Theory I: Database Foundations

Jan-Georg Smaus (Georg Lausen)

26.7.2011

1. Languages: Relational Algebra
   - Projection
   - Selection
   - Union and Difference
   - Join
   - Summary

Languages

Paradigms
- Relational algebra
- Relational calculus
- SQL: not explicitly considered in this theory course!

Relational Algebra

Basic Operators
- delete attributes: Projection.
- select tuples: Selection.
- combine relations: Join.
- set operators: Union, Difference.

Projection

<table>
<thead>
<tr>
<th>MatrId</th>
<th>Name</th>
<th>Address</th>
<th>Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>1223</td>
<td>Hans Eifrig</td>
<td>Seeweg 20</td>
<td>2</td>
</tr>
<tr>
<td>3434</td>
<td>Lisa Lustig</td>
<td>Bergstraße 11</td>
<td>4</td>
</tr>
<tr>
<td>1234</td>
<td>Maria Gut</td>
<td>Am Bächle 1</td>
<td>2</td>
</tr>
</tbody>
</table>

↓

Student'

<table>
<thead>
<tr>
<th>MatrId</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1223</td>
<td>Hans Eifrig</td>
</tr>
<tr>
<td>3434</td>
<td>Lisa Lustig</td>
</tr>
<tr>
<td>1234</td>
<td>Maria Gut</td>
</tr>
</tbody>
</table>
### Projection on tuples

- Let \( R(X) \) be a schema, where \( X = \{A_1, \ldots, A_k\} \).
- Let \( Y \) be a set of attributes, where \( \emptyset \subseteq Y \subseteq X \).
- Let \( \mu \in \text{Tup}(X) \) be a tuple over \( X \).
- \( \mu[Y] \) is called projection of \( \mu \) to \( Y \):
  \[
  \mu[Y] \in \text{Tup}(Y), \quad \mu[Y](A) = \mu(A), A \in Y.
  \]

### Projection on relations

- Let \( r \subseteq \text{Tup}(X) \) a relation and \( Y \subseteq X \).
- \( \pi[Y]r \) is called projection of \( r \) to \( Y \):
  \[
  \pi[Y]r = \{ \mu \in \text{Tup}(Y) | \exists \mu' \in r, \text{such that } \mu = \mu'[Y] \}.
  \]

**Example**

\[
\begin{array}{ccc}
A & B & C \\
\hline
a & b & c \\
\hline
a & a & c \\
c & b & d \\
\end{array}
\]

\( \pi[A,C](r) = \)

---

### Selection condition

- Let \( A, B \in X \), \( a \in \text{dom}(A) \), and \( \theta \in \{=, \neq, \leq, <, \geq, >\} \) a comparison operator.
- An (atomic) selection condition \( \alpha \) (on \( X \)) is of the form \( A \theta B \), resp. \( A \theta a \), resp. \( a \theta A \).
- A tuple \( \mu \in \text{Tup}(X) \) fulfills a selection condition \( \alpha \), if \( \mu(A) \theta \mu(B) \), resp. \( \mu(A) \theta a \), resp. \( a \theta \mu(A) \) hold.
- Atomic selection conditions can be generalized to formulas using \( \land, \lor, \neg, \) and \( (.) \).

**Example**

\[
X = \{A, B, C\}, \\
\mu_1 = (A \rightarrow 2, B \rightarrow 2, C \rightarrow 1), \mu_2 = (A \rightarrow 2, B \rightarrow 3, C \rightarrow 2) \\
\alpha_1 = (A = B), \alpha_2 = ((B > 1) \land (C > 1))
\]
Selection

- Let \( r \subseteq \text{Tup}(X) \) be a relation and \( \alpha \) a selection condition over \( X \).
- \( \sigma[\alpha]r \) is called selection of relation \( r \) by \( \alpha \):

\[
\sigma[\alpha]r = \{ \mu \in \text{Tup}(X) | \mu \in r \land \mu \text{ fulfills } \alpha \}.
\]

Example

\[
\begin{array}{ccc}
A & B & C \\
\hline
a & b & c \\
d & a & f \\
c & b & d \\
\end{array}
\quad \sigma[B = b](r) =
\begin{array}{ccc}
A & B & C \\
\hline
a & b & c \\
c & b & d \\
\end{array}
\]

Union and Difference

- Let \( X \) be a set of attributes and \( r \subseteq \text{Tup}(X), s \subseteq \text{Tup}(X) \) two relations.
- \( r \cup s = \{ \mu \in \text{Tup}(X) | \mu \in r \lor \mu \in s \} \).
- \( r - s = \{ \mu \in \text{Tup}(X) | \mu \in r, \text{where } \mu \not\in s \} \).

