CS 235: Introduction to Databases

Svetlozar Nestorov

Lecture Notes #7

Outline

- So far, we studied schema design.
- How to manipulate data?
- Relational algebra
  - Elegant theoretical framework
  - Not so elegant in practice – SQL
- Relational operators

Core Relational Algebra

- A small set of operators that allows us to manipulate relations in limited but useful ways.
  1. *Union, intersection,* and *difference*: the usual set operators.
     - Relation schemas must be the same.
  2. *Selection*: Pick certain rows from a relation.
  3. *Projection*: Pick certain columns.
  4. *Products and joins*: Combine relations in useful ways.
  5. *Renaming* of relations and their attributes.

Selection

- $R_1 = \sigma_C(R_2)$
  - where C is a condition involving the attributes of relation R_2.
- Example:

Relation *Sells*:

bar	*beer*	*price*
Spoon	Amstel	4
Spoon	Guinness	7
Whiskey	Guinness	7
Whiskey	Bud	5

SpoonMenu = $\sigma_{\text{bar=Spoon}}(Sells)$

bar	*beer*	*price*
Spoon	Amstel	4
Spoon	Guinness	7

Projection

- $R_1 = \pi_L(R_2)$
  - where L is a list of attributes from the schema of R_2.
- Example

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

beer	*price*
Amstel	4
Guinness	7
Bud	5

- Notice elimination of duplicate tuples.

Product

- $R = R_1 \times R_2$
  - pairs each tuple t_1 of R_1 with each tuple t_2 of R_2 and puts in R a tuple t_1t_2.
- Theta-Join: $R = R_1 \underset{C}{\bowtie} R_2$
  - is equivalent to $R = \sigma_C(R_1 \times R_2)$.

Example

Sells =

bar	*beer*	*price*
Spoon	Amstel	4
Spoon	Guinness	7
Whiskey	Guinness	7
Whiskey	Bud	5

Bars =

name	*addr*
Spoon	Wells
Whiskey	Rush

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

bar	*beer*	*price*	*name*	*addr*
Spoon	Amstel	4	Spoon	Wells
Spoon	Guinness	7	Spoon	Wells
Whiskey	Guinness	7	Whiskey	Rush
Whiskey	Bud	5	Whiskey	Rush

Natural Join

- $R = R_1 \bowtie R_2$
  - Equivalent to:
    1. theta-join of R_1 and R_2 with the condition that all attributes of the same name be equated.
    2. one column for each pair of equated attributes is projected out.
- What is the formula?
- Example:
  - Suppose the attribute *name* in relation *Bars* was changed to *bar*, to match the bar name in *Sells*.
  - *BarInfo* = *Sells* $\bowtie$ *Bars*

Natural Join Example

- *BarInfo* = *Sells* $\bowtie$ *Bars*

bar	*beer*	*price*	*addr*
Spoon	Amstel	4	Wells
Spoon	Guinness	7	Wells
Whiskey	Guinness	7	Rush
Whiskey	Bud	5	Rush

Renaming

- $\rho_{S(A_1,\ldots,A_n)}(R)$ produces a relation identical to R but named S and with attributes, in order, named $A_1,\ldots,A_n$.
- Example:

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

bar	*addr*
Spoon	Wells
Whiskey	Rush

- The name of the second relation is R.

Combining Operations

- Any algebra is defined as:
  - basis arguments
  - ways of constructing expressions
- For relational algebra:
  - Arguments = variables standing for relations + finite, constant relations.
  - Expressions constructed by applying one of the operators + parentheses.
- Query = expression of relational algebra.

Operator Precedence

- The normal way to group operators is:
  1. Unary operators σ, π, and ρ have highest precedence.
  2. Next highest are the *multiplicative* operators, $\bowtie$, $\bowtie_C$, and $\times$.
  3. Lowest are the *additive* operators, $\cup$, $\cap$, and —.
- But there is no universal agreement, so we always put parentheses *around* the argument of a unary operator, and it is a good idea to group all binary operators with parentheses *enclosing* their arguments.
- Example:

  Group $R \cup \sigma S \bowtie T$ as $R \cup (\sigma(S) \bowtie T)$.

Expressions and Schemas

- If $\cup$, $\cap$, — applied, schemas are the same, so the result has the same schema.
- Projection: use the attributes listed in the projection.
- Selection: no change in schema.
- Product $R \times S$: use attributes of R and S.
  - But if they share an attribute A, prefix it with the relation name, as $R.A$, $S.A$.
- Theta-join: same as product.
- Natural join: use attributes from each relation; common attributes are merged anyway.
- Renaming: whatever it says.

Example 1

- Find the bars that are either on Wells Street or sell Bud for less than $6.

Sells(bar, beer, price)

Bars(name, addr)

Example 2

- Find the bars that sell two different beers at the same price.

Sells(bar, beer, price)

Linear Notation for Expressions

- Invent new names for intermediate relations, and assign them values that are algebraic expressions.
- Renaming of attributes implicit in schema of new relation.

Example

- Find the bars that are either on Wells Street or sell Bud for less than $6.

Sells(bar, beer, price)

Bars(name, addr)

$R1(name) := \pi_{name}(\sigma_{addr = Wells}(Bars))$

$R2(name) := \pi_{bar}(\sigma_{beer=Bud \text{ AND } price<6}(Sells))$

$R3(name) := R1 \cup R2$