

Every conversion into BCNF may not be dependency preserving

Proof:

We only need to give a counter example: Consider the following schema;

a b c and $c \rightarrow b$

Clearly the above schema is in 3NF, because $ab \rightarrow c$ is a superkey dependency and, from $c \rightarrow b$ we can see that $b - c = b$, which is a subset of the primary key (such dependency is also allowed in 3NF).

But, the above schema is not in BCNF because $c \rightarrow b$ is neither super-key nor trivial dependency.

So we decompose above schema, keeping it lossless.

Only possible lossless decomposition is: ac and cb. (because, their intersection c is primary key for the 2nd table).

But clearly the dependency $ab \rightarrow c$ is lost.

Hence, proved.

Q.E.D

Motivation for XML

Data Exchange/Mediation:

Consider the following cases:

- a) Travel Sites requiring information from various airline companies about the plane schedule.
- b) A database stored in Oracle needs to be transferred to another open source organization using MYSQL.
- c) A document stored in MS-Office needs to be converted to another document in Open Office.

The common constraints are:

- i) The party giving up information can't reveal its data storage format / mechanism.
- ii) The party requiring the information can't keep on changing its data storage format just because its needs information from some other organization.

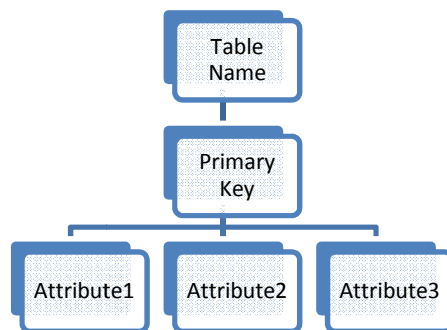
So, the need for an intermediary is apparent here.

XML serves as that intermediary. It is a markup language, in that sense it stores information additional to just the text, which qualifies/describes the various sections of the text.

XML falls in the class of semi-structured data, between structured(Databases) and un-structured data(simple text files).

Data in XML is stored in hierarchy similar to a tree. Each node is represented as a tag `< > </>` and metadata for the node can be stored within the `< >`, where as the data depending on this node is stored in between the tags `< > data</>`.

Example: Converting a table stored in DBMS, in order to store it into a XML file.



Other possible ways could have been to represent functional dependencies in the hierarchy.

But information like foreign key attributes can't be stored in an XML file. So this conversion might be *lossy*. While reverting back from XML to structured format some data loss might be inevitable.

Apart from this the conversion efficiency is also very poor. This is why Resident XML databases are hardly used in core databases.

Conversion from text files to XML files can be done by brute force methods or by creating customized software like crawlers.

XML files can easily be converted to simple text files by simply doing copy-paste, thereby removing the meanings attached to the various markups.