God Databasedesign:
På vei mot Normalformer

Martin Giese

24. november 2018

Agenda

- Hva er god databasedesign?
  - Forklart ved et dårlig eksempel
- Oppdateringsanomalier
- Repetisjon: Supernøkler, kandidatnøkler, primærnøkler, nøkkelattributter
- Repetisjon: Funksjonelle avhengigheter (FD-er)
- Normalformer, NF1 og NF2

Et Eksempel

Normalisering: et eksempel

FulltNavn	Adresse	Fakturanr	Ordrenr	Fakturadato	Produkt1	Kost1	Antall1	Produkt2	Kost2	Antall2	Produkt3	Kost3
Ola Nordmann	Problemveien 11 0313 Oslo	3201	AB123	13/11/2017	Ting	249,-	2	Tang	1200,-	1	Greie	599,-
Kari Nilsen	Moltke Moes vei 35 0851 Oslo	3202	QZ93	13/11/2017	Ting	249,-	4					
Ole Olsen	Sem Sælands vei 7 0371 Oslo	3203	33AR	14/11/2017	Gjenstand	25,-	10	Ting	249,-	5		
Ola Nordmann	Problemveien 11 0313 Oslo	3204	AB130	14/11/2017	Tang	1200,-	1					
John Doe	Gaustadbekkalléen 23B 0372 Oslo	3205	656	14/11/2017	Ting	249,-	3					

Atomære verdier: Adresse-eksempel

Gaustadalléen 23 B

0373 OSLO

Norge

Normalisering: et eksempel

FulltNavn	Adresse	Fakturanr	Ordrenr	Fakturadato	Produkt1	Kost1	Antall1	Produkt2	Kost2	Antall2	Produkt3	Kost3
Ola Nordmann	Problemveien 11 0313 Oslo	3201	AB123	13/11/2017	Ting	249,-	2	Tang	1200,-	1	Greie	599,-
Kari Nilsen	Moltke Moes vei 35 0851 Oslo	3202	QZ93	13/11/2017	Ting	249,-	4					
Ole Olsen	Sem Sælands vei 7 0371 Oslo	3203	33AR	14/11/2017	Gjenstand	25,-	10	Ting	249,-	5		
Ola Nordmann	Problemveien 11 0313 Oslo	3204	AB130	14/11/2017	Tang	1200,-	1					
John Doe	Gaustadbekkalléen 23B 0372 Oslo	3205	656	14/11/2017	Ting	249,-	3					

Fornavn	Etternavn	Adresse	Postnr	Poststed	Fakturanr	Ordrenr	Fakturadato	Produkt	Kost	Antall
Ola	Nordmann	Problemveien 11	0313	Oslo	3201	AB123	13/11/2017	Ting	249,-	2
Ola	Nordmann	Problemveien 11	0313	Oslo	3201	AB123	13/11/2017	Tang	1200,-	1
Ola	Nordmann	Problemveien 11	0313	Oslo	3201	AB123	13/11/2017	Greie	599,-	1
Kari	Nilsen	Moltke Moes vei 35	0851	Oslo	3202	QZ93	13/11/2017	Ting	249,-	4
Ole	Olsen	Sem Sælands vei 7	0371	Oslo	3203	33AR	14/11/2017	Gjenstand	25,-	10
Ole	Olsen	Sem Sælands vei 7	0371	Oslo	3203	33AR	14/11/2017	Ting	249,-	5
Ola	Nordmann	Problemveien 11	0313	Oslo	3204	AB130	14/11/2017	Tang	1200,-	1
John	Doe	Gaustadbekkalléen 23B	0372	Oslo	3205	656	14/11/2017	Ting	249,-	3

