

# Normalisering

# Læringsmål

Etter dagens forelesning skal du:

- forstå begrepet funksjonell avhengighet
- kunne finne og uttrykke funksjonelle avhengigheter i en relasjon
- forstå første, andre og tredje normalform, samt Boyce-Codd normalform, slik at du kan ...
  - beskrive hva som kjennetegner de forskjellige normalformene, og
  - gitt funksjonelle avhengigheter, normalisere en relasjonsdatabase slik at den oppfyller hver av disse.

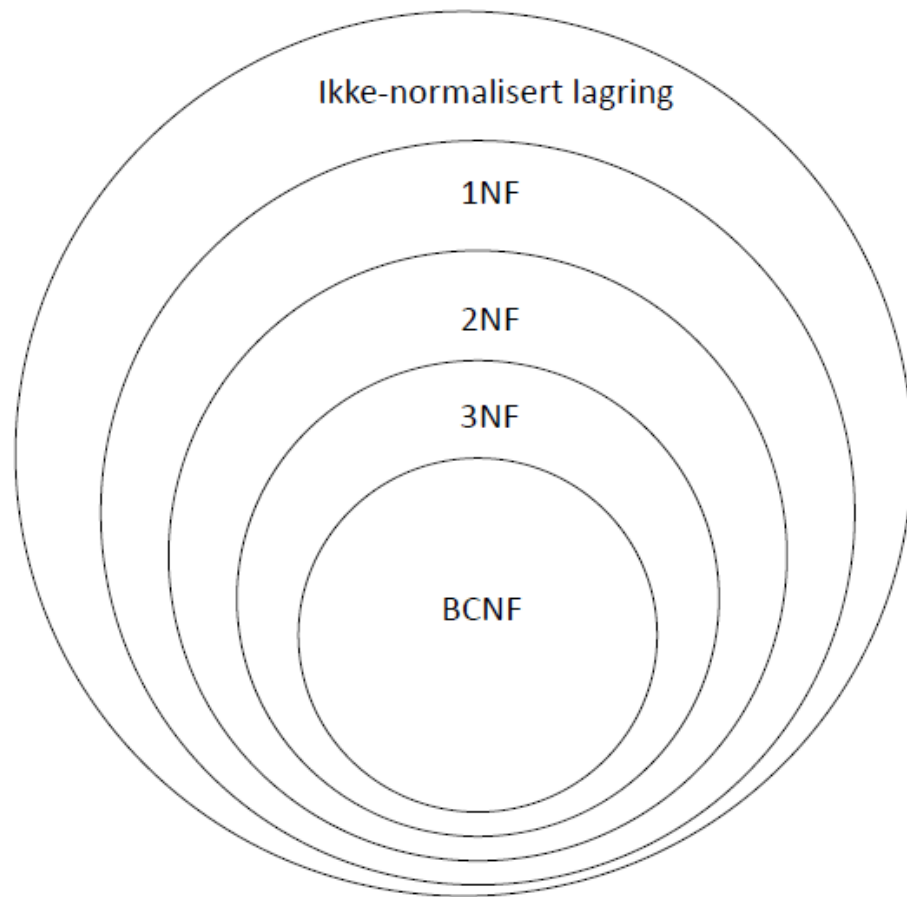
# Hva er normalisering?

- Normalisering er en formell metode for å designe relasjonsdatabaser via funksjonelle avhengigheter
- Mål:
  - Å unngå unødvendig dobbeltlagring av data (redundans) pga. at dobbeltlagring lett fører til inkonsistente data.
  - (Når er dobbeltlagring nødvendig: For å kople tabeller sammen via primærnøkkel og fremmednøkkel.)
- Middel:
  - Dele opp eksisterende tabeller
- Normaliseringen legger stor vekt på å finne kandidatnøkler.
- Det stilles formelle krav til sammenhengen mellom attributtene i en tabell. Jo strengere krav, desto høyere normalform. 1NF, 2NF, 3NF, BCNF, 4NF, 5NF,...?
- 4NF og 5NF kun av teoretisk betydning. Gjennomgås ikke.

