Normalformer or Normalisering
1NF, 2NF, 3NF, BCNF

Martin Giese

7. november 2018

Agenda

- Nytt eksempel
  - Med funksjonelle avhengigheter
- 1NF (veldig kort)
- 2NF, Grundig
  - Hva er vitsen? – anomalier
  - Få eksemplet til 2NF
  - Hva skjedde med anomaliene?
  - Hvordan går det generelt?
- 3NF
- BCNF

Eksempel: Bestillinger

Bestilling(<u>BestNr, ProdNr</u>, KundeNr, KundeAdr, AntBestilt)

Integritetsregler:

1. BestNr, ProdNr → KundeNr, KundeAdr, AntBestilt
   fordi (BestNr, ProdNr) er primærnøkkel i Bestilling
2. BestNr→KundeNr: hele bestilling for samme kunde
3. KundeNr→KundeAdr: kunde har kun en adresse
4. BestNr→KundeAdr: konsekvens av 2 og 3

5. ProdNr og KundeNr er fremmednøkler for andre relasjoner

Eksempeldata

Bestilling(<u>BestNr, ProdNr</u>, KundeNr, KundeAdr, AntBestilt)

BestNr	ProdNr	KundeNr	KundeAdr	AntBestilt
1	111	A	Askim	10
2	222	B	Bærum	20
2	202	B	Bærum	15
3	111	A	Askim	50
3	202	A	Askim	10

Sekundærinformasjon

Bestilling(<u>BestNr, ProdNr</u>, KundeNr, KundeAdr, AntBestilt)

BestNr	ProdNr	KundeNr	KundeAdr	AntBestilt
1	111	A	Askim	10
2	222	B	Bærum	20
2	202	B	Bærum	15
3	111	A	Askim	50
3	202	A	Askim	10

Sekundærinformasjon

Oppdateringsanomalier i eksemplet

- INSERT anomali:
  - Kan ikke starte med en «tom» bestilling uten et antall produkter
  - Kan ikke legge inn kundeadresse uten bestilling
- UPDATE anomalier:
  - Endring av kunden i en bestilling krever oppdatering flere steder
  - Endring av adressen til en kunde krever oppdatering flere steder
- DELETE anomali:
  - Sletting av en post i en bestilling kan medføre sletting av kundeinformasjon
  - Sletting av en bestilling kan medføre sletting av kundeinformasjon

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

Normalformer

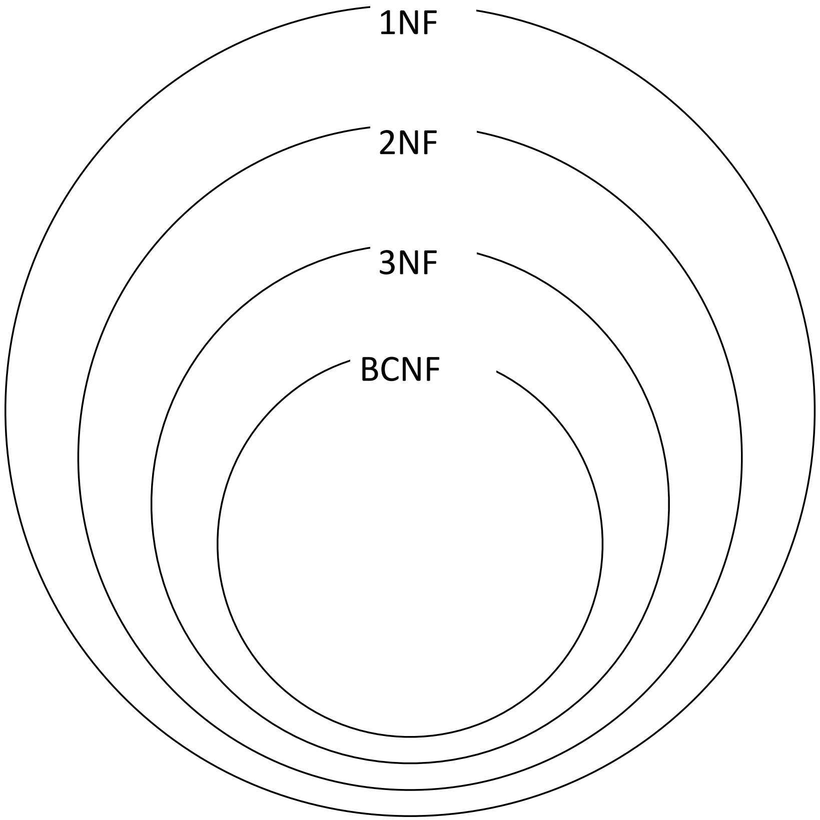
- Normalformer er et uttrykk for hvor godt vi har lykkes i en dekomposisjon
- Jo høyere normalform, jo færre oppdateringsanomalier
- Det finnes algoritmer for å omforme fra lavere til høyere normalformer

Utgangspunkt for normalformene 1NF–BCNF

- Alle integritetsregler er i form av FDer

  (i tillegg til domeneskranker og fremmednøkler)

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)

Andre normalform

- En relasjon R er 2NF hvis enhver ikketriviell FD $X \rightarrow A$ tilfredsstiller minst ett av følgende tre krav:
  - X inneholder en kandidatnøkkel
  - A er et nøkkelattributt
  - Ingen kandidatnøkler inneholder X

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

Hvordan ser brudd mot 2NF ut?

