

INF1300 – Introduksjon til databaser

Repetisjon: (nesten) alt du trenger å kunne om ORM og realisering

Mathias Stang (mjstang@ifi.uio.no)

21. november 2017

Agenda

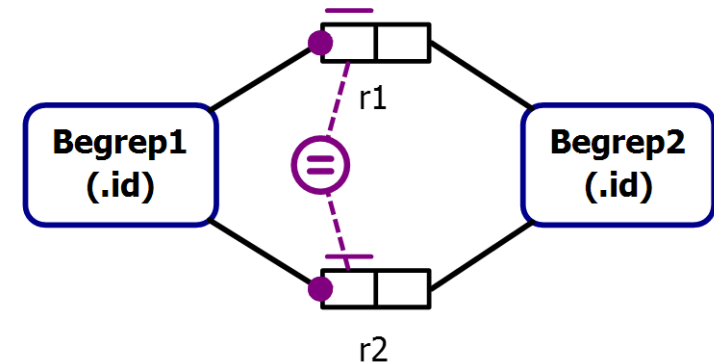
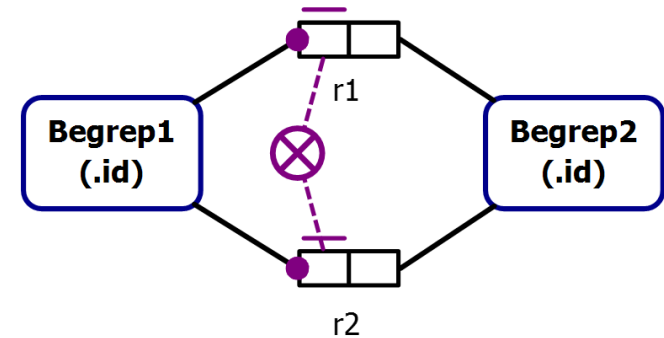
- Hensikten med ORM-modellering
- Hva er lov i ORM-modeller?
- Begreper og verdityper
- Roller
- Grunnleggende skranker
- Setningstyper
- Begrepsdannelse og objektifisering
- Mer avanserte skranker
- Underbegreper
- Realisering
- Andre tips og triks

Hvorfor modellerer vi?

- For å illustrere hvordan man tenker på en informativ, enkel og oversiktlig måte
- Å danne grunnlag for en database som kan løse det konkrete problemet (realiserbar modell)

Hva er egentlig lov i en ORM-modell?

- Gir det mening?
 - F.eks. skranker som motsier hverandre.
- Er det nødvendig?
 - Unngå redundans i modellen.



Begreper og verdityper

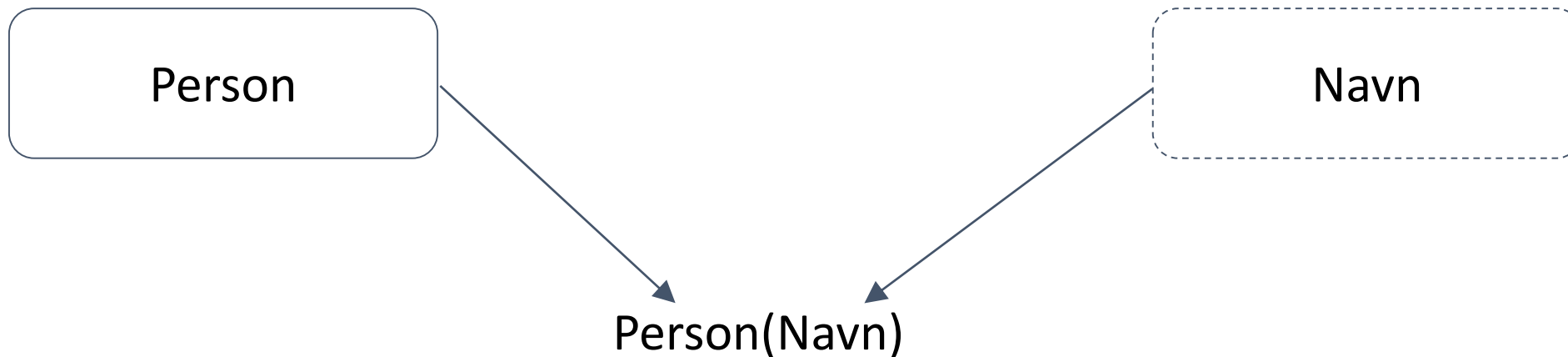
Fellesbetegnelse: «Objekter»

Begrep:

- Boks med heltrukket linje
- Modellerer et (abstrakt) konsept
- Ender opp som en relasjon
- Må være refererbart via verdityper

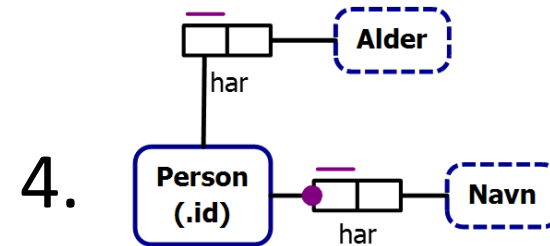
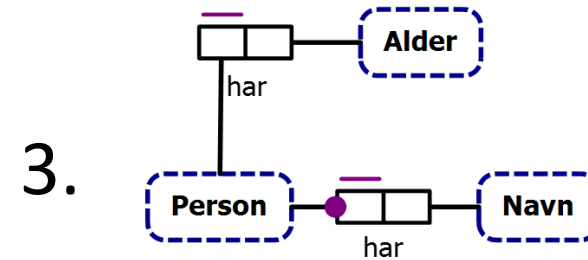
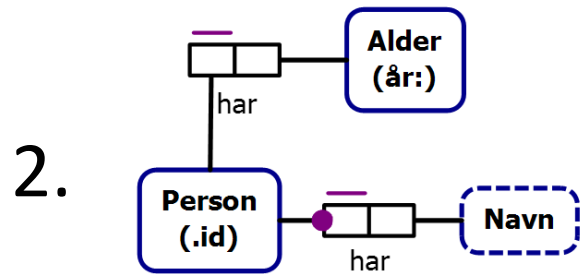
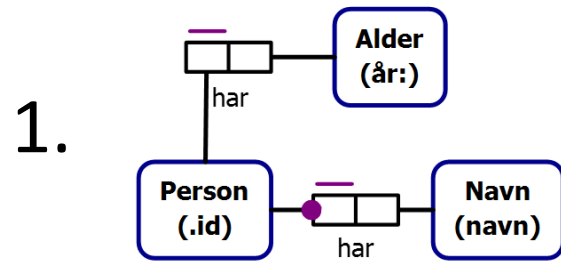
Verdtype:

- Boks med stiplet linje
- Modellerer en verdi (manifestering av et begrep)
- Ender opp som attributt(er)
- Kan ikke ha relasjoner til andre verdityper