## Normalisering 1NF, 2NF, 3NF, BCNF

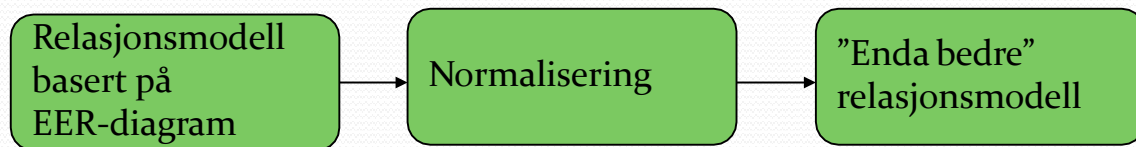
Normalformene bygger på hverandre. F.eks. så vil en relasjon på BCNF-form også være på 3NF, som igjen er 2NF etc.

Alle relasjoner (tabeller) vi snakker om har en primærnøkkel, og ingen deler av denne er NULL, og det er ingen repeterende grupper. Hvis kun dette er tilfredsstilt, sier vi at relasjonen er ikke-normalisert.

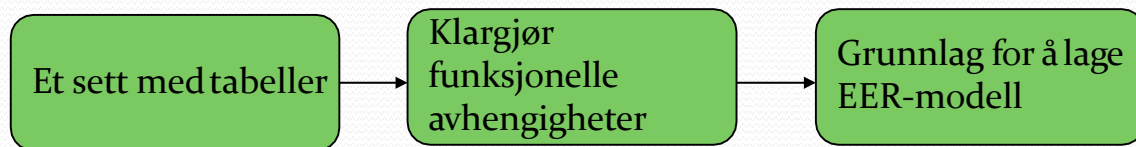


## Normaliseringens plass i utviklingsprosessen

- Som kvalitetssikring:



- For å forstå en eksisterende database:



# Databaseeksempel «Bestilling»

Relasjonsskjema:

Bestilling(Bestnr, Prodnr,  
Kundenr, Adresse, Antall)

Adresse er et eksempel på  
*sekundærinformasjon* om  
kunden.

Instansen:

Bestnr	Prodnr	Kundenr	Adresse	Antall
1	22	A	Gol	5
1	33	A	Gol	6
2	22	B	Hol	5
2	44	B	Hol	6
3	44	B	Hol	5

# Repetisjon: Nøkler

- **Supernøkkel**: En mengde attributter der verdiene i disse attributtene er unike (distinkte) i en relasjon
- **Kandidatnøkkel**: En *minimal* mengde attributter der verdiene er unike. Ved å fjerne et attributt, er det ikke lenger noen supernøkkel.
- **Primærnøkkel**: En spesielt utpekt kandidatnøkkel der ingen verdier kan være NULL
- **Nøkkelattributt**: Attributter som er med i en kandidatnøkkel

Bestnr	Prodnr	Kundenr	Adresse	Antall
1	22	1	Gol	5
2	33	2	Hol	6
2	44	2	Hol	5
3	22	1	Gol	6
3	44	1	Gol	5

## Eksempel:

**Supernøkler**: Alle nøkler som inneholder {Bestnr, Prodnr}

**Kandidatnøker**: {Bestnr, Prodnr}

**Primærnøkkel**: {Bestnr, Prodnr}

**Nøkkelattributter**: Bestnr, Prodnr

# Påkrevde integritetsregler

## Entitetsintegritet:

- Alle relasjonsskjemaer skal ha en og bare en primærnøkkel.
- Ingen av attributtene i primærnøkkelen får være **null**.

## Referanseintegritet

- Hvis fremmednøkkelene ikke er null, så skal de finnes et tuppel i den refererte relasjonen hvor primærnøkkelene har samme verdi som fremmednøkkelene.

Det kan også være andre integritetsregler.



# Hvorfor bry oss om normalformer?

## Anomalier (avvik, trøbbel!)

- INSERT anomali:
  - Kan ikke starte en tom bestilling uten antall produkter
  - Kan ikke legge inn adresse uten en bestilling
- UPDATE anomali:
  - Endring av kunde må gjøres flere steder
  - Samme for adresse
- DELETE anomali:
  - Sletting av post i bestilling kan sletting av kundeinformasjon
  - Sletting av bestilling kan slette kundeinformasjon (bestilling består av en post)

## Instans

Bestnr	Prodnr	Kunde nr	Adresse	Antall
1	22	1	Gol	5
2	33	2	Hol	6
2	44	2	Hol	5
3	22	1	Gol	6
3	44	1	Gol	5

**Definisjon:** Alle attributtene i relasjonen inneholder kun atomære verdier

- Atomære verdier er verdier «vi ikke kan dele opp». Dette må forstås i lys av datamodellen, hvilke bestandeler bryr vi oss om?
- Er «adresse» atomær, eller kan vi dele den opp? Skal vi bare skrive adresse-etiketter, eller ønsker vi å kunne gjøre spørringer hvor poststed inngår?

