

IN2090 – Databaser og datamodellering

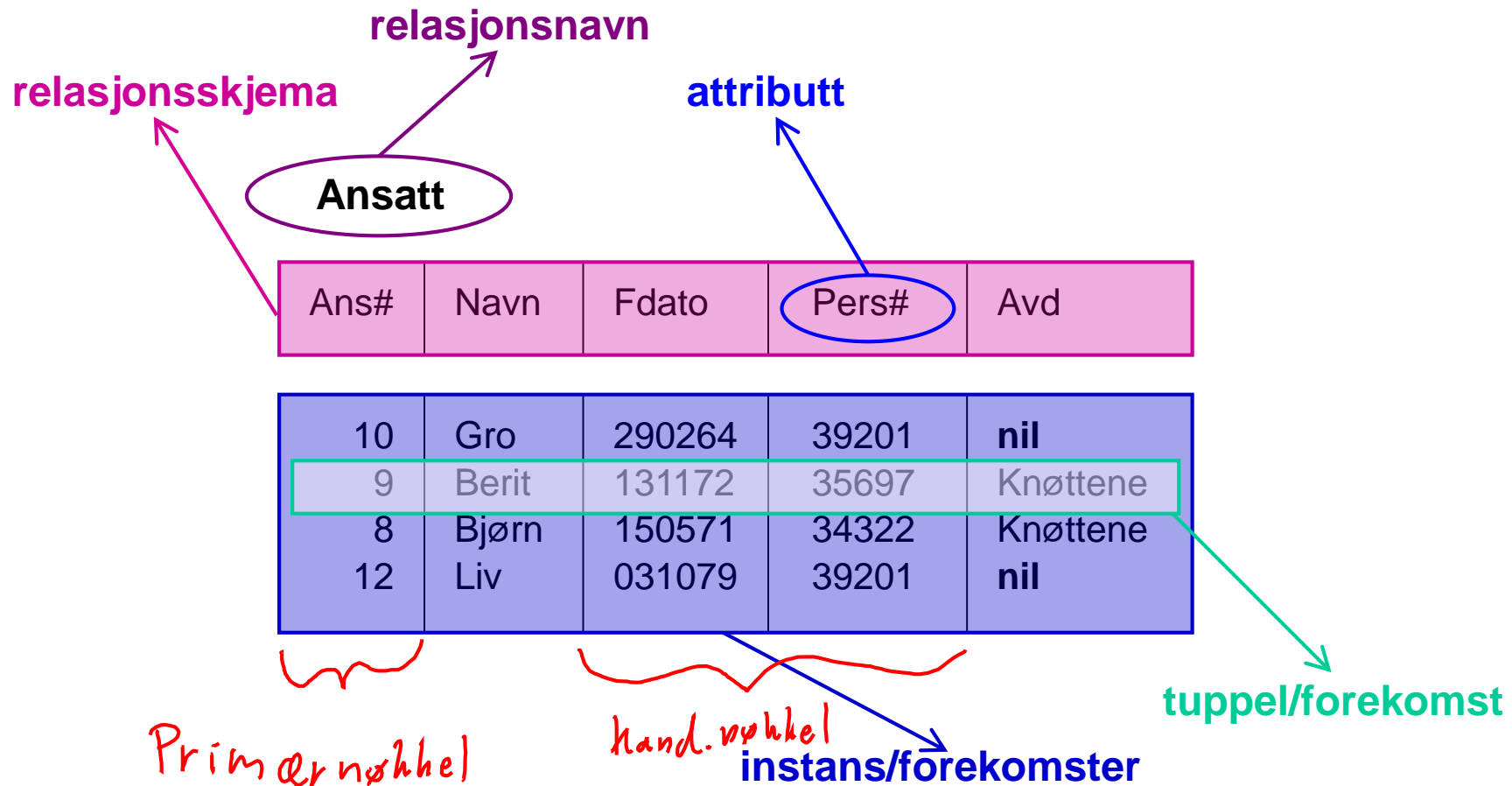
Realiseringsalgoritmen – fra ORM til relasjoner

Intro til mengdeskranker i ORM

Mathias Stang (mjstang@ifi.uio.no)

3. oktober 2018

Repetisjon: Relasjoner



Repetisjon: Nøkler

- **Primærnøkkel, kandidatnøkler**

- En mengde attributter der verdiene ikke kan gjentas
- Alle relasjoner skal ha én primærnøkkel
- Andre nøkler blir da kandidatnøkler

- **Fremmednøkler**

- Ett eller flere attributter der verdiene må finnes i en annen relasjon sin primærnøkkel (kan også peke på en annen relasjons kandidatnøkkel)

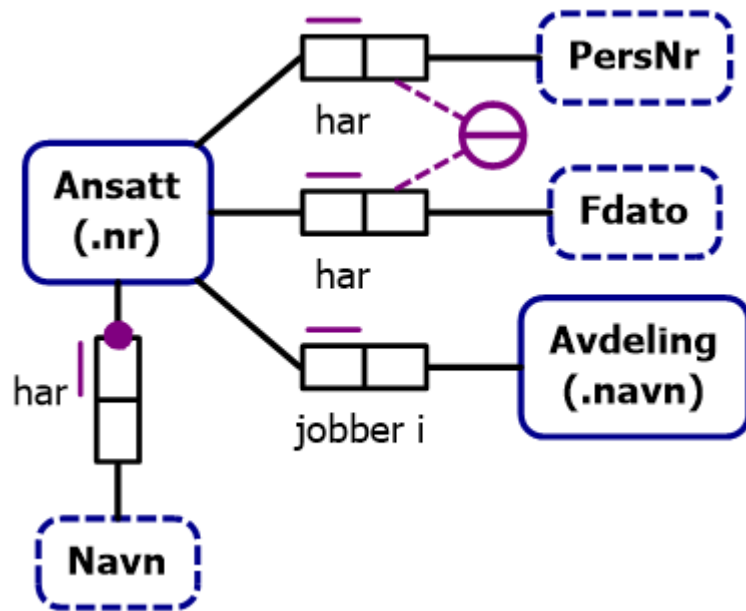
Mål: Gå fra ORM til relasjonsdatabaseskjema

Relasjonene vi lager skal skrives på formen:

Relasjonsnavn(attributt1, attributt2, ...)