- Anta at en relasjon R ikke er 2NF.

Hvordan ser brudd mot 2NF ut?

- Anta at en relasjon R ikke er 2NF.
- Finnes en ikketriviell FD X→A som *ikke* tilfredsstiller *noen av* disse:
  - X inneholder en kandidatnøkkel
  - A er et nøkkelattributt
  - Ingen kandidatnøkler inneholder X

Hvordan ser brudd mot 2NF ut?

- Anta at en relasjon R ikke er 2NF.
- Finnes en ikketriviell FD $X \rightarrow A$ som *ikke* tilfredsstiller *noen av* disse:
  - X inneholder en kandidatnøkkel
  - A er et nøkkelattributt
  - Ingen kandidatnøkler inneholder X
- Finnes en ikketriviell FD $X \rightarrow A$ slik att
  - X inneholder ingen kandidatnøkkel, OG
  - A er ikke-nøkkelattributt, OG
  - Det finnes en kandidatnøkkel W som inneholder X, $X \subseteq W$

Hvordan ser brudd mot 2NF ut?

- Anta at en relasjon R ikke er 2NF.
- Finnes en ikketriviell FD $X \rightarrow A$ slik att
  - X inneholder ingen kandidatnøkkel, OG
  - A er ikke-nøkkelattributt, OG
  - Det finnes en kandidatnøkkel W som inneholder X, $X \subseteq W$

Hvordan ser brudd mot 2NF ut?

- Anta at en relasjon R ikke er 2NF.
- Finnes en ikketriviell FD $X \rightarrow A$ slik att
  - X inneholder ingen kandidatnøkkel, OG
  - A er ikke-nøkkelattributt, OG
  - Det finnes en kandidatnøkkel W som inneholder X, $X \subseteq W$
- Finnes en ikketriviell FD $X \rightarrow A$ slik att
  - A er ikke-nøkkelattributt,
  - Det finnes en kandidatnøkkel W som strengt inneholder X, $X \subset W$

Hvordan ser brudd mot 2NF ut?

- Anta at en relasjon R ikke er 2NF.
- Finnes en ikketriviell FD $X \rightarrow A$ slik att
  - X inneholder ingen kandidatnøkkel, OG
  - A er ikke-nøkkelattributt, OG
  - Det finnes en kandidatnøkkel W som inneholder X, $X \subseteq W$
- Finnes en ikketriviell FD $X \rightarrow A$ slik att
  - A er ikke-nøkkelattributt,
  - Det finnes en kandidatnøkkel W som strengt inneholder X, $X \subset W$
    - (med $X = W$ hadde X vært kandidatnøkkel)

Omvendt…

- Anta en ikketriviell FD $X \rightarrow A$ slik att
  - A er ikke-nøkkelattributt
  - Det finnes en kandidatnøkkel W som strengt inneholder X, $X \subset W$
- Siden W er kandidatnøkkel er ingen delmengde av W det

Omvendt…

- Anta en ikketriviell FD $X \rightarrow A$ slik att
  - A er ikke-nøkkelattributt
  - Det finnes en kandidatnøkkel W som strengt inneholder X, $X \subset W$
- Siden W er kandidatnøkkel er ingen delmengde av W det
- Spesielt inneholder ikke X en kandidatnøkkel.

Omvendt…

- Anta en ikketriviell FD $X \rightarrow A$ slik att
  - A er ikke-nøkkelattributt
  - Det finnes en kandidatnøkkel W som strengt inneholder X, $X \subset W$
- Siden W er kandidatnøkkel er ingen delmengde av W det
- Spesielt inneholder ikke X en kandidatnøkkel.
- Det finnes altså en ikketriviell FD $X \rightarrow A$ som *ikke* tilfredsstiller *noen av* disse:
  - X inneholder en kandidatnøkkel
  - A er et nøkkelattributt
  - Ingen kandidatnøkler inneholder X

