Databases 2011
Lectures 05 — 07

Timothy G. Griffin

Computer Laboratory
University of Cambridge, UK

Databases, Easter 2011

Lecture 05 : Functional Dependencies (FDs)

Outline

- ER is for top-down and informal (but rigorous) design
- FDs are used for bottom-up and formal design and analysis
- update anomalies
- Reasoning about Functional Dependencies
- Heath's rule

Update anomalies

Big Table

sid	name	college	course	part	term_name
yy88	Yoni	New Hall	Algorithms I	IA	Easter
uu99	Uri	King's	Algorithms I	IA	Easter
bb44	Bin	New Hall	Databases	IB	Lent
bb44	Bin	New Hall	Algorithms II	IB	Michaelmas
zz70	Zip	Trinity	Databases	IB	Lent
zz70	Zip	Trinity	Algorithms II	IB	Michaelmas

- How can we tell if an insert record is consistent with current records?
- Can we record data about a course before students enroll?
- Will we wipe out information about a college when last student associated with the college is deleted?

Redundancy implies more locking ...

... at least for correct transactions!

Big Table

sid	name	college	course	part	term_name
yy88	Yoni	New Hall	Algorithms I	IA	Easter
uu99	Uri	King's	Algorithms I	IA	Easter
bb44	Bin	New Hall	Databases	IB	Lent
bb44	Bin	New Hall	Algorithms II	IB	Michaelmas
zz70	Zip	Trinity	Databases	IB	Lent
zz70	Zip	Trinity	Algorithms II	IB	Michaelmas

- Change New Hall to Murray Edwards College
  - Conceptually simple update
  - May require locking entire table.

Redundancy is the root of (almost) all database evils

- It may not be obvious, but redundancy is also the cause of update anomalies.
- By redundancy we do not mean that some values occur many times in the database!
  - A foreign key value may be have millions of copies!
- But then, what do we mean?

Functional Dependency

Functional Dependency (FD)

Let $R(\mathbf{X})$ be a relational schema and $\mathbf{Y} \subseteq \mathbf{X}$, $\mathbf{Z} \subseteq \mathbf{X}$ be two attribute sets. We say $\mathbf{Y}$ functionally determines $\mathbf{Z}$, written $\mathbf{Y} \to \mathbf{Z}$, if for any two tuples u and v in an instance of $R(\mathbf{X})$ we have

$$u.\mathbf{Y} = v.\mathbf{Y} \to u.\mathbf{Z} = v.\mathbf{Z}.$$

We call $\mathbf{Y} \to \mathbf{Z}$ a functional dependency.

A functional dependency is a <u>semantic</u> assertion. It represents a rule that should always hold in any instance of schema $R(\mathbf{X})$.

Example FDs

Big Table

sid	name	college	course	part	term_name
yy88	Yoni	New Hall	Algorithms I	IA	Easter
uu99	Uri	King's	Algorithms I	IA	Easter
bb44	Bin	New Hall	Databases	IB	Lent
bb44	Bin	New Hall	Algorithms II	IB	Michaelmas
zz70	Zip	Trinity	Databases	IB	Lent
zz70	Zip	Trinity	Algorithms II	IB	Michaelmas

- **sid** $\rightarrow$ **name**
- **sid** $\rightarrow$ **college**
- **course** $\rightarrow$ **part**
- **course** $\rightarrow$ **term_name**

Keys, revisited

Candidate Key

Let $R(\mathbf{X})$ be a relational schema and $\mathbf{Y} \subseteq \mathbf{X}$. $\mathbf{Y}$ is a candidate key if

1. The FD $\mathbf{Y} \rightarrow \mathbf{X}$ holds, and
2. for no proper subset $\mathbf{Z} \subset \mathbf{Y}$ does $\mathbf{Z} \rightarrow \mathbf{X}$ hold.