# Bestillingsbasen tilfredsstiller 1NF

Bestnr	Prodnr	Kundenr	Adresse	Antall
1	22	1	Gol	5
2	33	2	Hol	6
2	44	2	Hol	5
3	22	1	Gol	6
3	44	1	Gol	5

# Relasjoner som bryter med 1NF

filmid	tittel	sjangere
244	The Matrix	Action, Sci-Fi, Thriller
453	The Godfather	Krim, drama
1999	Star Wars: The rise of Skywalker	Action, Adventure, Fantasy

Bryter med 1NF fordi sjangere er en liste med verdier.

Vi fikser dette ved å splitte opp ikke-atomære verdier i separate liner/forekomster.

filmid	sjanger	tittel
244	Action	The Matrix
244	Sci-Fi	The Matrix
244	Thriller	The Matrix
453	Krim	The Godfather
453	Drama	The Godfather
1999	Action	Star Wars: The rise of Skywalker

Tilfredsstiller 1NF

(Men her gir det oss dupliserte data. Det må vi etter hvert fikse...)



# Funksjonelle avhengigheter

- Hvis verdiene i en attributt Y er bestemt av verdien i en annen attributt X, så er det en **funksjonell avhengighet**  $X \rightarrow A$  forkortet som f.d. («functional dependency»)
- X kan være sammensatt attributt  
«X bestemmer A» eller «A avhenger av X»

## Eksempler:

filmid  $\rightarrow$  Tittel i filmbasen

postnummer  $\rightarrow$  poststed (men ikke motsatt!)

Vi har **full funksjonell avhengighet (f.f.d.)** hvis A ikke avhenger av en delmengde av attributtene i X. En f.d. som ikke er full, kalles en partiell f.d.

# Normalformer og fd-er

Normalformene 2NF, 3NF og BCNF bestemmes av hvilke f.d.-er den har.

Gitt en relasjon R, så har den en mengde f.d-er på formen

$$X \rightarrow A.$$

Normalformene bestemmes av hvilke egenskaper disse har.

Vi utelukker trivielle f.d., der hvor A er delmengde av X.

# Andre normalform 2NF

En relasjon er på 2NF  
hvis og bare hvis  
den er på 1NF,  
og den ikke inneholder  
partielle avhengigheter  
(en ikke-nøkkel-attributt er  
f.d. av en del av  
primærnøkkel)

Dette betyr at for en f.a.  $X \rightarrow A$  så er

- $X$  en supernøkkel, eller
- $A$  er en nøkkelattributt, eller
- $X$  er ikke en delmengde av noen kandidatnøkler i  $R$

(Hvis  $X$  er supernøkkel, så er  $X \rightarrow A$  alltid en f.d.)

At vi IKKE har 2NF betyr at:

- $X$  inneholder ingen kandidatnøkkel OG
- $A$  er ikke nøkkelattributt OG
- $X$  er inneholdt i en kandidatnøkkel

# Eksempel på 1NF, men ikke 2NF

Eneste kandidatnøkkel her er  
{filmid, sjanger}

så det er også  
primærnøkkelen.

Tittel er en ikke-nøkkel-attributt  
som avhenger av filmid, ikke  
av hele nøkkelen, så  
relasjonen er ikke på 2NF.

filmid	sjanger	tittel
244	Action	The Matrix
244	Sci-Fi	The Matrix
244	Thriller	The Matrix
453	Krim	The Godfather
453	Drama	The Godfather
1999	Action	Star Wars: The rise of Skywalker



# Fra 1NF til 2NF

Vi splitter relasjonen flere nye relasjoner, også kalt dekomponering, slik at partielle f.d-er fjernes.

De nye tabellene får vi ved **projeksjon**

## Regel:

Hvis vi har relasjonen  $R(XYZ)$ , hvor attributtene til  $R$  er partisjonert i tre disjunkte ikke-tomme delmengder  $X, Y, Z$ , og vi har f.d.  $X \rightarrow Y$ , så kan vi dekomponere  $R$  i  $S(XY)$  og  $T(XZ)$ , slik at vi ikke får falske tupler ved naturlig forening (join) av  $S$  og  $T$ .