- Primærnøkler understrekes
- Om vi også har kandidatnøkler, skal primærnøklerne ha dobbel understreking, og kandidatnøklerne vanlig understreking
- Attributter som kan være null, settes i klammeparenteser
- Fremmednøkler listes opp på formen:
FraRelasjon(attributt) → TilRelasjon(attributt)

Mål: Gå fra ORM til relasjonsdatabaseskjema



Relasjoner:

Ansatt(ansattNr, navn, [fdato], [persNr], [avdeling])

Avdeling(navn)

Fremmednøkler:

Ansatt(avdeling) → Avdeling(navn)

Forutsetninger/forberedelser

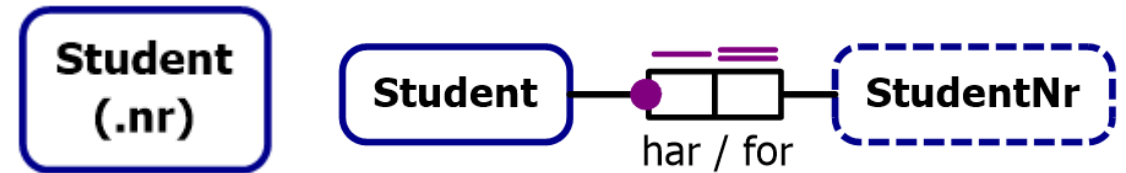
- A. ORM-diagrammet må være **representerbart**
- B. Alle **lange entydighetsskranger** er omformet til korte ved begrepsdannelse
- C. Diagrammet må ikke inneholde **synonyme broer**:
alle broer må ha en entydig begrepsrolle

Fortussetninger/forberedelser

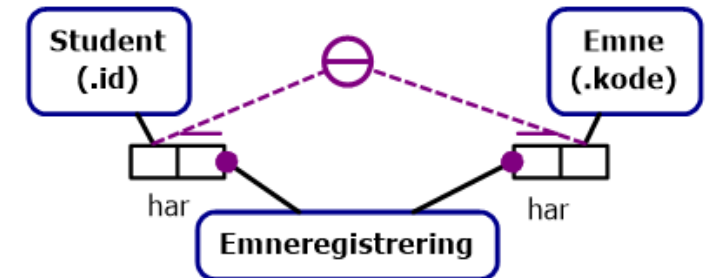
A: Representerbart ORM-diagram

Alle begreper må ha en representasjon på en av følgende måter:

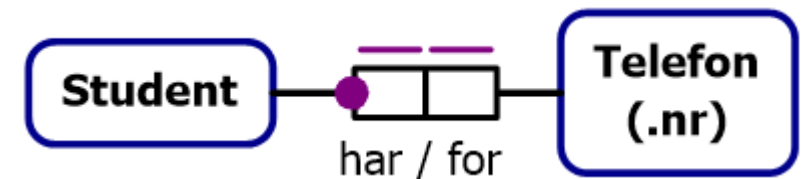
1. Verditype som er knyttet til begrepet med perfekt bro



2. Gjennom ekstern entydighetskranke og påkrevde roller (som i en begrepsdannelse)



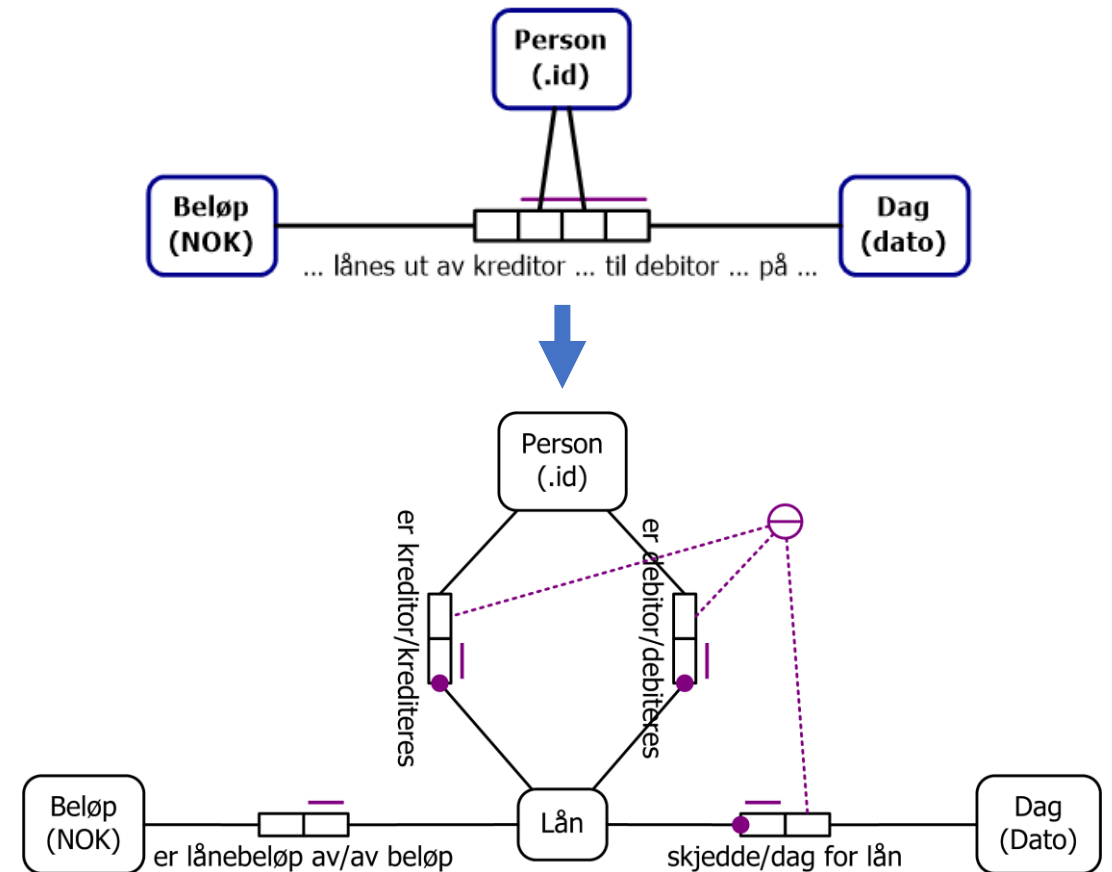
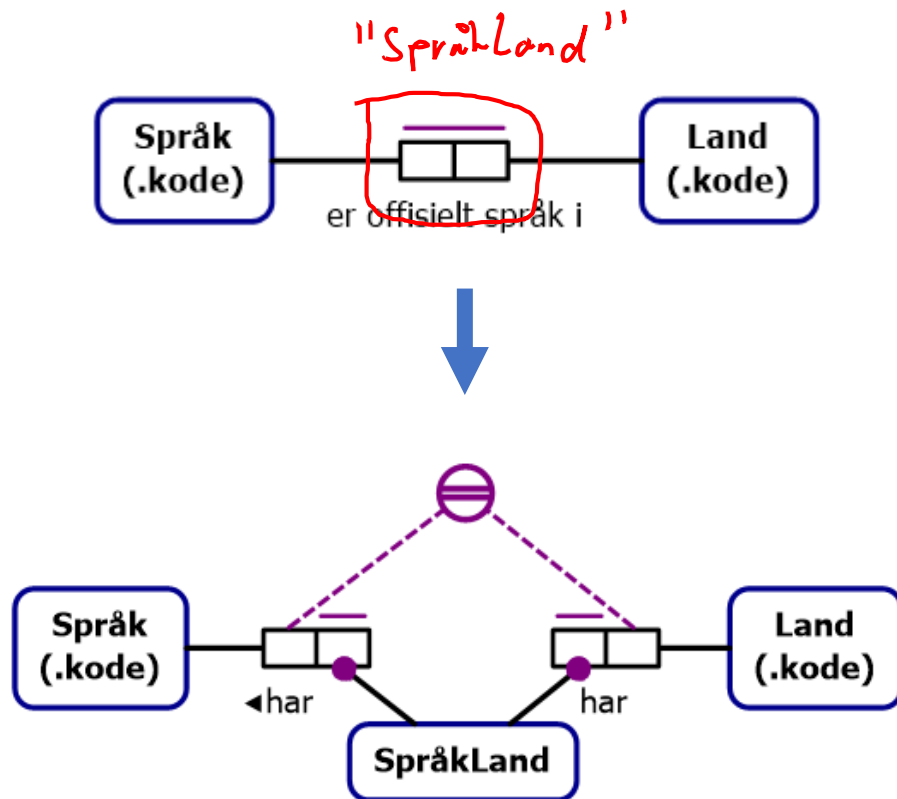
3. Representasjonen til et annet begrep som er knyttet til begrepet via en 1:1-faktatype med påkrevd rolle



