Normalization

- Improve the schema by decomposing relations and removing anomalies.
- Boyce-Codd Normal Form (BCNF): all FD's follow from the fact key $\rightarrow$ everything.
- Formally, $R$ is in BCNF if every nontrivial FD for $R$, say $X \rightarrow A$, has $X$ a superkey.
  - "Nontrivial" = right-side attribute not in left side.

BCNF properties

1. Guarantees no redundancy due to FD’s.
2. Guarantees no update anomalies = one occurrence of a fact is updated, not all.
3. Guarantees no deletion anomalies = valid fact is lost when tuple is deleted.

Example

Beers(name, manf, manfAddr).

- FD’s:
  - name $\rightarrow$ manf,
  - manf $\rightarrow$ manfAddr.
- Only key is name.
- manf $\rightarrow$ manfAddr violates BCNF with a left side unrelated to any key.

Decomposition into BCNF

- Setting: relation $R$, given FD’s $F$. Suppose relation $R$ has BCNF violation $X \rightarrow B$.
- We need only look among FD’s of $F$ for a BCNF violation.
- If there are no violations, then the relation is in BCNF.
- Don’t we have to consider implied FD’s?
- No, because…

Proof

- Let $Y \rightarrow A$ is a BCNF violation and follows from $F$.
- Then the computation of $Y^*$ used at least one FD $X \rightarrow B$ from $F$.
- $X$ must be a subset of $Y$.
- Thus, if $Y$ is not a superkey, $X$ cannot be a superkey either, and $X \rightarrow B$ is also a BCNF violation.
Decomposition Algorithm (1/2)

For every violation $X \rightarrow B$ among given FD’s:
1. Compute $X^+$.
   - Cannot be all attributes – why?
2. Decompose $R$ into $X^+$ and $(R \setminus X^+) \cup X$.

Decomposition Algorithm (2/2)

3. Find the FD’s for the decomposed relations.
   - Project the FD’s from $F = \{X \rightarrow B \mid X \subseteq X^+ \}$ that involve only attributes from $X^+$ or only from $(R \setminus X^+) \cup X$.

Example (1/3)

$R = \text{Drinkers}(\text{name, addr, beersLiked, manf, favoriteBeer})$

FD’s:
- $\text{name} \rightarrow \text{addr}$
- $\text{name} \rightarrow \text{favoriteBeer}$
- $\text{beersLiked} \rightarrow \text{manf}$

Pick BCNF violation $\text{name} \rightarrow \text{addr}$.  

Close the left side: $name^+ = name \cup addr \cup favoriteBeer$.

Decomposed relations:
- Drinkers1($\text{name, addr, favoriteBeer}$)
- Drinkers2($\text{name, beersLiked, manf}$)

Projected FD’s (skipping a lot of work):
- For Drinkers1: $name \rightarrow addr$ and $name \rightarrow favoriteBeer$.
- For Drinkers2: $beersLiked \rightarrow manf$.

Example (2/3)

- BCNF violations?
  - For Drinkers1, $name$ is key and all left sides of FD’s are superkeys.
  - For Drinkers2, $(\text{name, beersLiked})$ is the key, and $\text{beersLiked} \rightarrow \text{manf}$ violates BCNF.

Example (3/3)

- Decompose Drinkers2
- Close $\text{beersLiked}^+ = \text{beersLiked, manf}$.
- Decompose:
  - Drinkers3($\text{beersLiked, manf}$)
  - Drinkers4($\text{name, beersLiked}$)

Resulting relations are all in BCNF:
- Drinkers1($\text{name, addr, favoriteBeer}$)
- Drinkers3($\text{beersLiked, manf}$)
- Drinkers4($\text{name, beersLiked}$)

Third Normal Form (3NF)

- Sometimes we have a dilemma:
  - If you decompose, you can’t check the FD’s in the decomposed relations.
  - If you don’t decompose, you violate BCNF.
- Abstractly: $AB \rightarrow C$ and $C \rightarrow B$.
- In book: $\text{title city} \rightarrow \text{theatre}$ and $\text{theatre} \rightarrow \text{city}$.
- Another example: $\text{street city} \rightarrow \text{zip}$, $\text{zip} \rightarrow \text{city}$.
- Keys: $AB$ and AC, but $C \rightarrow B$ has a left side not a superkey.
- Suggests decomposition into $BC$ and $AC$.
  - But you can’t check the FD $AB \rightarrow C$ in these relations.
Example

• What can go wrong if we decompose:
  \[A = \text{street}, \quad B = \text{city}, \quad C = \text{zip}.\]

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

Join:

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

“Elegant” Workaround

• Define the problem away.
• A relation \( R \) is in 3NF iff for every nontrivial
  \( FD \ X \rightarrow A \), either:
  1. \( X \) is a superkey, or
  2. \( A \) is prime = member of at least one key.
• Thus, the canonical problem goes away: you don’t have to decompose because all
  attributes are prime.

