UNIVERSITETET
I OSLO

Relasjonsdatabasedesign

Normalformer

Hvordan dekomponere tapsfritt

Fagins teorem

Gitt en relasjon R(XYZ) med FDer F.
En dekomposisjon D={XY, XZ} er tapsfri mhp. F hvis og bare hvis minst en av følgende holder:

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

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

Eksempel: R(A,B,C,D), F={C→AD}
Dekomposisjon: R_1(A,C,D), R_2(B,C)

Normalformer

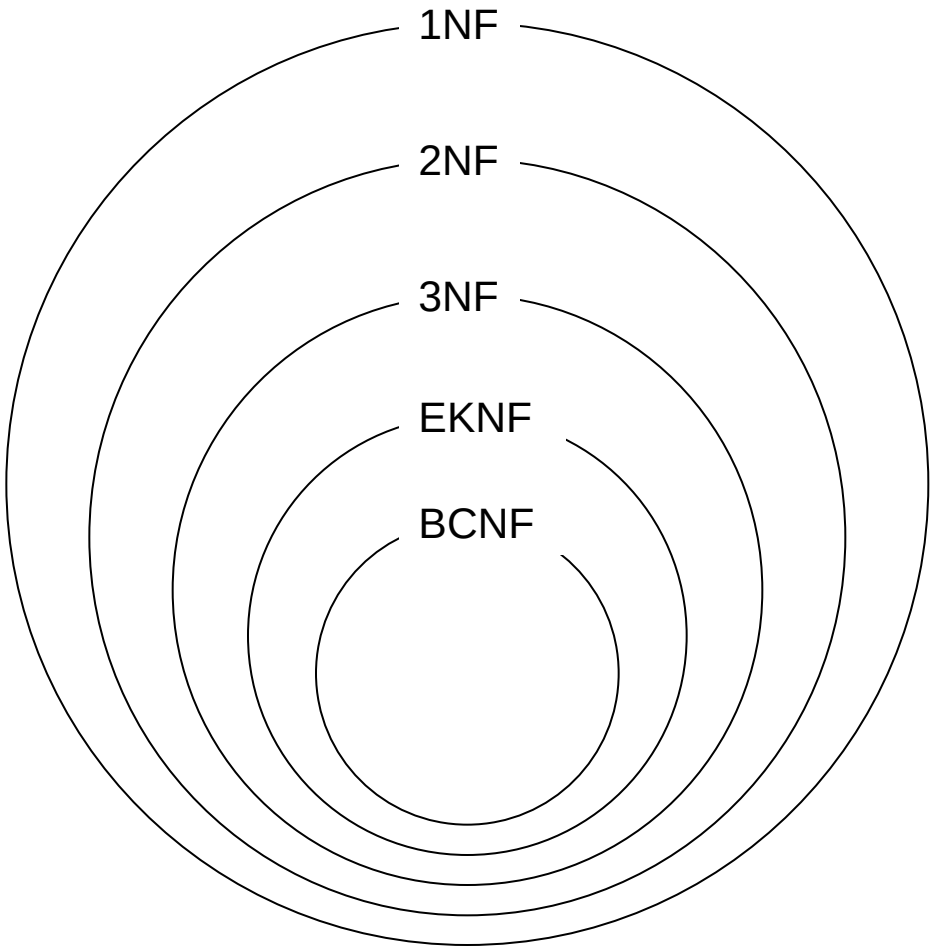
- Normalformer er et uttrykk for hvor godt vi har lykkes i en dekomposisjon
- Jo høyere normalform, jo færre oppdateringsanomalier
- Det fins algoritmer for å omforme fra lavere til høyere normalformer

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

Andre normalform

- En FD $X \rightarrow Y$ er **ikketriviell** 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 ikketrivielle FDer i R på formen $X \rightarrow A$, der X er en mengde attributter og A et attributt i R, tilfredsstiller minst ett av følgende:
  i. X er en supernøkkel i R
  ii. A er et nøkkelattributt i R
  iii. $X \not\subset K$ 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 ikketriviell 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,C), R_2(B,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 ikketrivielle 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 ikketriviell FD $X \rightarrow A$ hvor X ikke er en supernøkkel og A ikke er et nøkkel-attributt og $X \not\subset K$ for noen kandidatnøkkel K
- Eksempel: R(A,B,C), F={$A \rightarrow BC$, $B \rightarrow C$}
  Dekomposisjon: R_1(A,B), R_2(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 ikketriviell
  - 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) (kursorisk pensum)
- **Definisjon EKNF**: En relasjon R er på **elementary key normal form** hvis alle ikketrivielle 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 ikketriviell FD $X \rightarrow A$ hvor X ikke er en supernøkkel og A er et attributt i en ikke-elementær kandidatnøkkel
- Eksempel: R(A,B,C), F = {$C \rightarrow A$, $A \rightarrow C$}

  Dekomposisjoner: R_1(A,B), R_2(A,C) eller

  S_1(B,C), S_2(A,C).

Boyce-Codd Normalform

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

Egenskaper ved BCNF I

- 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 ikketriviell FD X$\rightarrow$A hvor X ikke er en supernøkkel og A er et attributt i en elementær kandidatnøkkel

- Når er en relasjon 3NF, men ikke BCNF?
  Svar: Når det fins en ikketriviell FD X$\rightarrow$A hvor X ikke er en supernøkkel og A er et attributt i en kandidatnøkkel

Eksempel

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

Dekomposisjon til BCNF: R_1(<u>B,C</u>), R_2(A,<u>C</u>)

Merk: FDen AB$\rightarrow$C kan ikke sjekkes i R_1 eller R_2 alene: Vi må ta en join av R_1 og R_2 før vi kan teste om AB$\rightarrow$C.

