Goal = BCNF = Boyce-Codd Normal Form = all FD’s follow from the fact “key → everything.”

- Formally, $R$ is in BCNF if for every nontrivial FD for $R$, say $X \rightarrow A$, then $X$ is a superkey.
  - “Nontrivial” = right-side attribute not in left side.

Why?
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.
Boyce-Codd Normal Form (Cont.)

- Example schema that is not in BCNF:
  \[ \text{in\_dep (ID, name, salary, \textit{dept\_name}, building, budget)} \]
  because:
  - \( \text{dept\_name} \rightarrow \text{building, budget} \)
    - holds on \text{in\_dep}
    - but
  - \( \text{dept\_name} \) is not a superkey
- When decompose \text{in\_dept} into \text{instructor} and \text{department}
  - \text{instructor} is in BCNF
  - \text{department} is in BCNF

Lossless Join

- Goal: All legal values can be stored in relations
  - Recover originals through join
- Formally: \( X, Y \) is a lossless join decomposition of \( R \) w.r.t. \( F \) if \( \forall r \in R \) satisfying dependencies in \( F \),
  \[ \pi_X(r) \Join \pi_Y(r) = r \]
- Does BCNF imply lossless join?
Decomposing a Schema into BCNF

- Let $R$ be a schema $R$ that is not in BCNF. Let $\alpha \rightarrow \beta$ be the FD that causes a violation of BCNF.
- We decompose $R$ into:
  - $(\alpha \cup \beta)$
  - $(R - (\beta - \alpha))$
- In our example of $in\_dep$,
  - $\alpha = dept\_name$
  - $\beta = building, budget$
and $in\_dep$ is replaced by
  - $(\alpha \cup \beta) = (dept\_name, building, budget)$
  - $(R - (\beta - \alpha)) = (ID, name, dept\_name, salary)$

Decomposition to Reach 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, not those that follow from $F$.
- Proof: If $Y \rightarrow A$ 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.
- In our example of $in\_dep$,
  - $\alpha = dept\_name$
  - $\beta = building, budget$
and $in\_dep$ is replaced by
  - $(\alpha \cup \beta) = (dept\_name, building, budget)$
  - $(R - (\beta - \alpha)) = (ID, name, dept\_name, salary)$
Lossless Decomposition

- BCNF Decomposition algorithm IS lossless!
  - *Only decompose until we reach BCNF, and no farther*

- **Proof sketch:**
  - When we decompose, we get:
    - \((\alpha \cup \beta)\)
    - \((R - (\beta - \alpha))\)
  - \(\pi_{\alpha \cup \beta}(r) \bowtie \pi_{R - (\beta - \alpha)}(r) = r?\)
    - Since \(\alpha\) is a superkey in left relation, only one possible value for \(\beta\) in each joined tuple!

---

Example

- \(R = (A, B, C)\)
  - \(F = (A \rightarrow B, B \rightarrow C)\)
- \(R_1 = (A, B), \ R_2 = (B, C)\)
  - Lossless-join decomposition:
    - \(R_1 \cap R_2 = \{B\} \text{ and } B \rightarrow BC\)
    - Dependency preserving
- \(R_1 = (A, B), \ R_2 = (A, C)\)
  - Lossless-join decomposition:
    - \(R_1 \cap R_2 = \{A\} \text{ and } A \rightarrow AB\)
    - Not dependency preserving
      (cannot check \(B \rightarrow C\) without computing \(R_1 \bowtie R_2)\)
3NF

One FD structure causes problems:
- If you decompose, you can’t check all the FD’s only in the decomposed relations.
- If you don’t decompose, you violate BCNF.

Abstractly: \( AB \rightarrow C \) and \( C \rightarrow B \).

Example 1: \( \text{title city} \rightarrow \text{theatre} \) and \( \text{theatre} \rightarrow \text{city} \).

Example 2: \( \text{street city} \rightarrow \text{zip} \), \( \text{zip} \rightarrow \text{city} \).

Keys: \( \{A, B\} \) and \( \{A, C\} \), but \( C \rightarrow B \) has a left side that is not a superkey.
- Suggests decomposition into \( BC \) and \( AC \).
  - But you can’t check the FD \( AB \rightarrow C \) in only these relations.

“Elegant” Workaround

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

- Consider a schema:
  \( dept\_advisor(s\_ID, i\_ID, dept\_name) \)

- With function dependencies:
  \( i\_ID \rightarrow dept\_name \)
  \( s\_ID, dept\_name \rightarrow i\_ID \)

- Two candidate keys = \{s\_ID, dept\_name\}, \{s\_ID, i\_ID\} 

- We have seen before that \( dept\_advisor \) is not in BCNF

- \( R \), however, is in 3NF
  - \( s\_ID, dept\_name \) is a superkey
  - \( i\_ID \rightarrow dept\_name \) and \( i\_ID \) is NOT a superkey, but:
    - \( \{dept\_name\} - \{i\_ID\} = \{dept\_name\} \) and
    - \( dept\_name \) is contained in a candidate key

Example

\[ A = street, B = city, C = zip. \]

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

<table>
<thead>
<tr>
<th>city</th>
<th>zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cambridge</td>
<td>02138</td>
</tr>
<tr>
<td>Cambridge</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>
Redundancy in 3NF

- Consider the schema $R$ below, which is in 3NF
  
  - $R = (J, K, L)$
  - $F = \{JK \rightarrow L, L \rightarrow K\}$
  - And an instance table:

<table>
<thead>
<tr>
<th>J</th>
<th>L</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>j1</td>
<td>l1</td>
<td>k1</td>
</tr>
<tr>
<td>j2</td>
<td>l1</td>
<td>k2</td>
</tr>
<tr>
<td>j3</td>
<td>l1</td>
<td>k2</td>
</tr>
<tr>
<td>null</td>
<td>l2</td>
<td>k2</td>
</tr>
</tbody>
</table>

- What is wrong with the table?
  
  - Repetition of information
  - Need to use null values (e.g., to represent the relationship $l_2, k_2$ where there is no corresponding value for $J$)

What 3NF Gives You

There are two important properties of a decomposition:

1. We should be able to recover from the decomposed relations the data of the original.
   - Recovery involves projection and join, which we shall defer until we've discussed relational algebra.

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.
   - 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.
3NF Synthesis

- Given a canonical cover $F_C$ for $F$
- Schema $S = \emptyset$
- $\forall A \rightarrow B \in F_C$
  - If there is no $R_i \in S$ such that $AB \subseteq R_i$
    - $S = S + AB$
- If there is no $R_i \in S$ containing a candidate key for $R$
  - $S = S + ($any candidate key for $R$)