Boyce Codd Normal Form

Patrick Kevorkian

Spring 2016

What is BCNF?

Developed by Edgar F. Codd and Raymond F. Boyce in 1974, it is a process of organizing columns and tables of a relational database in order to minimize data redundancy. Specifically a database schema is in this form if all redundancy based upon functional dependency has been removed. It is also referred to as 3.5 Normal Form because it is a stronger form than several other normalization processes.

Furthermore these conditions must hold:

X -> Y is a subset of X
X is a superkey for schema R

Interestingly enough this normalization definition predates Codd and Boyce by 3 years. Although it appears in a paper by Ian Heath in 1971 it is not named after him.

BCNF Calculator

This study has two main goals: first to understand Boyce Codd Normal Form and second to create an accompanying computer program.

- Program implements an algorithm in which a set of user defined attributes can be normalized in BCNF.
- User has a choice of up to 26 attributes and an unlimited amount of user defined functional dependencies.
- Clicking "Calculate BCNF" information is processed and the Decomposition along with the closures are displayed.
- In order to create a likable user interface and have ease of text manipulation the program was coded in C# for windows.
- Its aptly named "BCNF Calculator."

Key Terms

- Attribute- Column of a table that has the same type of value
- Closure- Given a set of attributes the closure of that set under the relation is the set of attributes that are functional determined by that set under the relation. Is stylized in the form $\{A\}^{+} = \{A, \ldots ,n\}$
- Decomposition- Break down relations according to BCNF conditions
- Relation- A set of attributes
- Relational Database- A database structured to recognize relations among stored items of information
- Subset- Part of a larger group of related things
- Superkey- A set of one or more attributes that can identify all other attributes uniquely.
- Schema- Structure of the database that defines the objects in it
- Normalization- The process of organizing data in a database
- Functional Dependency (FD)- Constraint that describes the relationship between two attributes
- Implicit Functional Dependency- Functional dependencies that show relations that aren't known before the computation of closures

Example

Here we see a table that is not normalized. A database of books and such is common and could be used for many applications. Its attributes include Author, Nationality, Book Title, Genre, and Number of pages. As the size of information grows there is a possibility for redundant data within the table. For instance an author can have more than one novel so the authors name and nationality could appear more than once. This repetition of data is redundant.

Author	Nationality	Book title	Genre	Number of pages
William Shakespeare	English	The Comedy of Errors	Comedy	100
Markus Winand	Austrian	SQL Performance Explained	Textbook	200
Jeffrey Ullman	American	A First Course in Database Systems	Textbook	500
Jennifer Widom	American	A First Course in Database Systems	Textbook	500

Functional Dependencies

However there are attributes that have a specific relationship to other attributes. For example an author will always have the same nationality, so listing this more than once does not add new information. Another example in this case is that book title determines the genre and the number of pages. This is stylized by an attribute (Left), an arrow, followed by an attribute (Right):

Author -> Nationality
Book Title -> Genre
Book Title -> Number of pages

Decomposition

Here we see that the data from the table above has been decomposed into three separate tables. X determining Y being a subset of X is held true because each table contains one functional dependency. In addition X being a superkey of schema R is held true because each table has unique values. Therefore this is in BCNF.

Author	Nationality
William Shakespeare	English
Markus Winand	Austrian
Jeffrey Ullman	American
Jennifer Widom	American

Book title	Genre	Number of pages
The Comedy of Errors	Comedy	100
SQL Performance Explained	Textbook	200
A First Course in Database Systems	Textbook	500

Author	Book title
William Shakespeare	The Comedy of Errors
Markus Winand	SQL Performance Explained
Jeffrey Ullman	A First Course in Database Systems
Jennifer Widom	A First Course in Database Systems

Algorithmic Implementation

An algorithm is a procedure or formula for solving a problem, in this case it calculates Boyce Codd normal Form on a series of attributes. Here is shown the steps taken to achieve the desired result.

1- User Input

A. User inputs the number of attributes for their schema (1-26)
B. User picks their functional dependencies. (A->B etc.)

2- Calculate Closures

A. Checks every functional dependency for what the attribute determines
B. If left side of a functional dependency is in right side of a closure then it is added to the right side of the closure for that attribute
C. Repeat step B until all functional dependencies are considered

3- Calculate Implicit Functional Dependencies

A. Given newly calculated closures pull what each attribute determines in a single relation that isn't itself

Example:

-Given attributes {A,B,C}
-Functional Dependencies of A->B, B->C
-Closures: $\{A\}^{+} = \{A, B, C\}$
$\{B\}^{+} = \{B, C\}$
$\{C\}^{+} = \{C\}$
$\{A,B\}^{+} = \{A, B, C\}$
$\{A,C\}^{+} = \{A, B, C\}$
$\{C,B\}^{+} = \{B, C\}$
$\{A,B,C\}^{+} = \{A, B, C\}$

Given these updated closures we can see what the implicit functional dependencies are:

A -> C
AB -> C
AC -> B

4- Normalize to Boyce Codd Normal Form

A. Find a functional dependency that violates the two BCNF conditions
B. Pick that dependency and decompose the relation into sub-relations. The first sub-relation being itself and second being everything that isn't in the right side of the first.
C. Repeat steps A and B (Recursively)von each sub-relation until each meet the BCNF conditions

Example: Continuing from the above we first consider step A since all of the FD's violate BCNF conditions we must pick one to decompose

Chosen FD: A -> B

Decomposed into: R1 (A,B)
R2 (A,C)

Repeat steps A and B on R1 and R2: Since both relations meet BCNF conditions they are considered to be in Boyce Codd Normal form

Boyce Codd Normal Form Relations(Tables): (A, B) and (A, C)