Dependencies and Normal Forms

Example 1

Present a real-life example (not an example of ABCD, etc), to demonstrate that the multiple-value dependencies and/or 4NF cannot be ignored in the database design. Explain why.

It is sufficient to given an example to show that without considering the multiple value dependencies, a table schema, even though in BCNF, may have some serious redundancies and/or update anomalies.

Consider the following table

employee(emp_id, project, dependent),

with one multi-value dependency:

$emp\_id \twoheadrightarrow project | dependend.$

Then the table is in BCNF but with serious redundency and update anomalies.

For an employee may have 4 dependents but work on 5 projects.

Example 2

Consider the following functional dependencies over the attribute set $R = ABCDEFGH$:

$A \to E$ $\quad$ $BE \to D$

$AD \to BE$ $\quad$ $BDH \to E$

$AC \to E$ $\quad$ $F \to A$

$E \to B$ $\quad$ $D \to H$

$BG \to F$ $\quad$ $CD \to A$

1. Find a minimal cover of the given set of FDs.
2. Find a join loss-less, dependency preserving and 3NF decomposition of R.
3. Check if all the resulting relations in the previous step are in BCNF. If you find a schema that is not, decompose it into a lossless BCNF.

(a) The following three steps lead to a minimal cover.

Right-reducing It is done by split AD $\rightarrow$ BE into AD $\rightarrow$ B and AD $\rightarrow$ E.

Left-reducing leads to the following set:

$A \rightarrow E$ $\quad BD \rightarrow E$

$A \rightarrow B$ $\quad F \rightarrow A$

$E \rightarrow B$ $\quad D \rightarrow H$

$BG \rightarrow F$ $\quad CD \rightarrow A$

$E \rightarrow D$

By eliminating redundant ones, we obtain the following minimal cover of the given set of FDs:

$A \rightarrow E$ $\quad BD \rightarrow E$

$E \rightarrow B$ $\quad F \rightarrow A$

$BG \rightarrow F$ $\quad D \rightarrow H$

$E \rightarrow D$ $\quad CD \rightarrow A$

(b) The following is a join lossless, dependency preserving and 3NF decomposition of the original database schema.

AE, EBD, BGF, FA, DH, CDA, BGC.

Note that BFC is a candicate key that was added later.

(c) All tables, except CDA, in the above database schema are in BCNF. CDA is not in BCNF because A $\rightarrow$ D but not A $\rightarrow$ C. CDA can be decomposited into {AD, AC}.

Example 3

Consider R = ABCDE. For each of the following instances of R, state whether (1) it violates the FD AE $\rightarrow$ C, and (2) it violates the MVD AC $\rightarrow\!\!\!\rightarrow$D:

1. an empty table

2.

A	B	C	D	E
a	2	3	4	5
2	a	3	5	5
a	3	3	6	5

1. An empty table satisfies both dependencies.

2. R satisfies AE $\rightarrow$ C but not AC $\rightarrow\!\!\!\!\rightarrow$D.

Example 4

Consider $R = ABCDEGHI$ and the following set F of functional dependencies:

H $\rightarrow$ GD

E $\rightarrow$ D

HD $\rightarrow$ CE

BD $\rightarrow$ A

1. Find a join loss-less, dependency preserving and 3NF decomposition of R.
2. Indicate whether your database schema is in BCNF with respect to F. Explain.

We first find a minimal cover of the FDs, as shown below.

Right reduced	Left Reduced:	Minimal Cover
H $\rightarrow$ G	H $\rightarrow$ G	H $\rightarrow$ G
H $\rightarrow$ D	H $\rightarrow$ D	E $\rightarrow$ D
E $\rightarrow$ D	E $\rightarrow$ D	H $\rightarrow$ C
HD $\rightarrow$ C	H $\rightarrow$ C	H $\rightarrow$ E
HD $\rightarrow$ E	H $\rightarrow$ E	BD $\rightarrow$ A
BD $\rightarrow$ A	BD $\rightarrow$ A	

Then construct a database $D' = \{HGCE, ED, BDA\}$.

Now, we need to check if D' contains any candidate key.

Since no FD in the minimal cover above contains H, B, or I in its right side, any candidate key shall contain these three attributes.

Further, it is not difficult to check that HBI is indeed a candidate key. Therefore, HBI is the only candidate key of R, and shall be added to D'.

Hence,

$$D = \{HGCE, ED, BDA, HBI\}$$

is a join loss-less, dependency preserving and 3NF decomposition of R.

(2) D is in BCNF since all the non-trivial FDs X $\rightarrow$ A in held in any relation $R_i \in D$, X is a key of R_i.

Example 5

In designing a relational database schema, why might we choose a non-BCNF design?

This is because in many cases, there exists no database schema that is both BCNF and dependence preserving. If one prefers to have a dependence preserving database schema, then one have to choose a normal form, such as 3NF, that is weaker than BCNF.