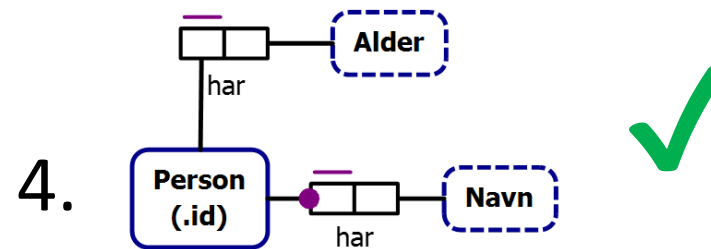
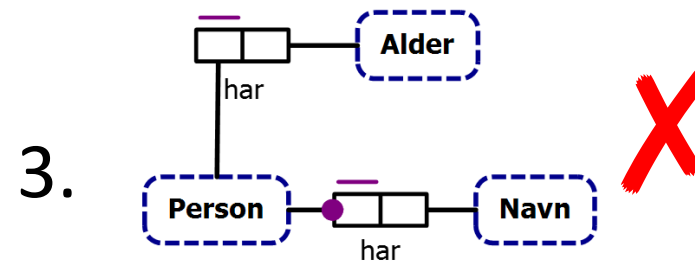
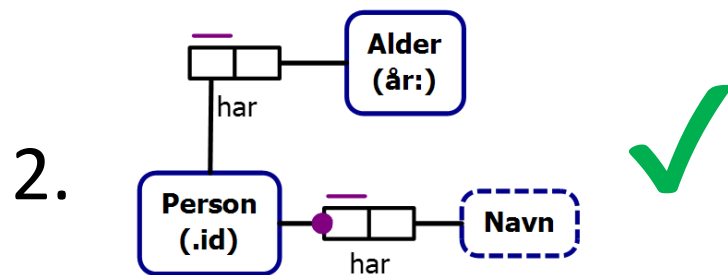
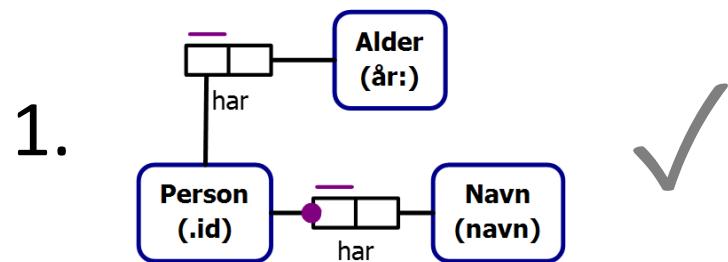
Begreper og verdityper

Begrep eller verditype?



Begreper og verdityper

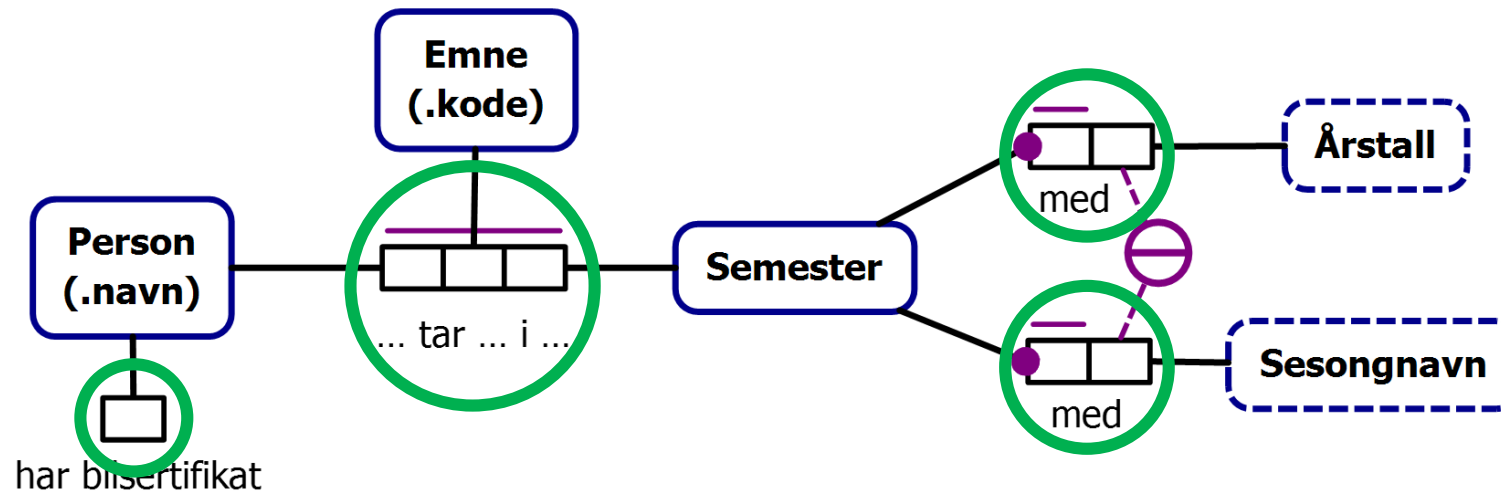
Begrep eller verditype?



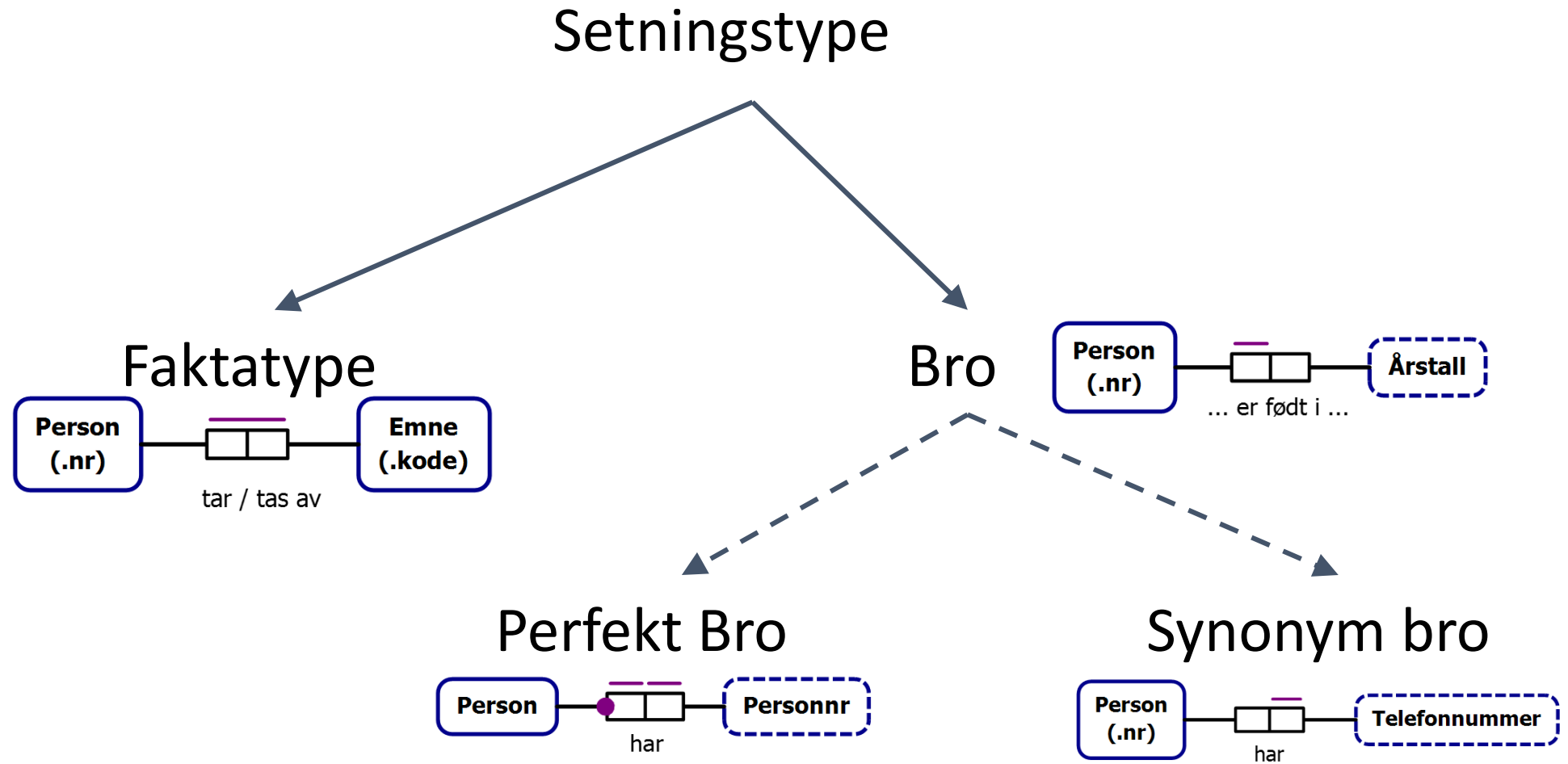
Roller

En rolle:

- Er en firkantet boks med heltrukket linje
- Har navn som beskriver rollen
- Knyttes alltid til ett og bare ett objekt (*begrep, verditype*)
- Danner grunnlag for setningstyper (relasjoner)

