CSC 4480

Chapter 19: Normalization

Problems Caused By Redundancy

- redundant storage
- update anomaly
- insertion anomaly
- deletion anomaly

Decomposition

- Decompose schema R by replacing the schema with two (or more) relation schema that each contain a subset of R and together include all attributes of R.
- Two questions to ask repeatedly:
  - do we need to decompose a relation?
  - what problems does a given decomposition cause?

Functional Dependencies

- the FD $X \rightarrow Y$ hold if, $\forall t_1, t_2$ in r:
  - If $t_1.X = t_2.X$ then $t_1.Y = t_2.Y$
- Closure of a set of F is $F^*$.
- Armstrong’s axioms:
  - Reflexivity: if $X \supseteq Y$, then $X \rightarrow Y$
  - Augmentation: if $X \rightarrow Y$ then $XZ \rightarrow YZ$
  - Transitivity: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$
- If we take the original FDs and then apply the axioms, we obtain the closure. (Theorem 1).

Boyce-Codd Normal Form

- For every FD $X \rightarrow A$ in F, one of following is true:
  - $A \in X$ (trivial)
  - $X$ is a superkey
- Each attribute depends on the key, the whole key, and nothing but the key.

Third Normal Form

- For every FD $X \rightarrow A$ in F, one of following is true:
  - $A \in X$ (trivial)
  - $X$ is a superkey
  - $A$ is part of some key for R.
(Desirable) Properties of Decompositions

- Lossless-Join: we can recover the original relation from the decomposed relations.
  - Let R be a relation. Decompose into R₁ and R₂. The attributes in common must contain a key to either R₁ or R₂.
- Dependency-Preserving: we can enforce each original FD by examining a single decomposed table.

Decomposition into BCNF

- see algorithm top of p. 623.
- BCNF is more normalized than 3NF, but, we can’t guarantee that we can generate a BCNF decomposition that is also dependency-preserving. But we can for 3NF.

Decomposition into 3NF

- The previous algorithm would generate 3NF. But we already said it is not DP.
- We will need to modify the algorithm.
- This requires the concept of “minimal cover for a set of FDs”.

Minimal Cover

- Formal definition p. 625.
- A minimal cover for a set F of FDs is a set of dependencies that is minimal in two respects:
  - Every dependency is as small as possible; each attribute on the left side is necessary, and the right side is a single attribute.
  - Every dependency in it is required for the closure to equal F⁺.
- An algorithm for obtaining a minimal cover is on p. 626, followed by algorithms for 3NF.

Multivalued Dependencies

<table>
<thead>
<tr>
<th>Course</th>
<th>Teacher</th>
<th>Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physics</td>
<td>Green</td>
<td>Mechanics</td>
</tr>
<tr>
<td>Physics</td>
<td>Green</td>
<td>Optics</td>
</tr>
<tr>
<td>Physics</td>
<td>Brown</td>
<td>Mechanics</td>
</tr>
<tr>
<td>Physics</td>
<td>Brown</td>
<td>Optics</td>
</tr>
</tbody>
</table>

- CTB is in BCNF
- Should be decomposed into CT and CB.

4NF

- A MVD X →→ Y holds over R if, for every legal R, each X value is associated with a set of Y values, and this set is independent of the values in other attributes.
- Definition of 4NF: For every MVD, one of the following is true:
  - Y⊆X or XY=R, or
  - X is a superkey
MVD

- If this holds, we don't need to worry about MVDs:
  - If a relation schema is in BCNF, and at least one of the keys consists of a single attribute, it is also in 4NF.
- This agrees with the previous statement that 4NF, 5NF problems only occur in all-key tables.

5NF

- Join dependency holds over R if \( R_1, R_2, ..., R_n \) is a lossless-join decomposition of R.
- A relation R is in 5NF if, for every JD that holds over R, one of these is true:
  - \( R_i = R \) for some i
  - The JD is implied by those FDs over R in which the left side is a key for R.
- If a schema is in 3NF and each key is a single attribute, it is also in 5NF.

Example

<table>
<thead>
<tr>
<th>Store</th>
<th>Manufacturer</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowes</td>
<td>Dewalt</td>
<td>Hammer</td>
</tr>
<tr>
<td>Lowes</td>
<td>Dewalt</td>
<td>Saw</td>
</tr>
<tr>
<td>Home Depot</td>
<td>Dewalt</td>
<td>Hammer</td>
</tr>
<tr>
<td>Home Depot</td>
<td>Dewalt</td>
<td>Saw</td>
</tr>
<tr>
<td>Lowes</td>
<td>Stanley</td>
<td>Hammer</td>
</tr>
<tr>
<td>Home Depot</td>
<td>Stanley</td>
<td>Hammer</td>
</tr>
</tbody>
</table>

Is this table in 3NF? 4NF? 5NF?

Example

- It depends. If this decomposition is true, then not in 4NF due to MVD.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Item</th>
<th>Store</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dewalt</td>
<td>Hammer</td>
<td>Lowes</td>
<td>Dewalt</td>
</tr>
<tr>
<td>Dewalt</td>
<td>Saw</td>
<td>Lowes</td>
<td>Stanley</td>
</tr>
<tr>
<td>Stanley</td>
<td>Hammer</td>
<td>Home Depot</td>
<td>Dewalt</td>
</tr>
<tr>
<td>Home Depot</td>
<td>Hammer</td>
<td>Home Depot</td>
<td>Stanley</td>
</tr>
</tbody>
</table>

- But, if a store can sell, say, just a hammer from Dewalt, then it is in 4NF.

Example

- But is it in 5NF?
- Probably, unless the following is true:

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Item</th>
<th>Store</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dewalt</td>
<td>Hammer</td>
<td>Lowes</td>
<td>Dewalt</td>
</tr>
<tr>
<td>Dewalt</td>
<td>Saw</td>
<td>Lowes</td>
<td>Stanley</td>
</tr>
<tr>
<td>Stanley</td>
<td>Hammer</td>
<td>Home Depot</td>
<td>Dewalt</td>
</tr>
<tr>
<td>Home Depot</td>
<td>Hammer</td>
<td>Home Depot</td>
<td>Stanley</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Store</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowes</td>
<td>Hammer</td>
</tr>
<tr>
<td>Lowes</td>
<td>Saw</td>
</tr>
<tr>
<td>Home Depot</td>
<td>Hammer</td>
</tr>
</tbody>
</table>