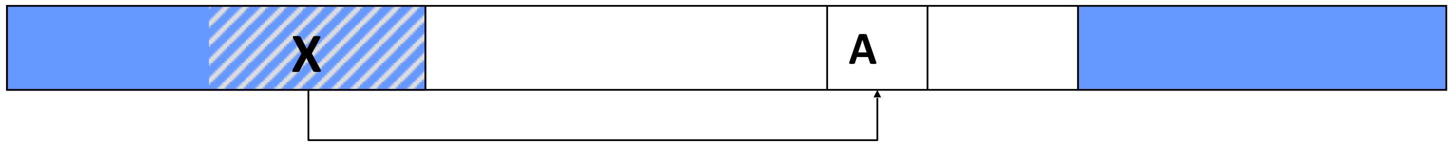
Omvendt…

- Anta en ikketriviell FD X→A slik att
  - A er ikke-nøkkelattributt
  - Det finnes en kandidatnøkkel W som strengt inneholder X, $X \subset W$
- Siden W er kandidatnøkkel er ingen delmengde av W det
- Spesielt inneholder ikke X en kandidatnøkkel.
- Det finnes altså en ikketriviell FD X→A som *ikke* tilfredsstiller *noen av* disse:
  - X inneholder en kandidatnøkkel
  - A er et nøkkelattributt
  - Ingen kandidatnøkler inneholder X
- Derfor er ikke relasjonen i 2NF.

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 Eksemplet

Bestilling(<u>BestNr, ProdNr</u>, KundeNr, KundeAdr, AntBestilt)

FDer:

1. BestNr, ProdNr → KundeNr, KundeAdr, AntBestilt
   fordi (BestNr, ProdNr) er primærnøkkel i Bestilling
2. BestNr→KundeNr: hele bestilling for samme kunde
   Brudd på 2NF!
3. KundeNr→KundeAdr: kunde har kun en adresse
4. BestNr→KundeAdr: konsekvens av 2 og 3
   Brudd på 2NF!

Normalisering

- **Normalisering** går ut på å dekomponere relasjoner med lav normalform til relasjoner med høyere normalform.
- **Regel:** Gitt en relasjon R(XYZ) med en FD X→Y.
  Hvis R dekomponeres til S(XY), T(XZ), vil vi *aldri* kunne få falske tupler ved naturlig join av S og T.
  - Hvis vi dekomponerer R(XYZ) til S(XY), T(XZ) og det *ikke* er slik at X→Y holder, vil vi generelt få falske tupler ved naturlig join av S og T.

Her uttrykker R(XYZ) at attributtene i R kan deles inn i tre (ikketomme) disjunkte mengder X, Y og Z.

Hvorfor?

- Gitt $R(X,Y,Z)$ med FD $X \rightarrow Y$, og *projeksjoner* $S(X,Y)$, $T(X,Z)$
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$

Hvorfor?

- Gitt $R(X,Y,Z)$ med FD $X \rightarrow Y$, og *projeksjoner* $S(X,Y)$, $T(X,Z)$
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join bevarer alle tupler i R:
  - $R(X,Y,Z) \supseteq S(X,Y) \bowtie T(X,Z)$

Hvorfor?

- Gitt R(X,Y,Z) med FD X→Y, og *projeksjoner* S(X,Y), T(X,Z)
  - <x,y> ∈ S(X,Y) hvis og bare hvis ∃z′ med <x,y,z′> ∈ R(X,Y,Z)
  - <x,z> ∈ T(X,Z) hvis og bare hvis ∃y′ med <x,y′,z> ∈ R(X,Y,Z)
- Naturlig join bevarer alle tupler i R:
  - R(X,Y,Z) ⊇ S(X,Y) ⋈ T(X,Z)
- La <x,y,z> ∈ R(X,Y,Z)

Hvorfor?

- Gitt $R(X,Y,Z)$ med FD $X \rightarrow Y$, og *projeksjoner* $S(X,Y)$, $T(X,Z)$
    - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
    - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join bevarer alle tupler i R:
    - $R(X,Y,Z) \supseteq S(X,Y) \bowtie T(X,Z)$
- La $\langle x,y,z \rangle \in R(X,Y,Z)$
    - Da er $\langle x,y \rangle \in S(X,Y)$ og $\langle x,z \rangle \in T(X,Z)$

Hvorfor?

- Gitt R(X,Y,Z) med FD $X \rightarrow Y$, og *projeksjoner* S(X,Y), T(X,Z)
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join bevarer alle tupler i R:
  - $R(X,Y,Z) \supseteq S(X,Y) \bowtie T(X,Z)$
- La $\langle x,y,z \rangle \in R(X,Y,Z)$
  - Da er $\langle x,y \rangle \in S(X,Y)$ og $\langle x,z \rangle \in T(X,Z)$
  - Naturlig join bruker X som join attributt

Hvorfor?