Fortussetninger/forberedelser

B: Begrepsdannelse av «lange» entydighetsskranger

Skranger over mer enn én rolle *må* «løses opp» gjennom begrepsdannelse



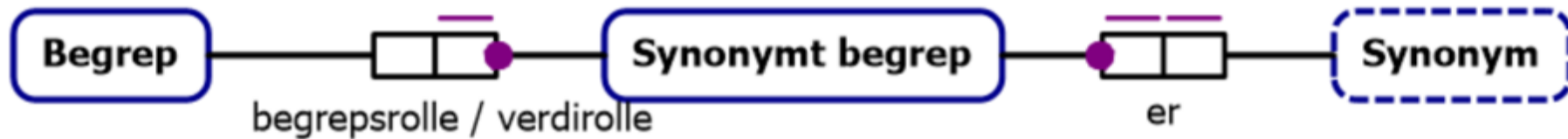
Fortussetninger/forberedelser

C: Fjerne «synonyme broer»

Synonym bro:
(1:n-bro)



må erstattes av:

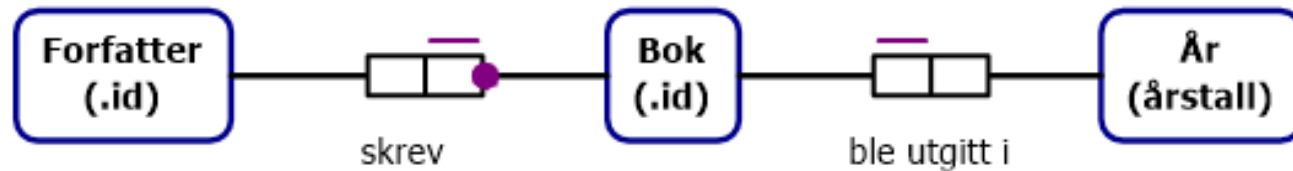


Realiseringsalgoritmen

1. For hvert begrep, lag en relasjon.
2. Velg primær representasjon for alle begreper. Valgt representasjon blir primærnøkler.
3. Behandle gjenstående broer og unære roller. De blir til attributer.
4. Behandle gjenstående faktatyper. De blir til attributer med fremmednøkler.
5. Bestem om undertrykkbare relasjoner kan undertrykkes.

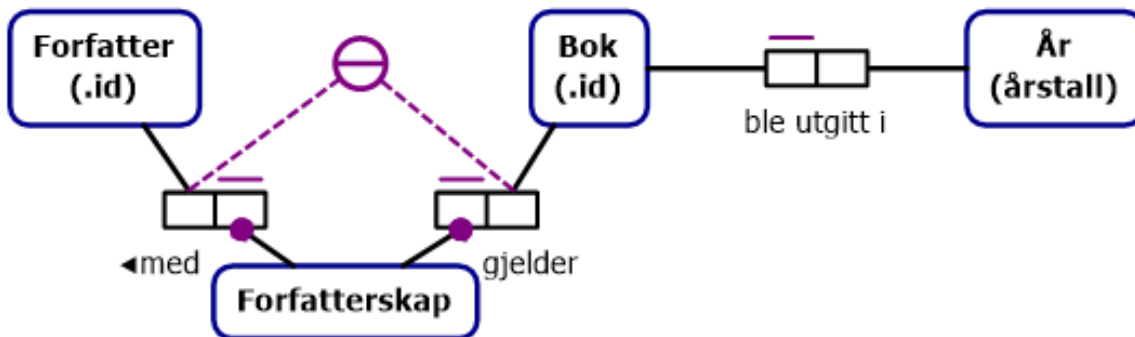
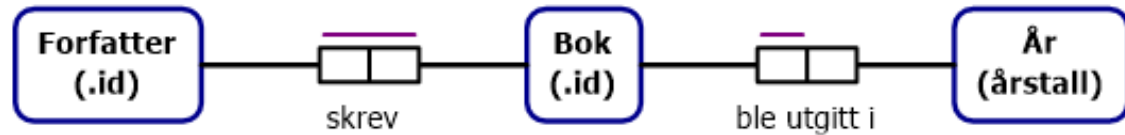
Realiseringsalgoritmen