Prime and Non-prime attributes

An attribute A is prime for $R(\mathbf{X})$ if it is a member of some candidate key for R. Otherwise, A is non-prime.

Database redundancy roughly means the existence of non-key functional dependencies!

Closure

By soundness and completeness

$$\text{closure}(F,\ \mathbf{X}) = \{A \mid F \vdash \mathbf{X} \rightarrow A\} = \{A \mid \mathbf{X} \rightarrow A \in F^+\}$$

Claim 2 (from previous lecture)

$\mathbf{Y} \rightarrow \mathbf{W} \in F^+$ if and only if $\mathbf{W} \subseteq \text{closure}(F,\ \mathbf{Y})$.

If we had an algorithm for closure(F, $\mathbf{X}$), then we would have a (brute force!) algorithm for enumerating F^+:

F^+

- for every subset $\mathbf{Y} \subseteq \text{atts}(F)$
  - for every subset $\mathbf{Z} \subseteq \text{closure}(F,\ \mathbf{Y})$,
    - output $\mathbf{Y} \rightarrow \mathbf{Z}$

Attribute Closure Algorithm

- Input : a set of FDs F and a set of attributes $\mathbf{X}$.
- Output : $\mathbf{Y} = \text{closure}(F,\ \mathbf{X})$

1. $\mathbf{Y} := \mathbf{X}$
2. while there is some $\mathbf{S} \rightarrow \mathbf{T} \in F$ with $\mathbf{S} \subseteq \mathbf{Y}$ and $\mathbf{T} \not\subseteq \mathbf{Y}$, then $\mathbf{Y} := \mathbf{Y} \cup \mathbf{T}$.

An Example (UW1997, Exercise 3.6.1)

$R(A,\ B, C, D)$ with F made up of the FDs

$$A, B \rightarrow C$$
$$C \rightarrow D$$
$$D \rightarrow A$$

What is F^+?

Brute force!

Let's just consider all possible nonempty sets **X** — there are only 15...

Example (cont.)

$$F = \{A, B \rightarrow C,\ \ C \rightarrow D,\ \ D \rightarrow A\}$$

For the single attributes we have

- $\{A\}^+ = \{A\}$,
- $\{B\}^+ = \{B\}$,
- $\{C\}^+ = \{A,\ C,\ D\}$,
  - $\{C\} \stackrel{C \rightarrow D}{\Longrightarrow} \{C,\ D\} \stackrel{D \rightarrow A}{\Longrightarrow} \{A,\ C,\ D\}$
- $\{D\}^+ = \{A,\ D\}$
  - $\{D\} \stackrel{D \rightarrow A}{\Longrightarrow} \{A,\ D\}$

The only new dependency we get with a single attribute on the left is $C \rightarrow A$.

Example (cont.)

$$F = \{A, B \to C, \quad C \to D, \quad D \to A\}$$

Now consider pairs of attributes.

- $\{A,\ B\}^+ = \{A,\ B,\ C,\ D\}$,
  - so $A, B \to D$ is a new dependency
- $\{A,\ C\}^+ = \{A,\ C,\ D\}$,
  - so $A, C \to D$ is a new dependency
- $\{A,\ D\}^+ = \{A,\ D\}$,
  - so nothing new.
- $\{B,\ C\}^+ = \{A,\ B,\ C,\ D\}$,
  - so $B, C \to A, D$ is a new dependency
- $\{B,\ D\}^+ = \{A,\ B,\ C,\ D\}$,
  - so $B, D \to A, C$ is a new dependency
- $\{C,\ D\}^+ = \{A,\ C,\ D\}$,
  - so $C, D \to A$ is a new dependency

Example (cont.)

$$F = \{A, B \to C, \quad C \to D, \quad D \to A\}$$

For the triples of attributes:

- $\{A,\ C,\ D\}^+ = \{A,\ C,\ D\}$,
- $\{A,\ B,\ D\}^+ = \{A,\ B,\ C,\ D\}$,
  - so $A, B, D \to C$ is a new dependency
- $\{A,\ B,\ C\}^+ = \{A,\ B,\ C,\ D\}$,
  - so $A, B, C \to D$ is a new dependency
- $\{B,\ C,\ D\}^+ = \{A,\ B,\ C,\ D\}$,
  - so $B, C, D \to A$ is a new dependency

And since $\{A,\ B,\ C,\ D\}+ = \{A,\ B,\ C,\ D\}$, we get no new dependencies with four attributes.

