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.
  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 $Sells$:
  
<table>
<thead>
<tr>
<th>bar</th>
<th>beer</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoon Amstel</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Spoon Guinness</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Whiskey Guinness</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Whiskey Bud</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>bar</th>
<th>beer</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoon Amstel</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Spoon Guinness</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>beer</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amstel</td>
<td>4</td>
</tr>
<tr>
<td>Guinness</td>
<td>7</td>
</tr>
<tr>
<td>Bud</td>
<td>5</td>
</tr>
</tbody>
</table>

  Notice elimination of duplicate tuples.

Projection
• $R_1 = \pi_L(R_2)$
  – where L is a list of attributes from the schema of $R_2$.
• Example
  $\pi_{\text{beer,price}}(Sells)$

<table>
<thead>
<tr>
<th>beer</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amstel</td>
<td>4</td>
</tr>
<tr>
<td>Guinness</td>
<td>7</td>
</tr>
<tr>
<td>Bud</td>
<td>5</td>
</tr>
</tbody>
</table>

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 $t_1 t_2$.
• Theta-Join: $R = R_1 \bowtie R_2$
  – is equivalent to $R = \sigma_C(R_1 \times R_2)$. 
Example

\[
\text{Sells} = \begin{array}{ccc}
\text{bar} & \text{beer} & \text{price} \\
\text{Spoon} & \text{Amstel} & 4 \\
\text{Spoon} & \text{Guinness} & 7 \\
\text{Whiskey} & \text{Bud} & 5 \\
\end{array}
\]

\[
\text{Bars} = \begin{array}{ccc}
\text{name} & \text{addr} \\
\text{Spoon} & \text{Wells} \\
\text{Whiskey} & \text{Rush} \\
\end{array}
\]

\[
\text{BarInfo} = \text{Sells} \bowtie_{\text{bar<->name}} \text{Bars}
\]

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 \( \text{Bars} \) was changed to \( \text{bar} \), to match the bar name in \( \text{Sells} \).
  - \( \text{BarInfo} = \text{Sells} \bowtie \text{Bars} \)

Renaming

- \( \rho_{A_1, \ldots, A_n} (R) \) produces a relation identical to \( R \) but named \( S \) and with attributes, in order, named \( A_1, \ldots, A_n \).
- Example:
  \[
  \rho_{R[\text{bar,addr}]} (\text{Bars}) = \begin{array}{cc}
\text{bar} & \text{addr} \\
\text{Spoon} & \text{Wells} \\
\text{Whiskey} & \text{Rush} \\
\end{array}
\]
  - 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 \( \sigma, \pi, \rho \) have highest precedence.
  2. Next highest are the multiplicative operators, \( \bowtie, @, \text{and} \times \).
  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 $6$.
  $Sells(bar, beer, price)$
  $Bars(name, addr)$

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 2

- Find the bars that sell two different beers at the same price.
  $Sells(bar, beer, price)$

Example

- Find the bars that are either on Wells Street or sell Bud for less than $6$.
  $Sells(bar, beer, price)$
  $Bars(name, addr)$
  $R1(name) := \pi_{name}(\sigma_{addr = Wells(Bars)})$
  $R2(name) := \pi_{bar}(\sigma_{beer=Bud \text{ AND } price<6}(Sells))$
  $R3(name) := R1 \cup R2$