KundeID	Fornavn	Etternavn	Adresse	Postnr	Poststed	Fakturanr	Ordrenr	Fakturadato	ProduktID	Produkt	Kost	Antall
1	Ola	Nordmann	Problemveien 11	0313	Oslo	3201	AB123	13/11/2017	1	Ting	249,-	2
1	Ola	Nordmann	Problemveien 11	0313	Oslo	3201	AB123	13/11/2017	2	Tang	1200,-	1
1	Ola	Nordmann	Problemveien 11	0313	Oslo	3201	AB123	13/11/2017	3	Greie	599,-	1
2	Kari	Nilsen	Moltke Moes vei 35	0851	Oslo	3202	QZ93	13/11/2017	1	Ting	249,-	4
3	Ole	Olsen	Sem Sælands vei 7	0371	Oslo	3203	33AR	14/11/2017	4	Gjenstand	25,-	10
3	Ole	Olsen	Sem Sælands vei 7	0371	Oslo	3203	33AR	14/11/2017	1	Ting	249,-	5
1	Ola	Nordmann	Problemveien 11	0313	Oslo	3204	AB130	14/11/2017	2	Tang	1200,-	1
4	John	Doe	Gaustadbekkalléen 23B	0372	Oslo	3205	656	14/11/2017	1	Ting	249,-	3

KundeID	Fakturanr	Ordrenr	Fakturadato	ProduktID	Antall
1	3201	AB123	13/11/2017	1	2
1	3201	AB123	13/11/2017	2	1
1	3201	AB123	13/11/2017	3	1
2	3202	QZ93	13/11/2017	1	4
3	3203	33AR	14/11/2017	4	10
3	3203	33AR	14/11/2017	1	5
1	3204	AB130	14/11/2017	2	1
4	3205	656	14/11/2017	1	3

KundeID	Fakturanr	ProduktID	Antall
1	3201	1	2
1	3201	2	1
1	3201	3	1
2	3202	1	4
3	3203	4	10
3	3203	1	5
1	3204	2	1
4	3205	1	3

Hva kjennetegner god relasjonsdatabasedesign?

A. Relasjonene samler beslektet informasjon
B. Så lite dobbeltlagring som mulig
C. Så få «glisne» relasjoner som mulig

Vi kan endre på skjemaet for å få det til, men:

D. Korrekt totalinformasjon kan gjenskapes nøyaktig ved join

Eksempel:
Grossistdatabase versjon I (GDB1)

Produkt(Kode, Produktnavn, Produsent, AntEnheter)
Bestilling(Kode, Kundenr, Navn, Adresse, AntBestilt)

Integritetsregler i tillegg til primærnøklene:

A. Til hvert kundenummer (Kundenr) skal det bare være ett navn og én adresse
B. Kode i Bestilling er fremmednøkkel til Kode i Produkt

A. Relasjonene samler beslektet informasjon:

- Tekstlig nærhet skal gjenspeile logisk nærhet (Med tekstlig nærhet menes her samlokalisering i en relasjon)
- Brudd på dette prinsippet har en tendens til å påtvinge duplisering av data innen en tabell og dermed forårsake **oppdateringsanomalier**

Eksempel på Bestilling

Bestilling

Kode	Kundenr	Navn	Adresse	AntBestilt
1	1	A	a	3
2	1	A	a	8
1	2	B	b	2

Innsettingsanomali

Bestilling

Kode	Kundenr	Navn	Adresse	AntBestilt
1	1	A	a	3
2	1	A	a	8
1	2	B	b	2
2	**2**	**B**	**c**	1

Kunde 2 har bestilt noe. Men nå har han en ny adresse!?

Slettingsanomali

Bestilling

Kode	Kundenr	Navn	Adresse	AntBestilt
1	1	A	a	3
2	1	A	a	8

Ordre 1 av kunde 1 ble kansellert. Hva heter hun og hvor bor hun?

Modifikasjonsanomali

Bestilling

Kode	Kundenr	Navn	Adresse	AntBestilt
1	1	A	**c**	3
2	1	A	a	8
1	2	B	b	2

Kunde 1 har flyttet – men vi har ikke oppdatert adressen overalt!

Sekundær Informasjon

Bestilling

Kode	Kundenr	Navn	Adresse	AntBestilt
1	1	A	a	3
2	1	A	a	8
1	2	B	b	2

Sekundær informasjon

«Navn» og «Adresse» er eksempler på **sekundær informasjon**:
Dette er nyttig informasjon om kundene, men den passer ikke naturlig inn i en tabell over bestillingene

B. Så lite dobbeltlagring som mulig:

- Oppdatering forenkles
- Plassbehovet minimaliseres

C. Så få "glisne" relasjoner som mulig:

- Unngår problemer med hvordan join på nil-verdier skal håndteres
- Unngår problemer med hvordan aggregeringsfunksjoner skal håndtere nil-verdier
- Plassbehovet minimaliseres