Example

\[
\begin{array}{ccc}
A & B & C \\
\hline
a & b & c \\
d & a & f \\
c & b & d \\
\end{array}
\quad \begin{array}{ccc}
A & B & C \\
\hline
b & g & a \\
d & a & f \\
\end{array}
\]

Join

- For sets of attributes \( X, Y \), we may also write \( XY \) instead of \( X \cup Y \).
- Let \( r \subseteq \text{Tup}(X), s \subseteq \text{Tup}(Y) \).
- The (natural) join \( r \bowtie s \) of \( r \) and \( s \) is defined:

\[
r \bowtie s = \{ \mu[X] \in r \land \mu[Y] \in s \}.
\]

Example

\[
\begin{array}{ccc}
A & B & C \\
\hline
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 6 \\
\end{array}
\quad \begin{array}{cc}
C & D \\
\hline
3 & 1 \\
6 & 2 \\
4 & 5 \\
\end{array}
\]

\[
r \bowtie s =
\begin{array}{ccc}
A & B & C \\
\hline
1 & 2 & 3 \\
\end{array}
\quad \begin{array}{cc}
C & D \\
\hline
3 & 1 \\
6 & 2 \\
4 & 5 \\
\end{array}
\]

\[
r \bowtie s =
\begin{array}{ccc}
A & B & C \\
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 6 \\
\end{array}
\quad \begin{array}{cc}
C & D \\
3 & 1 \\
6 & 2 \\
4 & 5 \\
\end{array}
\]

\[
r \bowtie s =
\begin{array}{ccc}
A & B & C \\
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 6 \\
\end{array}
\quad \begin{array}{cc}
C & D \\
3 & 1 \\
6 & 2 \\
4 & 5 \\
\end{array}
\]

\[
r \bowtie s =
\begin{array}{ccc}
A & B & C \\
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 6 \\
\end{array}
\quad \begin{array}{cc}
C & D \\
3 & 1 \\
6 & 2 \\
4 & 5 \\
\end{array}
\]
Observation about join
If $X_1 \cap X_2 = \emptyset$, then $r_1 \bowtie r_2 = r_1 \times r_2$.

Generalization of join
Let $X_i$, $1 \leq i \leq n$ be sets of attributes.

$$\bowtie_{i=1}^n r_i = \{ \mu \in \text{Tup}(\bigcup_{i=1}^n X_i) \mid \mu[X_i] \in r_i, 1 \leq i \leq n \}.$$ 

The relational algebra as query language
- In the algebra expressions we have seen, the operations are applied to relation instances (small letters $r, s, \ldots$), not relation names (capital letters $R, S, \ldots$).
- One can also build expressions based on the relation names. These expressions are then called queries and must be evaluated wrt. a database instance $I$. We write $I(Q)$ for the result of this evaluation, the answer. That is, to obtain $I(Q)$, one has to replace every relation name $R$ occurring in $Q$ by the relation instance $I(R)$.
- $I(Q)$ is again a relation. Recall that a query is formally given as a mapping (transformation) from a database instance to a relation instance.
- Not all computable transformations can be expressed in the relational algebra. Example: transitive closure.
Equivalence

Two algebra expressions $Q, Q'$ are called equivalent, $Q \equiv Q'$, if for any instance $I$ of a database:

$$I(Q) = I(Q').$$

Examples

Let attr($\alpha$) be the attributes in $\alpha$ and let $R, S, T \ldots$ be relation names whose formats are $X, Y, Z$.

- $Z \subseteq Y \subseteq X \implies \pi[Z](\pi[Y]R) \equiv \pi[Z]R$.
- $X \equiv Y \implies R \cap S \equiv R \bowtie S$. 