Databases
Lecture 8

Timothy G. Griffin

Computer Laboratory
University of Cambridge, UK

Databases, Lent 2009

Lecture 08: Multivalued Dependencies

Outline

- Multivalued Dependencies
- Fourth Normal Form (4NF)
- General integrity Constraints

Another look at Heath's Rule

Given $R(\mathbf{Z}, \mathbf{W}, \mathbf{Y})$ with FDs F

If $\mathbf{Z} \rightarrow \mathbf{W} \in F^+$, the

$$R = \pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R)$$

What about an implication in the other direction? That is, suppose we have

$$R = \pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R).$$

Q Can we conclude anything about FDs on R? In particular, is it true that $\mathbf{Z} \rightarrow \mathbf{W}$ holds?

A No!

We just need one counter example ...

$$R \quad = \quad \pi_{A,B}(R) \quad \bowtie \quad \pi_{A,C}(R)$$

A	B	C
a	b_1	c_1
a	b_2	c_2
a	b_1	c_2
a	b_2	c_1

A	B
a	b_1
a	b_2

A	C
a	c_1
a	c_2

Clearly $A \rightarrow B$ is not an FD of R.

A concrete example

course_name	lecturer	text
Databases	Tim	Ullman and Widom
Databases	Fatima	Date
Databases	Tim	Date
Databases	Fatima	Ullman and Widom

Assuming that texts and lecturers are assigned to courses independently, then a better representation would in two tables:

course_name	lecturer
Databases	Tim
Databases	Fatima

course_name	text
Databases	Ullman and Widom
Databases	Date

Time for a definition!

Multivalued Dependencies (MVDs)

Let $R(\mathbf{Z}, \mathbf{W}, \mathbf{Y})$ be a relational schema. A multivalued dependency, denoted $\mathbf{Z} \twoheadrightarrow \mathbf{W}$, holds if whenever t and u are two records that agree on the attributes of $\mathbf{Z}$, then there must be some tuple v such that

1. v agrees with both t and u on the attributes of $\mathbf{Z}$,
2. v agrees with t on the attributes of $\mathbf{W}$,
3. v agrees with u on the attributes of $\mathbf{Y}$.

A few observations

Note 1

Every functional dependency is multivalued dependency,

$$(\mathbf{Z} \rightarrow \mathbf{W}) \implies (\mathbf{Z} \twoheadrightarrow \mathbf{W}).$$

To see this, just let $v = u$ in the above definition.

Note 2

Let $R(\mathbf{Z}, \mathbf{W}, \mathbf{Y})$ be a relational schema, then

$$(\mathbf{Z} \twoheadrightarrow \mathbf{W}) \iff (\mathbf{Z} \twoheadrightarrow \mathbf{Y}),$$

by symmetry of the definition.

MVDs and lossless-join decompositions

Fun Fun Fact

Let $R(\mathbf{Z}, \mathbf{W}, \mathbf{Y})$ be a relational schema. The decomposition $R_1(\mathbf{Z}, \mathbf{W})$, $R_2(\mathbf{Z}, \mathbf{Y})$ is a lossless-join decomposition of R if and only if the MVD $\mathbf{Z} \twoheadrightarrow \mathbf{W}$ holds.

Proof of Fun Fun Fact

Proof of $(\mathbf{Z} \twoheadrightarrow \mathbf{W}) \implies R = \pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R)$

- Suppose $\mathbf{Z} \twoheadrightarrow \mathbf{W}$.
- We know (from proof of Heath's rule) that $R \subseteq \pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R)$. So we only need to show $\pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R) \subseteq R$.
- Suppose $r \in \pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R)$.
- So there must be a $t \in R$ and $u \in R$ with $\{r\} = \pi_{\mathbf{Z},\mathbf{W}}(\{t\}) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(\{u\})$.
- In other words, there must be a $t \in R$ and $u \in R$ with $t.\mathbf{Z} = u.\mathbf{Z}$.
- So the MVD tells us that then there must be some tuple $v \in R$ such that
  1. v agrees with both t and u on the attributes of $\mathbf{Z}$,
  2. v agrees with t on the attributes of $\mathbf{W}$,
  3. v agrees with u on the attributes of $\mathbf{Y}$.
- This v must be the same as r, so $r \in R$.

