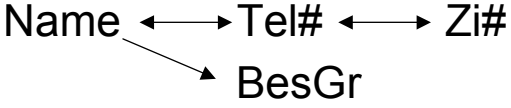
Kapitel 7.3 Boyce-Codd Normalform *BCNF*

Definition: Sei R = (A_1, A_2, ..., A_k) Relation mit Attr.
A_1, ... A_k
(A_{i1}, A_{i2}, ..., A_{im}) heißt Determinator von A_j,
wenn $A_j \notin \{A_{i1}, ..., A_{im}\}$ ist und A_j ist voll
funktional abhängig von $\{A_{i1}, ..., A_{im}\}$

Beispiele für Determinatoren:

PROFESSOREN = (Name, Tel#, Zi#, BesGr)



Sei Det die Menge der Determinatoren:

Det (BesGr) = {(Name), (Tel#), (Zi#)}
Det (Name) = {(Tel#), (Zi#)}
etc.

REPERTOIRE = (Name, Vorlesungen)

Det (Name) = ∅

Det (Vorlesungen) = ∅

BELEGUNG = (H#, Tag, Zeit, V#) *Vorles mehrfach pro Tag!*

Det (H#) = {(Tag, Zeit, V#)}

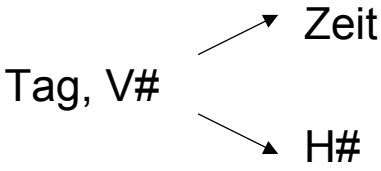
Det (Tag) = ∅

Det (Zeit) = ∅

Det (V#) = {(H#, Tag, Zeit)}

⇒ Tag, Zeit müssen in jedem Schlüsselkandidaten sein.

neue Annahme: "*Vorlesung nur **einmal** pro Tag*":



Det (H#) = {(Tag, V#)}

Det (Tag) = ∅

Det (Zeit) = {(Tag, V#)}

Det (V#) = {(H#, Tag, Zeit)}

⇒ Tag in jedem Schlüsselkandidat

1. Schl. Kand: (Tag, V#)
2. Schl. Kand: (H#, Tag, Zeit)

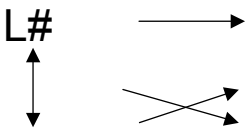
BELEGUNG nicht weiter zerlegbar!

Def: R ist in BCNF, wenn jeder Determinator Schlüsselkandidat ist.

Beispiele für BCNF:

1) L3 = (L#, LNAME, STATUS, STADT)

Kandidaten: L# und LNAME



in 3NF und
in BCNF

2) nicht in 3NF und nicht in BCNF:

LT1 = (L#, LNAME, T#, A)

Det (L#)	=	{(LNAME)}
Det (LNAME)	=	{(L#)}
Det (T#)	=	∅
Det (A)	=	{(L#, T#), (LNAME, T#)}

⇒ T# muß in jedem Schl. Kandidaten sein,
zwei Schlüsselkandidaten: (L#, T#) und (LNAME, T#)
also LNAME bzw L# nicht voll funktional abhängig,
also nicht einmal in 2NF

⇒ L# oder LNAME herausfaktorisieren

LT1 zerlegen in LT2 und LT3:

LT2 = (L#, T#, A); LT3 = (L#, LNAME)

bzw. in LT2' und LT3':

LT2’ = (LNAME, T#, A); LT3’ = (LNAME, L#)

LT2: Det (L#) = ∅
Det (T#) = ∅
Det (A) = {(L#, T#)} i.e. einziger Schlüssel

LT3: Det (L#) = {(LNAME)}
Det (LNAME) = {(L#**)}**
⇒ zwei Schlüsselkandidaten

⇒ LT2 und LT3 in BCNF !!

3) 3NF nach Codd, nicht BCNF

FLS = (F, L, S)

F: Fach
L: Lehrer
S: Schüler

1. Jeder Schüler hat nur 1 Lehrer pro Fach
2. Jeder Lehrer vertritt nur 1 Fach
3. Zu jedem Fach mehrere Lehrer

Det (F) = L von 2. L ⟶ F
Det (L) = {(S, F)} von 1. S,F ⟶ L
Det (S) = ∅

in 3NF nach Codd:

Schlüssel (S, F), L voll funktional abhängig von (S, F)

nicht in BCNF, weil Det (F) nicht Schlüsselkandidat

nicht in 3NF nach Kent:

weiterer Schlüssel Kandidat ist (S, L) und F ist nicht voll funktional abhängig

Zerlegung in: L ⟶ F

S, L

Frage: *Gibt es Unterschied zwischen 2NF nach Kent und BCNF?*

Allgemeiner Fall:

R = (A, B, C, D, E, F)

F voll funktional abhängig von allen Schlüsselkandidaten (A, B) und (C, D), aber auch von (B, C), ohne daß (B, C) Schlüsselkand. zu sein braucht.

⇒ nach Kent,
aber nicht BCNF

R1 = (B, C, F)
R2 = (A, B, C, D, E)

Hinweis: S, F, L
wenn Teilschlüssel von anderem Attribut abhängig ist, wähle anderen Schlüssel und löse Problem über 2NF nach Kent !

Abstraktes Problem: allg. Funktionen in Relationen verborgen!

bei Codd/Kent nur volle funktionale Abhängigkeit von **Schlüsseln** und **einstellige** transitive Abhängigkeiten.

f : A, B, ..., C $\longrightarrow$ D

aus Rel. herausfaktorisieren und Funktionen mit identischem Definitionsbereich (wird Schlüssel in neuer Relation) zusammenfassen (außer für Schlüsselkand.)

$\Rightarrow \forall$ Attr., die sich aus anderen durch Regel, d.h. Algorithmus berechnen lassen.

Beispiel:

PERSON = (P#, ..., Eink, Steuerkl, Steuer)
Eink, Steuerkl $\longrightarrow$ Steuer
faktorisieren, i.e. *Steuertabelle,* zusätzlich
Ehestatus, Anz. Kinder $\longrightarrow$ Steuerkl.
herausfaktorisieren

DIENSTREISE = (P#, Abfahrt, Rückkehr, Tagegeld)
f: Abfahrt, Rückkehr $\longrightarrow$ Tagegeld
f nicht durch Tabelle oder Ausdruck, sondern als bedingte Anweisung:
z. B. $\leq$ 12 h : Betrag A
12 - 24 h : Betrag B etc.

Weitere Normalformen:

siehe Spezialliteratur:

i.a. geht es um Zerlegung von R in R1, R2
so daß: R = R1 ω R2
R1, R2 "einfacher" als R und ω ist relationaler Operator

Allg. Problem: gesucht ist minimale Basis von Erzeugenden, um alle benötigten Relationen als Ausdrücke der rel. Algebra darstellen zu können

Probleme: Auswertung von R1 ω R2 immer wenn R benötigt wird

- Kosten : Speicher, CPU
- Semantik
- Integritätsbedingungen

Absichtliche Nicht-Normalisierung

z.B. weil Info fast immer gemeinsam benötigt wird

KNADR = (K#, Name, Straße, Stadt, Land, PLZ)