Hvordan unngå dobbeltlagring

- Splitt (dekomponer) relasjonene slik at dobbeltlagring blir borte!
  - Instanser for de dekomponerte relasjonene fremkommer fra opprinnelig instans ved projeksjon

Eksempel:
Grossistdatabase versjon 2 (GDB2)

Produkt(Kode, Produktnavn, Produsent, AntEnheter)

Kunde(Kundenr, Navn, Adresse)

Ordre(Kode, Kundenr, AntBestilt)

Integritetsregler i tillegg til primærnøklene:

- Kode i Ordre er fremmednøkkel til Produkt
- Kundenr i Ordre er fremmednøkkel til Kunde

Eksempelinstanser
Kunde, Ordre

Gammel tabell

Bestilling

Kode	Kundenr	Navn	Adresse	AntBestilt
1	1	A	a	3
2	1	A	a	8
1	2	B	b	2

Nye tabeller

Kunde

Kundenr	Navn	Adresse
1	A	a
2	B	b

Ordre

Kode	Kundenr	AntBestilt
1	1	3
2	1	8
1	2	2

Krav til dekomposisjoner

- Vi ønsker å kunne rekonstruere den opprinnelige instansen
- Dekomposisjon av relasjoner må derfor gjøres på en måte som sikrer at vi alltid kan gjenskape den opprinnelige instansen ved **join**
- Hvis vi ikke omdøper attributtene: **naturlig join**

Naturlig join angis med symbolet $\bowtie$

Kunde ⋈ Ordre

Kunde

Kundenr	Navn	Adresse
1	A	a
2	B	b

Ordre

Kode	Kundenr	AntBestilt
1	1	3
2	1	8
1	2	2

Kunde ⋈ Ordre = Bestilling

Kundenr	Navn	Adresse	Kode	AntBestilt
1	A	a	1	3
1	A	a	2	8
2	B	b	1	2

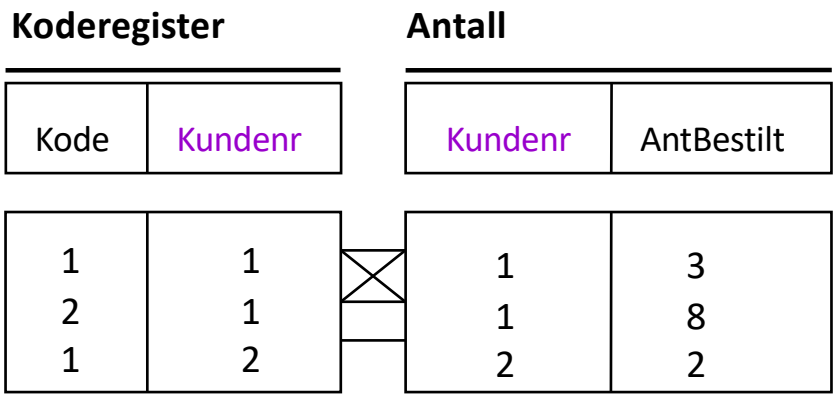
Eksempel:
Grossistdatabase, versjon 3 (GDB3)

Produkt(Kode, Produktnavn, Produsent, AntEnheter)
Kunde(Kundenr, Navn, Adresse)
Koderegister(Kode, Kundenr)
Antall(Kundenr, AntBestilt)

Integritetsregler:

- Kode i Koderegister er fremmednøkkel til Produkt
- Kundenr i Koderegister er fremmednøkkel til Kunde
- Kundenr i Antall er fremmednøkkel til Kunde

Eksempelinstanser
Koderegister, Antall



Kode	Kundenr	AntBestilt
1	1	3
1	1	8
2	1	8
2	1	3
1	2	2

Naturlig join på de to tabellene gir flere tupler
enn i den opprinnelige tabellen!
= **Falske tupler**

D. Korrekt totalinformasjon kan gjenskapes nøyaktig ved join:

- Ingen falske tupler genereres

Dekomposisjon

- Dekomposisjon bryter ned en relasjon i flere mindre relasjoner.
- Den bryter ned en tabelle i flere tabeller.
- Det skal være mulig å rekonstruere innholdet i den opprinnelige tabllen fra komponentene

- Dekomposisjon flytter sekundær informasjon fra en tabelle
- Den kan altså forhindre oppdateringsanomalier
- Og dermed forbedre et databaseskjema