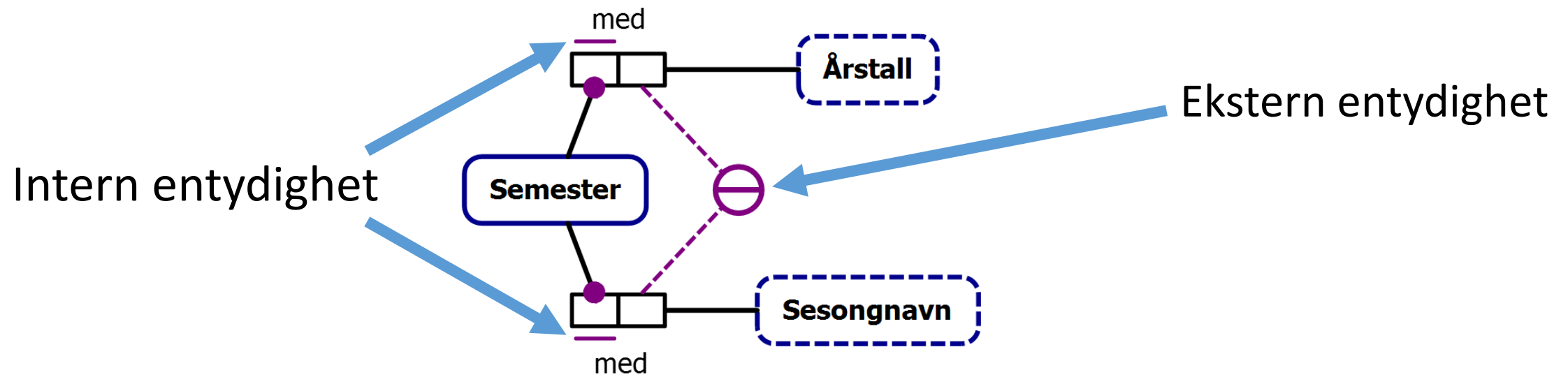


Setningstyper



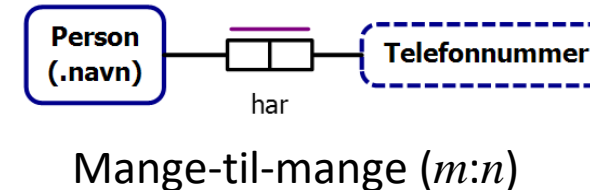
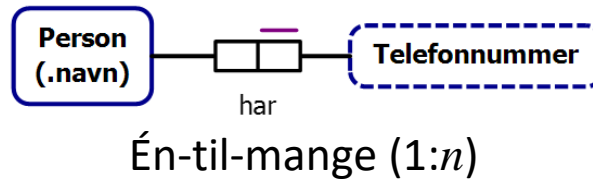
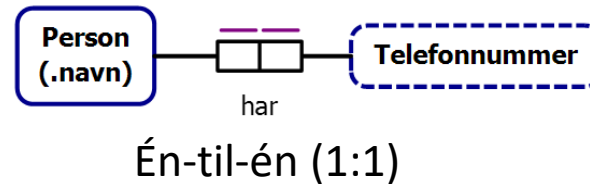
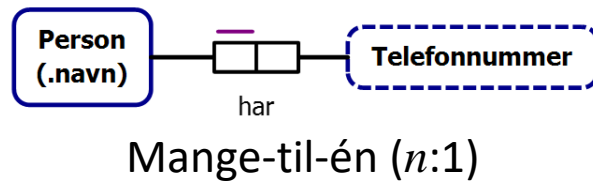
Entydighetsskranker

- Gir restriksjoner for hvilke kombinasjoner av objekter som kan forekomme i en relasjon
- Kan opptre både som intern entydighet og som ekstern entydighet

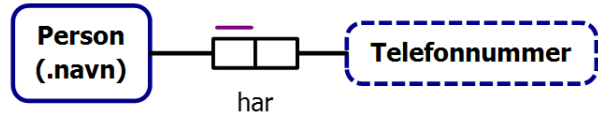


Intern entydighet

- Alle setningstyper må ha dette!
- Må være over n eller $n-1$ antall roller i relasjonen (hvor n er totalt antall roller)
- *Ble tidligere kalt «piler» etter gammel ORM-notasjon*

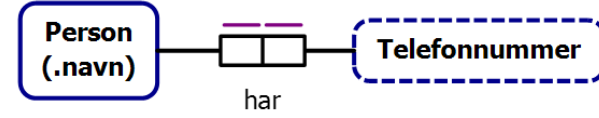


Intern entydighet



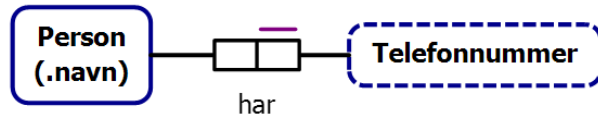
Mange-til-én ($n:1$)

Person	Tlf.nr	
Simen	654	
Lars	654	
Simen	458	⊘
Simen	654	⊘



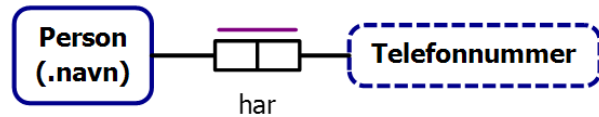
Én-til-én ($1:1$)

Person	Tlf.nr	
Simen	654	
Lars	654	⊘
Simen	458	⊘
Simen	654	⊘



Én-til-mange ($1:n$)

Person	Tlf.nr	
Simen	654	
Lars	654	⊘
Simen	458	
Simen	654	⊘

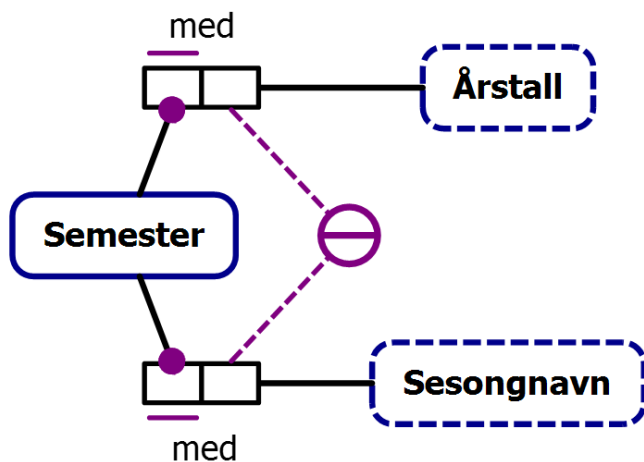


Mange-til-mange ($m:n$)

Person	Tlf.nr	
Simen	654	
Lars	654	
Simen	458	
Simen	654	⊘

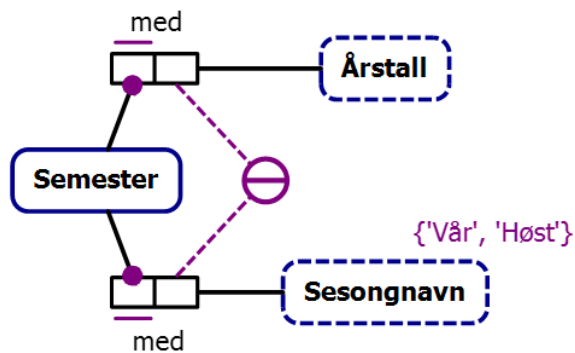
Ekstern entydighet

- Setter begrensninger for forekomster av objekter i forskjellige setningstyper
- Står (nesten) alltid over ett begrep, men på rollene som ikke spilles av begrepet.
- Ender opp enten som primærnøkkel eller kandidatnøkkel i realisering
- Svært viktig hvilken av rollene i setningstypene de kobles til



