Database Systems I

Design Theory (2)

Instructor: Ouldooz Baghban Karimi

CMPT 354 - Summer 2019
Closure of Attributes

\{A_1, A_2, \ldots, A_n\} is a set of attributes and \(S\) is a Set of FDs.

The **closure** of \(\{A_1, A_2, \ldots, A_n\}\) under the FDs in \(S\) is the set of attributes \(B\) such that every relation that satisfies all the FDs in set \(S\) also satisfies \(A_1 A_2 \ldots A_n \rightarrow B\)

- We denote the closure of a set of attributes \(A_1 A_2 \ldots A_n\) by \(\{A_1, A_2, \ldots, A_n\}^+\)
Algorithm: Closure of Attributes

Start with $X = \{A_1, \ldots, A_n\}$ and set of FDs $F$

Repeat until $X$ does not change; do:

if $B_1 \ldots B_m \rightarrow C$ is in $F$

and $\{B_1, \ldots, B_m\} \subseteq X$

then add $C$ to $X$

Return $X$ as $X^+$
Example

\[
\text{Student}(\text{ssn, sname, address, gpa, schcode, schname, schcity, priority})
\]

\[
S = \begin{cases}
\text{ssn} \rightarrow \text{sname, address, gpa} \\
\text{schcode} \rightarrow \text{schname, schcity} \\
\text{gpa} \rightarrow \text{priority}
\end{cases}
\]

\[
\{\text{ssn, schcode}\}^+ = \{\text{ssn, schcode}, \}
\]
Example

$R(A, B, C, D, E, F)$

\[
\{ 
\begin{align*}
A, B & \rightarrow C \\
A, D & \rightarrow E \\
B & \rightarrow D \\
A, F & \rightarrow B \\
\end{align*}
\}
\]

Compute $\{A, B\}^+ = \{A, B, \}$

Compute $\{A, F\}^+ = \{A, F, \}$
Armstrong’s Axioms

• Set of rules to derive any FD that follows from a given set (Inference Rules)

• Reflexivity. If \( \{B_1, B_2, \ldots, B_m\} \subseteq \{A_1, A_2, \ldots, A_n\} \) then \( A_1 A_2 \ldots A_n \Rightarrow B_1 B_2 \ldots B_m \) (trivial FDs)

• Augmentation. If \( A_1 A_2 \ldots A_n \Rightarrow B_1 B_2 \ldots B_m \), then \( A_1 A_2 \ldots A_n C_1 C_2 \ldots C_k \Rightarrow B_1 B_2 \ldots B_m C_1 C_2 \ldots C_k \) for any attributes \( A_1 A_2 \ldots A_n \Rightarrow B_1 B_2 \ldots B_m \)

• Transitivity. If \( A_1 A_2 \ldots A_n \Rightarrow B_1 B_2 \ldots B_m \) and \( B_1 B_2 \ldots B_m \Rightarrow C_1 C_2 \ldots C_k \) then \( A_1 A_2 \ldots A_n \Rightarrow C_1 C_2 \ldots C_k \)
Armstrong’s Axioms

• **Sound** and Complete Rules

• **Sound**: Do not produce FDs not in the closure

• **Complete**: Do not miss any FDs in the closure
Closing Set of FDs

• A choice of which FDs to represent the full set of FDs for a relation

• Given set of FDs $S$, any set of FDs equivalent to $S$ is a basis for $S$

• **Minimal Basis** for a relation is basis $B$ that satisfies
  • All the FDs in $B$ have singleton right sides
  • If any FD is removed from $B$ the result is no longer a basis
  • If for any FD in $B$ we remove one or more attributes from the left side of $F$, the result is no longer a basis
Closure of FDs

• Can we compute the FD closure?
  Yes. Slowly. Expensive: Exponential in the number of attributes

• Can we check if $X \rightarrow Y$ is in the closure of $F$?
  $X^+ = \text{attribute closure of } X$
  Expand $X$ using Axioms
  Check if $Y$ is implied in the attribute closure
Example

- \( F = \{ A \rightarrow B, \ B \rightarrow C, \ CB \rightarrow E \} \)
- Is \( A \rightarrow E \) in the closure?

\[
\begin{align*}
A \rightarrow B & \quad \text{given} \\
A \rightarrow AB & \quad \text{Augmentation A} \\
A \rightarrow BB & \quad \text{Apply} \quad A \rightarrow B \quad \text{(Transitivity)} \\
A \rightarrow BC & \quad \text{Apply} \quad B \rightarrow C \quad \text{(Transitivity)} \\
A \rightarrow E & \quad \text{Apply} \quad BC \rightarrow E \quad \text{(Transitivity)}
\end{align*}
\]
Example

Find all FDs implied by

\{ A, B \rightarrow C, A, D \rightarrow B, B \rightarrow D \}

Requirements
1. Non-trivial FD
2. The right-hand side contains a single attribute

Step 1: Compute $X^+$, for every set of attributes $X$

Step 2: Enumerate all FDs $X \rightarrow Y$, so that $Y \subseteq X^+$ and $X \cap Y = \emptyset$
**Projecting FDs**

\[ R_1 = \pi_L(R) \]

What FDs hold in \( R_1 \)?

