BCNF

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

Recall

A primary goal of database design is to decide what tables to create. Usually, there are two principles:

1. Capture all the information that needs to be captured by the underlying application.
2. Achieve the above with little redundancy.

The first principle is enforced with an entity relationship (ER) diagram, while the second with normalization.

We are finally ready to discuss redundancy. This lecture will reveal:

- When does redundancy exist?
- How to remove it?

BCNF

Let F be the set of functional dependencies we have collected, and F^+ the closure of F.

Definition

A relation R is in Borce-Codd Normal Form (BCNF) if F^+ has no FD $X \to A$ such that

1. Attribute A and all the attributes of set X appear in R (in simple words: the FD "concerns" R)
2. $A \notin X$ (in simple words: the FD is not trivial)
3. X does not contain any candidate key of R.

If $X \to A$ exists, we say that it is a BCNF-violating FD of R.

Example

Assume that there are 4 attributes A, B, C, D, and $F = \{A \rightarrow B, B \rightarrow C\}$. F^+ consists of:

$A \rightarrow A$, $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow B$, $B \rightarrow C$, $C \rightarrow C$, $D \rightarrow D$, $AB \rightarrow A$, $AB \rightarrow B$, $AB \rightarrow C$, $AC \rightarrow A$, $AC \rightarrow B$, $AC \rightarrow C$, $AD \rightarrow A$, $AD \rightarrow B$, $AD \rightarrow C$, $AD \rightarrow D$, $BC \rightarrow B$, $BC \rightarrow C$, $BD \rightarrow B$, $BD \rightarrow C$, $BD \rightarrow D$, $CD \rightarrow C$, $CD \rightarrow D$, $ABC \rightarrow A$, $ABC \rightarrow B$, $ABC \rightarrow C$, $ABD \rightarrow A$, $ABD \rightarrow B$, $ABD \rightarrow C$, $ABD \rightarrow D$, $BCD \rightarrow B$, $BCD \rightarrow C$, $BCD \rightarrow D$, $ABCD \rightarrow A$, $ABCD \rightarrow B$, $ABCD \rightarrow C$, $ABCD \rightarrow D$.

Continue onto the next slide.

Example

Question: Is relation $R(A, B, C)$ in BCNF?

Answer:

- First, derive all the candidate keys of R: only A.
- There is a FD $B \rightarrow C$ in F^+ such that
  - $B \rightarrow C$ concerns R (because both B and C are in R).
  - it is not trivial.
  - its left side (i.e., B) does not contain any candidate key of R.
- Hence, R is not in BCNF.

Example

Question: Is relation $R(A, C, D)$ in BCNF?

Answer:

- First, derive all the candidate keys of R: only AD.
- There is a FD $A \rightarrow C$ in F^+ such that
  - $A \rightarrow C$ concerns R.
  - it is not trivial.
  - its left side (i.e., A) does not contain any candidate key of R.
- Hence, R is not in BCNF.

Example

Question: Is relation $R(A, B)$ in BCNF?

Answer:

- Derive all the candidate keys of R: only A.
- F^+ has no BCNF-violating FD of R.
- Hence, R is in BCNF.

Similarly, you can verify that $R_1(B, C)$, $R_2(A, B)$ and $R_3(C, D)$ are all in BCNF.

Recall that the red values are redundant:

cid	title	year	dept
$c1$	database	2010	cs
$c1$	database	2011	cs
$c1$	database	2012	cs

A relation R can have such redundancy if and only if R is not in BCNF.

Think

Why?

We now proceed to explain how to remove redundancy. Intuitively, if a table R has redundancy, it means that some attributes in R should not be put together in the same table. Hence, naturally, we will seek ways to project R into two tables R_1, R_2, each of which has only some, but not all, attributes of R. In other words, we will not store R, but instead will store only R_1 and R_2.

This process is also called decomposing R into R_1, R_2.

