Normalization

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>123 Main St.</td>
</tr>
<tr>
<td>2</td>
<td>Jane</td>
<td>456 State St.</td>
</tr>
</tbody>
</table>

Bad Schema Design

- Update anomaly
- Delete anomaly

Questions To Be Answered

- How do we decide whether a schema is bad?
- How do we decompose a table to turn a bad schema into a good one?

Functional Dependency (FD)

- A functional dependency on table R is the assertion that two records having the same values for attributes \( \{A_1, \ldots, A_n\} \) must also have the same value for attribute B
- \( \{A_1, \ldots, A_n\} \rightarrow B \), or \( \{A_1, \ldots, A_n\} \) functionally determine B

Example: FD
FD with Multiple Attributes

\[
\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow B_1 \\
\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow B_2 \\
\ldots \\
\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow B_m \\
\underbrace{\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow \{B_1, B_2, B_3, \ldots, B_m\}} \\
A \rightarrow B
\]

Trivial Functional Dependency

FD: \(\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow \{B_1, B_2, B_3, \ldots, B_m\}\)

- FD is trivial if all \(B\)'s are in \(A\)
- FD is nontrivial if at least one \(B\) is not in \(A\)
- FD is completely nontrivial if no \(B\) is in \(A\)

Key

- \(A\) is a key of table \(R\) if
  - \(A\) functionally determines all attributes of \(R\)
  - No proper subset of \(A\) functionally determines all attributes of \(R\)

A Few Things about Keys

- A table may have multiple keys
- A key may consist of multiple attributes
- Superset of a key is called a super key
- A key has to be minimal, but not necessarily minimum
- The definition doesn't say anything about uniqueness

Example: Key

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>address</th>
<th>assignment</th>
<th>due</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>123 Main St.</td>
<td>HW1</td>
<td>2009-06-22</td>
<td>A-</td>
</tr>
<tr>
<td>1</td>
<td>John</td>
<td>123 Main St.</td>
<td>HW2</td>
<td>2009-07-10</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>Jane</td>
<td>456 State St.</td>
<td>HW1</td>
<td>2009-06-22</td>
<td>A</td>
</tr>
</tbody>
</table>

Boyce-Codd Normal Form (BCNF)

- A table \(R\) is in BCNF if for every nontrivial \(FD A \rightarrow B\) in \(R\), \(A\) is a super key of \(R\).

  Or

  *The key, the whole key, and nothing but the key, so help me Codd.*
Decompose into BCNF

- Given table \( R \) with FD's F
- Look among F for a BCNF violation \( A \rightarrow B \)
- Compute \( A^+ \)
- Decompose \( R \) into:
  - \( R_1 = A^+ \)
  - \( R_2 = (R - A^+) \cup A \)
- Continue decomposition with \( R_1 \) and \( R_2 \) until all resulting tables are BCNF

Closure of Attributes \( A^+ \)

- Given
  - a set of attributes \( A \)
  - a set of functional dependencies \( S \)
- Closure of \( A \) under \( S \), \( A^+ \), is the set of all possible attributes that are functionally determined by \( A \) based on the functional dependencies inferable from \( S \)

Simple Closure Example

- \( R: \{A,B,C\} \)
  - \( S: \{A \rightarrow B, B \rightarrow C\} \)
- \( (A)^+ ?? \)
- \( (B)^+ ?? \)
- \( (C)^+ ?? \)

Armstrong’s Axioms

- Reflexivity
  - If \( B \subseteq A \), then \( A \rightarrow B \)
- Transitivity
  - If \( A \rightarrow B \) and \( B \rightarrow C \), then \( A \rightarrow C \)
- Augmentation
  - If \( A \rightarrow B \), then \( AC \rightarrow BC \) for any \( C \)

Two More FD Rules

- Union
  - If \( A \rightarrow B \) and \( A \rightarrow C \), then \( A \rightarrow BC \)
- Decomposition
  - If \( A \rightarrow BC \), then \( A \rightarrow B \) and \( A \rightarrow C \)

Computing \( A^+ \)

- Initialize \( A^+ = A \)
- Search in \( S \) for \( B \rightarrow C \) where
  - \( B \subseteq A^+ \)
  - \( C \in A^+ \)
- Add \( C \) to \( A^+ \)
- Repeat until nothing can be added to \( A^+ \)
Computing A+ Example

- \( R(\ A, B, C, D, E, F) \)
- \( S: AB \rightarrow C, BC \rightarrow AD, D \rightarrow E, CF \rightarrow B \)

- \( \{A,B\}^+ ?? \)
- \( \{A,B\} \) a key ??
- How do we find out the key(s) from \( R ?? \)

Example: BCNF Decomposition

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>address</th>
<th>assignment</th>
<th>due</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>123 Main St.</td>
<td>HW1</td>
<td>2009-06-22</td>
<td>A-</td>
</tr>
<tr>
<td>1</td>
<td>John</td>
<td>123 Main St.</td>
<td>HW2</td>
<td>2009-07-10</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>Jane</td>
<td>456 State St.</td>
<td>HW1</td>
<td>2009-06-22</td>
<td>A</td>
</tr>
</tbody>
</table>

Example: BCNF Decomposition

- \( \text{class_records} \)

Motivation for 3NF

- We lose the FD \( \{\text{street_address, city}\} \rightarrow \text{zip} \) after decomposition, or in other words, it becomes unenforceable.

An Unenforceable FD

Before decomposition:

<table>
<thead>
<tr>
<th>street</th>
<th>city</th>
<th>zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>545 Tech Sq.</td>
<td>Cambridge</td>
<td>02138</td>
</tr>
<tr>
<td>545 Tech Sq.</td>
<td>Cambridge</td>
<td>02139</td>
</tr>
</tbody>
</table>

Data error like this can be detected.

After decomposition:

<table>
<thead>
<tr>
<th>street</th>
<th>zip</th>
<th>city</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>545 Tech Sq.</td>
<td>02138</td>
<td>Cambridge</td>
<td>02138</td>
</tr>
<tr>
<td>545 Tech Sq.</td>
<td>02139</td>
<td>Cambridge</td>
<td>02139</td>
</tr>
</tbody>
</table>

The same data error can no longer be detected.

Third Normal Form (3NF)

- A table \( R \) is in 3NF if for every nontrivial \( FD \ A \rightarrow B \) in \( R, \)
  - \( A \) is a super key of \( R \)
  - or \( B \) is part of a key of \( R \)