Decomposition Properties

1. We should be able to recover from the
   decomposed relations the data of the
   original.
   • Recovery involves projection and join (next
     time).
2. We should be able to check that the FD’s
   for the original relation are satisfied by
   checking the projections of those FD’s in
   the decomposed relations.

3NF vs. BCNF

• Without proof, we assert that it is always
  possible to decompose into BCNF and
  satisfy (1).
• Also without proof, we can decompose
  into 3NF and satisfy both (1) and (2).
• But it is not possible to decompose into
  BCNF and get both (1) and (2).
  • Street-city-zip is an example of this point.

Multivalued Dependencies

• The multivalued dependency \( X \rightarrow\rightarrow Y \) holds in a relation \( R \) if whenever we have
  two tuples of \( R \) that agree in all the
  attributes of \( X \), then we can swap their \( Y \)
  components and get two new tuples that
  are also in \( R \).

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>phones</th>
<th>beersLiked</th>
</tr>
</thead>
<tbody>
<tr>
<td>sue</td>
<td>a</td>
<td>p1</td>
<td>b1</td>
</tr>
<tr>
<td>sue</td>
<td>a</td>
<td>p2</td>
<td>b2</td>
</tr>
</tbody>
</table>

Example

• \( \text{Drinkers(name, addr, phones, beersLiked)} \)
• MV D \( \text{name } \rightarrow \rightarrow \text{phones}. \)
• If \( \text{Drinkers} \) has the two tuples:

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>phones</th>
<th>beersLiked</th>
</tr>
</thead>
<tbody>
<tr>
<td>sue</td>
<td>a</td>
<td>p1</td>
<td>b1</td>
</tr>
<tr>
<td>sue</td>
<td>a</td>
<td>p2</td>
<td>b2</td>
</tr>
</tbody>
</table>

it must also have the same tuples with \( \text{phones} \)
components swapped:

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>phones</th>
<th>beersLiked</th>
</tr>
</thead>
<tbody>
<tr>
<td>sue</td>
<td>a</td>
<td>p1</td>
<td>b2</td>
</tr>
<tr>
<td>sue</td>
<td>a</td>
<td>p2</td>
<td>b1</td>
</tr>
</tbody>
</table>
MVD Rules
• Every FD is an MVD: if $X \rightarrow Y$, then swapping $Y$'s between tuples that agree on $X$ doesn't create new tuples.
• Example, in Drinkers: $\text{name} \rightarrow \rightarrow \text{addr}$.
• Complementation: if $X \rightarrow \rightarrow Y$, then $X \rightarrow Z$, where $Z$ is all attributes not in $X$ or $Y$.
• Example: since $\text{name} \rightarrow \rightarrow \text{phones}$ holds in Drinkers, so does $\text{name} \rightarrow \rightarrow \text{addr beersLiked}$.

Splitting Doesn’t Hold
• Sometimes you need to have several attributes on the right of an MVD.
• For example: Drinkers($\text{name}$, $\text{areaCode}$, $\text{phones}$, $\text{beersLiked}$, $\text{beerManf}$)

<table>
<thead>
<tr>
<th>name</th>
<th>areaCode</th>
<th>phones</th>
<th>beersLiked</th>
<th>beerManf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leo</td>
<td>773</td>
<td>555-1111</td>
<td>Bud</td>
<td>A.B.</td>
</tr>
<tr>
<td>Leo</td>
<td>773</td>
<td>555-1111</td>
<td>Honkers</td>
<td>G.I.</td>
</tr>
<tr>
<td>Leo</td>
<td>800</td>
<td>555-9999</td>
<td>Bud</td>
<td>A.B.</td>
</tr>
<tr>
<td>Leo</td>
<td>800</td>
<td>555-9999</td>
<td>Honkers</td>
<td>G.I.</td>
</tr>
</tbody>
</table>

• $\text{name} \rightarrow \rightarrow \text{areaCode phones}$ holds, but neither $\text{name} \rightarrow \rightarrow \text{areaCode}$ nor $\text{name} \rightarrow \rightarrow \text{phones}$ do.

Fourth Normal Form (4NF)
• Eliminate redundancy due to multiplicative effect of MVD’s.
• Roughly: treat MVD’s as FD’s for decomposition, but not for finding keys.
• Formally: $R$ is in Fourth Normal Form if whenever MVD $X \rightarrow \rightarrow Y$ is nontrivial ($Y$ is not a subset of $X$, and $X \cup Y$ is not all attributes), then $X$ is a superkey.
  – Remember, $X \rightarrow Y$ implies $X \rightarrow \rightarrow Y$, so 4NF is more stringent than BCNF.
• Decompose $R$, using 4NF violation $X \rightarrow \rightarrow Y$, into $XY$ and $X \cup (R-Y)$. 