- Gitt R(X,Y,Z) med FD $X \rightarrow Y$, og *projeksjoner* S(X,Y), T(X,Z)
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join bevarer alle tupler i R:
  - $R(X,Y,Z) \supseteq S(X,Y) \bowtie T(X,Z)$
- La $\langle x,y,z \rangle \in R(X,Y,Z)$
  - Da er $\langle x,y \rangle \in S(X,Y)$ og $\langle x,z \rangle \in T(X,Z)$
  - Naturlig join bruker X som join attributt
  - Så $\langle x,y,z \rangle \in S(X,Y) \bowtie T(X,Z)$

Hvorfor?

- Gitt R(X,Y,Z) med FD $X \rightarrow Y$, og *projeksjoner* S(X,Y), T(X,Z)
    - $<x,y> \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $<x,y,z'> \in R(X,Y,Z)$
    - $<x,z> \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $<x,y',z> \in R(X,Y,Z)$
- Naturlig join gir ikke flere tupler enn R:
    - $R(X,Y,Z) \subseteq S(X,Y) \bowtie T(X,Z)$

Hvorfor?

- Gitt $R(X,Y,Z)$ med FD $X \rightarrow Y$, og *projeksjoner* $S(X,Y)$, $T(X,Z)$
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join gir ikke flere tupler enn R:
  - $R(X,Y,Z) \subseteq S(X,Y) \bowtie T(X,Z)$
- La $\langle x,y,z \rangle \in S(X,Y) \bowtie T(X,Z)$

Hvorfor?

- Gitt R(X,Y,Z) med FD $X \rightarrow Y$, og *projeksjoner* S(X,Y), T(X,Z)
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join gir ikke flere tupler enn R:
  - $R(X,Y,Z) \subseteq S(X,Y) \bowtie T(X,Z)$
- La $\langle x,y,z \rangle \in S(X,Y) \bowtie T(X,Z)$
  - Da er $\langle x,y \rangle \in S(X,Y)$ og $\langle x,z \rangle \in T(X,Z)$

Hvorfor?

- Gitt $R(X,Y,Z)$ med FD $X \rightarrow Y$, og *projeksjoner* $S(X,Y)$, $T(X,Z)$
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join gir ikke flere tupler enn R:
  - $R(X,Y,Z) \subseteq S(X,Y) \bowtie T(X,Z)$
- La $\langle x,y,z \rangle \in S(X,Y) \bowtie T(X,Z)$
  - Da er $\langle x,y \rangle \in S(X,Y)$ og $\langle x,z \rangle \in T(X,Z)$
  - Da finnes y' og z' med $\langle x,y,z' \rangle \in R(X,Y,Z)$ og med $\langle x,y',z \rangle \in R(X,Y,Z)$

Hvorfor?

- Gitt R(X,Y,Z) med FD X→Y, og *projeksjoner* S(X,Y), T(X,Z)
  - <x,y> ∈ S(X,Y) hvis og bare hvis ∃z′ med <x,y,z′> ∈ R(X,Y,Z)
  - <x,z> ∈ T(X,Z) hvis og bare hvis ∃y′ med <x,y′,z> ∈ R(X,Y,Z)
- Naturlig join gir ikke flere tupler enn R:
  - R(X,Y,Z) ⊆ S(X,Y) ⋈ T(X,Z)
- La <x,y,z> ∈ S(X,Y) ⋈ T(X,Z)
  - Da er <x,y> ∈ S(X,Y) og <x,z> ∈ T(X,Z)
  - Da finnes y′ og z′ med <x,y,z′> ∈ R(X,Y,Z) og med <x,y′,z> ∈ R(X,Y,Z)
  - Husk FD X→Y: lik X medfører lik Y: y = y′!

Hvorfor?

- Gitt $R(X,Y,Z)$ med FD $X \rightarrow Y$, og *projeksjoner* $S(X,Y)$, $T(X,Z)$
  - $<x,y> \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $<x,y,z'> \in R(X,Y,Z)$
  - $<x,z> \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $<x,y',z> \in R(X,Y,Z)$
- Naturlig join gir ikke flere tupler enn R:
  - $R(X,Y,Z) \subseteq S(X,Y) \bowtie T(X,Z)$
- La $<x,y,z> \in S(X,Y) \bowtie T(X,Z)$
  - Da er $<x,y> \in S(X,Y)$ og $<x,z> \in T(X,Z)$
  - Da finnes y' og z' med $<x,y,z'> \in R(X,Y,Z)$ og med $<x,y',z> \in R(X,Y,Z)$
  - Husk FD $X \rightarrow Y$: lik X medfører lik Y: $y = y'$!
  - Dermed: $<x,y,z> \in R(X,Y,Z)$

Hvorfor?