1: Hvert begrep gir opphav til en relasjon



Relasjoner:

- **Forfatter, Bok, År**



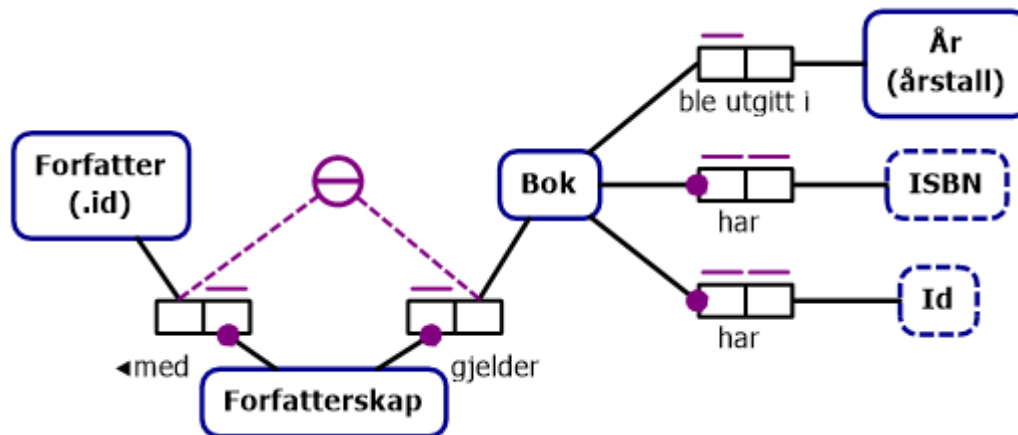
Relasjoner:

- **Forfatter, Forfatterskap, Bok, År**

Realiseringsalgoritmen

2: Identifiser primærnøkler

Velg primær representasjon for alle begreper. Valgt representasjon blir primærnøkler.



Relasjoner:

Forfatter(id)

Forfatterskap(forfatter, bok)

Bok(ISBN, id)

År(årstall)

Fremmednøkler (vi kommer tilbake til dette i steg 4):

Forfatterskap(forfatter) → Forfatter(id)

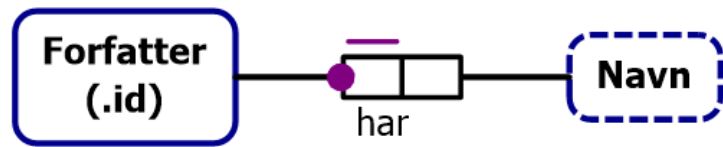
Forfatterskap(bok) → Bok(id)

Realiseringsalgoritmen

3: Resterende broer og unære roller (1/2)

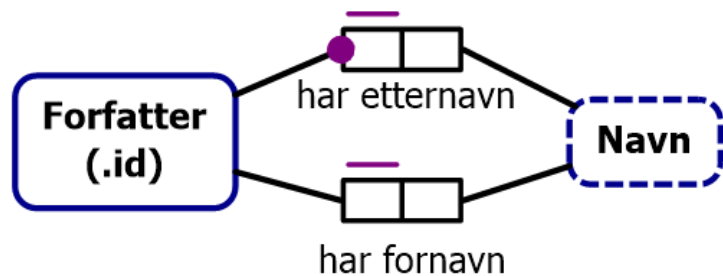
Behandle gjenstående broer og unære roller.

Hver bro / unære rolle gir ett attributt i tabellen til det tilhørende begrepet.



Relasjonen *Forfatter* får attributtet *navn*:

Forfatter(forfatterId, navn)



Relasjonen *Forfatter* får to nye attributter (i tillegg til primærnøkkel):

Forfatter(forfatterId, etternavn, [fornavn])

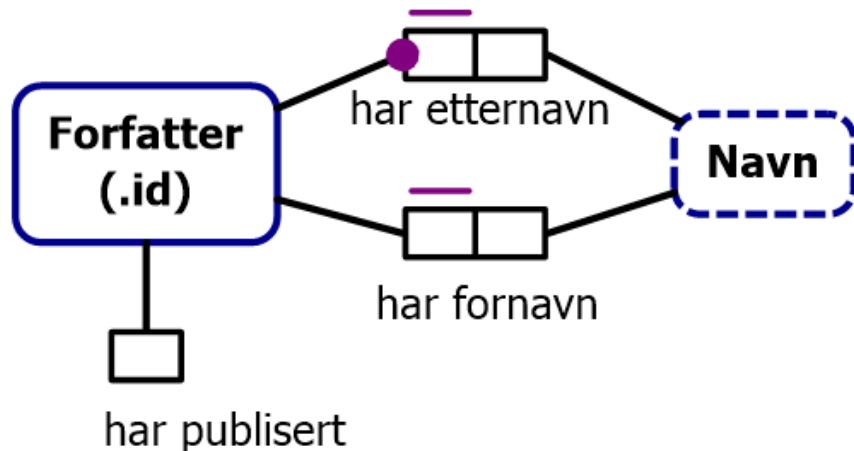
- Merk: Hvis en begrepsrolle er påkrevd, er det ikke tillatt med nullverdier i det tilhørende attributtet
- Her er begrepsrollen «har fornavn» ikke påkrevd
- Vi kan markere at nullverdier er tillatt ved å sette klammer rundt attributtet fornavn

Realiseringsalgoritmen

3: Resterende broer og unære roller (2/2)

Behandle gjenstående broer og unære roller.

Hver bro / unære rolle gir ett attributt i tabellen til det tilhørende begrepet.