# Fra 1NF til 2NF: Film-eksempel

«Tittel» avhenger av bare filmid.

Bruker **regel** med

$X = \{\text{filmid}\}$ ,  $Y = \{\text{tittel}\}$ ,  $Z = \{\text{sjanger}\}$

Naturlig forening bringer relasjonene tilbake til utgangspunktet.

(Den siste tabellen har ikke flere attributter enn primærnøkkelen. men de kunne f.eks. prosent av personer som mente filmen primært var av den gitte sjangeren...)

filmid	sjanger	tittel
244	Action	The Matrix
244	Sci-Fi	The Matrix
244	Thriller	The Matrix
453	Krim	The Godfather
453	Drama	The Godfather
1999	Action	Star Wars: The rise of Skywalker

filmid	tittel
244	The Matrix
453	The Godfather
1999	Star Wars: The rise of Skywalker

filmid	sjanger
244	Action
244	Sci-Fi
244	Thriller
453	Krim
453	Drama
1999	Action

# Brudd på 2NF i Bestillings-eksempel

Bestilling(Bestnr, Prodnr, Kundenr, Adresse, Antall)

Her har vi følgende f.d.:

1. Bestnr, Prodnr  $\rightarrow$  Kundenr, Adresse, Antall
2. Bestnr  $\rightarrow$  Kundenr. Hele bestillingen er til samme kunde, **dette bryter med 2NF!**
3. Kundenr  $\rightarrow$  Adresse: Kunde har kun en adresse
4. Bestnr  $\rightarrow$  Adresse: konsekvens av 2 og 3, også **brudd på 2NF.**

# Bruk av regel på Bestilling-eksempel

Vi ser på f.d. som ikke tilfredsstiller 2NF, og hvilke dekomponeringer det gir:

- $\text{Bestnr} \rightarrow \text{Kundenr}$ 
  - $S(\underline{\text{Bestnr}}, \text{Kundenr}), T(\underline{\text{Bestnr}}, \underline{\text{Prodnr}}, \text{Adresse}, \text{Antall})$
  - (gi hensiktsmessige navn til S og T)
- $\text{Bestnr} \rightarrow \text{Adresse}$ 
  - $S(\underline{\text{Bestnr}}, \text{Adresse}), T(\underline{\text{BestNr}}, \underline{\text{ProdNr}}, \text{Kundenr}, \text{Antall})$
- Også denne:  $\text{Bestnr} \rightarrow \text{Kundenr}, \text{Adresse}$ :
  - $S(\underline{\text{Bestnr}}, \text{Kundenr}, \text{Adresse}), T(\underline{\text{Bestnr}}, \underline{\text{Prodnr}}, \text{Antall})$

Det er kun den siste hvor begge S og T tilfredsstiller 2NF.

# Fikser 2NF anomaliene?

La oss se på den siste, som gir oss 2NF:

S(Bestnr, Kundenr, Adresse), T(Bestnr, Prodnr, Antall)

- INSERT : Tom bestilling kan gjøres, men kundedresse uten bestilling IKKE OK
- UPDATE : Endring kunde bare ett sted ok, men endring av adresse krever flere oppdateringer
- DELETE: Sletting av post i bestilling sletter ikke kundeinformasjon OK, sletting av bestilling kan slette kundeinformasjon

Så vil må gjøre mer for å fikse anomaliene. Vi fortsetter på normaliseringsstigen!

# Normalisering til 2NF når en runde med **regelen** ikke er nok

Når vi har et brudd  $X \rightarrow A$  på 2NF  
( $X$  utgjør en del av en kandidatnøkkel,  $A$  er ikke nøkkelattributt), velg et «minimalt» brudd, dvs. med  $X$  slik at det ikke finnes en delmengde av  $X$  som også gir brudd. Vi partisjonerer attributtene i  $W, X, A, Y$ , med kandidatnøkkel  $WX$ , og minimal f.d.  $X \rightarrow A$

- Dekomponer til  $S(\underline{X}, A)$  og  $T(\underline{W}, \underline{X}, Y)$  etter **regelen**.  $S$  er nå på 2NF
- Fortsett med å dekomponere  $T$  hvis den har brudd på 2NF.
- Dette må til slutt endre, siden vi får færre og færre f.d., og vi har bare et endelig antall fra starten.