Example (cont.)

We generated 11 new FDs:

$$\begin{array}{rcl} C & \to & A \\ A, C & \to & D \\ B, C & \to & D \\ B, D & \to & C \\ A, B, C & \to & D \\ B, C, D & \to & A \end{array} \qquad \begin{array}{rcl} A, B & \to & D \\ B, C & \to & A \\ B, D & \to & A \\ C, D & \to & A \\ A, B, D & \to & C \end{array}$$

Can you see the Key?

$\{A,\ B\}$, $\{B,\ C\}$, and $\{B,\ D\}$ are keys.

Note: this schema is already in 3NF! Why?

Semantic Closure

Notation

$$F \models \mathbf{Y} \to \mathbf{Z}$$

means that any database instance that that satisfies every FD of F, must also satisfy $\mathbf{Y} \to \mathbf{Z}$.

The semantic closure of F, denoted F^+, is defined to be

$$F^+ = \{\mathbf{Y} \to \mathbf{Z} \mid \mathbf{Y} \cup \mathbf{Z} \subseteq \text{atts}(F) and \wedge F \models \mathbf{Y} \to \mathbf{Z}\}.$$

The membership problem is to determine if $\mathbf{Y} \to \mathbf{Z} \in F^+$.

Reasoning about Functional Dependencies

We write $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$ when $\mathbf{Y} \rightarrow \mathbf{Z}$ can be derived from F via the following rules.

Armstrong's Axioms

Reflexivity If $\mathbf{Z} \subseteq \mathbf{Y}$, then $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$.

Augmentation If $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$ then $F \vdash \mathbf{Y}, \mathbf{W} \rightarrow \mathbf{Z}, \mathbf{W}$.

Transitivity If $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$ and $F \models \mathbf{Z} \rightarrow \mathbf{W}$, then $F \vdash \mathbf{Y} \rightarrow \mathbf{W}$.

Logical Closure (of a set of attributes)

Notation

$$\text{closure}(F,\ \mathbf{X}) = \{A \mid F \vdash \mathbf{X} \rightarrow A\}$$

Claim 1

If $\mathbf{Y} \rightarrow \mathbf{W} \in F$ and $\mathbf{Y} \subseteq \text{closure}(F,\ \mathbf{X})$, then $\mathbf{W} \subseteq \text{closure}(F,\ \mathbf{X})$.

Claim 2

$\mathbf{Y} \rightarrow \mathbf{W} \in F^{+}$ if and only if $\mathbf{W} \subseteq \text{closure}(F,\ \mathbf{Y})$.

Soundness and Completeness

Soundness

$$F \vdash f \implies f \in F^+$$

Completeness

$$f \in F^+ \implies F \vdash f$$

Proof of Completeness (soundness left as an exercise)

Show $\neg(F \vdash f) \implies \neg(F \models f)$:

- Suppose $\neg(F \vdash \mathbf{Y} \to \mathbf{Z})$ for $R(\mathbf{X})$.
- Let $\mathbf{Y}^+ = \text{closure}(F, \mathbf{Y})$.
- $\exists B \in \mathbf{Z}$, with $B \notin \mathbf{Y}^+$.
- Construct an instance of R with just two records, u and v, that agree on $\mathbf{Y}^+$ but not on $\mathbf{X} - \mathbf{Y}^+$.
- By construction, this instance does not satisfy $\mathbf{Y} \to \mathbf{Z}$.
- But it does satisfy F! Why?
  - let $\mathbf{S} \to \mathbf{T}$ be any FD in F, with $u.[\mathbf{S}] = v.[\mathbf{S}]$.
  - So $\mathbf{S} \subseteq \mathbf{Y}+$. and so $\mathbf{T} \subseteq \mathbf{Y}+$ by claim 1,
  - and so $u.[T] = v.[T]$