Projection of functional dependencies from S that

- Follow from S
- Involve only attributes of \( R_1 \)

• To find the set \( T \) of FDs that hold in \( R_1 \)
  - For each set of attributes \( X \) that is a subset of the attributes of \( R_1 \), compute \( X^+ \). Add to \( T \) all non-trivial FDs \( X \rightarrow A \) such that \( A \) is both in \( X^+ \) and an attribute of \( R \)
  - Construct a minimal basis by modifying \( T \) with repeating the following steps
    - Remove any FD following from other FDs
    - Let \( Y \rightarrow B \) be an FD in \( T \), with at least two attributes in \( Y \)
      let \( Z \) be \( Y \) with one of its attributes removed
      If \( Z \rightarrow B \) follows from the FDs in \( T \) (including \( Y \rightarrow B \)), then replace \( Y \rightarrow B \) by \( Z \rightarrow B \)
Example

\[ R(A, B, C, D) \]
\[
\begin{align*}
& A \rightarrow B \\
& B \rightarrow C \\
& C \rightarrow D
\end{align*}
\]

\[ \pi_{A, C, D}(R) = R_1(A, C, D) \]

Since \{A\}^+ includes all attributes of \(R_1\), there is no point in considering any superset of \{A\}

\[
\begin{align*}
& A \rightarrow C \\
& C \rightarrow D
\end{align*}
\]
Design of Relations

• Remember our initial example. Is this a better design now?

<table>
<thead>
<tr>
<th>Student</th>
<th>Course</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>354</td>
<td>AQ3149</td>
</tr>
<tr>
<td>Mary</td>
<td>354</td>
<td>AQ3149</td>
</tr>
<tr>
<td>Sam</td>
<td>354</td>
<td>AQ3149</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>354</td>
</tr>
<tr>
<td>Mary</td>
<td>354</td>
</tr>
<tr>
<td>Sam</td>
<td>354</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>354</td>
<td>AQ3149</td>
</tr>
<tr>
<td>454</td>
<td>TASC1 9204</td>
</tr>
<tr>
<td>371</td>
<td>WMC 3520</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

• Argue
  • Redundancy
  • Update anomaly
  • Delete anomaly
  • Insert anomaly

Question: How do we find this decomposition?
Anomalies

• When we try to cram too much into a single relation

  • **Redundancy**: Information may be repeated unnecessarily in several tuples

  • **Update Anomalies**: We may change information in one tuple but leave the same information unchanged in another

  • **Deletion Anomalies**: If a set of values becomes empty, we may lose other information as a side effect
Decomposing Relations

• The accepted way to eliminate these anomalies is to decompose relations

• Given a relation $R(A_1, A_2, \ldots, A_n)$, we may decompose $R$ into two relations $S(B_1, B_2, \ldots, B_m)$ and $T(C_1, C_2, \ldots, C_k)$ such that

  • $\{A_1, A_2, \ldots, A_n\} = \{B_1, B_2, \ldots, B_m\} \cup \{C_1, C_2, \ldots, C_k\}$

  • $S = \pi_{B_1, B_2, \ldots, B_m}(R)$

  • $T = \pi_{C_1, C_2, \ldots, C_k}(R)$
<table>
<thead>
<tr>
<th>title</th>
<th>year</th>
<th>length</th>
<th>genre</th>
<th>studioName</th>
<th>starName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>sciFi</td>
<td>Fox</td>
<td>Carrie Fisher</td>
</tr>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>sciFi</td>
<td>Fox</td>
<td>Mark Hamill</td>
</tr>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>sciFi</td>
<td>Fox</td>
<td>Harrison Ford</td>
</tr>
<tr>
<td>Gone With the Wind</td>
<td>1939</td>
<td>231</td>
<td>drama</td>
<td>MGM</td>
<td>Vivian Leigh</td>
</tr>
<tr>
<td>Wayne’s World</td>
<td>1992</td>
<td>95</td>
<td>comedy</td>
<td>Paramount</td>
<td>Dana Carvey</td>
</tr>
<tr>
<td>Wayne’s World</td>
<td>1992</td>
<td>95</td>
<td>comedy</td>
<td>Paramount</td>
<td>Mike Meyers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>year</th>
<th>length</th>
<th>genre</th>
<th>studioName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>sciFi</td>
<td>Fox</td>
</tr>
<tr>
<td>Gone With the Wind</td>
<td>1939</td>
<td>231</td>
<td>drama</td>
<td>MGM</td>
</tr>
<tr>
<td>Wayne’s World</td>
<td>1992</td>
<td>95</td>
<td>comedy</td>
<td>Paramount</td>
</tr>
</tbody>
</table>
Lossless Decomposition