Egenskaper ved BCNF II

- Hvis alle FDer etter dekomposisjonen kan sjekkes lokalt i en av de nye relasjonene, sier vi at dekomposisjonen er **FD-bevarende**.
  - I eksempelet forrige side er dekomposisjonen **ikke FD-bevarende**.
- Man kan alltid dekomponere til BCNF, men ikke alltid FD-bevarende.
- Man kan alltid dekomponere FD-bevarende til EKNF.

Eksempel: Studenter

Student(stud#, brukernavn, emnekode, karakter)
FDer: stud# $\rightarrow$ brukernavn
brukernavn $\rightarrow$ stud#
stud#, emnekode $\rightarrow$ karakter

Hvilke av normalformene 1NF-BCNF tilfredsstiller denne?

Det fins algoritmer som…

- Tester om en dekomposisjon er tapsfri (chasealgoritmen)
- Tester om en dekomposisjon er FD-bevarende (ikke tatt med i dette foilsettet)
- Dekomponerer relasjoner tapsfritt og FD-bevarende
  - Garanterer EKNF, men gir ikke alltid BCNF
- Dekomponerer relasjoner tapsfritt til BCNF
  - Er ikke alltid FD-bevarende

Tapsfri dekomposisjon til BCNF

Gitt en relasjon R med et sett FDer F

1. Hvis $X \rightarrow A$ er et brudd på BCNF:
   A. Beregn X^+
   B. Dekomponer R i S og T der S er lik X^+ og T er lik X samt alle attributtene i R som ikke er i X^+.
2. Fortsett på samme måte med S og T inntil alle relasjonene i dekomposisjonen tilfredsstiller BCNF.

Eksempel: Studenter

Student(stud#, brukernavn, emnekode, karakter)

FDer: stud# $\rightarrow$ brukernavn

brukernavn $\rightarrow$ stud#

stud#, emnekode $\rightarrow$ karakter

Hvordan dekomponere denne til BCNF?

Minimale overdekninger

La F være en mengde FDer over relasjon R.

En **minimal overdekning** for F er en mengde FDer G som er ekvivalent med F og som tilfredsstiller følgende krav:

- Alle høyresidene i G er atomære
- Venstresidene er minst mulige
- Ingen av FDene i G er overflødige

Algoritme for å finne minimale overdekninger

1. Initialiser G til F.
2. Lag atomære høyresider ved hjelp av splitting.
3. Gjør venstresidene minimale:
   For hver $X \rightarrow A$ i G og hver $B \in X$:
   1. Beregn $(X-B)^+$ med hensyn på G.
   2. Hvis $A \in (X-B)^+$, erstatt $X \rightarrow A$ med $(X-B) \rightarrow A$ i G.
4. Fjern overflødige FDer:
   For hver $X \rightarrow A$ i G:
   1. Beregn X^+ med hensyn på G uten å bruke $X \rightarrow A$.
   2. Hvis $A \in X^+$, fjern $X \rightarrow A$ fra G.

Tapsfri FD-bevarende dekomposisjon til EKNF

Gitt en relasjon R med et sett FDer F.

1. Finn en minimal overdekning G for F.
2. For hver X som opptrer som venstreside i G, finn alle $X \rightarrow A_i$ i G og lag en relasjon R_X som inneholder X samt alle høyresidene A_i.
3. Hvis ingen av R_X-ene inneholder en kandidatnøkkel for R, utvid med en relasjon R_0 der attributtene er en kandidatnøkkel.

Relasjonene i dekomposisjonen er minst EKNF.
(I beste fall oppnås BCNF i alle relasjonene.)

Eksempel

- R(A,B,C,D,E,F,G,H,I)
- FDer: AB$\rightarrow$CH, DE$\rightarrow$FG, D$\rightarrow$H, E$\rightarrow$B, H$\rightarrow$AG
- Hva er høyeste normalform som R tilfredsstiller?
- Dekomponer R tapsfritt og FD-bevarende til EKNF.

Dekomposisjoner og sjekk av FDer

- Hvis dekomposisjonen ikke er FD-bevarende, må de FDene som går «på tvers», sjekkes ved å joine relasjoner
- I tillegg skal fremmednøkler overholdes
- FDer som gjelder internt i en relasjon, sjekkes lokalt i relasjonen
  I noen ganske få tilfeller er ikke dette tilstrekkelig selv når dekomposisjonen er FD-bevarende!

Eksempel: Varemagasindatabasen

Vareplassering(reolnr, hyllenr, varenr, #varer, avdeling)

Integritetsregler:

1) En reol består av et visst antall hyller. Hver hylle inneholder bare én type varer, og #varer er hvor mange enheter av denne varen som hyllen inneholder:
   reolnr, hyllenr $\rightarrow$ varenr, #varer

2) Hver reol inneholder varer fra bare én avdeling:
   reolnr $\rightarrow$ avdeling

3) En vare selges av bare én avdeling:
   varenr $\rightarrow$ avdeling

FD-bevarende tapsfri dekomposisjon til EKNF og eksempelinstanser

EKNF-algoritmen (s. 24) gir følgende dekomposisjon:

Hylle(reolnr, hyllenr, varenr, #varer)
Reol(reolnr, avdeling)
Vare(varenr, avdeling)

(Dekomposisjonen er faktisk ikke bare EKNF, men også BCNF.)

