

IN2090 – Databaser og datamodellering

07 – Normalformer

Leif Harald Karlsen (Evgenij Thorstensen)

leifhka@ifi.uio.no



Universitetet i Oslo

Normalformene 1NF-BCNF

- ◆ Normalformene vi skal se på danner et hierarki:

$$BCNF \subseteq 3NF \subseteq 2NF \subseteq 1NF$$

- ◆ Det vil si: Hvis et skjema oppfyller 3NF, oppfyller det også 2NF og 1NF
- ◆ Høyere NF gir færre anomalier, men flere tabeller og flere joins
- ◆ Gitt et skjema, så finnes det en algoritme som lager et ekvivalent skjema på hvilken NF man ønsker
- ◆ For **alle** NF er det slik at et skjema oppfyller en gitt NF hvis alle tabeller oppfyller kravene

1NF

- ◆ En tabell er på 1NF hvis alle attributter er **atomære**
- ◆ Altså ikke attributter med arrays/JSON/osv.
- ◆ Heller ikke strenger som inneholder flere verdier
- ◆ Av og til ullent begrep (PostgreSQL støtter f.eks. arrays og JSON som datatyper)
- ◆ Poenget er at verdiene man ønsker å bruke i joins, sammenlikninger, ol. skal være i egne kolonner
- ◆ Dette slik at man skal slippe kompleks manipulering av datastrukturer eller strenger for enkle operasjoner
- ◆ Vi antar derfor at 1NF alltid er oppfylt

1NF, eksempler på brudd

Brnavn	...	Veiledere	Adresse
evgenit	...	[arild, martingi]	Gateveien 1b, 0123 Oslo
peternl	...	[abc]	Stedplassen 2, 1234 Bergen

Brnavn	...	Studenter
arild	...	[evgenit]
abc	...	[peternl]

Nesten alltid lurere å

- ◆ lage en egen tabell `Veiledning(Student, Veileder)`
- ◆ splitte strengen opp i `Student(..., Gate, Postnummer, Poststed)`

- ◆ En tabell oppfyller 2NF hvis
 - ◆ den oppfyller 1NF og
 - ◆ alle attributter A som ikke er nøkkelattributter, **ikke** er funksjonelt avhengige av en delmengde av en kandidatnøkkel
- ◆ Nøkkelattributt: Attributt som er med i en kandidatnøkkel.
- ◆ Alternativt: En tabell **bryter** 2NF hvis det finnes et ikke-nøkkelattributt A som **er avhengig** av en delmengde av en kandidatnøkkel.

2NF, eksempel

Følgende tabell er på 1NF, men ikke 2NF:

R(Brnavn, Navn, Etternavn, Adresse, Kurskode, Tittel, Beskrivelse, AntSP, Karakter)

- ◆ Brnavn \rightarrow Navn, Etternavn, Adresse
- ◆ Kurskode \rightarrow Tittel, Beskrivelse, AntSP
- ◆ Brnavn, Kurskode \rightarrow Karakter

- ◆ Kandidatnøkkel: Brnavn, Kurskode
- ◆ Navn er avhengig av Brnavn og Brnavn er en del av nøkkelen. Brudd på 2NF.

- ◆ En tabell oppfyller 3NF hvis
 - ◆ den oppfyller 2NF og
 - ◆ alle ikke-nøkkelattributter **kun** er avhengige av kandidatnøkler
- ◆ Alternativt: En tabell bryter 3NF hvis det finnes et ikke-nøkkelattributt som **er avhengig** av noe som **ikke** er en kandidatnøkkel.
- ◆ En tabell på 2NF og ikke 3NF kan kun skje dersom en ikke-nøkkelattributt A er avhengig av en (eller fler) ikke-nøkkelattributt(er) X som selv er avhengig av en kandidatnøkkel
- ◆ Altså, om K er en kandidatnøkkel kunne vi over ha FDene $K \rightarrow X$ og $X \rightarrow A$.
- ◆ Altså, A avhenger ikke direkte av kandidatnøkkel, men transitivt

3NF, eksempel

Følgende tabell er på 2NF, men ikke 3NF:

`Ansatt(Id, Navn, Avdeling, AvdelingsKode)`

Gitt FDene:

- ◆ `Id → Navn, AvdelingsKode`
- ◆ `AvdelingsKode → Avdeling`

Her har vi:

- ◆ `Id` er en kandidatnøkkel (transitivitet)
- ◆ Alle attributter avhenger av hele kandidatnøkkelen, altså `Id` (2NF)
- ◆ men `Avdeling` er avhengig av `AvdelingsKode`, som ikke er en nøkkelattributt.
- ◆ `Avdeling` dobbeltlagres for hver ansatt (burde skilles ut i egen tabell)

- ◆ Kort for Boyce-Codd normalform
- ◆ En tabell oppfyller BCNF hvis alle attributter kun er avhengige av en kandidatnøkkel
- ◆ Samme som 3NF, men unntaket for nøkkelattributter er **fjernet**
- ◆ Unntaket blir sjeldent brukt, og som regel er tabeller på 3NF også på BCNF
- ◆ Huskeregel for BCNF: *The key, the whole key, and nothing but the key, (so help me Codd)*

BCNF, eksempel

Følgende tabell er på 3NF, men ikke BCNF¹:

Person	Type	Nærmeste
David	Optiker	Ørneøyet
David	Frisør	Kutt og krøll
Kari	Bokhandler	Merlins bøker
Erik	Bakeri	Deiglig
Erik	Frisør	Klipperud

Med følgende FDer:

- ◆ Person, Type → Nærmeste
- ◆ Nærmeste → Type

Her har vi:

- ◆ Kandidatnøkler: {Person, Type} og {Person, Nærmeste}
- ◆ Alle ikke-nøkkelattributter (ingen slike) avhenger kun av kandidatnøkler (altså 3NF)
- ◆ Men, nøkkel-attributtet Type avhenger av Nærmeste som ikke er en kandidatnøkkel

¹Inspirert av eksempel fra https://en.wikipedia.org/wiki/Boyce%E2%80%93Codd_normal_form

Normalformer i praksis

- ◆ Jo høyere normalform, jo færre anomalier
- ◆ Men jo høyere normalform, jo fler relasjoner i skjemaet
- ◆ BCNF fjerner (nesten) alle anomalier og er et godt kompromiss
- ◆ De fleste ordentlige skjemaene (Northwind, filmdb, osv.) er på BCNF

Takk for nå!

Neste uke vil omhandle algoritmer for å avgjøre normalformer og tapsfri dekomposisjon.