Relational Database Design Theory
Introduction to Databases
CompSci 316 Spring 2017

Announcements (Wed. Feb. 1)
• Homework #1 due Monday 02/06 (11:59 pm)

Review: Motivation

• redundancy is bad
  • user name is recorded multiple times
  • Leads to update, insertion, deletion anomalies
  • Have a systematic approach to detecting and removing redundancy in designs
  • Dependencies, decompositions, and normal forms

Review: Functional dependencies

• A functional dependency (FD) $X \rightarrow Y$
  • $X$ and $Y$ are sets of attributes in a relation $R$
  • whenever two tuples in $R$ agree on all the attributes in $X$, they must also agree on all attributes in $Y$

Review: Attribute closure

• Given
  • $R$
    • a set of FD's $F$ that hold in $R$, and
    • a set of attributes $Z$ in $R$

  • The closure of $Z$ (denoted $Z^+$) with respect to $F$ is the set of all attributes $A_1, A_2, ...$ functionally determined by $Z$
    • that is, $Z \rightarrow A_1, A_2, ...$

Review: Superkeys and Keys

Given a relation $R$ and set of FD’s $F$

• Compute $K^+$ with respect to $F$
  • If $K^+$ contains all the attributes of $R$, $K$ is a super key
  • If $K$ is also minimal (no proper subset is a superkey), $K$ is a key
Review: Motivation of BCNF decomposition

- Non-key FDs cause redundancy

\[
\begin{align*}
\text{X} & \rightarrow \text{Y} \\
\text{a} & \rightarrow \text{b} \\
\text{a} & \rightarrow \text{c}
\end{align*}
\]

Here \( X \rightarrow Y \)

Detect such FDs where \( X \) is not a superkey, and decompose into two relations

1. One relation gets \( X, Y \) (\( X \) is a superkey; these make it lossless)
2. The other one gets \( X, Z \) (in general \( Z \) = everything else)

- Multivalued dependencies

\[
\begin{align*}
\text{X} \rightarrow \text{Y} \\
\text{a} & \rightarrow \text{b} \\
\text{a} & \rightarrow \text{c}
\end{align*}
\]

Check yourself! If in one of the two new relations, the common join attributes is a superkey, then lossless

- Complete MVD + FD rules

- Check your knowledge!
An elegant solution: chase

• Given a set of FD’s and MVD’s \( \mathcal{D} \), does another dependency \( d \) (FD or MVD) follow from \( \mathcal{D} \)?

• Procedure
  • Start with the premise of \( d \), and treat them as “seed” tuples in a relation
  • Apply the given dependencies in \( \mathcal{D} \) 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 \( A \rightarrow B \) and \( B \rightarrow C \) imply that \( A \rightarrow C \)?

Another proof by chase

• In \( R(A, B, C, D) \), does \( A \rightarrow B \) and \( B \rightarrow C \) imply that \( A \rightarrow C \)?

Counterexample by chase

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

In general, with both MVD’s and FD’s, chase can generate both new tuples and new equalities

4NF decomposition algorithm

• Find a 4NF violation
  • A non-trivial MVD \( X \rightarrow Y \) in \( R \) where \( X \) is not a superkey
  • Decompose \( R \) into \( R_1 \) and \( R_2 \), where
    • \( R_1 \) has attributes \( X \cup Y \)
    • \( R_2 \) has attributes \( X \cup Z \) (where \( Z \) contains \( R \) 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

User (uid, gid, place)
4NF violation: uid → gid

Member (uid, gid)
4NF

Visited (uid, place)
4NF

Summary

- Philosophy behind BCNF, 4NF: Data should depend on the key, the whole key, and nothing but the key!
  - You could have multiple keys though

- Other normal forms
  - 3NF: More relaxed than BCNF; will not remove redundancy if doing so makes FDs harder to enforce
  - 2NF: Slightly more relaxed than 3NF
  - 1NF: All column values must be atomic

Next: Project Mixer!