Eksempelinstanser:

Hylle			
reolnr	**hyllenr**	**varenr**	**#varer**
1	1	25	45
2	6	70	20

Reol	
reolnr	**avdeling**
1	Sko
2	Frukt

Vare	
varenr	**avdeling**
70	Sko
25	Frukt

Join av eksempelinstansene

Hylle ⋈ Reol				
reolnr	**hyllenr**	**varenr**	**#varer**	**avdeling**
1	1	25	45	Sko
2	6	70	20	Frukt

Hylle ⋈ Vare				
reolnr	**hyllenr**	**varenr**	**#varer**	**avdeling**
1	1	25	45	Frukt
2	6	70	20	Sko

Hylle ⋈ Reol ⋈ Vare				
reolnr	**hyllenr**	**varenr**	**#varer**	**avdeling**

Inneholder ingen tupler!

Chase av eksempelinstansene

Finnes det noen global instans som kan forklare innholdet i Hylle, Reol, Vare?

reolnr	hyllenr	varenr	#varer	avdeling
1	1	25	45	~~a_1~~ Sko
2	6	70	20	~~a_2~~ Frukt
1	h_3	v_3	n_3	Sko
2	h_4	v_4	n_4	Frukt
r_5	h_5	70	n_5	Sko
r_6	h_6	25	n_6	Frukt

Chaser med reolnr $\rightarrow$ avdeling:

Da holder ikke varenr $\rightarrow$ avdeling, så svaret er **nei**.

Hva er problemet?

- Alle integritetsregler er oppfylt:
  - Dekomposisjonen er tapsfri og FD-bevarende
  - Hver av instansene oppfyller alle lokale FDer
  - Alle fremmednøkler er oppfylt
- Men: Ingen instans av den opprinnelige relasjonen Vareplassering passer overens med disse instansene!

Støyinstanser

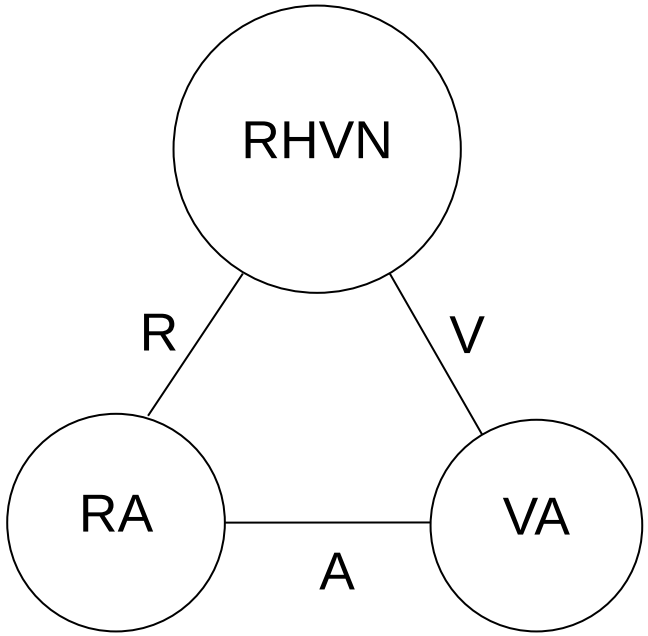
- Hvis det ikke finnes noen global instans som kan forklare de lokale instansene, kaller vi de lokale instansene for **støyinstanser**

Snittgrafen til en dekomposisjon

Hylle(R,H,V,N)

Reol(R,A)

Vare(V,A)



Snittgrafen har én node for hver relasjon og kanter mellom noder med felles attributter.

Snittgrafer og støyinstanser

- Støyinstanser kan bare opptre hvis snittgrafen er syklisk[1]. Da er det ikke alltid tilstrekkelig å teste at lokale FDer holder; man må i tillegg sjekke med join eller chase-algoritmen at det finnes en global instans der de lokale instansene er projeksjonen av den globale instansen.

[1]Det finnes tilfeller hvor snittgrafen er syklisk, men det likevel ikke kan bli støyinstanser. Så vi har støyinstans $\Rightarrow$ syklisk snittgraf, men implikasjonen gjelder ikke i den andre retningen.

Snittgrafer og støyinstanser (forts.)

- I slike tilfeller trenger vi integritetsregler som beskriver **aksessveier** – dvs. hvordan relasjonene henger sammen utover det som fremgår av fremmednøklene.

- I varemagasindatabasen kan regelen formuleres slik: Hvis det finnes et tuppel (r,h,v,n) i Hylle og et tuppel (r,a) i Reol, så finnes det også et tuppel (v,a) i Vare.

Hvilken normalform skal vi velge?

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

Hvilken normalform skal vi velge? (Forts.)

- I de tilfellene der vi har en dekomposisjon med en syklisk snittgraf, må vi vurdere om vi skal la være å dekomponere selv om den opprinnelige relasjonen bryter 2NF eller 3NF. Vi har valget mellom
  - *Enten*: Gå tilbake til den opprinnelige relasjonen og godta dobbeltlagring som vi må ta hensyn til ved oppdateringer slik at FDene overholdes
  - Eller: Beholde BCNF eller EKNF (3NF) og foreta join mellom relasjoner for å sjekke at de lokale instansene ikke inneholder støy etter oppdateringer. Dette gjelder også for FD-bevarende dekomposisjoner.