- Gitt $R(X,Y,Z)$ med FD $X \rightarrow Y$, og *projeksjoner* $S(X,Y)$, $T(X,Z)$
  - $\langle x,y \rangle \in S(X,Y)$ hvis og bare hvis $\exists z'$ med $\langle x,y,z' \rangle \in R(X,Y,Z)$
  - $\langle x,z \rangle \in T(X,Z)$ hvis og bare hvis $\exists y'$ med $\langle x,y',z \rangle \in R(X,Y,Z)$
- Naturlig join gir nøyaktig de samme tuplene som R:
  - $R(X,Y,Z) = S(X,Y) \bowtie T(X,Z)$

Bruk av regelen på Eksemplet

Bestilling(BestNr, ProdNr, KundeNr, KundeAdr, AntBestilt)

FD:

- **BestNr → KundeNr**
  - S(BestNr, KundeNr), T(BestNr, ProdNr, KundeAdr, AntBestilt)
  - BstKnd(BestNr, KundeNr), Best(BestNr, ProdNr, KundeAdr…)
- **BestNr → KundeAdr**
  - BstKnd(BestNr, KundeAdr), Best(…, KundeNr, AntBestilt)
- **BestNr → KundeNr, KundeAdr**
  - BestKunde(BestNr, KundeNr, KundeAdr)
  - BestAnt(BestNr, ProdNr, AntBestilt)

Anomalier fikset?

BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)
BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)

- INSERT anomali:
  - Kan starte med en «tom» bestilling uten et antall produkter ✅
  - Kan ikke legge inn kundeadresse uten bestilling ❌
- UPDATE anomalier:
  - Endring av kunden i en bestilling krever oppdatering ett sted ✅
  - Endring av adressen til en kunde krever oppdatering flere steder ❌
- DELETE anomali:
  - Sletting av en post i en bestilling kan ikke medføre sletting av kundeinformasjon ✅
  - Sletting av en bestilling kan medføre sletting av kundeinformasjon ❌

Normalisering til 2NF

- Husk brudd på 2NF:
  - FD X→A hvor X utgjør noen av, men ikke alle, attributtene i en av kandidatnøklene og A er et ikke-nøkkelattributt.
- Observasjon: det er bare endelig mange ikke-nøkkelattributter
  - Kan vi fikse brudd for ikke-nøkkelattributter ett og ett?
  - Må da sjekke at dekomponering ikke fører til nye brudd…

Normalisering til 2NF (forts.)

- La X→A være et brudd mot 2NF:
  - FD X→A hvor X utgjør noen av, men ikke alle, attributtene i en av kandidatnøklene og A er et ikke-nøkkelattributt.
- Men «minimalt», altså ikke X′ →A for X′ ⊂ X
- R(<u>W,X</u>,A,Y) med kandidatnøkkel WX, og minimal FD X→A
- Dekomponer til S(<u>X</u>,A) og T(<u>W,X</u>,Y) etter regelen
  - S er nå i 2NF, siden X er kandidatnøkkel for S.
  - T inneholder ikke lenger A, fortsett med flere brudd for andre ikke-nøkkelattributter i Y

Tredje normalform

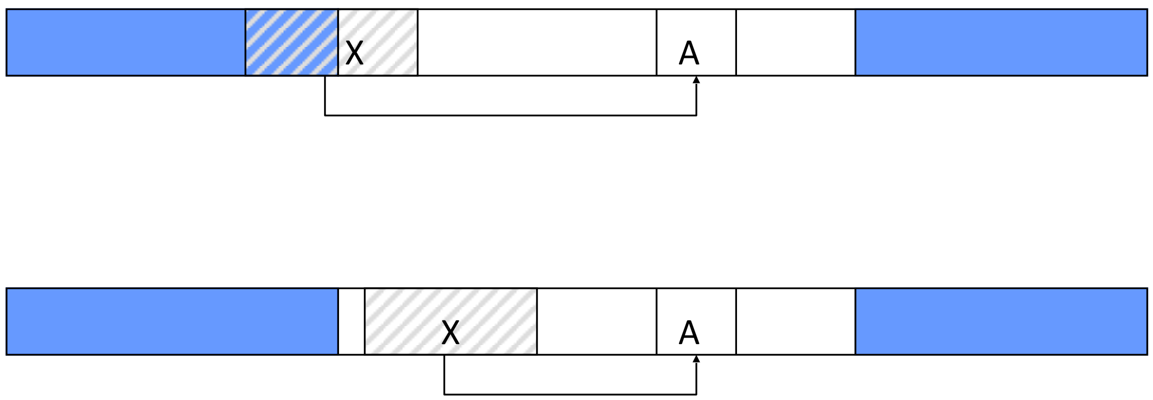
- En relasjon R er 3NF hvis enhver ikketriviell FD X→A tilfredsstiller minst ett av følgende to krav:
  - X inneholder en kandidatnøkkel
  - A er et nøkkelattributt

- R bryter 3NF hvis det finnes en ikketriviell FD X→A hvor X ikke inneholder noen kandidatnøkkel og A er et ikke-nøkkelattributt.

