Relational Database Design

CPS 216
Advanced Database Systems

Relational design: a review

- Identifying tuples: keys
- Generalizing the key concept: FDs
- Non-key FDs: redundancy
- Avoiding redundancy: BCNF decomposition
- Preserving FDs: 3NF

BNCF = no redundancy?

- Student (SID, CID, club)
  - Suppose your classes have nothing to do with the clubs you join
  - FDs?
    - None
  - BNCF?
    - Yes
  - Redundancies?
    - Tons!

<table>
<thead>
<tr>
<th>SID</th>
<th>CID</th>
<th>club</th>
</tr>
</thead>
<tbody>
<tr>
<td>140</td>
<td>CPS 216</td>
<td>ballet</td>
</tr>
<tr>
<td>140</td>
<td>CPS 216</td>
<td>sumo</td>
</tr>
<tr>
<td>142</td>
<td>CPS 214</td>
<td>sumo</td>
</tr>
<tr>
<td>142</td>
<td>CPS 214</td>
<td>ballet</td>
</tr>
<tr>
<td>123</td>
<td>CPS 216</td>
<td>chess</td>
</tr>
<tr>
<td>123</td>
<td>CPS 216</td>
<td>pdf</td>
</tr>
</tbody>
</table>

Multi-valued dependencies

- A multi-valued dependency (MVD) has the form $X \rightarrow Y$, where $X$ and $Y$ are sets of attributes in a relation $R$
- $X \rightarrow Y$ means that whenever two tuples in $R$ agree on all the attributes of $X$, then we can swap their $Y$ components and get two new tuples that are also in $R$

$xyz\begin{array}{ccc}
   a & b & c \\
   a & b & c \\
   a & b & c \\
   a & b & c \\
   a & b & c \\
\end{array}$

MVD examples

Student (SID, CID, club)

- $SID \rightarrow CID$
- $SID \rightarrow club$
  - Intuition: given SID, CID and club are “independent”
- $SID, CID \rightarrow club$
  - Trivial: LHS $\cup$ RHS = all attributes of $R$
- $SID, CID \rightarrow SID$
  - Trivial: LHS $\supseteq$ RHS

Complete MVD + FD rules

- FD reflexivity, augmentation, and transitivity
- MVD complementation:
  If $X \rightarrow Y$, then $X \rightarrow \text{attrs}(R) - X - Y$
  Try proving dependencies
- MVD augmentation:
  If $X \rightarrow Y$ and $V \subseteq W$, then $XW \rightarrow YV$
  with these!!
- MVD transitivity:
  If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z - Y$
- Replication (FD is MVD):
  If $X \rightarrow Y$, then $X \rightarrow \rightarrow Y$
- Coalescence:
  If $X \rightarrow Y$ and $Z \subseteq Y$ and there is some $W$ disjoint from $Y$ such that $W \rightarrow Z$, then $X \rightarrow Z$
An elegant solution: chase

- Given a set of FDs and MVDs $D$, does another dependency $d$ (FD or MVD) follow from $D$?
- Procedure
  - Start with the hypotheses of $d$, and treat them as "seed" tuples in a relation
  - Apply the given dependencies in $D$ repeatedly
    - If we apply an FD, we infer equality of two symbols
    - If we apply an MVD, we infer more tuples
  - If we infer the conclusion of $d$, we have a proof
  - Otherwise, if nothing more can be inferred, we have a counterexample

Proof by chase

- In $R(A, B, C, D)$, does $A \rightarrow\rightarrow B$ and $B \rightarrow\rightarrow C$ imply $A \rightarrow\rightarrow C$?

<table>
<thead>
<tr>
<th>Have</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \rightarrow B$</td>
<td>$B \rightarrow C$</td>
</tr>
<tr>
<td>$b_1 = b_2$</td>
<td>$c_1 = c_2$</td>
</tr>
</tbody>
</table>

Another proof by chase

- In $R(A, B, C, D)$, does $A \rightarrow B$ and $B \rightarrow C$ imply $A \rightarrow C$?