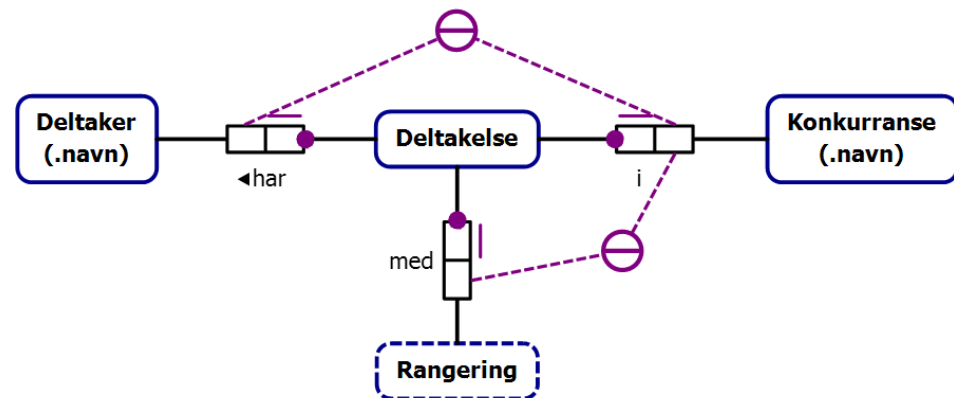
Semester(Årstall, Sesongnavn)

Ekstern entydighet



Semester(Årstall, Sesongnavn)

Årstall	Sesongnavn
2013	Høst
2014	Vår
2014	Høst
2014	Høst

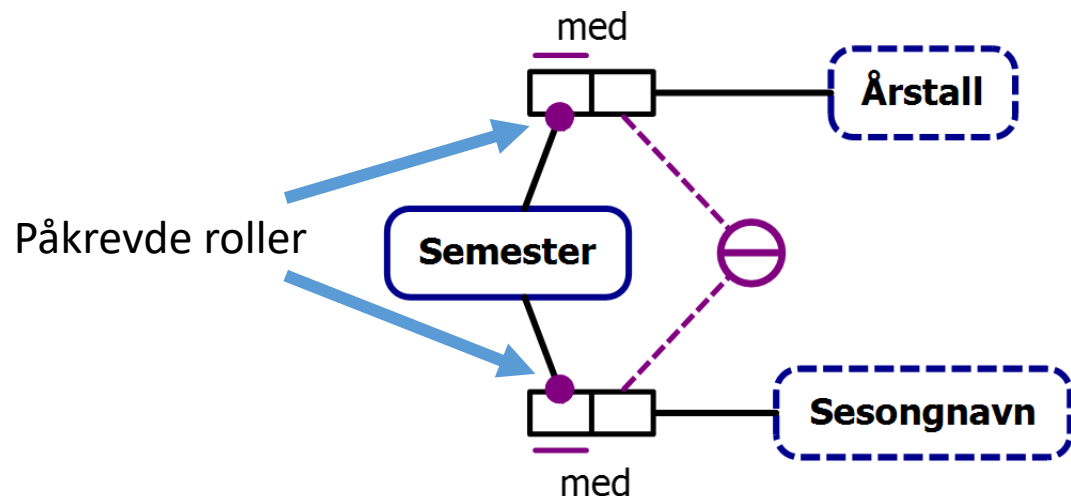


Deltakelse(Deltaker, Konkurransen, Rangering)

Deltaker	Konkurransen	Rangering
Simen	Ski	5
Ole	Løping	2
Kari	Ski	5
Kari	Ski	3
Simen	Ski	3
Kari	Løping	1

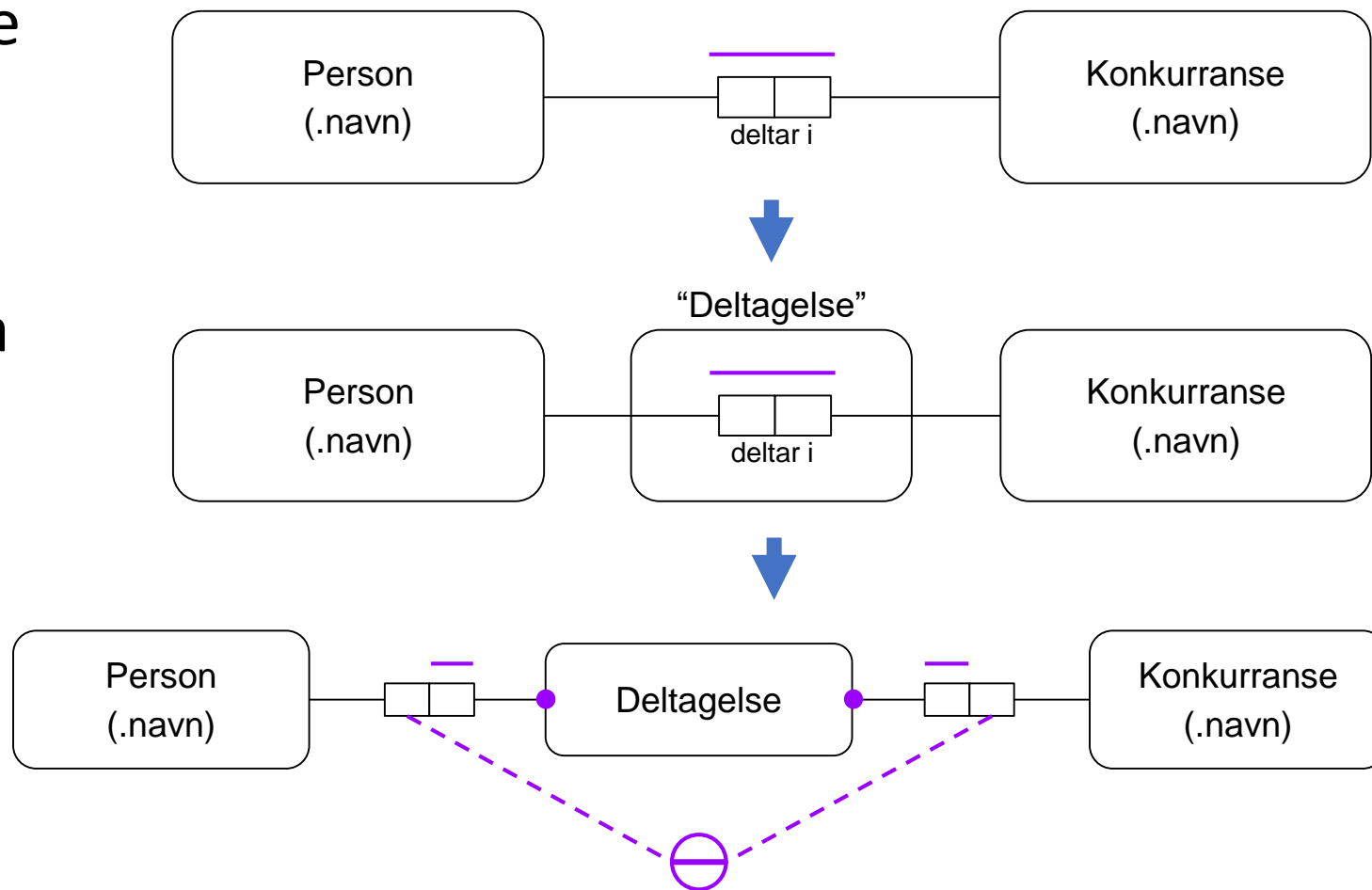
Påkrevde roller

- Objektet skal spille denne rollen
- I SQL resulterer dette i NOT NULL i CREATE-spørringen
- «Attributtet kan aldri stå tomt»
- Alltid involvert i forbindelse med primærnøkkel
- *Tidligere kalt «totale roller»*



Begrepsdannelse og objektifisering

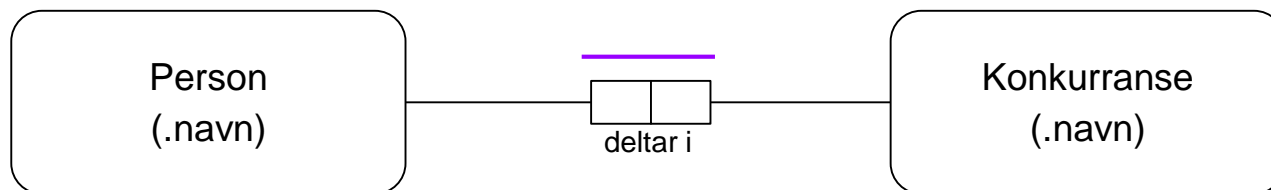
- Mange-til-mange kan ikke realiseres direkte
- Vi må lage nytt begrep på rolleparet for å realisere
- Det kan være lurt å sette opp en begrepsdannelse for å forstå hva som skjer