Brudd på tredje, men ikke andre, normalform

FD X→A hvor ingen kandidatnøkler inneholder X, X ikke inneholder noen kandidatnøkkel, og A er et ikke-nøkkelattributt.

Eksempler:



Eksempel på brudd mot 3NF

Eksempel på brudd mot 3NF

- Eksemplet etter normalisering til 2NF:

  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)

Eksempel på brudd mot 3NF

- Eksemplet etter normalisering til 2NF:
  - BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)
  - BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Fortsatt FD KundeNr→KundeAdr i BestKunde!
  - Ikke brudd mot 2NF
  - Ikketriviell, KundeAdr er ikke-nøkkelattributt, ingen kandidatnøkler inneholder Adresse.
  - Brudd mot 3NF

Videre dekomponering

Videre dekomponering

- I 2NF:

  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)

Videre dekomponering

- I 2NF:

  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Bruke samme regel med KundeNr→KundeAdr

Videre dekomponering

- I 2NF:

  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Bruke samme regel med KundeNr$\rightarrow$KundeAdr

  Kunde(<u>KundeNr</u>,KundeAdr)

Videre dekomponering

- I 2NF:
  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)
  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Bruke samme regel med KundeNr→KundeAdr
  Kunde(<u>KundeNr</u>,KundeAdr)
  BestKunde(<u>BestNr</u>, KundeNr)

Videre dekomponering

- I 2NF:

  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)

- Bruke samme regel med KundeNr→KundeAdr

  Kunde(<u>KundeNr</u>,KundeAdr)

  BestKunde(<u>BestNr</u>, KundeNr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)

Videre dekomponering

- I 2NF:
  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)
  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Bruke samme regel med KundeNr→KundeAdr
  Kunde(<u>KundeNr</u>,KundeAdr)
  BestKunde(<u>BestNr</u>, KundeNr)
  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Denne er i 3NF

Videre dekomponering

- I 2NF:
  - BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)
  - BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Bruke samme regel med KundeNr→KundeAdr
  - Kunde(<u>KundeNr</u>,KundeAdr)
  - BestKunde(<u>BestNr</u>, KundeNr)
  - BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)
- Denne er i 3NF

Videre dekomponering

- I 2NF:

  BestKunde(<u>BestNr</u>, KundeNr, KundeAdr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)

- Bruke samme regel med KundeNr→KundeAdr

  Kunde(<u>KundeNr</u>,KundeAdr)

  BestKunde(<u>BestNr</u>, KundeNr)

  BestAnt(<u>BestNr, ProdNr</u>, AntBestilt)

- Denne er i 3NF

Anomalier fikset?

Kunde(KundeNr,KundeAdr)
BestKunde(BestNr, KundeNr)
BestAnt(BestNr, ProdNr, AntBestilt)

- INSERT anomali:
  - Kan starte med en «tom» bestilling uten et antall produkter ✅
  - Kan legge inn kundeadresse uten bestilling ✅
- UPDATE anomalier:
  - Endring av kunden i en bestilling krever oppdatering ett sted ✅
  - Endring av adressen til en kunde krever oppdatering ett sted ✅
- DELETE anomali:
  - Sletting av en post i en bestilling kan ikke medføre sletting av kundeinformasjon ✅
  - Sletting av en bestilling kan ikke medføre sletting av kundeinformasjon ✅

Boyce-Codd normalform

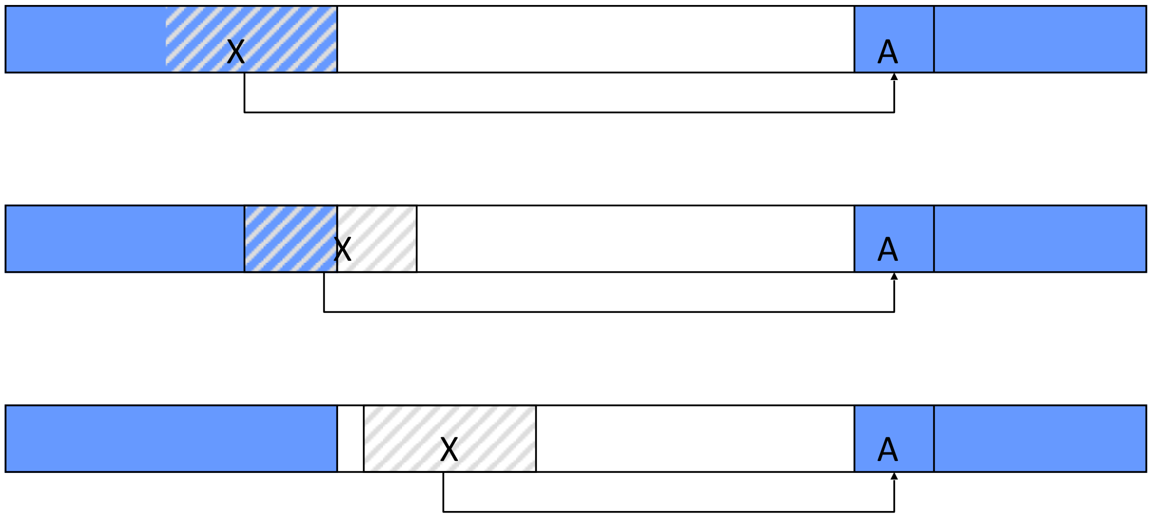
- En relasjon R er BCNF hvis enhver ikketriviell FD X→A tilfredsstiller følgende krav:
  - X inneholder en kandidatnøkkel

