

La differenza tra BCNF e 3NF (BCNF implica 3NF)

Utilizzando la definizione BCNF:

"Se e solo se per ciascuna delle sue dipendenze $X \rightarrow Y$, vale almeno una delle seguenti condizioni:"

1. $X \rightarrow Y$ è una dipendenza funzionale banale ($Y \subseteq X$), o
2. X è una super chiave per lo schema R

e la definizione 3NF:

"Se e solo se, per ciascuna delle sue dipendenze funzionali $X \rightarrow A$, vale almeno una delle seguenti condizioni:"

1. X contiene A (cioè $X \rightarrow A$ è una dipendenza funzionale banale), oppure
2. X è una superchiave, o
3. Ogni elemento di $A-X$, la differenza di insieme tra A e X , è un attributo primo (cioè, ogni attributo in $A-X$ è contenuto in una chiave candidata)

Vediamo la seguente differenza, in termini semplici:

- In BCNF: ogni chiave parziale (attributo prime) può dipendere solo da una superchiave,

Mentre

- In 3NF: una chiave parziale (attributo prime) può anche dipendere da un attributo che non è una superchiave (ad esempio, un altro attributo chiave parziale / attributo primo o anche un attributo non primo).

Dove

1. Un attributo principale (definito in inglese come *prime attribute*) è un attributo trovato in una chiave candidata e
2. Una chiave candidata è una superchiave minima per tale relazione, e
3. Una superchiave è un insieme di attributi di una variabile di relazione per i quali sostiene che in tutte le relazioni assegnate a tale variabile, non ci sono due tuple distinte (righe) che hanno gli stessi valori per gli attributi in questo set. In modo equivalente, una superchiave può anche essere definita come un insieme di attributi di uno schema di relazione da cui tutti gli attributi dello schema dipendono funzionalmente. (Una superchiave contiene sempre una chiave candidata/una chiave candidata è sempre un sottoinsieme di una superchiave. È possibile aggiungere qualsiasi attributo in una relazione per ottenere una delle superchiavi.)

Cioè, nessun sottoinsieme parziale (qualsiasi sottoinsieme non banale tranne l'insieme completo) di una chiave candidata può essere funzionalmente dipendente da qualcosa di diverso da una superchiave.

Una tabella/relazione che non è in BCNF è soggetta ad anomalie come le anomalie di aggiornamento.

Purtroppo:

- BCNF non può sempre essere ottenuta, mentre
- 3NF può sempre essere ottenuto.

The subtle difference is that 3NF makes a distinction between key and non-key attributes (also called non-prime attributes) whereas BCNF does not.

Una relazione, R , è in 3NF se e solo se per ogni FD non banale ($X \rightarrow A$) soddisfatto da R almeno UNA delle seguenti condizioni è vera:

- a) X è una superchiave per R , o
- b) A è un attributo chiave per R

BCNF richiede (a) ma non tratta (b) come un caso speciale a sé stante. In altre parole, BCNF richiede che ogni determinante non banale sia una superchiave anche se i suoi attributi dipendenti fanno parte di una chiave.

Una relazione, R , è in BCNF se e solo se per ogni FD non banale ($X \rightarrow A$) soddisfatta da R la seguente condizione è vera:

- (a) X è una superchiave per R

BCNF è quindi più severa. La differenza è così sottile che ciò che molte persone descrivono informalmente come 3NF è BCNF. 3NF potrebbe essere descritto più accuratamente come "i dati non chiave dipendono dalle chiavi ... e nient'altro che le chiavi".

Insomma:

BCNF è una forma normale in cui per ognuna delle dipendenze funzionali non banali di una tabella, è una superchiave; 3NF è una forma normale in cui la tabella è in 2NF e ogni attributo non primo non dipende transitivamente da ogni chiave della tabella.

(Esempio intuitivo da Stack Overflow)

La tua pizza può avere esattamente tre tipi di condimento:

1. un tipo di formaggio
2. un tipo di carne
3. un tipo di verdura

Quindi, ordiniamo due pizze e scegliamo i seguenti condimenti:

Pizza	Topping	Topping Type
1	mozzarella	cheese
1	pepperoni	meat
1	olives	vegetable
2	mozzarella	meat
2	sausage	cheese
2	peppers	vegetable

La mozzarella non può essere sia un formaggio che una carne e la salsiccia non è un formaggio! Dobbiamo evitare questo tipo di errori, fare in modo che la mozzarella sia sempre formaggio. Dovremmo usare una tabella separata per questo, quindi scriviamo questo fatto in un solo posto.

Il semplice esempio fa capire che:

"BCNF agisce in modo diverso da 3NF solo quando ci sono più chiavi candidate sovrapposte. Il motivo è che la dipendenza funzionale $X \rightarrow Y$ è ovviamente vera se Y è un sottoinsieme di X . Quindi, in ogni tabella che ha una sola chiave candidata ed è in 3NF, è già in BCNF perché non c'è colonna (chiave o non chiave) che sia funzionalmente dipendente da qualcosa oltre a quella chiave."

Pizza	Topping
1	mozzarella
1	pepperoni
1	olives
2	mozzarella
2	sausage
2	peppers

Topping	Topping Type
mozzarella	cheese
pepperoni	meat
olives	vegetable
sausage	meat
peppers	vegetable

Poiché ogni pizza deve avere esattamente uno di ogni tipo di topping, sappiamo che $(Pizza, Topping Type)$ è una chiave candidata. Sappiamo anche intuitivamente che un dato topping non può appartenere a diversi tipi contemporaneamente. Quindi $(Pizza, Topping)$ deve essere unico e quindi è anche una chiave candidata. Quindi, abbiamo due chiavi candidate sovrapposte.

Ho mostrato un'anomalia in cui abbiamo contrassegnato mozzarella come il tipo di topping sbagliato.

Sappiamo che questo è sbagliato, ma la regola che lo rende sbagliato è una dipendenza $Topping \rightarrow Topping Type$ che non è una dipendenza valida per BCNF per questa tabella. È una dipendenza da qualcosa di diverso da un'intera chiave candidata.

Quindi, per risolvere questo problema, togliamo *Topping Type* dalla tabella *Pizzas* e lo rendiamo un attributo non chiave in una tabella *Toppings*.