Schema Refinement and Normal Forms

Yanlei Diao
UMass Amherst
Consider an Example

- Consider relation obtained from **Hourly_Emps**:
  - Hourly_Emps\((ssn, name, lot, rating, hrly\_wages, hrs\_worked)\)
  - Denote the schema by listing all its attributes: SNLRWH
Example (Contd.)

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>L</th>
<th>R</th>
<th>W</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-22-3666</td>
<td>Attishoo</td>
<td>48</td>
<td>8</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>231-31-5368</td>
<td>Smiley</td>
<td>22</td>
<td>8</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>131-24-3650</td>
<td>Smethurst</td>
<td>35</td>
<td>5</td>
<td>7</td>
<td>30</td>
</tr>
<tr>
<td>434-26-3751</td>
<td>Guldu</td>
<td>35</td>
<td>5</td>
<td>7</td>
<td>32</td>
</tr>
<tr>
<td>612-67-4134</td>
<td>Madayan</td>
<td>35</td>
<td>8</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>

- Rating (R) determines hourly wages (W):
  - **Redundant storage**
  - **Update**: Can we change W in just the 1st tuple of rating 8?
  - **Insertion**: Insert an employee without knowing the hourly wage for his rating? Insert the hourly wage for rating 10 with no employee?
  - **Deletion**: Delete all employees with rating 5.
**Will Two Smaller Tables be Better?**

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>L</th>
<th>R</th>
<th>W</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-22-3666</td>
<td>Attishoo</td>
<td>48</td>
<td>8</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>231-31-5368</td>
<td>Smiley</td>
<td>22</td>
<td>8</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>131-24-3650</td>
<td>Smethurst</td>
<td>35</td>
<td>5</td>
<td>7</td>
<td>30</td>
</tr>
<tr>
<td>434-26-3751</td>
<td>Guldu</td>
<td>35</td>
<td>5</td>
<td>7</td>
<td>32</td>
</tr>
<tr>
<td>612-67-4134</td>
<td>Madayan</td>
<td>35</td>
<td>8</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>

**Hourly_Emps2**

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>L</th>
<th>R</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-22-3666</td>
<td>Attishoo</td>
<td>48</td>
<td>8</td>
<td>40</td>
</tr>
<tr>
<td>231-31-5368</td>
<td>Smiley</td>
<td>22</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>131-24-3650</td>
<td>Smethurst</td>
<td>35</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>434-26-3751</td>
<td>Guldu</td>
<td>35</td>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>612-67-4134</td>
<td>Madayan</td>
<td>35</td>
<td>8</td>
<td>40</td>
</tr>
</tbody>
</table>

**Wages**

<table>
<thead>
<tr>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>
The Evils of Redundancy

- Redundant storage causes several operation anomalies:
  - Insert/delete/update anomalies

- Functional dependencies, a new type of integrity constraint, can be used to identify schemas with such problems.
  - IC’s we have seen: attribute constraints, key constraints, foreign key constraints, general constraints
  - A new type of IC: functional dependencies
Functional Dependencies (FDs)

- A **functional dependency** $X \rightarrow Y$ holds over relation $R$ if:
  - $X$ and $Y$ are two sets of attributes of $R$;
  - $\forall$ allowable instance $r$ of $R$:
    \[ t_1 \in r, t_2 \in r, \pi_X(t_1) = \pi_X(t_2) \text{ implies } \pi_Y(t_1) = \pi_Y(t_2) \]
- An FD holds for **all** allowable instances of a schema.
- Key constraint is a special form of FD:
  - $K$ is a candidate key for $R$ means that $K \rightarrow R$.
  - $K \rightarrow R$ does not require $K$ to be *minimal*!
**FDs in the Hourly_Emps Example**

- **Hourly_Emps**\((ssn, name, office, rating, hrly_wages, hrs_worked)\)
  - Denoted by SNLRWH

- Some FDs on Hourly_Emps:
  - \(ssn\) is the key:  \(S \rightarrow SNLRWH\)
  - \(rating\) determines \(hrly_wages\):  \(R \rightarrow W\)
Reasoning About FDs

- Given some FDs, we can usually infer additional FDs:
  - \( \text{ssn} \rightarrow \text{did}, \text{did} \rightarrow \text{building} \) implies \( \text{ssn} \rightarrow \text{building} \)

- Given a set of FDs \( F \), \textit{closure of } \( F \) (\( F^+ \)) is the set of all FDs that are implied by \( F \).
  - All FDs in \( F^+ \) hold over the relation \( R \).
Axioms and Rules

- Armstrong’s Axioms ($X, Y, Z$ are sets of attributes):
  - **Reflexivity**: If $X \subseteq Y$, then $Y \rightarrow X$
  - **Augmentation**: If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any $Z$
  - **Transitivity**: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

- A few additional rules (that follow from AA):
  - **Union**: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
  - **Decomposition**: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$

- Computing the closure $F^+$ using the axioms/rules:
  - Compute for all FD’s.
  - Size of closure is exponential in number of attrs!
Attribute Closure

- What if we just want to check if a given FD $X \rightarrow Y$ is in $F^+$?
- Simple algorithm for attribute closure $X^+$:
  - $X^+ := \{X\}$
  - DO if there is $U \rightarrow V$ in $F$, s.t. $U \subseteq X^+$, then $X^+ = X^+ \cup V$
    UNTIL no change
- Check if a given FD $X \rightarrow Y$ is in $F^+$:
  - Simply check if $Y \subseteq X^+$.

Does $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D \rightarrow E\}$ imply $A \rightarrow E$?
- Is $A \rightarrow E$ in the closure $F^+$?
- Equivalently, is $E$ in $A^+$?
Normal Forms

- **Role of FDs in detecting redundancy**: R(A,B,C)
  - *No FDs hold*: No redundancy here.
  - *Given A → B*: Two tuples have the same A value will have the same B value!

- **Normal forms**: If a reln does not have certain kinds of FDs, certain *redundancy-related problems* are known not to occur.
Boyce-Codd Normal Form (BCNF)

- Rewrite every FD in the form of $X \rightarrow A$, $X$ is a set of attributes, $A$ is a **single** attribute
  - Use the decomposition rule

- Reln $R$ with FDs $F$ is in **BCNF** if $\forall X \rightarrow A$ in $F^+$:
  1. $A \in X$ (called a **trivial** FD), or
  2. $X$ is a **superkey** (i.e., contains a key) for $R$.

- In **BCNF**, the only non-trivial FDs are key constraints!
Can we infer the value marked by ‘?’?

- If \( X \rightarrow A \), then the relation is not in BCNF
- A reln in BCNF can’t have \( X \rightarrow A \)

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y_1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>y_2</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>

Relation in BCNF:

- Every field of every tuple records information that can’t be inferred using FD’s from other fields.
- *No redundancy can be detected using FDs!*