• A decomposition $R$ to $(R_1, R_2)$ is **lossless** if $R = R_1 \Join R_2$
• Means we do get $R$ back: decomposition has a lossless join

\[
R(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_p)
\]

\[
R_1(A_1, \ldots, A_n, B_1, \ldots, B_m)
\]

\[
R_2(A_1, \ldots, A_n, C_1, \ldots, C_p)
\]

• If \{A_1, \ldots, A_n\} $\rightarrow$ \{B_1, \ldots, B_m\}, then the decomposition is **lossless**
Example

- Neither of $A \rightarrow B$ and $B \rightarrow A$ hold
- If we try to reconstruct $R$ by the natural join of the projected relations, we get two additional bogus tuples, $(1,2,5)$ and $(4,2,3)$
Decomposing Relations

• First Normal Form (1NF): All tables are flat

• Second Normal Form: Disused!

• Boyce-Codd Normal Form (BCNF): No bad FDs

• Third, Fourth, and Fifth Normal Forms: Further Reading!
Boyce-Codd Normal Form

• A simple condition under which the anomalies can be guaranteed not to exist

• A relation $R$ is in $BCNF$ if and only if
  • Whenever there is a nontrivial FD $A_1A_2…A_n \rightarrow B_1B_2…B_m$ for $R$, it is the case that 
    $\{A_1, A_2,…, A_n\}$ is a superkey for $R$

Means: The left side of every nontrivial FD must be a superkey!

Equivalently: $\forall$ sets of attributes $X$, either ($X^+ = X$) or ($X^+ = \text{all attributes}$)
BCNF Example

- Any two-attribute relation is in BCNF
  
  • $A \rightarrow B$ holds but $B \rightarrow A$ does not hold. $A$ will be key, and every non-trivial FD will have $A$ on the left
  
  • $B \rightarrow A$ holds but $A \rightarrow B$ does not hold. $B$ will be key, and every non-trivial FD will have $B$ on the left
  
  • Both $B \rightarrow A$ and $A \rightarrow B$ hold. Both $A$ and $B$ are keys, and every non-trivial FD will have at least one of $A$ or $B$ on the left
  
- Therefore, there can be no BCNF violation
BCNF Example

\{\text{title, year, studioName, president, presAddr}\}

\{
\text{title, year} \rightarrow \text{studioName} \\
\text{studioName} \rightarrow \text{president} \\
\text{president} \rightarrow \text{presAddr}
\}

\{\text{title, year}\} \text{ is the only key for this relation}

• Resulting Schemas:
  \{\text{title, year, studioName}\}
  \{\text{studioName, president}\}
  \{\text{president, presAddr}\}
Decomposition into BCNF

BCNFDecomp(R):

Find a non-trivial bad FD: \( X \rightarrow Y \)

if (not found) then Return \( R \)

Split \( R \) into \( R_1=X^+ \) and \( R_2=X+\{\text{rest of attributes}\} \)

Return BCNFDecomp(\( R_1 \)), BCNFDecomp(\( R_2 \))
Decomposition Properties

• We like a decomposition to have the following properties

  • **Elimination of Anomalies** by decomposition

  • **Recoverability of Information.** Can we recover the original relation from the tuples in its decomposition?

  • **Preservation of Dependencies.** If we check the projected FDs in the relations of the decomposition, can we be sure that when we reconstruct the original relation from the decomposition by joining, the result will satisfy the original FDs?

• **BCNF gives us (1) and (2), but does not necessarily give us all three**
Third Normal Form

• A relation \( R \) is in **Third Normal Form (3NF)** if

  • Whenever \( A_1 A_2 \ldots A_n \rightarrow B_1 B_2 \ldots B_m \) is a nontrivial FD either
    \( \{A_1, A_2, \ldots, A_n\} \) is a superkey
    or those of \( B_1 B_2 \ldots B_m \) that are not among \( A \)'s are each a member of some key (not necessarily the same key)

• An attribute that is a member of some key is often said to be **prime**

• 3NF is a relaxation of BCNF with the prime clause
I have used materials from the following resources in preparation of this course:

- **Database Systems: The Complete Book**
- Database Systems (Kifer, Bernstein, Lewis)
- Course offerings
  - W 4111 (Eugene Wu - Columbia): [https://w4111.github.io/](https://w4111.github.io/)
  - CS 186 (Joe Hellerstein - Berkeley): [https://sites.google.com/site/cs186fall17/](https://sites.google.com/site/cs186fall17/)