- R bryter BCNF hvis det finnes en ikketriviell FD X→A hvor X ikke inneholder noen kandidatnøkkel

Brudd på Boyce-Codd, men ikke tredje, normalform

FD X→A hvor A er et nøkkelattributt og X ikke inneholder noen kandidatnøkkel.

Eksempler:



Normalisering til 3NF

- Det lar seg alltid gjøre å normalisere (dekomponere) til 3NF
- Men hvis en relasjon er på 3NF og ikke på BCNF, betyr det at det kan finnes noen funksjonelle avhengigheter som må sjekkes ved alle innsettinger og oppdateringer
- (I tillegg må selvfølgelig primær- og kandidatnøkler alltid sjekkes)

Normalisering til BCNF

- Det lar seg alltid gjøre å normalisere (dekomponere) til BCNF
- Men hvis vi har dekomponert til BCNF, kan det være at vi etterpå har noen funksjonelle avhengigheter som går på tvers av relasjonene, og der vi ved alle innsettinger og oppdateringer må joine de involverte relasjonene for å kunne sjekke at de funksjonelle avhengighetene fortsatt er oppfylt.

Normalisering til 3NF vs. BCNF

- Vi kan alltid normalisere til 3NF uten å få funksjonelle avhengigheter på tvers av relasjonene.
- Derfor vil man som regel nøye seg med å normalisere bare til 3NF i de tilfellene hvor alternativet er normalisering til BCNF med funksjonelle avhengigheter på tvers.
- I praksis har de fleste relasjoner ikke så mange ulike kandidatnøkler. Derfor vil relasjoner i 3NF ofte allerede være i BCNF.

Normalformer

Gitt en relasjon R, med et sett FD-er på formen X → A, der X og A er et sett av attributter

1NF:

- R inneholder kun atomære verdier/attributter

2NF:

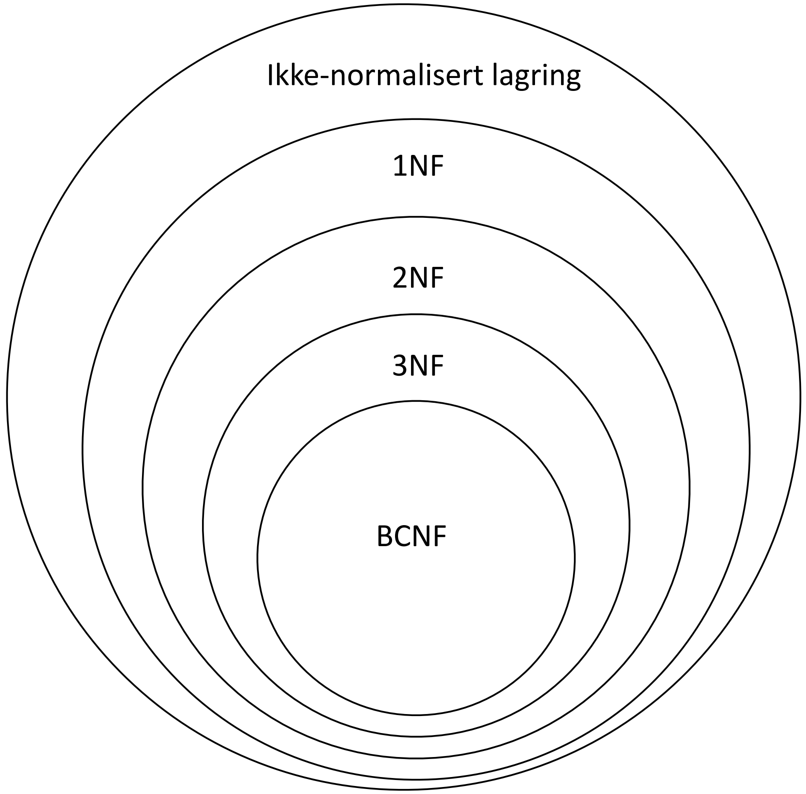
- X er en supernøkkel i R, eller
- A er et nøkkelattributt, eller
- X er *ikke* en delmengde av noen nøkler i R