Repetisjon Nøkler

Nøkler

- X er en **supernøkkel** i R hvis $X \subseteq R$, og ingen instans av R får inneholde to forskjellige tupler t_1 og t_2 hvor $t_1[X] = t_2[X]$
- X er en **kandidatnøkkel** i R hvis X er en supernøkkel i R, og for alle A i X er X−A ikke en supernøkkel i R (dvs. X er en minimal supernøkkel)
- En **primærnøkkel** X er en spesielt utpekt kandidatnøkkel i R

Eksempel:

Filmgenre(<u>filmid, genre</u>, title)

filmid	genre	title
85908	Action	The Matrix
85908	Sci-Fi	The Matrix
85908	Thriller	The Matrix
26103	Action	Planet of the Apes
26103	Sci-Fi	Planet of the Apes
1320611	Action	Planet of the Apes
1320611	Sci-Fi	Planet of the Apes

Supernøkler: {filmid, genre},
{filmid, genre, title}

Kandidatnøkkel: {filmid, genre}

Primærnøkkel: {filmid, genre}

Nøkkelattributt

- Et **nøkkelattributt** er et attributt som er med i en kandidatnøkkel

- Et ikke-nøkkelattributt er et attributt som *ikke* er med i noen kandidatnøkkel

Eksempel:

Filmgenre(filmid, genre, title)

filmid	genre	title
85908	Action	The Matrix
85908	Sci-Fi	The Matrix
85908	Thriller	The Matrix
26103	Action	Planet of the Apes
26103	Sci-Fi	Planet of the Apes
1320611	Action	Planet of the Apes
1320611	Sci-Fi	Planet of the Apes

Kandidatnøkkel: {filmid, genre}

Nøkkelattributter: filmid, genre

Ikke-nøkkelattributt: title

Eksempel 2

Student(<u>fnr</u>, <u>id</u>, navn, adresse)

Primærnøkkel: fnr

Kandidatnøkler: fnr, id

Supernøkler: 12 stk

{fnr}, {fnr,id}, {fnr,navn}, {fnr,adresse}, {fnr, navn, adresse}, {fnr,id,navn}, {fnr,id,adresse}, {fnr,id,navn,adresse}, {id}, {id,navn}, {id,adresse}, {id,navn,adresse}

Nøkkelattributter: fnr, id

Ikke-nøkkelattributter: navn, adresse

Repetisjon:
Funksjonelle avhengigheter

Funksjonelle avhengigheter

- Y er **funksjonelt avhengig** av X hvis vi for enhver lovlig instans av R har at hvis instansen inneholder to tupler t_1 og t_2 hvor $t_1[X] = t_2[X]$, så må $t_1[Y] = t_2[Y]$
  - Da skriver vi $X \rightarrow Y$
- Omtales også som **FD-er** (**f**unctional **d**ependencies)
- Vi sier at «Y følger av X» eller «X bestemmer Y»

Eksempel:

Person(PID, Navn, Postnr, Poststed)

PID	Navn	Postnr	Poststed
1	Ola	0372	OSLO
2	Kari	5006	BERGEN
3	Per	0372	OSLO
4	Ola	1383	ASKER
5	Jo	1384	ASKER
6	Nils	0372	OSLO
7	Stein	0010	OSLO

Funksjonelle avhengigheter:

- Postnr → Poststed

Funksjonelle avhengigheter

- Merk: Hvis X er en supernøkkel, så holder $X \rightarrow Y$ for enhver Y
  - Altså: Hvis X er en primærnøkkel/kandidatnøkkel, holder $X \rightarrow Y$ for alle Y
- Og motsatt: Hvis $X \rightarrow Y$ for enhver Y, så er X en kandidatnøkkel

- FD-en $X \rightarrow Y_1, Y_2, \ldots, Y_n$ kan også representeres som FD-ene

  $X \rightarrow Y_1, \quad X \rightarrow Y_2, \quad \ldots, \quad X \rightarrow Y_n$

  (slik at høyresidene består bare av ett attributt)

Eksempel:

Person(<u>PID</u>, Navn, Postnr, Poststed)

PID	Navn	Postnr	Poststed
1	Ola	0372	OSLO
2	Kari	5006	BERGEN
3	Per	0372	OSLO
4	Ola	1383	ASKER
5	Jo	1384	ASKER
6	Nils	0372	OSLO
7	Stein	0010	OSLO

