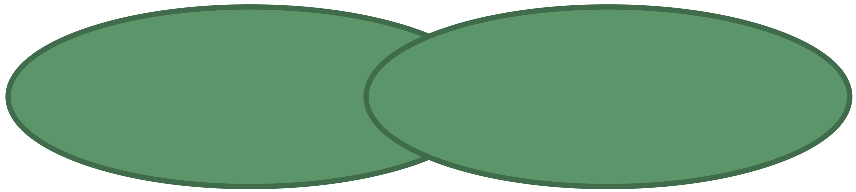
2.4.2. Relational algebra definition

- **Relational algebra**, an offshoot of algebra of sets via operators
- Operators operate on one or more relations to create a new relation

2.4.3 Overview of relational algebra

- 4 kinds of operations of the traditional RA:
  - Set operations: UNION, INTERSECTION, DIFFERENCE
  - Operations that remove parts of a relation: SELECTION and PROJECTION
  - Operations that combine the tuples of two relations: Cartesian PRODUCT, JOIN
  - Renaming operation

2.4.4. Set Operations on Relations - Set Union



Sells1

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

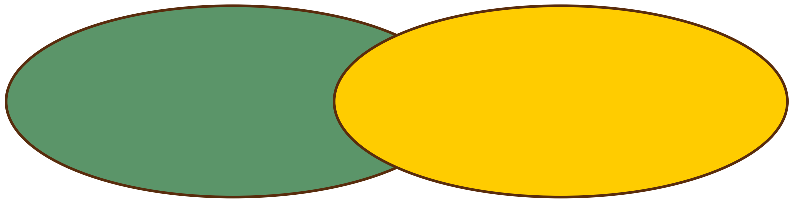
Sells2

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

Sells1 U Sells2

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

2.4.4. Set Operations on Relations - Set Except



Sells1

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

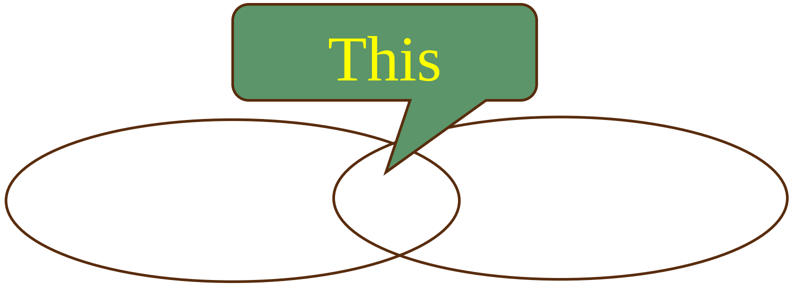
Sells2

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

Sells1 \ Sells2
or Sells1 – Sells2

bar	beer	price
Sue's	Bud	2.50

2.4.4. Set Operations on Relations - Set Intersection



Sells1

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

Sells2

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

Sells1 ∩ Sells2

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

2.4.5. 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 := $\pi_{beer,\ price}$(Sells):

beer	price
Bud	2.50
Miller	2.75
Miller	3.00

Extended Projection

- Using the same π_L operator, we allow the list L to contain arbitrary expressions involving attributes:
  1. Arithmetic on attributes, e.g., $A+B\text{->}C$.
  2. Duplicate occurrences of the same attribute.

Example: Extended Projection

R = (

A	B
1	2
3	4

)

$\pi_{A+B\text{->}C,\ A,\ A}$ (R) =

C	A1	A2
3	1	1
7	3	3

2.4.6. 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 := $\sigma_{bar=\text{“Joe's”}}$(Sells):

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

2.4.7. Cartesian Product

- R3 := R1 X 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 X 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

)

2.4.8. 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
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

)

Bars(

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

)

BarInfo := Sells ⋈ Bars

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

BarInfo(

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

)

2.4.9 Theta Join

- $R \bowtie_{\theta} S$

- The result of theta join consists of all combinations of tuples in two relations R and S that satisfy θ condition

Example: Theta Join

R

A	B
1	1
1	2
2	3

S

C	D
2	2
3	2
4	1

$R \bowtie_{B>=C} S$

A	B	C	D
1	2	2	2
2	3	2	2
2	3	3	2

2.4.11. Renaming

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

Example: Renaming

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.

)

2.4.11 Combining Operations to Form Queries

- Movies (title, year, length, genre, studioName)
- **What are titles and years of movies made by Fox that are at least 100 minutes long?**
- To answer above question, see the steps represented as an expression tree:

$$\pi_{\textit{title, year}}$$

$$\cap$$

$$\sigma_{\textit{length} >= 100} \qquad \sigma_{\textit{studioName} = \text{'Fox'}}$$

Movies **Movies**

2.4.12 Relationships among operations

- Some operations can be expressed by other operations
  - $\mathrm{R} \cap S = R - (R - S)$
  - $R \bowtie_C S = \sigma_C(R \times S)$

2.4.13 Exercises

- A database schema consist of 4 relations:

  (Exercise 2.4.1, page 52-55)

  - Product(maker, model, type)
  - PC(model, speed, ram, hd, price)
  - Laptop(model, speed, ram, hd, screen, price)
  - Printer(model, color, type, price)

2.4.13 Exercises

- Samples data for 4 relations

Product

maker	model	type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
D	1008	pc
D	1009	pc
D	1010	pc
D	3004	printer
D	3005	printer
E	1011	pc
E	1012	pc
E	1013	pc
E	2001	laptop
E	2002	laptop

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649
1005	3.20	512	250	630
1006	3.20	1024	320	1049

Laptop

model	speed	ram	hd	screen	price
2001	2.00	2048	240	20.1	3673
2002	1.73	1024	80	17.0	949
2003	1.80	512	60	15.4	549
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

Printer

model	color	type	price
3001	true	ink-jet	99
3002	false	laser	239
3003	true	laser	899
3004	true	ink-jet	120
3005	false	laser	120

2.4.13 Exercises

- Write expression of relational algebra to answer the following queries:
  a) What types of product made by manufacturer A?
  b) What PC models have a speed of at least 3.00?
  c) What types of product made by both manufacturers A and D?
  d) What types of product made by manufacturer A but not by manufacturer D?
  e) Which manufacturers make laptops with a hard disk of at least 100GB?
  f) Find the model numbers and price of all products (of any type) made by manufacturer B?
  g) Find the model number of all color laser printers?
  h) Find those manufacturers that sell Laptops, but not PC's.
  i) Find those hard-disk sizes that occur in two or more PC's.
  j) Find the manufacturers of the computer with the highest available speed.
  k) Find those manufacturers of at least two different computers (PC or laptop) with speeds of at least 2.80.

Summary

- RA is more useful than C or Java because it is less powerful.
- RA is an algebra: its atomic operands are:
  - Variables that stand for relations
  - Constants, which are finite relations
- The six primitive operators of RA are: Selection, Projection, Product, Union, Difference and Rename
- Other operators of RA are: Natural Join, Theta Join, …

Thank You !