Unære roller realiseres på samme måte som broer:

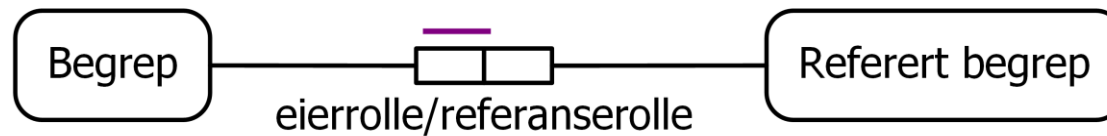
Forfatter(forfatterId, etternavn, [fornavn], har_publisert)

- Nullverdier er aldri tillatt for boolske attributter som stammer fra unære faktatyper
- Rollen i en unær faktatype kan aldri være påkrevd

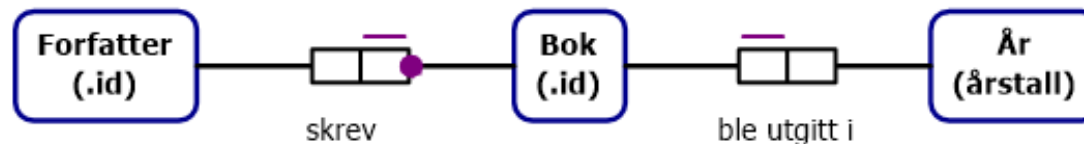
Realiseringsalgoritmen

4: Behandling av faktatyper (1/2)

Behandle gjenstående faktatyper. De blir fremmednøkler.



- I binære faktatyper velges en **entydig** rolle som **eierrolle**; den andre kalles **referanserollen**
 - Hvis begge rollene er entydige og en av dem er påkrevd, velges den påkrevde rollen som eierrolle
- Relasjonen til eierrollens begrep får en fremmednøkkel til det refererte begrepets relasjon



Relasjoner:

Forfatter(id)

Bok(id, forfatterId, [årstall])

År(årstall)

Fremmednøkler:

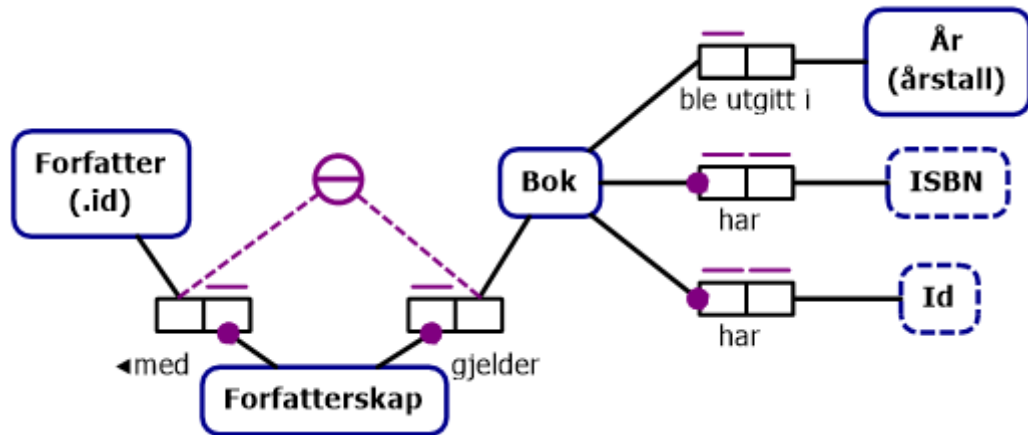
Bok(forfatterId) → Forfatter(id)

Bok(årstall) → År(årstall)

Realiseringsalgoritmen

4: Behandling av faktatyper (2/2)

Behandle gjenstående faktatyper. De blir fremmednøkler.



Faktatyper som inngår i begrepsdannelser (med ekstern entydighetskranke), gir også opphav til fremmednøkler

Relasjoner:

Forfatter(id)

Forfatterskap(forfatter, bok)

Bok(ISBN, id, [utgitt])

År(årstall)

Fremmednøkler:

Forfatterskap(forfatter) → Forfatter(id)

Forfatterskap(bok) → Bok(id)

Bok(utgitt) → År(årstall)

Realiseringsalgoritmen

5: Undertrykkelse

Bestem om undertrykbare relasjoner kan undertrykkes.

Relasjoner:

Forfatter(id)

Forfatterskap(forfatter, bok)

Bok(ISBN, id, [utgitt])

År(årstall)

Fremmednøkler:

Forfatterskap(forfatter) → Forfatter(id)

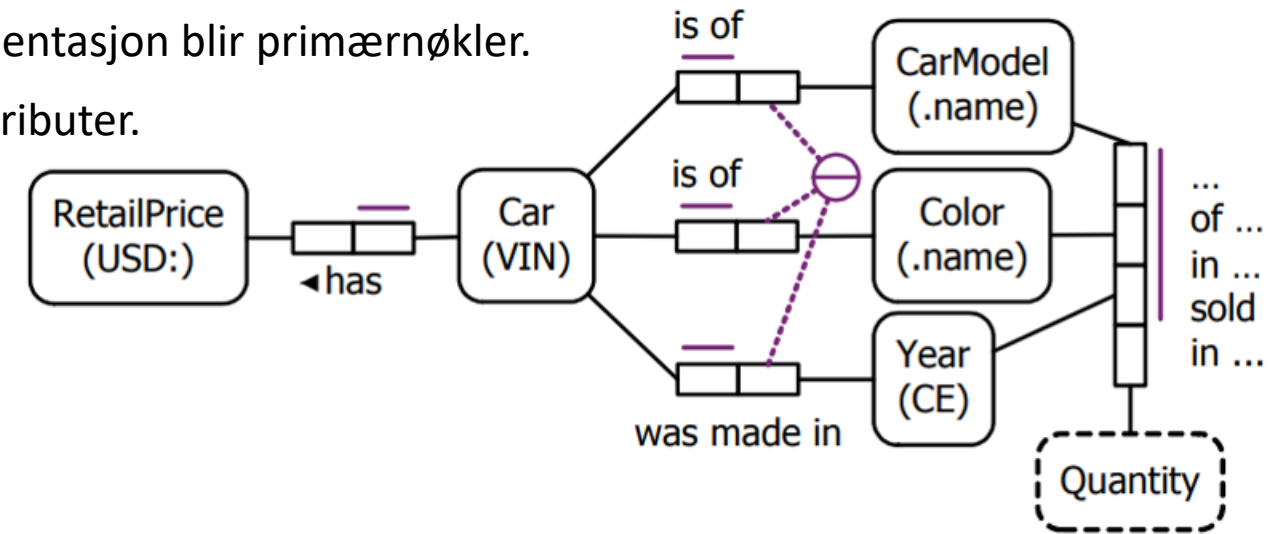
Forfatterskap(bok) → Bok(id)

Bok(utgitt) → År(årstall)

- Vi ser at relasjonen År ikke tilfører noe nyttig informasjon
- Relasjoner som kun består av en primærnøkkel, og som har minst én fremmednøkkel som peker til seg, kan strykes (undertrykkes)
- Blir som om vi i utgangspunktet hadde modellert begrepet som en verditype
- Undertrykkelse er valgfritt!

