



UNIVERSITETET
I OSLO

Relasjonsdatabasedesign

Oppdateringsanomalier

Dekomponering

Normalformer

Hva kjennetegner god relasjonsdatabasedesign?

- Beslektet informasjon legges i samme skjema
 - Ubeslektet informasjon legges *ikke* i samme skjema
 - Brudd på dette vil ofte medføre dobbeltlagring av data og dermed forårsake **oppdateringsanomalier**
- Så lite dobbeltlagring som mulig
 - Plassbehovet minimaliseres
 - Oppdatering forenkles
- Så få «glisne» relasjoner som mulig
 - Plassbehovet minimaliseres
 - Unngår problemer med håndtering av nil-verdier
- Korrekt totalinformasjon kan gjenskapes nøyaktig
 - Ingen falske data genereres

Eksempel:

Grossistdatabase versjon 1

Produkt(Kode, Produktnavn, Produsent, #Enheter)

Bestilling(Kode, Kundenr, Navn, Adresse, #Bestilt)

Skranke:

- 1) Alle verdier i Kode i Produktskjemaet skal være unike
- 2) For hvert par av Kundenr, Kode i Bestillingskjemaet skal det være bare en mulig verdi av #Bestilt
- 3) Verdien i Kundenr bestemmer verdiene i Navn og Adresse
- 4) Kode i Bestillingskjemaet er fremmednøkkel til (refererer til) verdier i Kode i Produktskjemaet

Grossistdatabase versjon 1 med integritetsregler

Produkt(Kode, Produktnavn, Produsent, #Enheter)

Bestilling(Kode, Kundenr, Navn, Adresse, #Bestilt)

Integritetsregler:

- 1) Kode → Produktnavn Produsent #Enheter (i Produktskjemaet)
- 2) Kundenr Kode → #Bestilt (i Bestillingskjemaet)
- 3) Kundenr → Navn Adresse (i Bestillingskjemaet)
- 4) Kode i Bestilling-skjemaet er fremmednøkkel til Produkt

Merk at {Kode} er en kandidatnøkkel i Produkt
Den er den eneste slike i Produkt og blir derfor primærnøkkel

Merk at 2 og 3 gir at {Kode, Kundenr} er supernøkkel i Bestilling
Dette er dessuten en kandidatnøkkel, og den eneste slike i
Bestilling, og den blir derfor primærnøkkel

Eksempelekstensjon Bestilling

Bestilling

Kode	Kundenr	Navn	Adresse	#Bestilt
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 er unødvendig i en tabell over bestillingene

Oppdateringsanomalier

- **Innsettingsanomalier**
 - Opprettholde konsistente verdier
 - Håndtere sekundær informasjon
 - Håndtere nil i kandidat- og fremmednøkler
- **Slettingsanomalier**
 - Unngå tap av sekundær informasjon
- **Modifiseringsanomalier**
 - Opprettholde konsistente verdier
 - Oppdatere sekundær informasjon

Hvordan unngå oppdateringsanomalier

- Unngå/fjern oppdateringsanomalier og dobbeltlagring ved å splitte (dekomponere) relasjonene slik at dobbeltlagring blir borte!

Dekomposisjon av et relasjonsskjema

- Gitt et relasjonsskjema $R(A_1, A_2, \dots, A_n)$
En **dekomposisjon** $D = \{R_1, R_2, \dots, R_m\}$ av R er en samling med skjemaer R_1, R_2, \dots, R_m som er slik at følgende to krav holder:
 - alle attributtene i hver R_k er også attributt i R
 - samtlige attributter A_1, A_2, \dots, A_n kan gjenfinnes i minst ett av skjemaene R_1, R_2, \dots, R_m

Eksempel:

Grossistdatabase versjon 2

Produkt(Kode, Produktnavn, Produsent, #Enheter)

Kunde(Kundenr, Navn, Adresse)

Bestilling(Kode, Kundenr, #Bestilt)

Integritetsregler:

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

Eksempelekstensjoner

Kunde, Bestilling

Ny
modell

Kunde

Kundenr	Navn	Adresse
1	A	a
2	B	b

Bestilling

Kode	Kundenr	#Bestilt
1	1	3
2	1	8
1	2	2

Gammel
modell

Bestilling

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

Naturlig join

- Vi ønsker å kunne rekonstruere den opprinnelige ekstensjonen
- **Naturlig join** er en operasjon der to relasjoner slås sammen til en ved at to og to tupler pares hvis og bare hvis de har like verdier i attributter med likt navn

Kunde ⋈ Bestilling

Kunde

Kundenr	Navn	Adresse
1	A	a
2	B	b

Bestilling

Kode	Kundenr	#Bestilt
1	1	3
2	1	8
1	2	2

Kunde ⋈ Bestilling

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

Eksempel:

Grossistdatabase, versjon 3

Produkt(Kode, Produktnavn, Produsent, #Enheter)

Kunde(Kundenr, Navn, Adresse)

Koderegister(Kode, Kundenr)

Antall(Kundenr, #Bestilt)

Integritetsregler:

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

Eksempelekstensjoner

Kundereg, Bestilling

Koderegister

Kode	Kundenr
1	1
2	1
1	2

Antall

Kundenr	#Bestilt
1	3
1	8
2	2

Koderegister \bowtie Antall

Kode	Kundenr	#Bestilt
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**

Retningslinjer for dekomposisjon av relasjoner

- **Motivasjon:** Fjerne oppdateringsanomalier
- **Teknikk:** Dekomponere problemrelasjoner
- **Betingelse:** Opprinnelig ekstensjon skal alltid kunne rekonstrueres nøyaktig ved naturlig join
- **Problemer:**
 - Hvordan vurdere objektivt om en samling relasjoner er god/dårlig?
 - Hvordan sikre at en dekomposisjon aldri gir falske tupler?

Normalformer

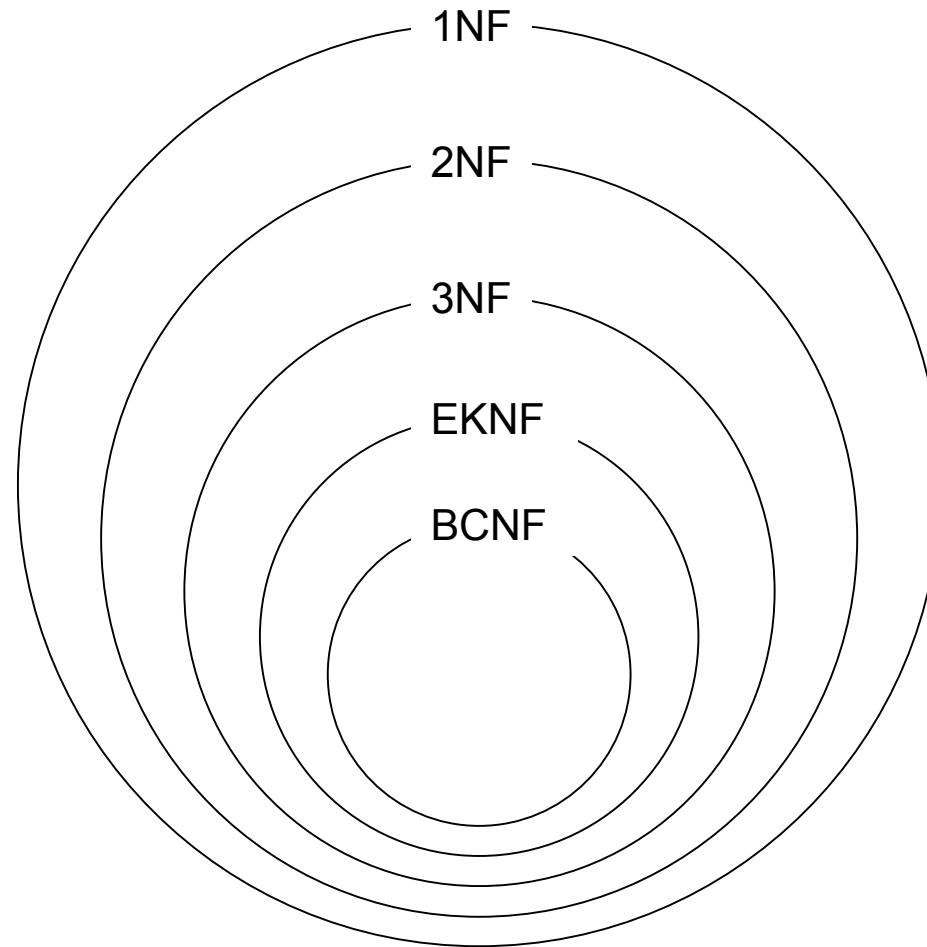
- Normalformer er et uttrykk for hvor godt vi har lykket i en dekomposisjon
- Jo høyere normalform, jo færre oppdateringsanomalier
- Det fins algoritmer for å omforme fra lavere til høyere normalformer
- Det fins algoritmer for å sjekke om en dekomposisjon er ***tapsfri***, dvs. at naturlig join aldri vil gi falske tupler