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 hver enkelt relasjon*.
  Men: Noen FDer kan gå på tvers av relasjoner
- EKNF (3NF): Ingen FDer går på tvers av relasjonene, men det kan være noen relasjoner der interne FDer gir opphav til dobbeltlagring
- Dekomposisjon med syklisk snittgraf: Da kan det være at dekomposisjonen tillater støyinstanser, dvs. instanser som ikke kan forklares ved en overordnet, «global» instans
- 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 relasjoner)

Høyere normalformer

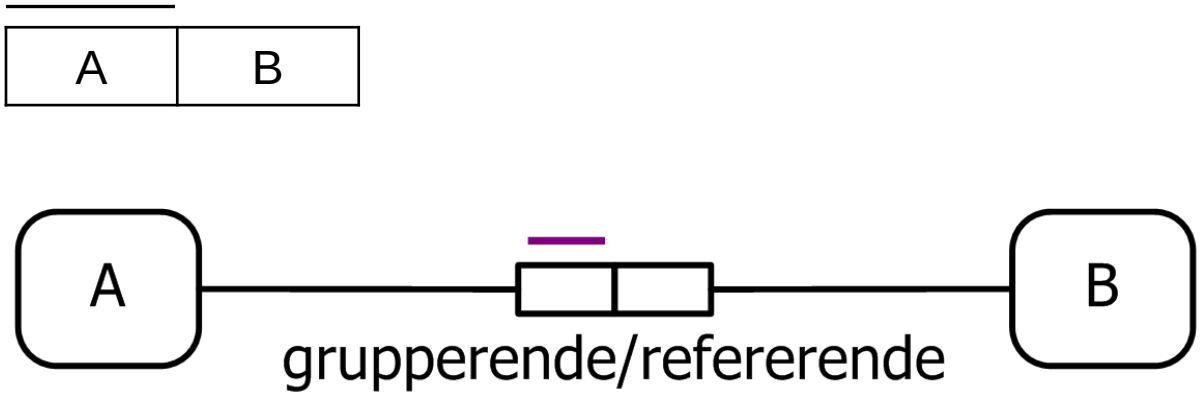
- Det finnes relasjoner som tilfredsstiller BCNF, men likevel kan gi oppdateringsanomalier. Dette skyldes andre integritetsregler enn det som kan uttrykkes ved FDer.
- Et eksempel er flerverdiavhengigheter (MVDer).
- Høyere normalformer kan eliminere også noen av disse oppdateringsanomaliene.

Ekstramateriale (ikke pensum)

ORM og normalisering

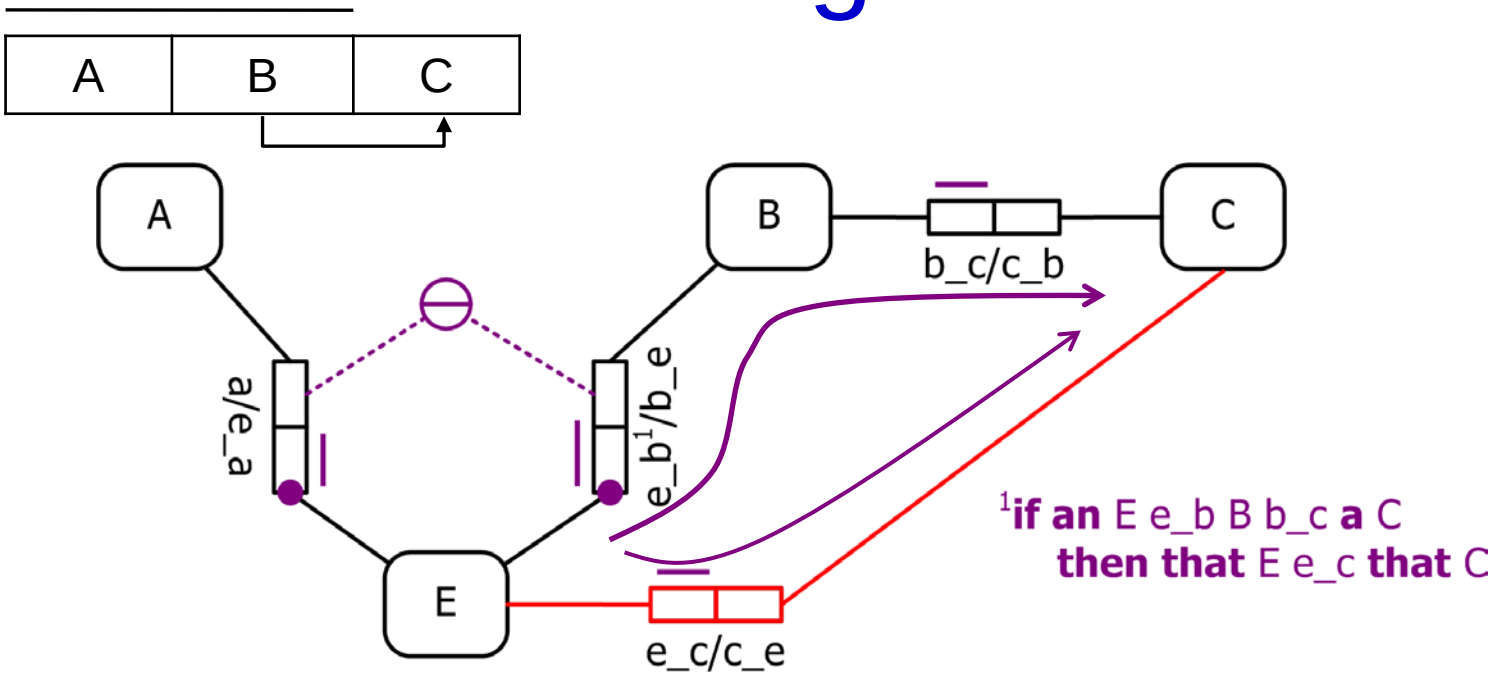
- Gruppering av ”korrekte” ORM-diagrammer gir **alltid** 3NF og **som regel** BCNF. Unntak skyldes alltid problemer knyttet til ekvivalente stier (EOPer).
- Hvis databasesystemet håndhever alle joinskranker (herunder de ekvivalente stiene), vil det både håndtere dobbeltlagring korrekt og eliminere støyinstanser

ORM og 1NF



ORM-diagrammet svarer til FDen A→B. ORM-diagrammer bygges «nedenfra og opp», dvs. man starter med enkle begreper (grunnbegreper) og bygger mer kompliserte begreper (begrepsdannelse) ved hjelp av disse. Siden det er representasjonene av grunnbegrepene som blir til attributter under grupperingen, blir den grupperte ORM-modellen alltid 1NF.

