1. Formal Design
   Motivation
   Functional Dependencies
   Decomposition
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?)
## 20.1 Motivation

### Relations and anomalies

#### City

<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>

#### Continent

<table>
<thead>
<tr>
<th>ConName</th>
<th>CCode</th>
<th>ConSurface</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>D</td>
<td>3234</td>
<td>100</td>
</tr>
<tr>
<td>Europe</td>
<td>RU</td>
<td>3234</td>
<td>20</td>
</tr>
<tr>
<td>Asia</td>
<td>RU</td>
<td>44400</td>
<td>80</td>
</tr>
</tbody>
</table>
Having removed anomalies

<table>
<thead>
<tr>
<th>City</th>
<th>Country</th>
<th>Location</th>
<th>Continent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityNo</td>
<td>CityName</td>
<td>CCode</td>
<td>CCode</td>
</tr>
<tr>
<td>7</td>
<td>Freiburg</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>9</td>
<td>Berlin</td>
<td>D</td>
<td>RU</td>
</tr>
<tr>
<td>40</td>
<td>Moscow</td>
<td>RU</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>St.Petersburg</td>
<td>RU</td>
<td></td>
</tr>
</tbody>
</table>
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 $\mathcal{F}$ be a set of functional dependencies over $X$. The set of all relations $r \in \text{Rel}(X)$, which fulfill all functional dependencies in $\mathcal{F}$, is called $\text{Sat}(X, \mathcal{F})$. 
20.2.2 Membership Test

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

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

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

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

Correctness and Completeness

- Every functional dependency derivable by the Armstrong axioms is an element of the closure (correctness).
- Every functional dependency in $\mathcal{F}^+$ is derivable by the Armstrong axioms (completeness)
**Membership test**

Starting from $\mathcal{F}$ apply (A1)–(A3) until $Y \rightarrow Z$ is derived, or $\mathcal{F}^+$ is derived and $Y \rightarrow Z \not\in \mathcal{F}^+$.  

In practice this test is too complex and therefore other tests have been developed.
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:

\[
r = \pi[Y_1]r \bowtie \ldots \bowtie \pi[Y_k]r.
\]
Example

- \( X = \{A, B, C\} \) and \( \mathcal{F} = \{A \rightarrow B, A \rightarrow C\} \).
- \( r \in \text{Sat}(X, \mathcal{F}) \):

\[
\begin{array}{ccc}
A & B & C \\
\hline
a_1 & b_1 & c_1 \\
a_2 & b_1 & c_2
\end{array}
\]

- \( \rho_1 = \{AB, BC\} \) and \( \rho_2 = \{AB, AC\} \).
- \( r \quad \pi[AB]r \bowtie \pi[BC]r \),
- \( r \quad \pi[AB]r \bowtie \pi[AC]r \).
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.