Funksjonelle avhengigheter:

- Postnr → Poststed
- PID → Navn, Postnr, Poststed

Alternativt skriver vi:

- Postnr → Poststed
- PID → Navn
- PID → Postnr
- PID → Poststed

Funksjonelle avhengigheter oppsummert

Enkel definisjon: Et attributt er «avhengig» av / bestemmes av et annet attributt

Eksempel: Student(<u>id</u>, <u>brukernavn</u>, navn, adresse, postnr, poststed)

Funksjonelle avhengigheter:

- id → brukernavn, navn, adresse, postnr, poststed
- brukernavn → id, navn, adresse, postnr, poststed

Ny integritetsregel: Et postnr bestemmer et poststed

- postnr → poststed

FD-er: Enda et eksempel

Ordre(ordrenr, kundenr, kundenavn, antall, sum, mva)

I tillegg har vi følgende integritetsregler:

- Ordrenr er unikt
- Kundenr bestemmer kundenavn
- Mva-verdi følger av sum

Ordrenr → kundenr, kundenavn, antall, sum, mva
Kundenr → kundenavn
Sum → mva

Trivielle FDer

- Hvis $Y \subseteq X$, så har vi *alltid* at $X \rightarrow Y$.

- For eksempel: Kundenr $\rightarrow$ Kundenr.
- Eller: Kundenr, Ordrenr $\rightarrow$ Kundenr.

- FDer hvor høyresiden er inneholdt i venstresiden, kalles **trivielle**.

$\frac{3}{2} \sim \frac{6}{4} \sim \frac{9}{6}$

$\frac{n}{a} \sim \frac{p}{q}$ hvis $n \cdot q = a \cdot p$

↑ Normalform

Normalformer

- har bestemt form
- alle br. er ~ med en i nevner form.

Normalformer

- Normalformer er et uttrykk for hvor godt vi har lykkes i en dekomposisjon
- Jo høyere normalform, jo færre oppdateringsanomalier
- Det finnes algoritmer for å omforme fra lavere til høyere normalformer

Utgangspunkt for normalformene 1NF-BCNF

- Alle integritetsregler er i form av FDer

  (i tillegg til domeneskranker og fremmednøkler)

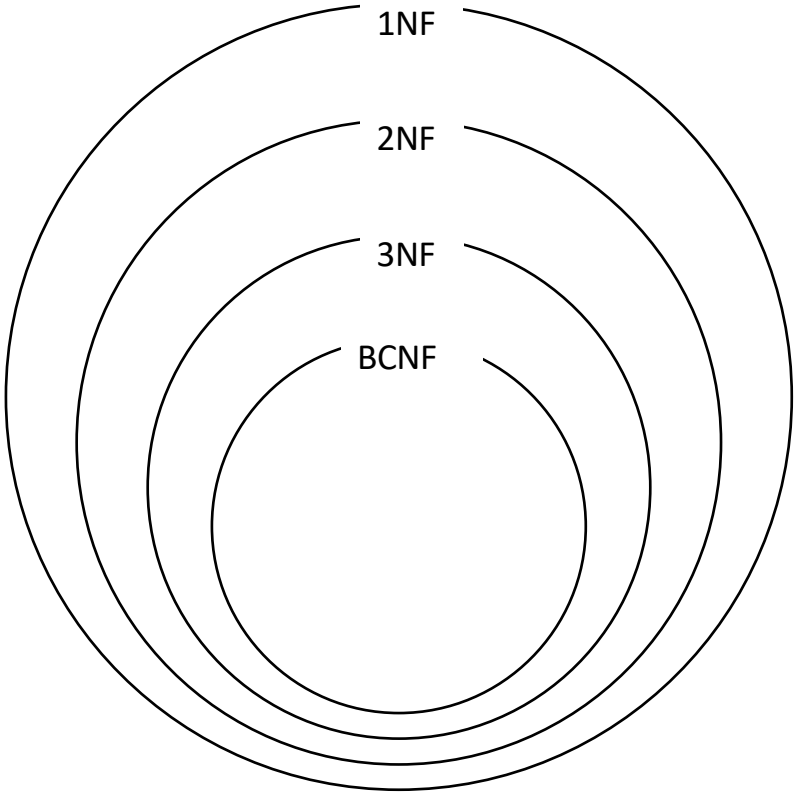
GDB1 med integritetsregler

Produkt(<u>Kode</u>, Produktnavn, Produsent, AntEnheter)
Bestilling(<u>Kode, Kundenr</u>, Navn, Adresse, AntBestilt)