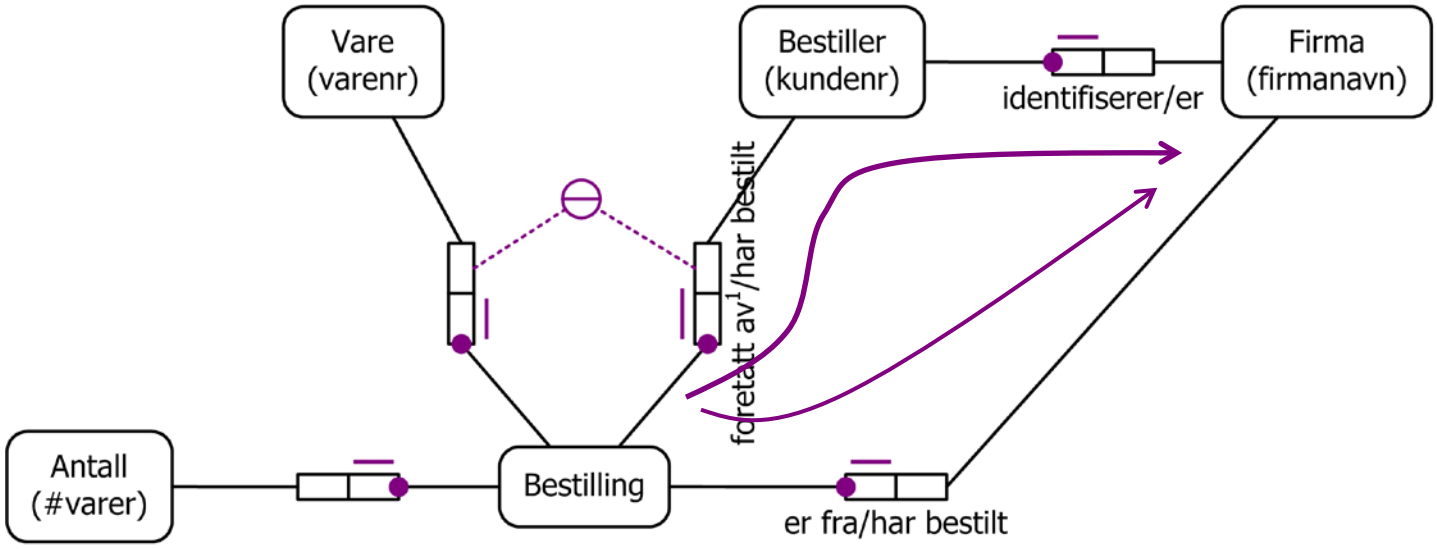
ORM og 2NF



Gruppert ORM-modell: R1(A,B,C), R2(B,C)

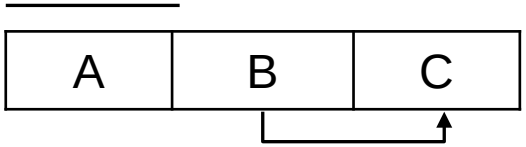
Hvis E→C (AB→C) representeres i tillegg til E→B, B→C i ORM-diagrammet, vil 2NF bli brutt i den grupperte modellen. Vi kan enkelt unngå dette ved å utelate den faktatypen som svarer til E→C (i rødt) under grupperingen. Vanligvis vil denne faktatypen være en feilmodellering i den forstand at den uttrykker en sammenheng som er mer presist uttrykt ved faktatypen mellom B og C. Gruppering uten faktatypen mellom E og C gir R1(A,B), R2(B,C).

ORM og 2NF – eksempel



[1]**Hvis en** Bestilling **er** *foretatt av* **en** Bestiller **som** *identifiserer* **et** Firma**,**
så *er* **samme** Bestilling *fra* **samme** Firma