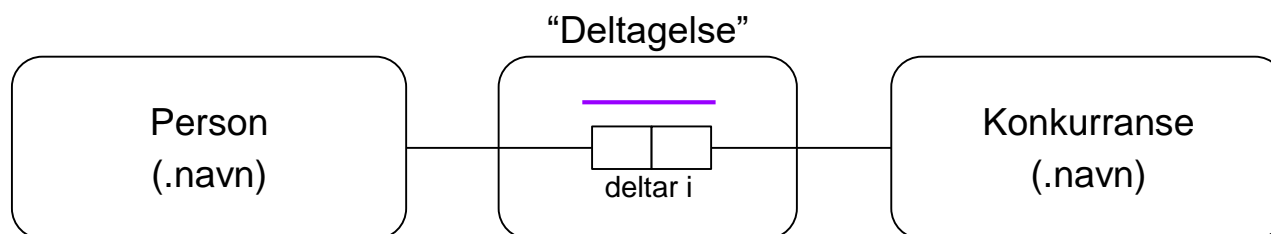
Begrepsdannelse og objektifisering

Enkelt $m:n$ rollepar

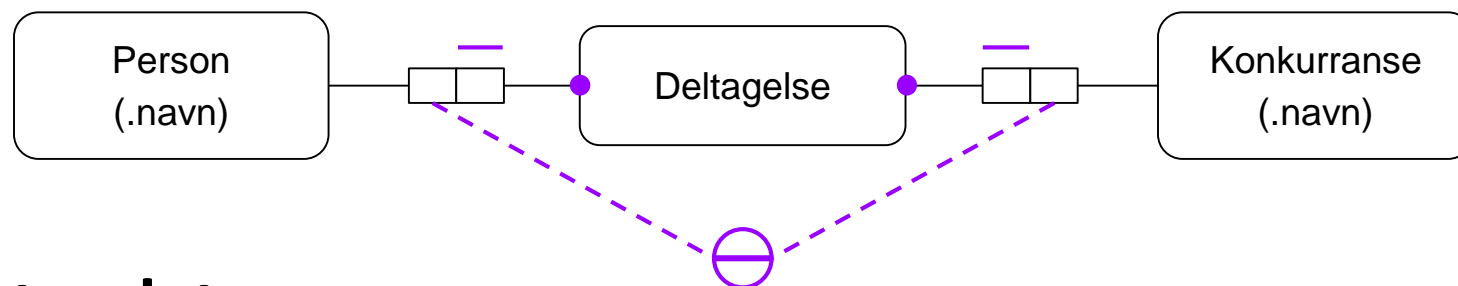
(dette er bra, for det gir enklest modell!)



Objektifisering

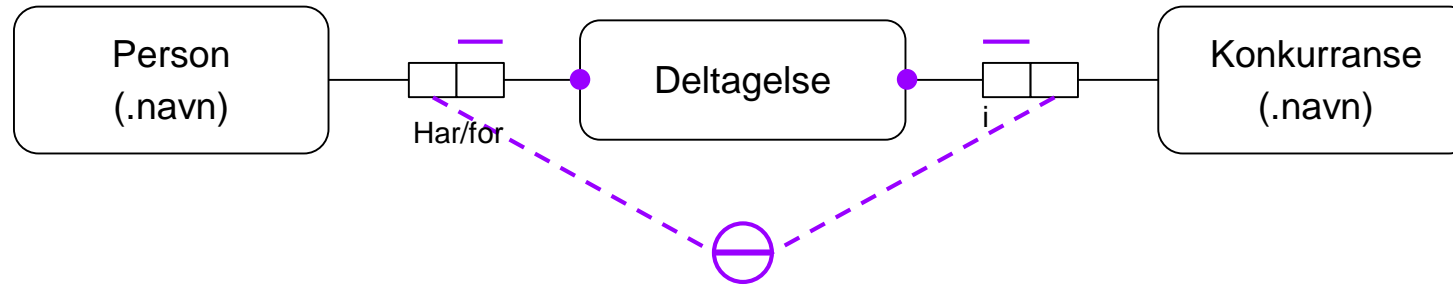


Begrepsdannelse



VIKTIG: Disse 3 formene betyr det samme

Realisering av mange-til-mange



Relasjoner:

Person (personnavn)

Konkurransen (konkurransenavn)

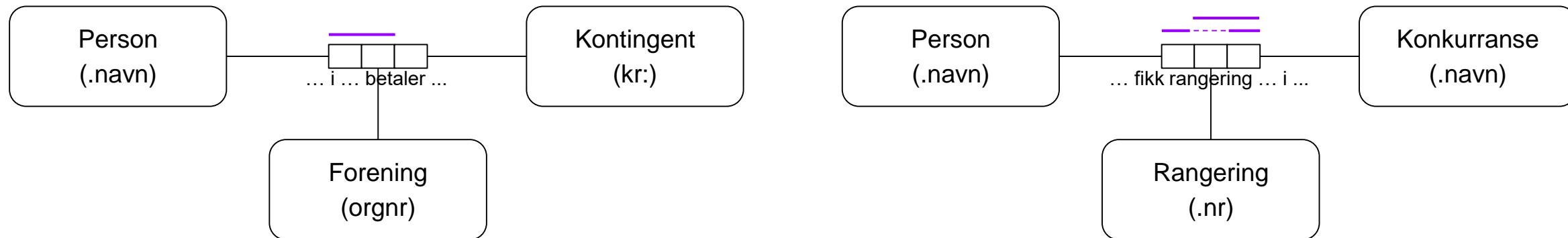
Deltagelse (personnavn, konkuransenavn)

Fremmednøkler:

Deltagelse(personnavn) → Person(personnavn)

