Theory I: Database Foundations

Jan-Georg Smaus (Georg Lausen)

2.8.2011

1. Formal Design
   - Motivation
   - Functional Dependencies
   - Decomposition

20.1 Motivation

Relations and anomalies

<table>
<thead>
<tr>
<th>CityNo</th>
<th>CityName</th>
<th>CCode</th>
<th>CSurface</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Freiburg</td>
<td>D</td>
<td>357</td>
</tr>
<tr>
<td>9</td>
<td>Berlin</td>
<td>D</td>
<td>357</td>
</tr>
<tr>
<td>40</td>
<td>Moscow</td>
<td>RU</td>
<td>17075</td>
</tr>
<tr>
<td>43</td>
<td>St.Petersburg</td>
<td>RU</td>
<td>17075</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConName</td>
</tr>
<tr>
<td>Europe</td>
</tr>
<tr>
<td>Europe</td>
</tr>
<tr>
<td>Asia</td>
</tr>
</tbody>
</table>

Formal Design

- We want to distinguish good from bad database design.
- What kind of additional information do we need?
- Can we transform a bad into a good design?
- (At which cost?)
20.2 Functional Dependencies

Definition

- Let a relation schema be given by its format $X$ and let $Y, Z \subseteq X$.
- Let $r \in \text{Rel}(X)$. $r$ fulfills a functional dependency $Y \rightarrow Z$, if for all $\mu, \nu \in r$:
  $$\mu[Y] = \nu[Y] \Rightarrow \mu[Z] = \nu[Z].$$
- Let $F$ be a set of functional dependencies over $X$. The set of all relations $r \in \text{Rel}(X)$, which fulfill all functional dependencies in $F$, is called $\text{Sat}(X, F)$.

20.2.2 Membership Test

- The functional dependency $Y \rightarrow Z$, is implied by $F$, written $F \models Y \rightarrow Z$, if for each relation $r$, whenever $r \in \text{Sat}(X, F)$ then $r$ fulfills $Y \rightarrow Z$.
- The set $F^+ = \{ Y \rightarrow Z \mid F \models Y \rightarrow Z \}$ is called closure of $F$.
- The question "$Y \rightarrow Z \in F^+$?" is called membership test.

Key

Let $X = \{A_1, \ldots, A_n\}$. $Y \subseteq X$ is called key of $X$ (wrt. $F$), if
- $Y \rightarrow A_1 \ldots A_n \in F^+$,
- $Z \subset Y \Rightarrow Z \rightarrow A_1 \ldots A_n \notin F^+$.

Armstrong axioms

Let $r \in \text{Sat}(X, F)$.

(A1) Reflexivity: If $Z \subseteq Y \subseteq X$, then $r$ fulfills functional dependency $Y \rightarrow Z$.

(A2) Augmentation: If $Y \rightarrow Z \in F$, $V \subseteq X$, then $r$ fulfills functional dependency $YV \rightarrow ZV$.

(A3) Transitivity: If $Y \rightarrow Z, Z \rightarrow V \in F$, then $r$ fulfills functional dependency $Y \rightarrow V$.

20.3 Decomposition

Let \( \rho = \{ Y_1, \ldots, Y_k \} \) a decomposition of \( X \), i.e., \( Y_1 \cup \ldots \cup Y_k = X \). Let \( \mathcal{F} \) be a set of functional dependencies.

- Let \( r \in \text{Sat}(X, \mathcal{F}) \) and let \( r_i = \pi[Y_i]r, 1 \leq i \leq k \).
  - \( \rho \) is called lossless, if for any \( r \in \text{Sat}(X, \mathcal{F}) \) it holds that:
    \[
    r = \pi[Y_1]r \bowtie \ldots \bowtie \pi[Y_k]r.
    \]
Theorem

Let a format $X$ and set $\mathcal{F}$ of functional dependencies. Let $\rho = (Y_1, Y_2)$ be a decomposition of $X$.

$\rho$ is lossless, iff

$$(Y_1 \cap Y_2) \rightarrow (Y_1 \setminus Y_2) \in \mathcal{F}^+, \text{ or } (Y_1 \cap Y_2) \rightarrow (Y_2 \setminus Y_1) \in \mathcal{F}^+.$$ 

- There is a similar notion called dependency-preserving.
- The aim of good database design is to decompose relations to remove redundancies.