Consequences of Armstrong's Axioms

Union If $F \models \mathbf{Y} \rightarrow \mathbf{Z}$ and $F \models \mathbf{Y} \rightarrow \mathbf{W}$, then $F \models \mathbf{Y} \rightarrow \mathbf{W}, \mathbf{Z}$.

Pseudo-transitivity If $F \models \mathbf{Y} \rightarrow \mathbf{Z}$ and $F \models \mathbf{U}, \mathbf{Z} \rightarrow \mathbf{W}$, then $F \models \mathbf{Y}, \mathbf{U} \rightarrow \mathbf{W}$.

Decomposition If $F \models \mathbf{Y} \rightarrow \mathbf{Z}$ and $\mathbf{W} \subseteq \mathbf{Z}$, then $F \models \mathbf{Y} \rightarrow \mathbf{W}$.

Exercise : Prove these using Armstrong's axioms!

Proof of the Union Rule

Suppose we have

$$F \models \mathbf{Y} \rightarrow \mathbf{Z},$$
$$F \models \mathbf{Y} \rightarrow \mathbf{W}.$$

By augmentation we have

$$F \models \mathbf{Y}, \mathbf{Y} \rightarrow \mathbf{Y}, \mathbf{Z},$$

that is,

$$F \models \mathbf{Y} \rightarrow \mathbf{Y}, \mathbf{Z}.$$

Also using augmentation we obtain

$$F \models \mathbf{Y}, \mathbf{Z} \rightarrow \mathbf{W}, \mathbf{Z}.$$

Therefore, by transitivity we obtain

$$F \models \mathbf{Y} \rightarrow \mathbf{W}, \mathbf{Z}.$$

Example application of functional reasoning.

Heath's Rule

Suppose $R(A,\ B,\ C)$ is a relational schema with functional dependency $A \rightarrow B$, then

$$R = \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R).$$

Proof of Heath's Rule

We first show that $R \subseteq \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R)$.

- If $u = (a,\ b,\ c) \in R$, then $u_1 = (a,\ b) \in \pi_{A,B}(R)$ and $u_2 = (a,\ c) \in \pi_{A,C}(R)$.
- Since $\{(a,\ b)\} \bowtie_A \{(a,\ c)\} = \{(a,\ b,\ c)\}$ we know $u \in \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R)$.

In the other direction we must show $R' = \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R) \subseteq R$.

- If $u = (a,\ b,\ c) \in R'$, then there must exist tuples $u_1 = (a,\ b) \in \pi_{A,B}(R)$ and $u_2 = (a,\ c) \in \pi_{A,C}(R)$.
- This means that there must exist a $u' = (a,\ b',\ c) \in R$ such that $u_2 = \pi_{A,C}(\{(a,\ b',\ c)\})$.
- However, the functional dependency tells us that $b = b'$, so $u = (a,\ b,\ c) \in R$.

Closure Example

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

$$\begin{array}{l} A, B \rightarrow C \\ B, C \rightarrow D \\ D \rightarrow E \\ C, F \rightarrow B \end{array}$$

What is the closure of $\{A, B\}$?

$$\begin{array}{lll} \{A, B\} & \stackrel{A,B \rightarrow C}{\Longrightarrow} & \{A, B, C\} \\ & \stackrel{B,C \rightarrow D}{\Longrightarrow} & \{A, B, C, D\} \\ & \stackrel{D \rightarrow E}{\Longrightarrow} & \{A, B, C, D, E\} \end{array}$$

So $\{A, B\}^+ = \{A, B, C, D, E\}$ and $A, B \rightarrow C, D, E$.