Consider table CLASS:

cid	title	year	dept
$c1$	database	2010	cs
$c1$	database	2011	cs
$c2$	circuit	2010	ee

cid → title
cid → dept

Do we lose information by decomposing CLASS into: R_1(cid, title) and R_2(year, dept)?

R_1

cid	title
$c1$	database
$c2$	circuit

R_2

year	dept
2010	cs
2011	cs
2010	ee

Consider table CLASS:

cid	title	year	dept
$c1$	database	2010	cs
$c1$	database	2011	cs
$c2$	circuit	2010	ee

cid $\rightarrow$ title
cid $\rightarrow$ dept

How about decomposing CLASS into: R_1(cid, title, year) and R_2(year, dept)?

R_1

cid	title	year
$c1$	database	2010
$c1$	database	2011
$c2$	circuit	2010

R_2

year	dept
2010	cs
2011	cs
2010	ee

Consider table CLASS:

cid	title	year	dept
$c1$	database	2010	cs
$c1$	database	2011	cs
$c2$	circuit	2010	ee

cid $\rightarrow$ title
cid $\rightarrow$ dept

How about decomposing CLASS into: R_1(cid, title) and R_2(cid, year, dept)?

R_1

cid	title
$c1$	database
$c2$	circuit

R_2

cid	year	dept
$c1$	2010	cs
$c1$	2011	cs
$c2$	2010	ee

Implication?

Lossless Decomposition

Decomposing R into R_1, R_2 is lossless (i.e., losing no information) if and only if X contains a candidate key of either R_1 or R_2, where X is the set of common attributes of R_1 and R_2.

Example 1. Decomposing CLASS(cid, title, year, dept) into R_1(cid, title) and R_2(cid, year, dept) is lossless because $X = \{\text{cid}\}$ contains a candidate key of R_1 (i.e., cid).

Example 2. Decomposing CLASS(cid, title, year, dept) into R_1(cid, title, year) and R_2(year, dept) is lossy because $X = \{\text{year}\}$ is neither a candidate key of R_1 nor a candidate key of R_2. This decomposition, obviously, must not be performed!

Decomposing a Non-BCNF Relation

Let F be the set of functional dependencies we have collected, and F^+ the closure of F.

Consider a table R with schema S that is not in BCNF. To decompose R, find a FD $X \rightarrow A$ in F^+ that causes R to violate BCNF.

Decompose R into R_1, R_2 where

- R_1 includes all the attributes in $X \cup \{A\}$.
- R_2 includes all the attributes in $S - \{A\}$.

Think

This decomposition must be lossless. Why?

Example

There are 4 attributes A, B, C, D, and $F = \{A \rightarrow B, B \rightarrow C\}$. Decompose $R(A, B, C, D)$ into BCNF tables.

- R has only one candidate key: AD.
- Find a BCNF-violating FD of R: $A \rightarrow B$.
- Decompose R into $R_1(AB)$ and $R_2(ACD)$.

R_1 is already in BCNF, but R_2 is not, and hence, needs to be decomposed further (see next).

Example

There are 4 attributes A, B, C, D, and $F = \{A \to B, B \to C\}$. Decompose $R(A, B, C, D)$ into BCNF tables.

- $R_2(ACD)$ has only one candidate key: AD.
- Find a BCNF-violating FD of R: $A \to C$.
- Decompose R into $R_3(AC)$ and $R_4(AD)$.

R_3 and R_4 are both in BCNF. We are done: the original R has been replaced with $R_1(AB), R_3(AC)$ and $R_4(AD)$, where no redundancy can occur!

Here is an exercise for you, the completion of which will (finally) finish the whole design of the tiny database that was used as an example in our lectures about ER-diagrams!

Consider the table CLASS(cid, title, year, dept) and the FDs we have collected:

$$\text{cid} \rightarrow \text{title}$$
$$\text{cid} \rightarrow \text{dept}$$

Decompose CLASS into BCNF tables.