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
 - Funksjonsverdien til et tuppel for et gitt attributt skal være en slik atomær verdi (eller nil)
- Alle relasjoner er automatisk på 1NF

Andre normalform

- Repetisjon: En FD $X \rightarrow Y$ er ikke-triviell hvis Y ikke er inneholdt i X , dvs. hvis $Y - X \neq \emptyset$
- **Definisjon 2NF** (Codd 1972):
En relasjon R er på **andre normalform** hvis alle ikke-trivielle FDer i R på formen $X \rightarrow A$, der X er en mengde attributter og A et attributt i R , tilfredsstillers minst ett av følgende:
 - i. X er en supernøkkel i R
 - ii. A er et nøkkelattributt i R
 - iii. $X \not\subseteq K$ for noen kandidatnøkkel K i R

Egenskaper ved 2NF

- $2NF \subseteq 1NF$ siden alle relasjoner er 1NF
- Når er en relasjon 1NF, men ikke 2NF?

Svar: Når det fins en ikke-triviell FD $X \rightarrow A$ og en kandidatnøkkel K hvor $X \subset K$, $X \neq K$ og A ikke er et nøkkelattributt

Eksempel: $R(A, B, C, D)$, $F = \{BC \rightarrow D, C \rightarrow A\}$

Dekomposisjon: $R_1(A, \underline{C})$, $R_2(\underline{B}, \underline{C}, D)$

- 2NF er mest av historisk interesse;
vi gjør sjelden feil som bryter 2NF

Tredje normalform

- **Definisjon 3NF** (Codd 1972):
En relasjon R er på **tredje normalform** hvis alle ikke-trivielle FDer i R på formen $X \rightarrow A$ tilfredsstiller minst ett av følgende:
 - i. X er en supernøkkel i R
 - ii. A er et nøkkelattributt i R

Egenskaper ved 3NF

- $3NF \subseteq 2NF$ fordi kravene til 3NF er en skjerping av kravene til 2NF

- Når er en relasjon 2NF, men ikke 3NF?

Svar: Når det fins en ikke-triviell FD $X \rightarrow A$ hvor X ikke er en supernøkkel og A ikke er et nøkkelattributt og $X \not\subseteq K$ for noen kandidatnøkkel K

Eksempel: $R(A, B, C)$, $F = \{A \rightarrow BC, B \rightarrow C\}$

Dekomposisjon: $R_1(\underline{A}, B)$, $R_2(\underline{B}, C)$

- Også 3NF er lett å oppnå

Elementære FDer og nøkler

- En FD $X \rightarrow A$ kalles *elementær* dersom
 - A er ett attributt
 - $X \rightarrow A$ er ikke-triviell
 - X er minimal
(dvs. at hvis $Y \subseteq X$ og $Y \rightarrow A$, så er $Y = X$)
- En kandidatnøkkel K kalles *elementær* hvis det finnes en elementær FD $K \rightarrow B$ i R

Elementary Key Normal Form

- **EKNF** (Zaniolo 1982) er ikke pensum, men er tatt med for fullstendighets skyld:
- En relasjon R er på **EKNF** hvis alle ikke-trivielle FDer i R på formen $X \rightarrow A$ tilfredsstiller minst ett av følgende:
 - i. X er en supernøkkel i R
 - ii. A er et attributt i en elementær kandidatnøkkel i R

Egenskaper ved EKNF

- $EKNF \subseteq 3NF$ fordi kravene til EKNF er en skjerping av kravene til 3NF
- Når er en relasjon 3NF, men ikke EKNF?