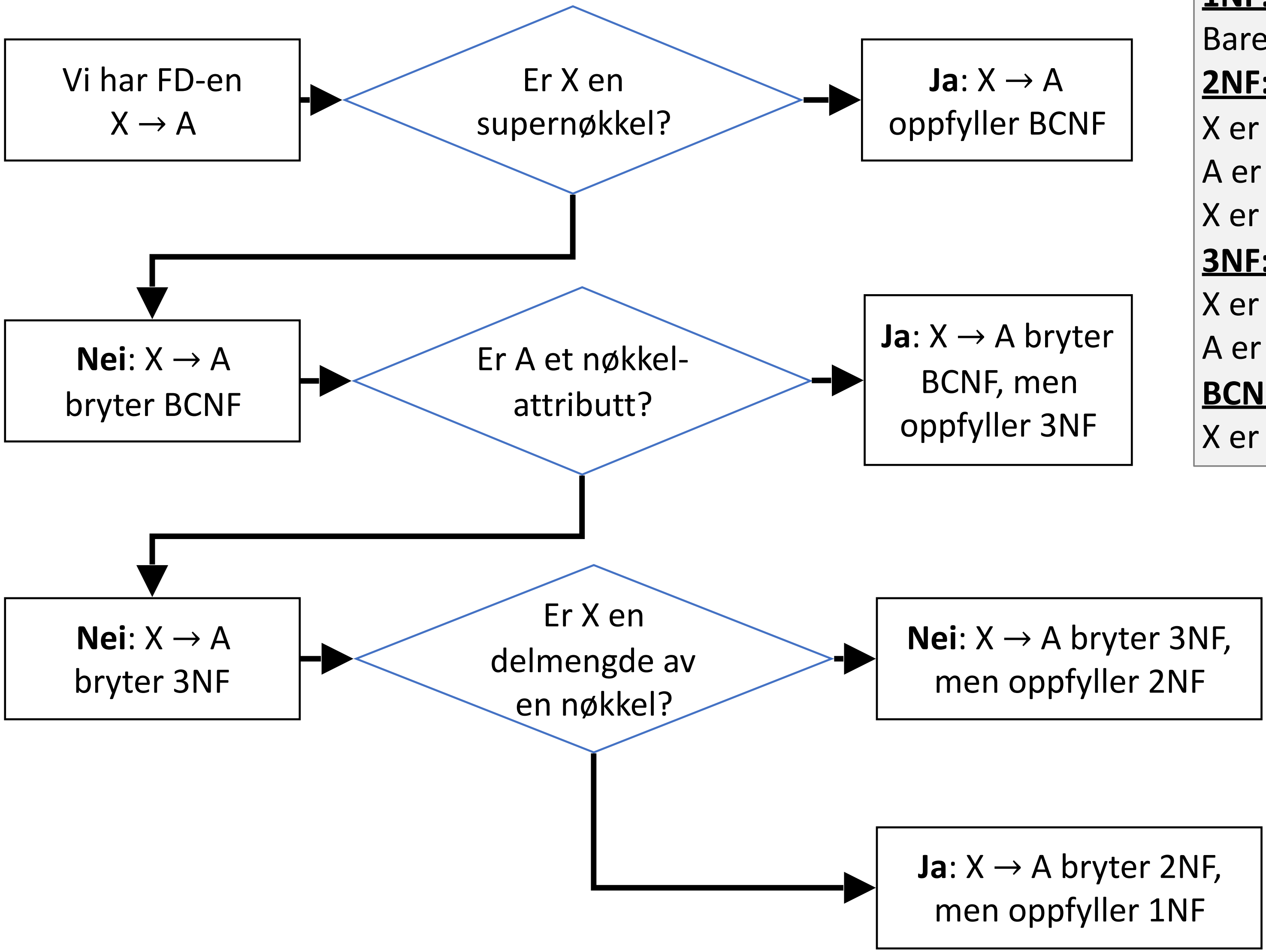
3NF:

- X er en supernøkkel i R, eller
- A er et nøkkelattributt

BCNF:

- X er en supernøkkel i R





1NF:
Bare atomære verdier/attributter
2NF:
X er en supernøkkel i R, eller
A er et nøkkelattributt, eller
X er *ikke* en delmengde av noen nøkler i R
3NF:
X er en supernøkkel i R, eller
A er et nøkkelattributt
BCNF:
X er en supernøkkel i R