Et eksempel av hele realiseringsprosessen

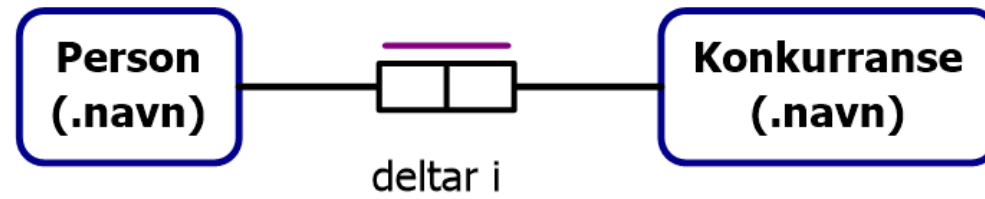
1. For hvert begrep, lag en relasjon.
2. Velg primær representasjon for alle begreper. Valgt representasjon blir primærnøkler.
3. Behandle gjenstående broer og unære roller. De blir til attributer.
4. Behandle gjenstående faktatyper. De blir fremmednøkler.
5. Bestem om undertrykkbare relasjoner kan undertrykkes.



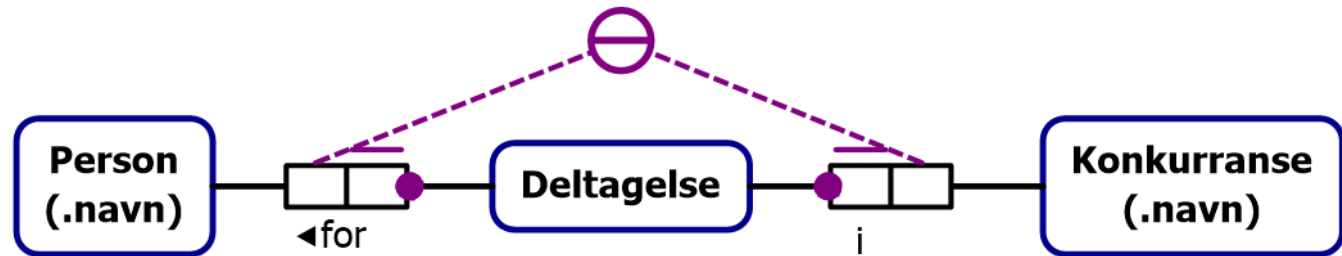
Detta rakk vi ikke gjennomgå, blir lagt ut som ukeoppgave for uke 7. Løsningsforslag blir publisert 10. okt.

Begrepsdannelse og objektifisering

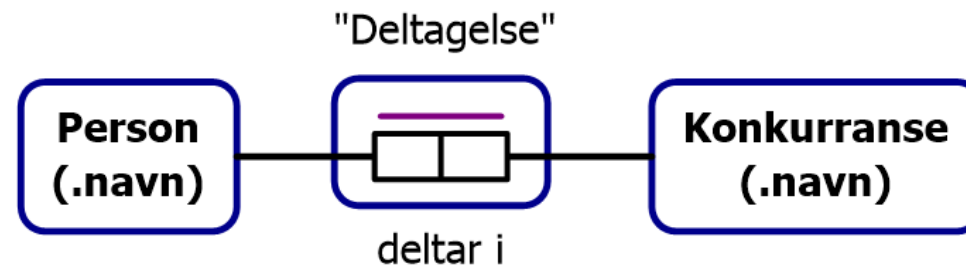
Mange-til-mange-skranker
kan gi opphav til nytt begrep



Vi kan bruke begrepsdannelse
for å lage dette begrepet



Men ORM gir oss også en syntaks
for å **objektifisere** faktatypen. Dette
er gyldig ORM-syntaks og betyr det
samme som begrepsdannelsen