Svar: Når det fins en ikke-triviell FD $X \rightarrow A$ hvor X ikke er en supernøkkel og A er et nøkkelattributt i en ikke-elementær kandidatnøkkel

Eksempel: $R(A,B,C)$, $F = \{C \rightarrow A, A \rightarrow C\}$

(Her kan A =epostadresse, B =emnekode og C =fødselsnummer på studenter)

Mulig dekomposisjon: $R_1(\underline{A}, \underline{B})$, $R_2(\underline{A}, \underline{C})$

Boyce-Codd Normalform

- **Definisjon BCNF** (Boyce & Codd 1974):
En relasjon R er på **Boyce-Codd normalform** hvis alle ikke-trivielle FDer i R på formen $X \rightarrow A$ tilfredsstiller følgende:
 - i. X er en supernøkkel i R

Egenskaper ved BCNF

- $BCNF \subseteq EKNF$ fordi kravene til BCNF er en skjerping av kravene til EKNF
- Når er en relasjon EKNF, men ikke BCNF?
- Svar: Når det fins en ikke-triviell FD $X \rightarrow A$ hvor X ikke er en supernøkkel og A er et attributt i en elementær kandidatnøkkel
- Eksempel: $R(A,B,C)$, $F = \{AB \rightarrow C, C \rightarrow A\}$
Dekomposisjon: $R_1(\underline{B}, \underline{C})$, $R_2(\underline{A}, \underline{C})$
- Merk at FDen $AB \rightarrow C$ går på tvers av R_1 og R_2 (Dvs. at vi må ta en join av R_1 og R_2 før vi kan teste om $AB \rightarrow C$)

Oppsummering normalisering

- Brudd på normalformer gir opphav til dobbeltlagring
- Jo høyere normalform, desto mindre dobbeltlagring
- BCNF: Alle FDer er i form av kandidatnøkler.
Dvs. ingen dobbeltlagring *innen skjemaet*.
Men: Noen FDer kan gå på tvers av skjemaer
- EKNF (3NF): Ingen FDer går på tvers av skjemaer, men det kan være noen FDer innen skjemaer som gir opphav til dobbeltlagring
- Krav til dekomposisjon: Den må være **tapsfri**, dvs. aldri kunne gi opphav til falske tupler
- Det er en fordel om dekomposisjonen også er **FD-bevarende**
(dvs. at ingen FDer går på tvers av skjemaer)

Hvorfor BCNF ikke alltid er gunstig

- Dersom vi har en dekomposisjon som ikke er FD-bevarende, må vi foreta en join mellom skjemaene i forbindelse med oppdateringer for å sjekke at integritetsreglene overholdes
- Vi har valget mellom
 - Enten:* Gå tilbake til 3NF (EKNF) og bryte BCNF og godta dobbeltlagring som vi må ta hensyn til ved oppdateringer slik at FD-ene overholdes
 - Eller:* Beholde BCNF og foreta join mellom relasjoner for å sjekke at FDer overholdes ved oppdateringer

Hvordan dekomponere tapsfritt

- **Fagins teorem**

Gitt et skjema $R(XYZ)$ med FDer F .

En dekomposisjon $D=\{XY, YZ\}$ er tapsfri mhp.

F hvis og bare hvis minst en av følgende holder:

1) $Y \rightarrow X \in F^+$

2) $Y \rightarrow Z \in F^+$

Her er F^+ mengden av alle FDer som kan avledes av F

Det fins algoritmer som...

- Tester om en dekomposisjon er tapsfri
- Tester om en dekomposisjon er FD-bevarende
- Dekomponerer skjemaer tapsfritt og FD-bevarende
 - Garanterer EKNF, men gir ikke alltid BCNF
- Dekomponerer skjemaer tapsfritt til BCNF
 - Er ikke alltid FD-bevarende