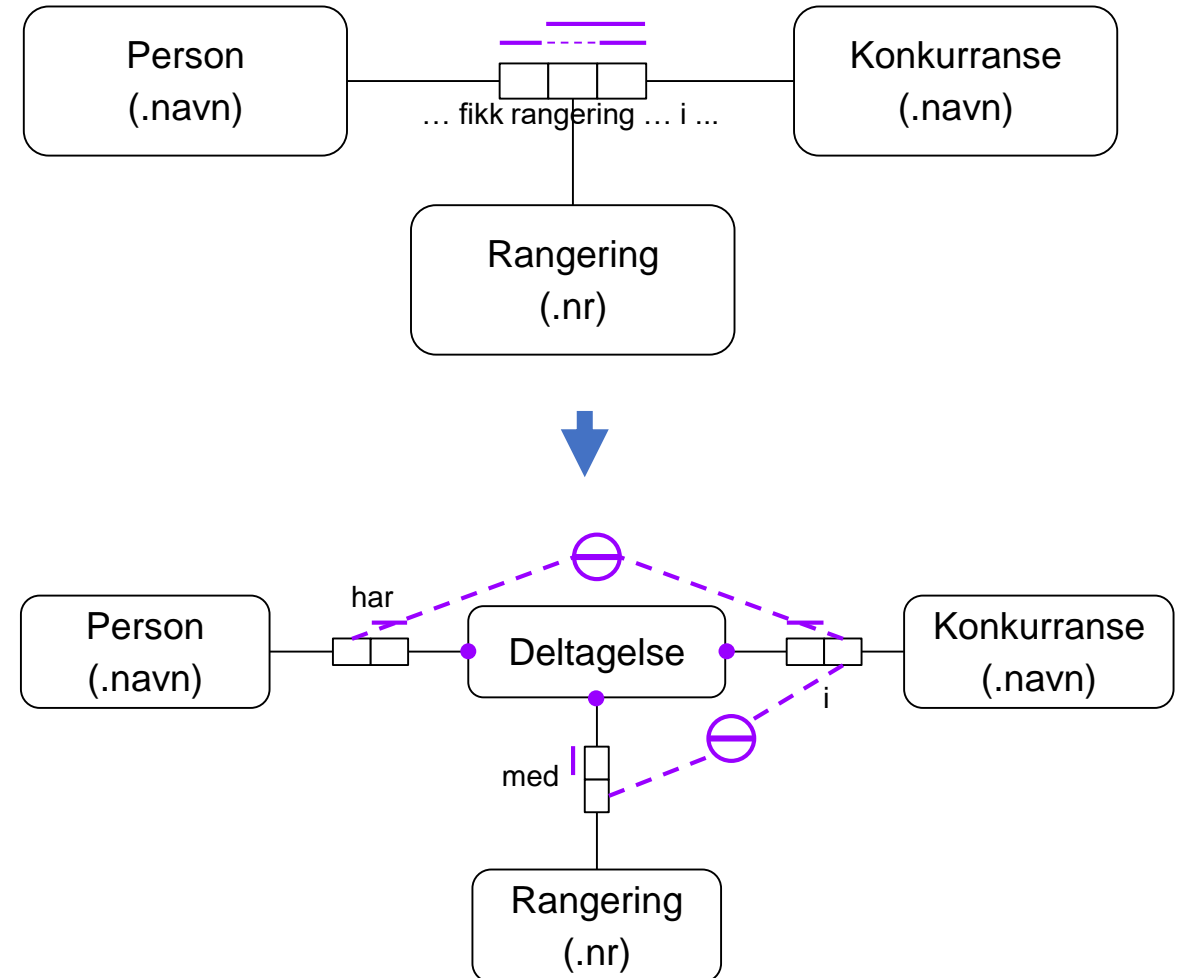
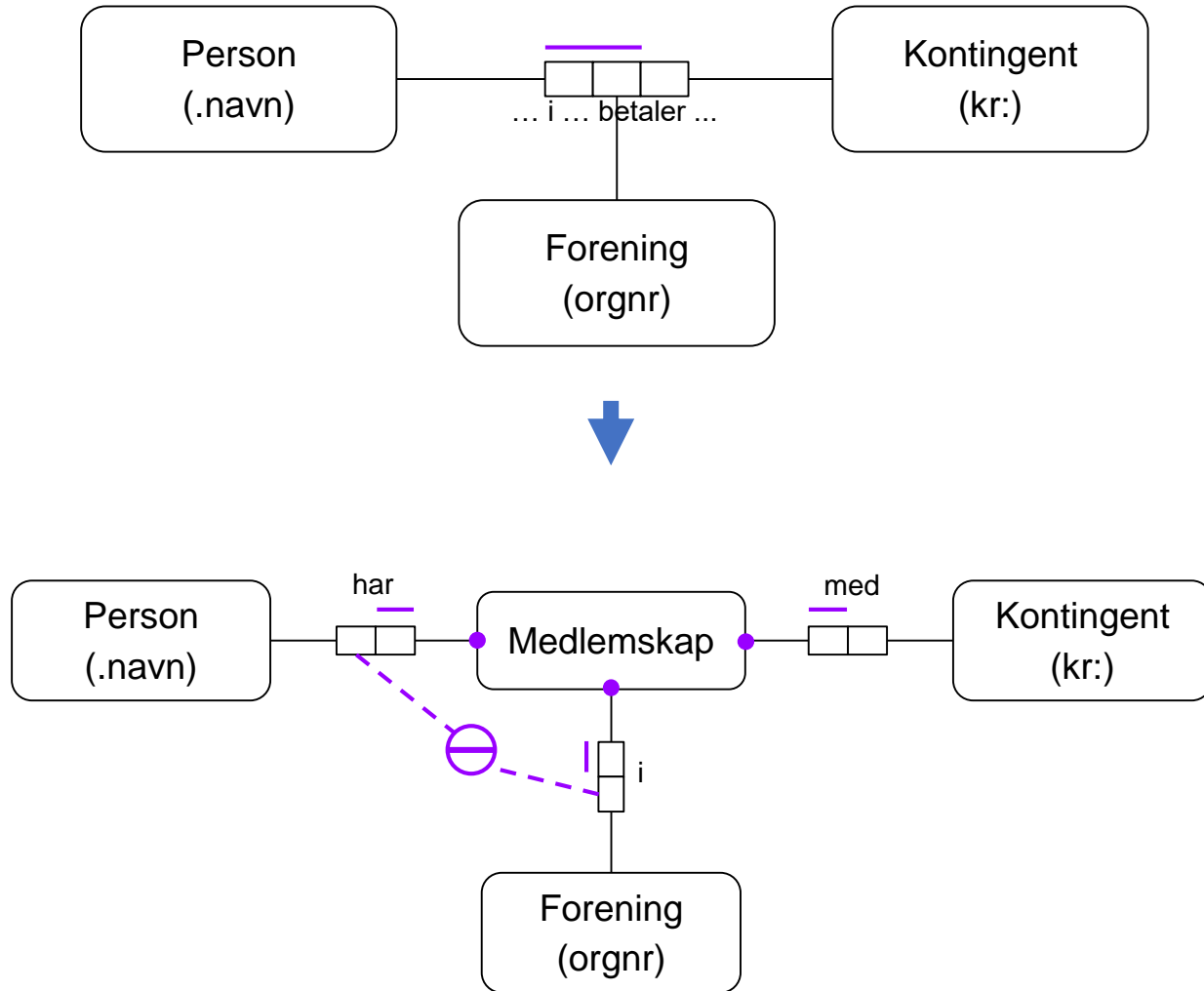
Deltagelse(konkurransenavn) → Konkurransen(konkurransenavn)

Ternære faktatyper



- **Elementære** setninger med aritet 3: *(se forelesning 22. aug og 25. aug)*
 - Person betaler kontingent i forening *(tidligere obligoppgave)*
 - Student fikk karakter i emne *(eksamen 2014/prøveeksamen 2015)*
 - Person fikk rangering i konkurransen *(tilsvarende eksamen 2013)*

Begrepsdannelse av ternære faktatyper



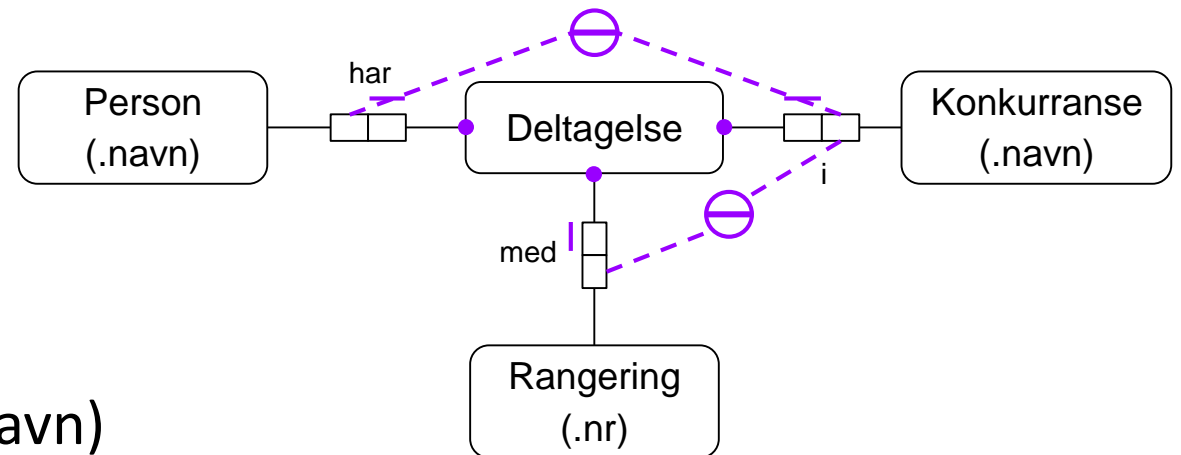
Realisering av ternære faktatyper

Deltagelse (personnavn, konkurransenavn, rangeringnr)

Person (personnavn)

Rangering (rangeringnr)