Lecture 06 : Normal Forms

Outline

- First Normal Form (1NF)
- Second Normal Form (2NF)
- 3NF and BCNF
- Multi-valued dependencies (MVDs)
- Fourth Normal Form

First Normal Form (1NF)

We will assume every schema is in 1NF.

1NF

A schema $R(A_1 : S_1, \ A_2 : S_2, \ \cdots, \ A_n : S_n)$ is in First Normal Form (1NF) if the domains S_1 are elementary — their values are atomic.

name
Timothy George Griffin

$\Longrightarrow$

first_name	**middle_name**	**last_name**
Timothy	George	Griffin

Second Normal Form (2NF)

Second Normal Form (2CNF)

A relational schema R is in 2NF if for every functional dependency $\mathbf{X} \rightarrow A$ either

- $A \in \mathbf{X}$, or
- $\mathbf{X}$ is a superkey for R, or
- A is a member of some key, or
- $\mathbf{X}$ is not a proper subset of any key.

3NF and BCNF

Third Normal Form (3CNF)

A relational schema R is in 3NF if for every functional dependency $\mathbf{X} \rightarrow A$ either

- $A \in \mathbf{X}$, or
- $\mathbf{X}$ is a superkey for R, or
- A is a member of some key.

Boyce-Codd Normal Form (BCNF)

A relational schema R is in BCNF if for every functional dependency $\mathbf{X} \rightarrow A$ either

- $A \in \mathbf{X}$, or
- $\mathbf{X}$ is a superkey for R.

Is something missing?

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 $A \to B$ 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! MVDs

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

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

Inclusions

Clearly BCNF $\subseteq$ 3NF $\subseteq 2NF$. These are proper inclusions:

In 2NF, but not 3NF

$R(A,\ B,\ C)$, with $F = \{A \rightarrow B,\ B \rightarrow C\}$.

In 3NF, but not BCNF

$R(A,\ B,\ C)$, with $F = \{A, B \rightarrow C,\ C \rightarrow B\}$.

- This is in 3NF since AB and AC are keys, so there are no non-prime attributes
- But not in BCNF since C is not a key and we have $C \rightarrow B$.

The Plan

Given a relational schema $R(\mathbf{X})$ with FDs F :

- Reason about FDs
  - Is F missing FDs that are logically implied by those in F?
- Decompose each $R(\mathbf{X})$ into smaller $R_1(\mathbf{X}_1),\ R_2(\mathbf{X}_2),\ \cdots R_k(\mathbf{X}_k)$, where each $R_i(\mathbf{X}_i)$ is in the desired Normal Form.

Are some decompositions better than others?

Desired properties of any decomposition

Lossless-join decomposition

A decomposition of schema $R(\mathbf{X})$ to $S(\mathbf{Y} \cup \mathbf{Z})$ and $T(\mathbf{Y} \cup (\mathbf{X} - \mathbf{Z}))$ is a lossless-join decomposition if for every database instances we have $R = S \bowtie T$.

Dependency preserving decomposition

A decomposition of schema $R(\mathbf{X})$ to $S(\mathbf{Y} \cup \mathbf{Z})$ and $T(\mathbf{Y} \cup (\mathbf{X} - \mathbf{Z}))$ is dependency preserving, if enforcing FDs on S and T individually has the same effect as enforcing all FDs on $S \bowtie T$.

We will see that it is not always possible to achieve both of these goals.

Lecture 07 : Schema Decomposition

Outline

- General Decomposition Method (GDM)
- The lossless-join condition is guaranteed by GDM
- The GDM does not always preserve dependencies!