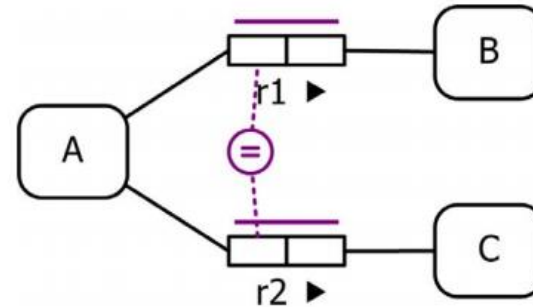


ORM: Mengdeskranker

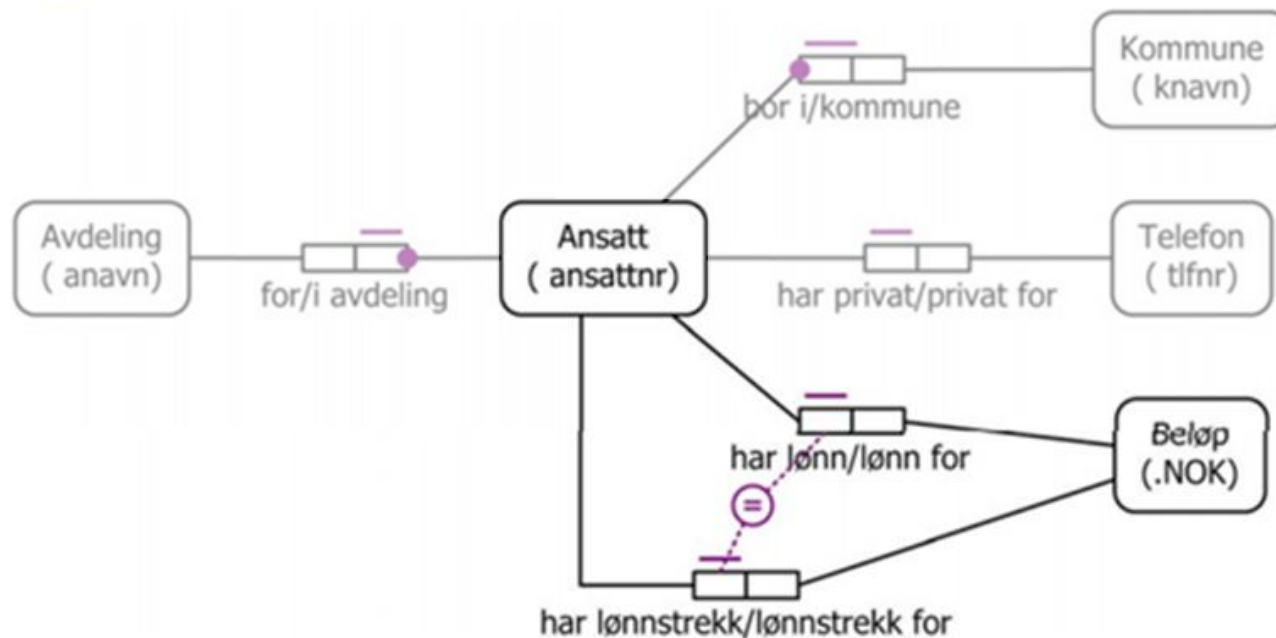
- Mengdeskrankene begrenser mengden av forekomster i en eller flere roller i forhold til forekomstene i andre roller
- Finnes i følgende varianter:
 - Likhetskranker (equality constraint)
 - Ulikhetskranker (exclusion constraint)
 - Delmengdeskranker (subset constraint)

Mengdeskranke: Likhetskranke

⊜ Likhetskranke



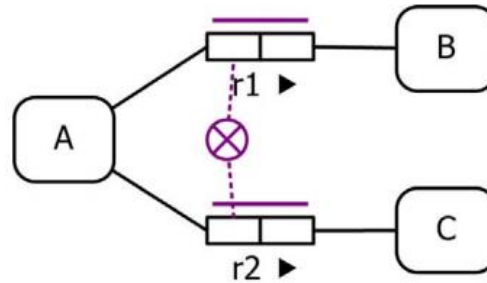
A skal ha rollen r1
hvis og bare hvis
A har rollen r2.



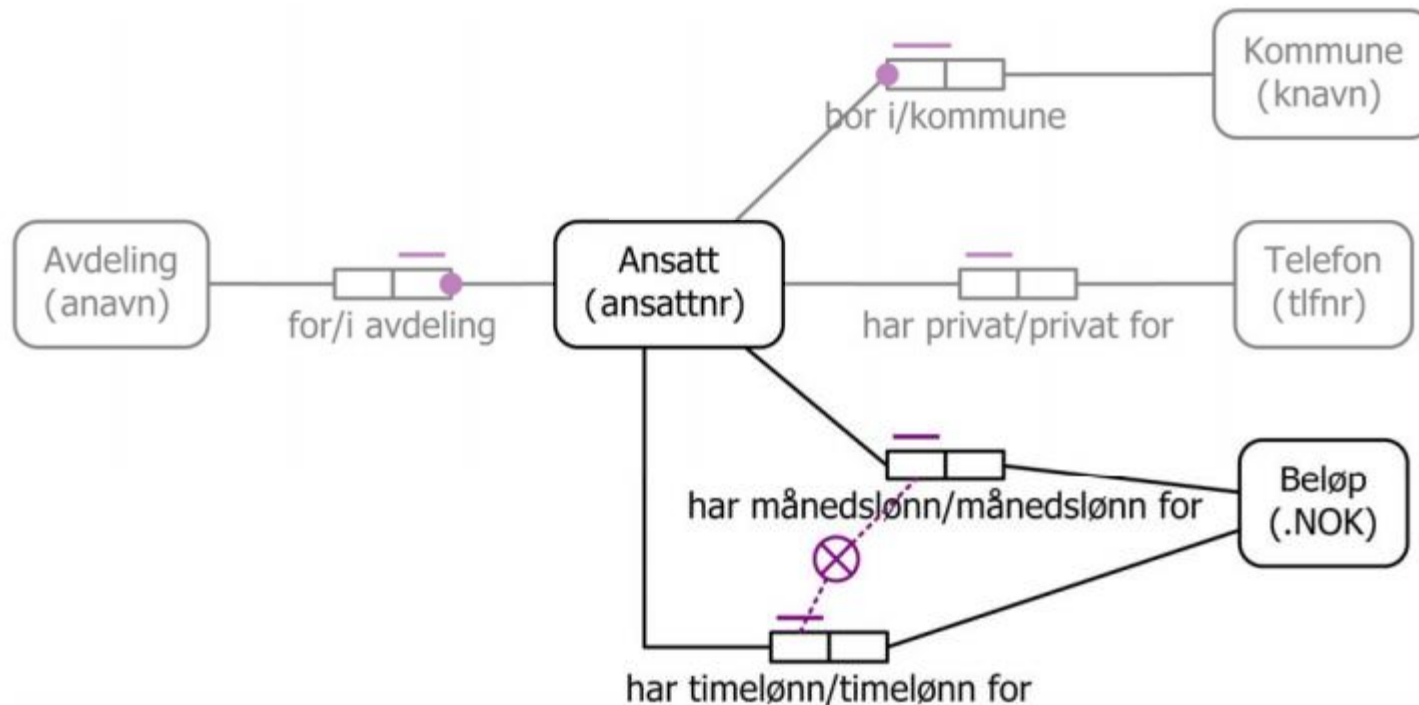
Konsekvens: En ansatt som **har lønn**
må også **ha lønnstrekk** (og motsatt)

Mengdeskranke: Ulikhetskranke

⊗ Ulikhetskranke



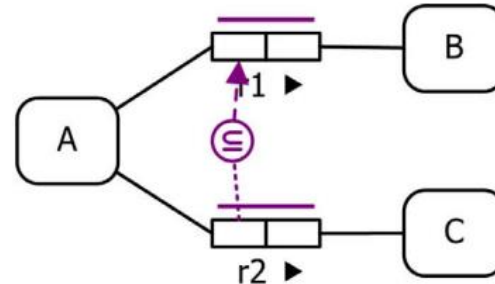
A skal ikke ha både rollen r1 og r2.