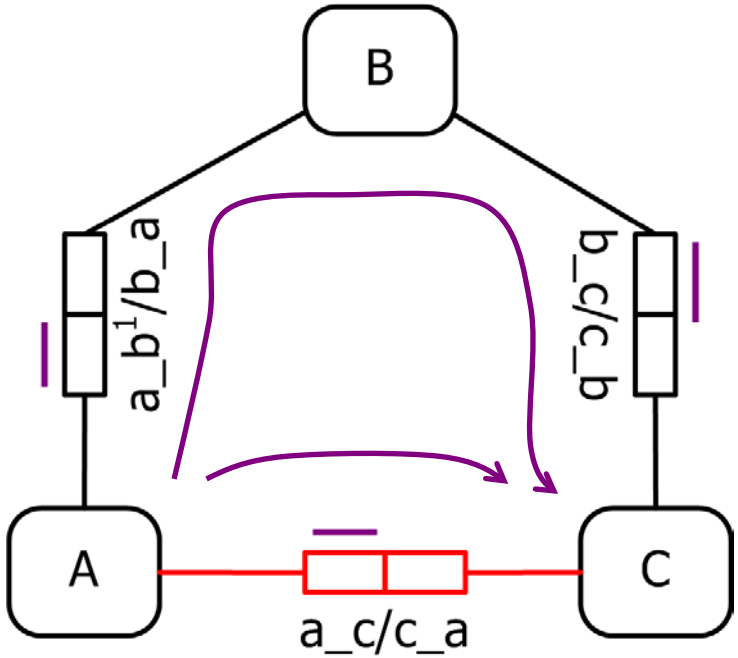
ORM og 3NF



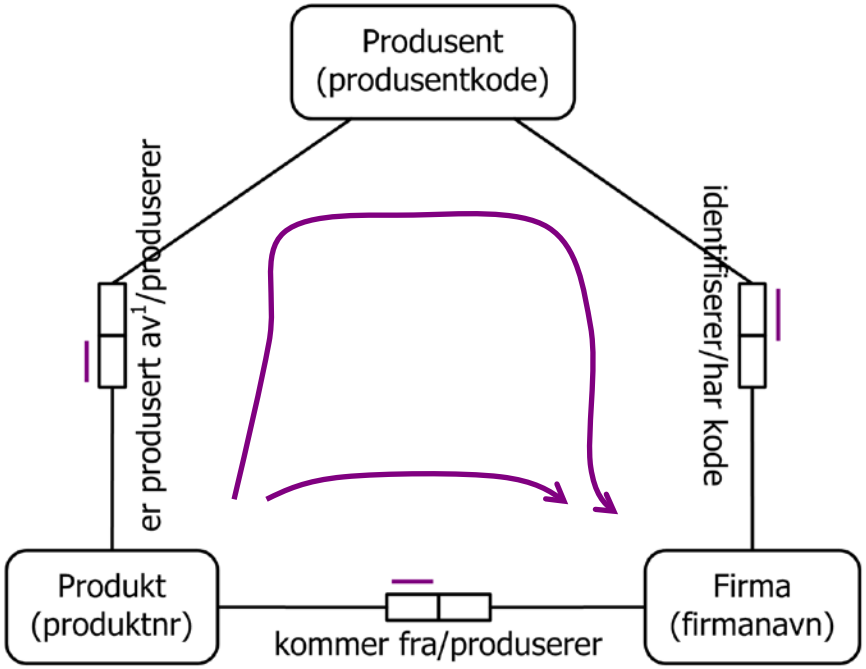
Gruppert ORM-modell:

R1(A,B,C), R2(B,C)

Hvis A→C representeres i tillegg til A→B, B→C, vil 3NF bli brutt i den grupperte modellen. Den røde faktatypen er vanligvis en feilmodellering, den uttrykker en sammenheng som er mer presist uttrykt ved de to andre faktatypene. Gruppering uten faktatypen mellom A og C gir R1(A,B), R2(B,C).

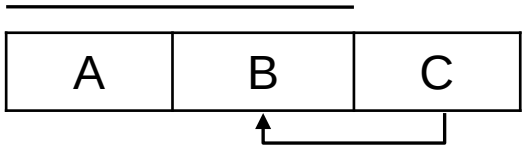


ORM og 3NF – eksempel



[1]**Hvis et** Produkt *er produsert av* **en** Produsent **som** *identifiserer* **et** Firma,
så *kommer* **samme** produkt *fra* **samme** Firma

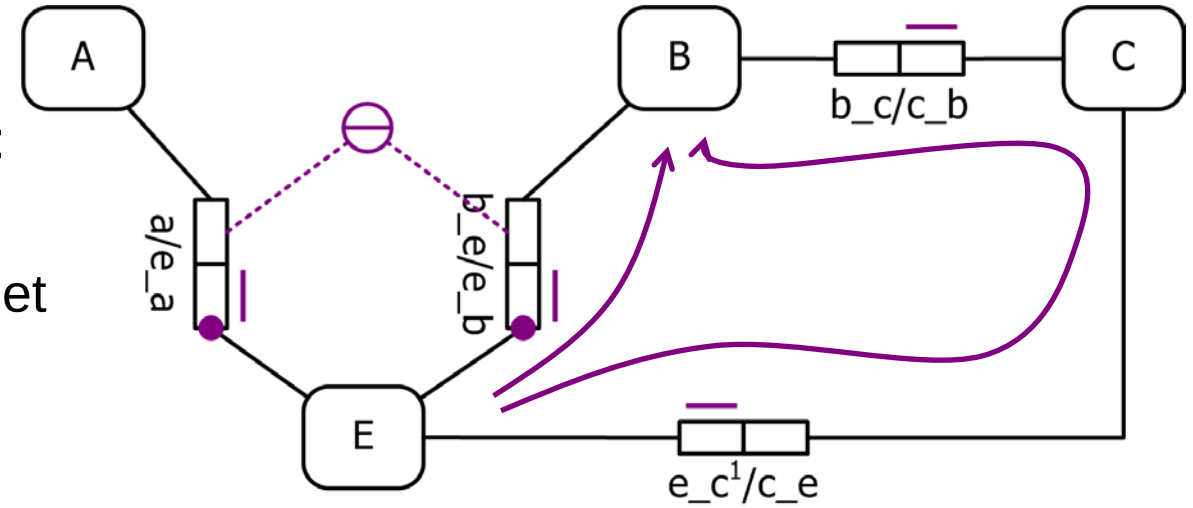
ORM og BCNF



Gruppert ORM-modell:

R1(A,B,C), R2(C,B)

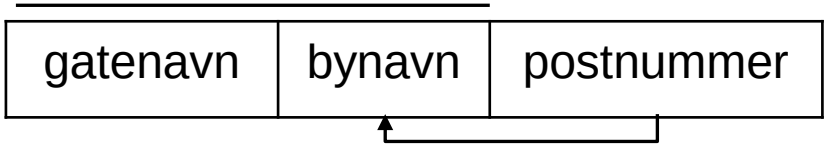
BCNF er brutt. Systemet må sjekke EOPen (E-C-B, E-B) under oppdateringer.