Konkurranse (konkurransenavn)



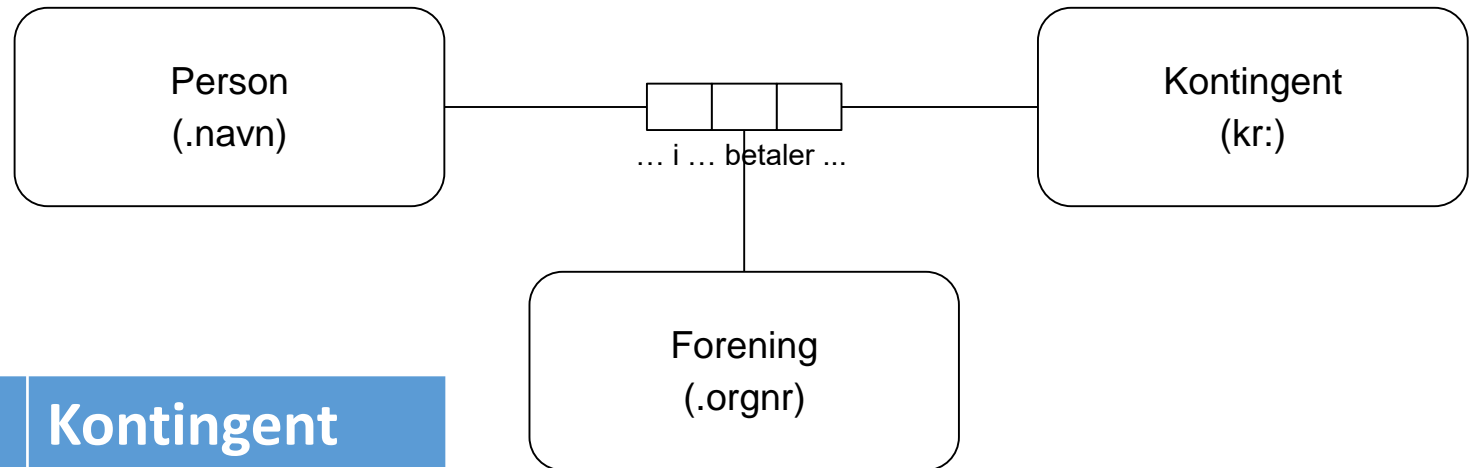
Deltagelse(personnavn) → Person(personnavn)

Deltagelse(konkurransenavn) → Konkurranse(konkurransenavn)

Deltagelse(rangeringnr) → Rangering (rangeringnr)

Ternære faktatyper

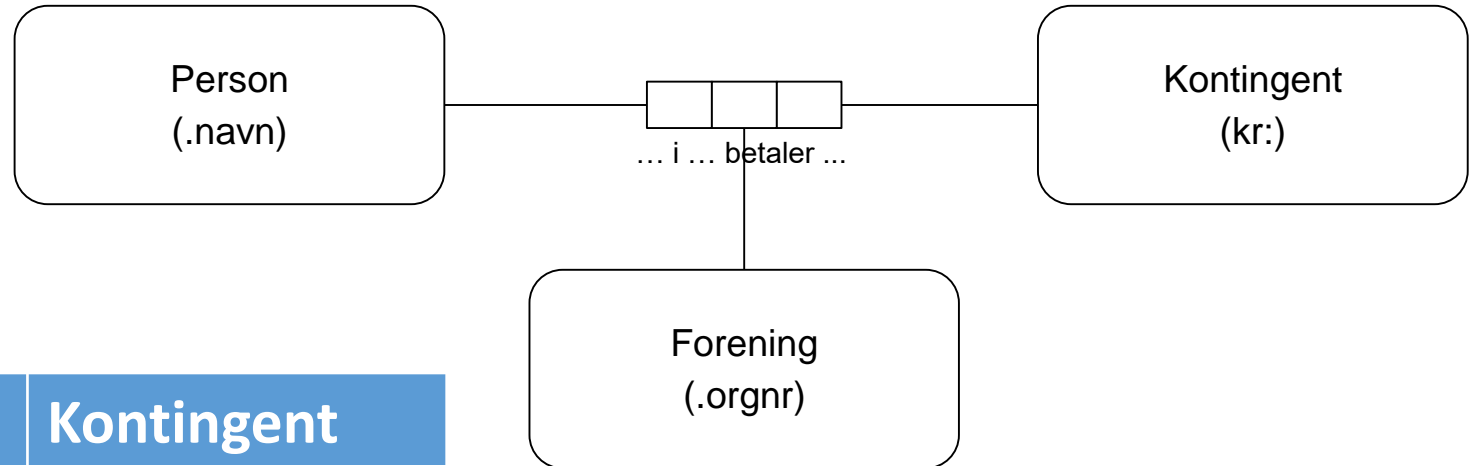
- Bruk forekomsttabeller for å teste interne entydighetskranker



Person	Forening	Kontingent
Deniz	Navet	50
Stein Michael	Navet	50
Mathias	FUI	100
Deniz	Navet	100

Ternære faktatyper

- Bruk forekomsttabeller for å teste interne entydighetskranker

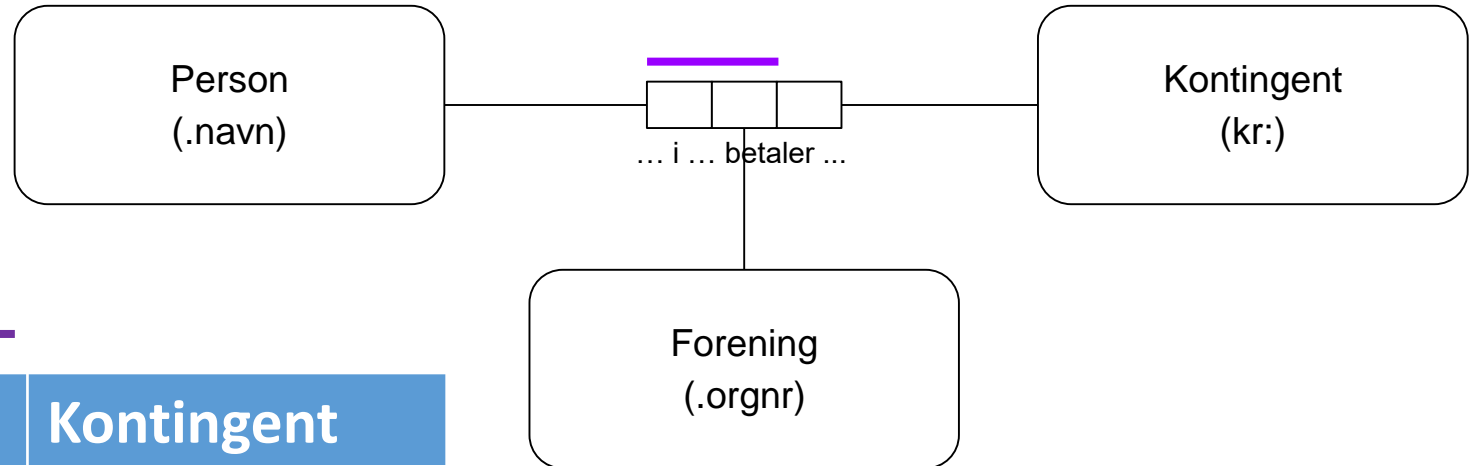


Person	Forening	Kontingent
Deniz	Navet	50
Stein Michael	Navet	50
Mathias	FUI	100
Deniz	Navet	100