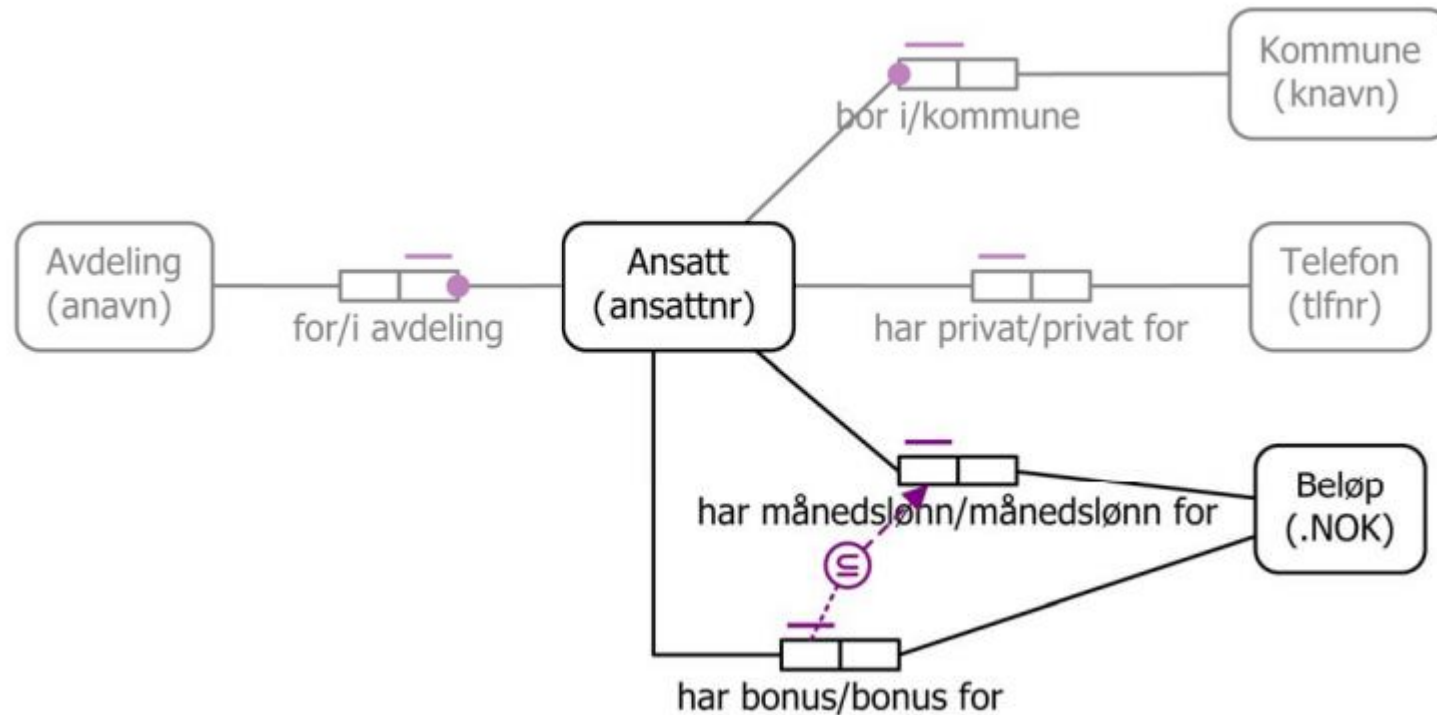
Konsekvens: En ansatt kan ikke ha både månedslønn og timelønn

Mengdeskranker: Delmengdeskranker

⊆ Delmengdeskranker



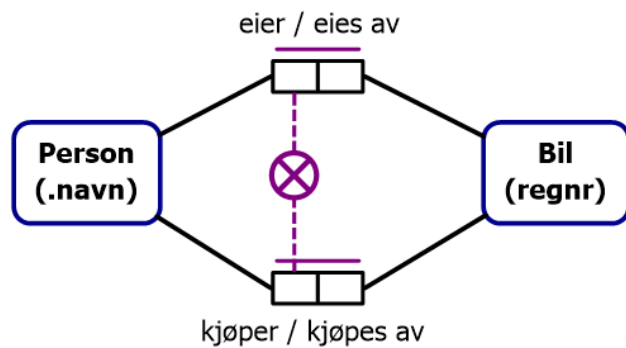
Hvis A har rollen r2,
så skal A også ha rollen r1.



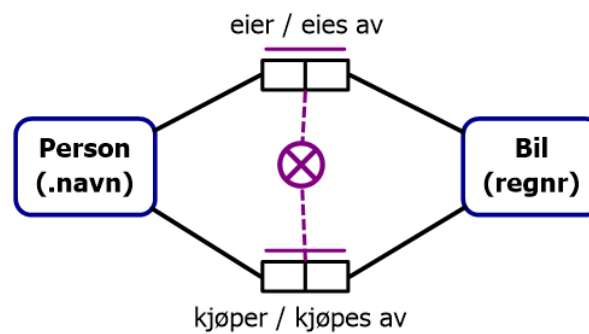
Konsekvens: Ansatt **har bonus**
bare hvis Ansatt **har månedslønn**

Hvor plasseres den eksterne skranken?

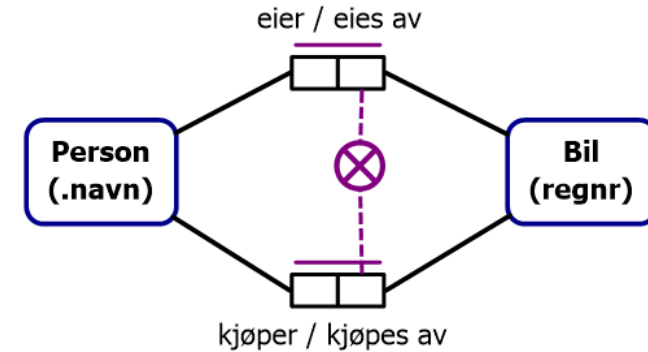
Skranken kan settes på ulike måter som endrer betydningen av modellen!



Person som eier bil
kan ikke også kjøpe bil



Person kan ikke kjøpe
en bil som han/hun eier



En bil kan ikke både
eies og kjøpes