Proof of Fun Fun Fact (cont.)

Proof of $R = \pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R) \implies (\mathbf{Z} \twoheadrightarrow \mathbf{W})$

- Suppose $R = \pi_{\mathbf{Z},\mathbf{W}}(R) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(R)$.
- Let t and u be any records in R with $t.\mathbf{Z} = u.\mathbf{Z}$.
- Let v be defined by $\{v\} = \pi_{\mathbf{Z},\mathbf{W}}(\{t\}) \bowtie \pi_{\mathbf{Z},\mathbf{Y}}(\{u\})$ (and we know $v \in R$ by the assumption).
- Note that by construction we have
  1. $v.\mathbf{Z} = t.\mathbf{Z} = u.\mathbf{Z}$,
  2. $v.\mathbf{W} = t.\mathbf{W}$,
  3. $v.\mathbf{Y} = u.\mathbf{Y}$.
- Therefore, $\mathbf{Z} \twoheadrightarrow \mathbf{W}$ holds.

Fourth Normal Form

Trivial MVD

The MVD $\mathbf{Z} \twoheadrightarrow \mathbf{W}$ is trivial for relational schema $R(\mathbf{Z}, \mathbf{W}, \mathbf{Y})$ if

1. $\mathbf{Z} \cap \mathbf{W} \neq \{\}$, or
2. $\mathbf{Y} = \{\}$.

4NF

A relational schema $R(\mathbf{Z}, \mathbf{W}, \mathbf{Y})$ is in 4NF if for every MVD $\mathbf{Z} \twoheadrightarrow \mathbf{W}$ either

- $\mathbf{Z} \twoheadrightarrow \mathbf{W}$ is a trivial MVD, or
- $\mathbf{Z}$ is a superkey for R.

Note : 4NF $\subset$ BCNF $\subset$ 3NF $\subset$ 2NF

General Decomposition Method Revisited

GDM++

1. Understand your FDs and MVDs F (compute F^+),
2. find $R(\mathbf{X}) = R(\mathbf{Z}, \mathbf{W}, \mathbf{Y})$ (sets $\mathbf{Z}$, $\mathbf{W}$ and $\mathbf{Y}$ are disjoint) with either FD $\mathbf{Z} \rightarrow \mathbf{W} \in F^+$ or MVD $\mathbf{Z} \twoheadrightarrow \mathbf{W} \in F^+$ violating a condition of desired NF,
3. split R into two tables $R_1(\mathbf{Z}, \mathbf{W})$ and $R_2(\mathbf{Z}, \mathbf{Y})$
4. wash, rinse, repeat

Summary

We always want the lossless-join property. What are our options?

	3NF	BCNF	4NF
Preserves FDs	Yes	Maybe	Maybe
Preserves MVDs	Maybe	Maybe	Maybe
Eliminates FD-redundancy	Maybe	Yes	Yes
Eliminates MVD-redundancy	No	No	Yes

General integrity constraints

- Suppose that C is some constraint we would like to enforce on our database.
- Let $Q_{\neg C}$ be a query that captures all violations of C.
- Enforce (somehow) that the assertion that is always $Q_{\neg C}$ empty.

Example

- $C = \mathbf{Z} \rightarrow \mathbf{W}$, and FD that was not preserved for relation $R(\mathbf{X})$,
- Let Q_R be a join that reconstructs R,
- Let Q'_R be this query with $\mathbf{X} \mapsto \mathbf{X}'$ and
- $Q_{\neg C} = \sigma_{\mathbf{W} \neq \mathbf{W}'}(\sigma_{\mathbf{Z}=\mathbf{Z}'}(Q_R \times Q'_R))$

Assertions in SQL

```
create view C_violations as ....

create assertion check_C
       check not (exists C_violations)
```