# Fra 1NF til 2NF, bestillings-eksempel

## Minimal f.d

S(Bestnr, Kundenr, Adresse),  
T(Bestnr, Prodnr, Antall) er

## Instans

Bestnr	Prodnr	Kundenr	Adresse	Antall
1	22	1	Gol	5
2	33	2	Hol	6
2	44	2	Hol	5
3	22	1	Gol	6
3	44	1	Gol	5

# 3NF

En relasjon er på tredje normalform (3NF), hvis og bare hvis den er på 2NF, og den ikke inneholder *transitive* avhengigheter. «Alle f.d. utgår fra primærnøkkelen.»

Dette kan karakteriseres slik:

For en f.d.  $X \rightarrow A$  i  $R$ , så er

- $X$  supernøkkel i  $R$ , ELLER
- $A$  er en nøkkelattributt

Relasjonen **bryter** 3NF dersom det finnes ikke-triviell f.d.  $X \rightarrow A$  hvor  $X$  ikke inneholder noen kandidatnøkkel **og**  $A$  er et ikke-nøkkelattributt.



# Eksempel på 2NF, men ikke 3NF

De to relasjonene i  
filmbasen er allerede  
på 3NF.

# Eksempel på brudd mot 3NF

Bestilling var normalisert til 2NF:

S(Bestnr, Kundenr, Adresse), T(Bestnr, Prodnr, Antall)

Vi har f.d Kundenr  $\rightarrow$  Adresse i S. Den er ikke-triviell, og Adresse er ikke nøkkelattributt, siden den er ikke inneholdt i noen kandidatnøkler. **Brudd på 3NF.**

# Fra 2NF til 3NF

S(Bestnr, Kundenr, Adresse), T(Bestnr, Prodnr, Antall)

Vi bruker **regelen**, med f.d. Kundenr  $\rightarrow$  Adresse, på relasjonen S:

K(Kundenr, Adresse)

B(Bestnr, Kundenr)

T(Bestnr, Prodnr, Antall)

Denne kan vi sjekke at er på 3NF!

Her kan vi også gi relasjonene mer beskrivende navn

# Fikser 3NF anomaliene?

K(Kundenr, Adresse)

B(Bestnr, Kundenr)

T(Bestnr, Prodnr, Antall)

- INSERT : Tom bestilling kan gjøres, og kundeadresse uten bestilling er OK
- UPDATE : Endring kunde bare ett sted ok, og endring av adresse kun ett sted
- DELETE: Sletting av post i bestilling sletter ikke kundeinformasjon OK, og sletting av bestilling sletter ikke kundeinformasjon

Da er vi kanskje i mål ?

# BCN (Boyce-Codd) normalform

En relasjon  $R$  er BCNF hvis den er NF3, og for enhver ikke-triviell f.d.  $X \rightarrow A$ , så inneholder  $X$  en kandidatnøkkel.

Motsatt: Den bryter BCNF hvis det finnes ikke-triviell f.d.  $X \rightarrow A$  hvor  $X$  ikke inneholder noen kandidatnøkkel.

# Noen «facts»

- En kan alltid normalisere til 3NF
- Hvis en relasjon er på 3NF, men ikke BCNF, så kan det finnes f.d. som må sjekkes ved innsetninger og oppdateringer
- En kan alltid normalisere til BCNF, men det kan gi noen funksjonelle avhengigheter på tvers av relasjonene, og ved innsetninger og oppdateringer, må joine de involverte relasjonene for å sjekke at de f.d. fortsatt er oppfylt. Dette kan unngås med 3NF
- En vil vanligvis nøye seg med 3NF hvis BCNF innfører funksjonelle avhengigheter på tvers.

# Eksempel på eksamensoppgave

- En studentforening tjener noen kroner på å distribuere selvlagede kompendier tilpasset fag på universitetet de er tilknyttet. Oversikten over alle kompendier de tilbyr er en implementasjon av følgende relasjon:
    - (Fag, Årstall, Tittel, Format, Pris)
  - Det lages høyst et kompendium per fag per år, og hvert kompendium kommer kun i et format (bok, hefte eller elektronisk). Alle kompendier i samme format koster det samme, uansett fag.
- I. Beskriv alle (fulle) funksjonelle avhengigheter i relasjonen over.
  - II. Forklar hvorfor relasjonen er i 2NF.
  - III. Forklar hvorfor relasjonen ikke er i 3NF.