General Decomposition Method (GDM)

GDM

1. Understand your FDs 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 FD $\mathbf{Z} \to \mathbf{W} \in F^+$ 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

Reminder

For $\mathbf{Z} \to \mathbf{W}$, if we assume $\mathbf{Z} \cap \mathbf{W} = \{\}$, then the conditions are

1. $\mathbf{Z}$ is a superkey for R (2NF, 3NF, BCNF)
2. $\mathbf{W}$ is a subset of some key (2NF, 3NF)
3. $\mathbf{Z}$ is not a proper subset of any key (2NF)

The lossless-join condition is guaranteed by GDM

- This method will produce a lossless-join decomposition because of (repeated applications of) Heath's Rule!
- That is, each time we replace an S by S_1 and S_2, we will always be able to recover S as $S_1 \bowtie S_2$.
- Note that in GDM step 3, the FD $\mathbf{Z} \rightarrow \mathbf{W}$ may represent a key constraint for R_1.

But does the method always terminate? Please think about this ....

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

Return to Example — Decompose to BCNF

$R(A,\ B, C, D)$

$$F = \{A, B \to C,\ \ C \to D,\ \ D \to A\}$$

Which FDs in F^+ violate BCNF?

$$\begin{array}{rcl} C & \to & A \\ C & \to & D \\ D & \to & A \\ A, C & \to & D \\ C, D & \to & A \end{array}$$

Return to Example — Decompose to BCNF

Decompose $R(A,\ B, C, D)$ to BCNF

Use $C \to D$ to obtain

- $R_1(C,\ D)$. This is in BCNF. Done.
- $R_2(A,\ B,\ C)$ This is not in BCNF. Why? A, B and B, C are the only keys, and $C \to A$ is a FD for R_1. So use $C \to A$ to obtain
  - $R_{2.1}(A,\ C)$. This is in BCNF. Done.
  - $R_{2.2}(B,\ C)$. This is in BCNF. Done.

Exercise : Try starting with any of the other BCNF violations and see where you end up.

The GDM does not always preserve dependencies!

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

$$\begin{array}{rcl} A, B & \rightarrow & C \\ D, E & \rightarrow & C \\ B & \rightarrow & D \end{array}$$

- $\{A,\ B\}^+ = \{A,\ B,\ C,\ D\}$,
- so $A, B \rightarrow C, D$,
- and $\{A,\ B,\ E\}$ is a key.

- $\{B,\ E\}^+ = \{B,\ C,\ D,\ E\}$,
- so $B, E \rightarrow C, D$,
- and $\{A,\ B,\ E\}$ is a key (again)

Let's try for a BCNF decomposition ...

Decomposition 1

Decompose $R(A,\ B,\ C,\ D,\ E)$ using $A, B \rightarrow C, D$:

- $R_1(A,\ B,\ C,\ D)$. Decompose this using $B \rightarrow D$:
  - $R_{1.1}(B,\ D)$. Done.
  - $R_{1.2}(A,\ B,\ C)$. Done.
- $R_2(A,\ B,\ E)$. Done.

But in this decomposition, how will we enforce this dependency?

$$D, E \rightarrow C$$

Decomposition 2

Decompose $R(A, B, C, D, E)$ using $B, E \rightarrow C, D$:

- $R_3(B, C, D, E)$. Decompose this using $D, E \rightarrow C$
  - $R_{3.1}(C, D, E)$. Done.
  - $R_{3.2}(B, D, E)$. Decompose this using $B \rightarrow D$:
    - $R_{3.2.1}(B, D)$. Done.
    - $R_{3.2.2}(B, E)$. Done.
- $R_4(A, B, E)$. Done.

But in this decomposition, how will we enforce this dependency?

$$A, B \rightarrow C$$

Summary

- It always is possible to obtain BCNF that has the lossless-join property (using GDM)
  - But the result may not preserve all dependencies.
- It is always possible to obtain 3NF that preserves dependencies and has the lossless-join property.
  - Using methods based on “minimal covers” (for example, see EN2000).