Integritetsregler:

1. Kode $\rightarrow$ Produktnavn, Produsent, AntEnheter (i Produkt)
   fordi Kode er primærnøkkel i Produkt
2. Kode, Kundenr $\rightarrow$ Navn, Adresse, AntBestilt (i Bestilling)
   fordi (Kode, Kundenr) er primærnøkkel i Bestilling
3. Kundenr $\rightarrow$ Navn, Adresse (i Bestilling)
   fordi det til hver verdi av Kundenr er maksimalt én verdi i hver av Navn og Adresse (integritetsregel A på lysark 3)
4. Kode i Bestilling er fremmednøkkel til Produkt
   (integritetsregel B på lysark 3)

Normalformer, oversikt



Første normalform

- **Definisjon 1NF** (Codd 1972):
  - Alle domener består av atomære verdier
  - Verdien av et gitt attributt i et tuppel for et gitt attributt skal være en slik atomær verdi (eller nil)

Atomære verdier?

- Litt vagt hva «atomær verdi» betyr
  - Strenger består av tegn, men er vel OK!?
- Dårlige ideer:
  - Flere verdier i ett attributt
    - Attributt «Farger» med verdier «rødt, hvitt, blått»
  - Sammensatte verdier i ett attributt, når det er plausibelt at en vil trenge delene
    - Attributt «Navn» med verdi «Alf Prøysen»
    - Attributt «Adresse» med verdi «Postboks 1080 Blindern
N-0316 Oslo»

Andre normalform

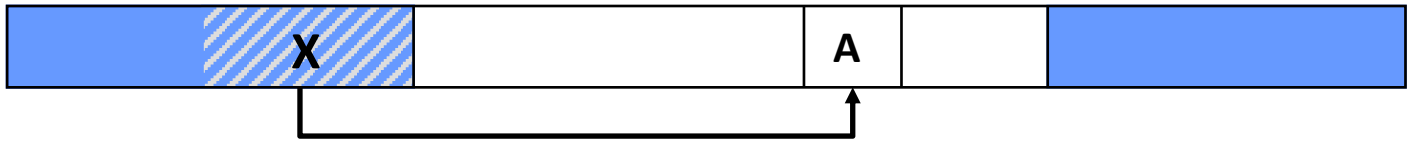
- En relasjon R er 2NF hvis enhver ikketriviell FD X→A tilfredsstiller minst ett av følgende tre krav:
  - X inneholder en kandidatnøkkel
  - A er et nøkkelattributt
  - Ingen kandidatnøkler inneholder X
- R bryter 2NF hvis det finnes en ikketriviell FD X→A hvor A er et ikke-nøkkelattributt og det finnes en kandidatnøkkel W slik at $X \subset W$ (ekte delmengde, dvs. X er inneholdt i, men ikke lik, W).

"**Ikketriviell FD X→A**" betyr at $X \subseteq R$ og $A \in R$, men $A \notin X$.

Brudd på andre normalform

FD X→A hvor X utgjør noen av, men ikke alle, attributtene i en av kandidatnøklene og A er et ikke-nøkkelattributt.

Eksempel:



X er skravert (lyseblått/grått).
Kandidatnøkler er markert med lyseblått.

Brudd på 2NF i GDB1

Produkt(Kode, Produktnavn, Produsent, AntEnheter)
Bestilling(Kode, Kundenr, Navn, Adresse, AntBestilt)

FDer:

1. **Kode → Produktnavn, Produsent, AntEnheter** (i Produkt)
   fordi Kode er primærnøkkel
2. **Kode, Kundenr → Navn, Adresse, AntBestilt** (i Bestilling)
   fordi (Kode, Kundenr) er primærnøkkel
3. **Kundenr → Navn, Adresse** (i Bestilling)
   fordi det til hver verdi av Kundenr er maksimalt én verdi i hver av Navn og Adresse
   Denne bryter 2NF (se forrige lysark), så Bestilling er på 1NF, men ikke 2NF.