<table>
<thead>
<tr>
<th>Have</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \rightarrow B$</td>
<td>$B \rightarrow C$</td>
</tr>
<tr>
<td>$b_1 = b_2$</td>
<td>$c_1 = c_2$</td>
</tr>
</tbody>
</table>

Counterexample by chase

- In $R(A, B, C, D)$, does $A \rightarrow\rightarrow BC$ and $CD \rightarrow B$ imply $A \rightarrow B$?

<table>
<thead>
<tr>
<th>Have</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \rightarrow\rightarrow BC$</td>
<td>$b_1 = b_2$</td>
</tr>
</tbody>
</table>

4NF

- A relation $R$ is in Fourth Normal Form (4NF) if
  - For every non-trivial MVD $X \rightarrow\rightarrow Y$ in $R$, $X$ is a super key
  - That is, all FDs and MVDs follow from “key $\rightarrow$ other attributes”

- 4NF is stronger than BCNF
  - Because every FD is also an MVD

4NF decomposition algorithm

- Find a 4NF violation
  - A non-trivial MVD $X \rightarrow\rightarrow Y$ in $R$ where $X$ is not a super key
- Decompose $R$ into $R_1$ and $R_2$, where
  - $R_1$ has attributes $X \cup Y$
  - $R_2$ has attributes $X \cup Z$ ($Z$ contains attributes not in $X$ or $Y$)
- Repeat until all relations are in 4NF

- Almost identical to BCNF decomposition algorithm
- Any decomposition on a 4NF violation is lossless
4NF decomposition example

\[
\begin{array}{|c|c|c|}
\hline
\text{ID} & \text{name} & \text{cost} \\
\hline
1 & Part & Assembly \\
\hline
\end{array}
\]

Student (SID, CID, club)

\[\text{4NF violation: } \text{SID} \rightarrow \rightarrow \text{CID}\]

Enroll (SID, CID)

Join (SID, club)

\[
\begin{array}{|c|c|c|}
\hline
\text{SID} & \text{CID} & \text{club} \\
\hline
142 & CPS 216 & ballet \\
142 & CPS 216 & sumo \\
142 & CPS 214 & sumo \\
123 & CPS 216 & chess \\
123 & CPS 216 & golf \\
\hline
\end{array}
\]

3NF, BCNF, and 4NF

<table>
<thead>
<tr>
<th>Preserves FDs?</th>
<th>3NF</th>
<th>BCNF</th>
<th>4NF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundancy due to FDs?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Redundancy due to MVDs?</td>
<td>Possible</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Recap

- Another source of redundancy: MVDs
- Reasoning about FDs and MVDs: chase
- Avoiding redundancy due to MVDs: 4NF

A complete design example

- Information about parts and assemblies for a manufacturing company; e.g.:
  - A bicycle consists of one frame and two wheels; the cost of assembly is $30
  - A frame is just a basic part
  - A wheel consists of one tire, one rim, and 48 spokes; the cost of assembly is $40
  - Everything has a part ID and a name

Entities and relationships

- Entities
  - Parts (with ID and name)
  - Assemblies (with ID, name, and cost)
- Relationships
  - An assembly as a whole is a part (with an assembly cost)
  - An assembly consists of some number of one or more subparts

Identify constraints

- ID is a key for parts and assemblies
- An assembly has one or more subparts
- A part can serve as a subpart for zero or more assemblies
Design relational schema

- Entities to relations
  - Part (ID, name)
  - Assembly (ID, cost)
    - ID is inherited from Part, name is not repeated
- Relationships to relations
  - ComposedOf (assemblyID, partID, number)
    - Use keys as “links”

Encode constraints

- Part (ID, name)
  - ID is a key
- Assembly (ID, cost)
  - ID is a key
- ComposedOf (assemblyID, partID, number)
  - {assemblyID, partID} is a key
- Any missing constraints?

Apply relational design theory

- Part (ID, name)
  - ID is a key
- Assembly (ID, cost)
  - ID is a key
- ComposedOf (assemblyID, partID, number)
  - {assemblyID, partID} is a key
- 3NF? BCNF? 4NF?
  - Yes, yes, yes

Populate schema with data

- Part
  - ID
  - name
  - Assembly
    - ID
    - cost
- ComposedOf
  - assemblyID
  - partID
  - number

Good design principles

- Avoid redundancy
- Avoid decomposing too much
- KISS
  - Focus on the task and avoid over-design