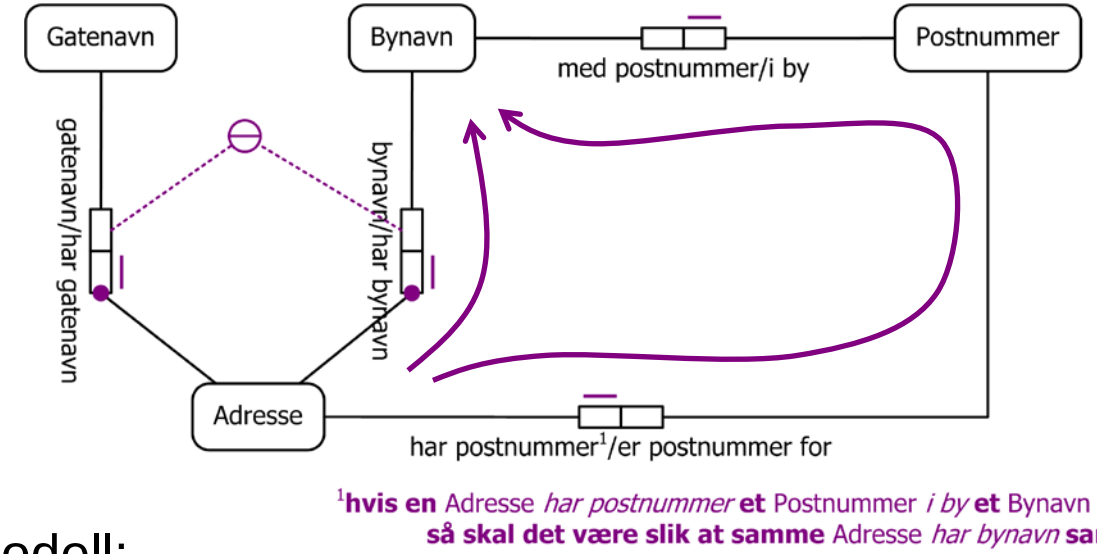


[1]**if an** E e_c C c_b **a** B
then that E e_b **that** B

Her sier EOPen noe vesentlig om virksomhetsområdet. Vi kan ikke utelate noen av faktatypene uten å få et ORM-diagram med andre egenskaper enn det opprinnelige!

ORM og BCNF – eksempel





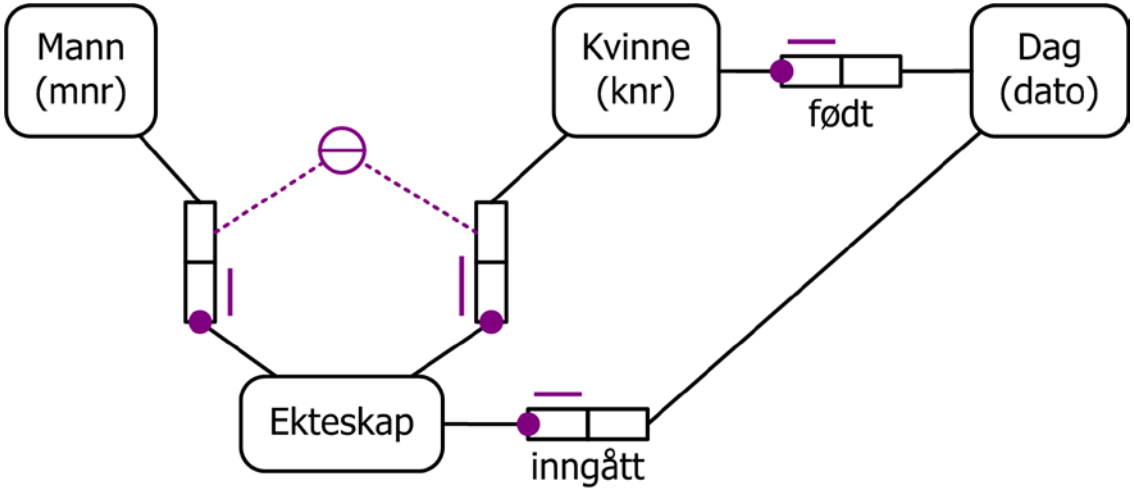
Gruppert ORM-modell:

Adresse(gatenavn, bynavn, postnummer)
Postnummer(postnummer, bynavn)

pluss en integritetsregel som ivaretar sjekk av EOPen (dvs. at oppslag på likt postnummer gir likt bynavn i de to tabellene).

Sykler i ORM-diagrammet

Sykler uten EOPer i et ORM-diagram gir **ikke** normalformsbrudd i ORM-grupperingen. Sammenlikn diagrammet under med diagrammet på s. 40 (brudd på 2NF). Eneste strukturelle forskjell er fraværet av EOPen:

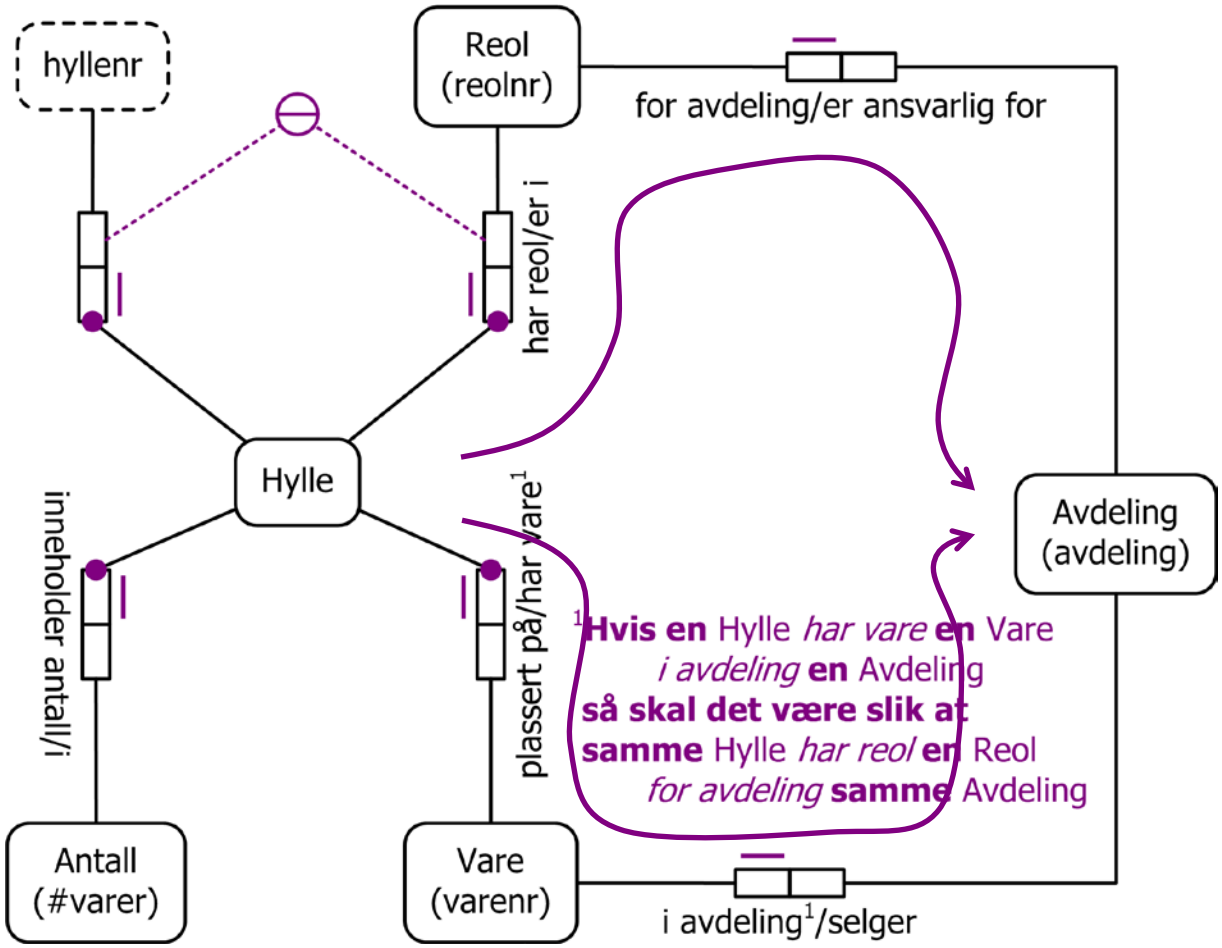


Her vil begrepet Dag kunne gjenfinnes som to urelaterte attributter (men med samme domene) i den grupperte ORM-modellen:

Ekteskap(mnr,knr,dato), Kvinne(knr,dato)

ORM og støyinstanser I

ORM-modellering av varemagasineksempelet:



ORM og støyinstanser II

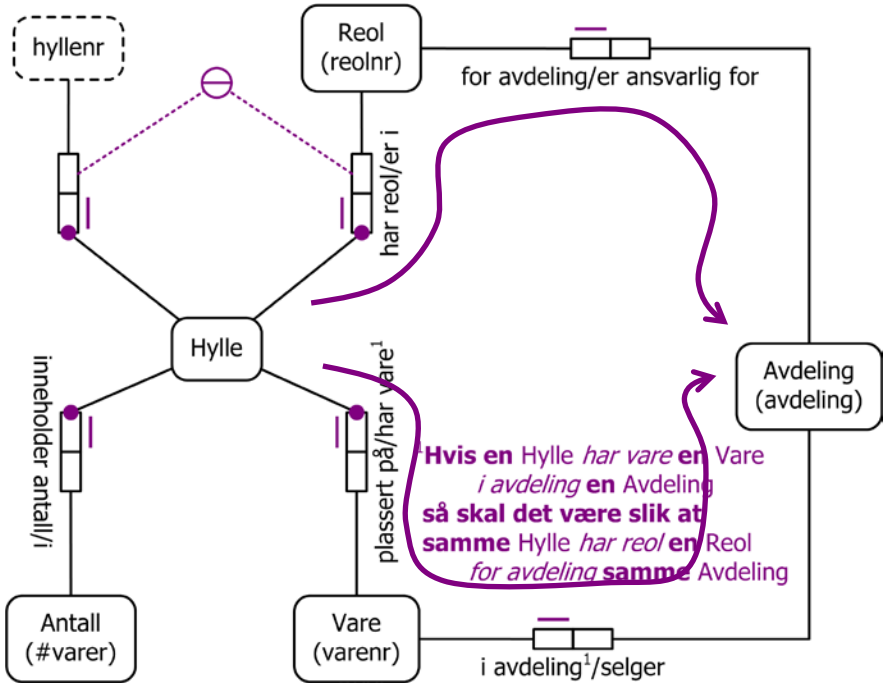
Gruppert ORM-modell er BCNF:

Hylle(<u>reolnr, hyllenr</u>, varenr, #varer)

Reol(<u>reolnr</u>, avdeling)

Vare(<u>varenr</u>, avdeling)

I tillegg må EOPen overholdes: Hvis vi slår opp på et reolnr i Reol, så får vi samme avdeling som om vi slår opp på reolnr i Hylle, velger et tilhørende varenr, og slår opp på dette i Vare). **Hvis EOPen overholdes, vil ikke støyinstanser kunne oppstå!**



ORM og støyinstanser III

- Sykler i snittgrafen til en gruppert ORM-modell opptrer bare hvis vi har EOPer eller andre joinskranker i ORM-diagrammet.
- Dersom systemet overholder alle joinskrankene, får vi ikke støyinstanser. Da vil det ved oppdateringer bli foretatt sjekker som oppdager/hindrer innsetting av støyinstanser.