Identifiser hvilke forekomster som ikke skal kunne gjentas

Ternære faktatyper

- Bruk forekomsttabeller for å teste interne entydighetskranker

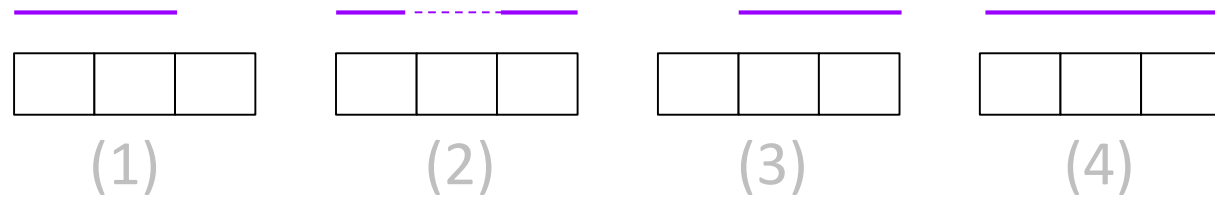


Person	Forening	Kontingent
Deniz	Navet	50
Stein Michael	Navet	50
Mathias	FUI	100
Deniz	Navet	100

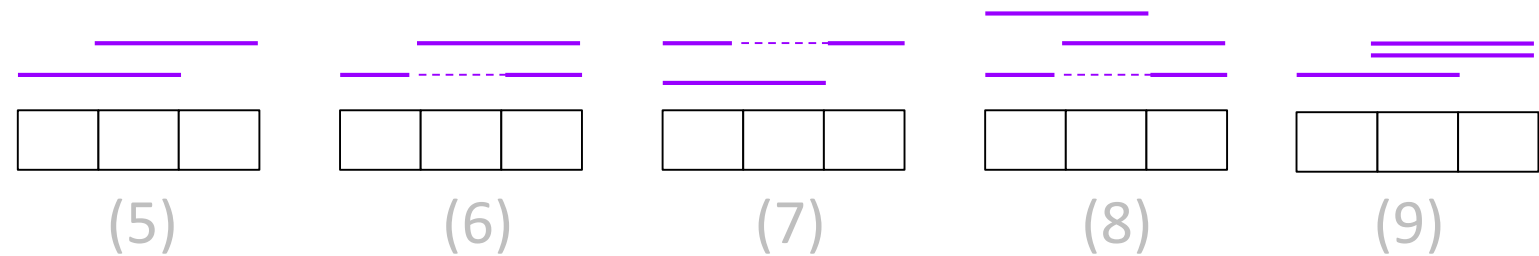
Sett skranken over rollene som skal være unike i forekomsttabellen

Eksempler på entydighetsskranger på ternære og n-ære faktatyper

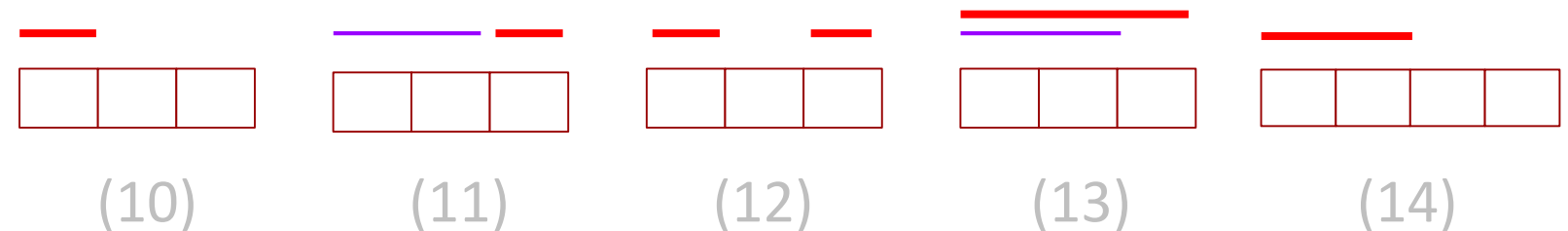
Gyldige skranger:



Gyldige kombinasjoner:



Ugyldig:

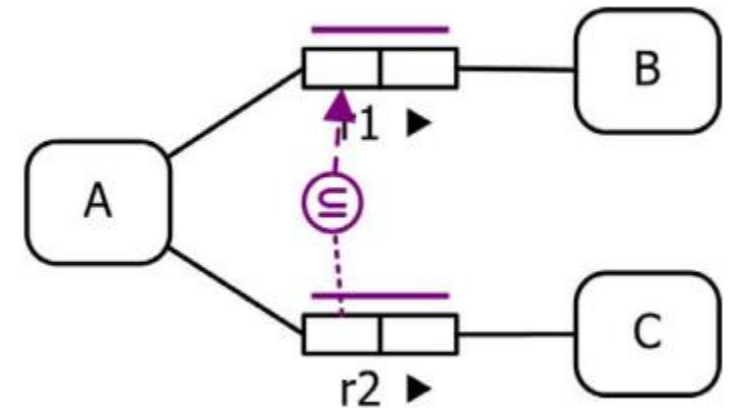
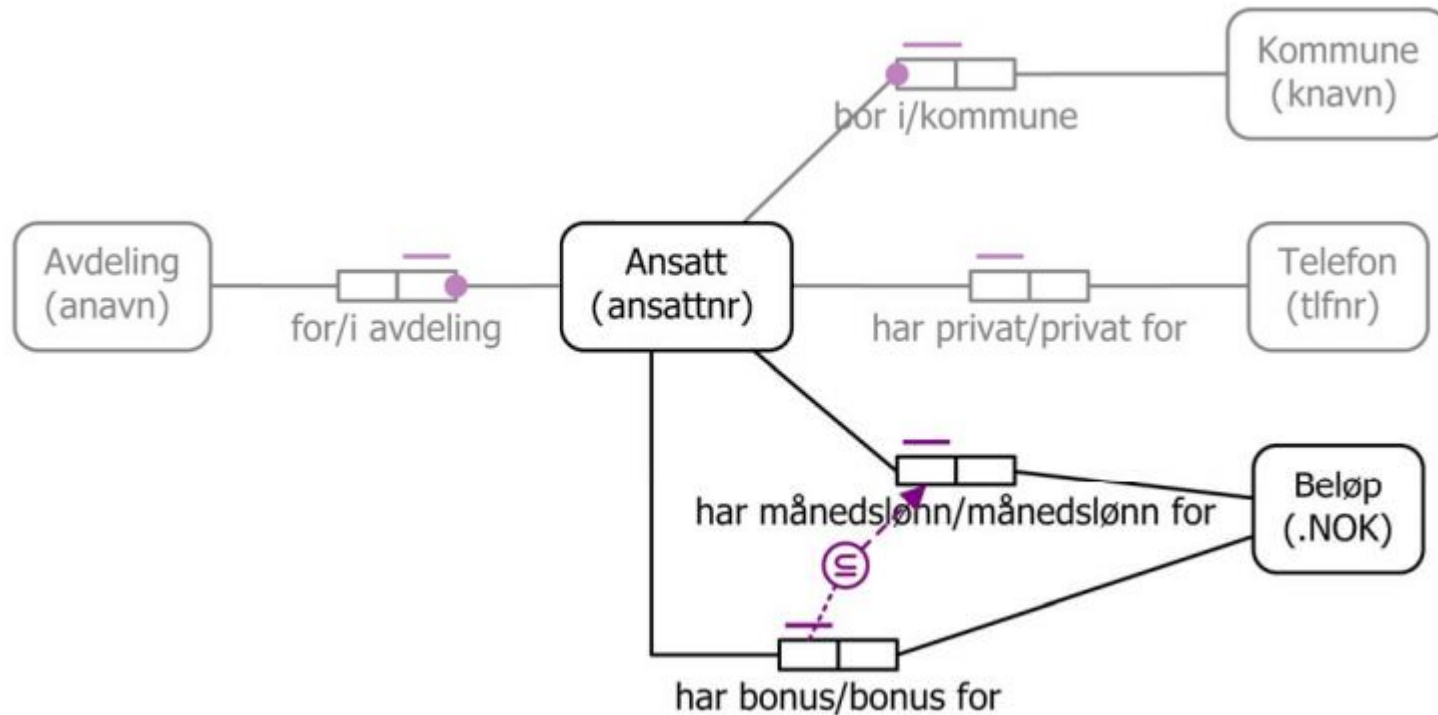


Merk: n-1-regelen (se forelesning «ORM 2» (29. aug) s. 12)

Mer avanserte skranker

Mengdeskranker

⊆ Delmengdeskranker

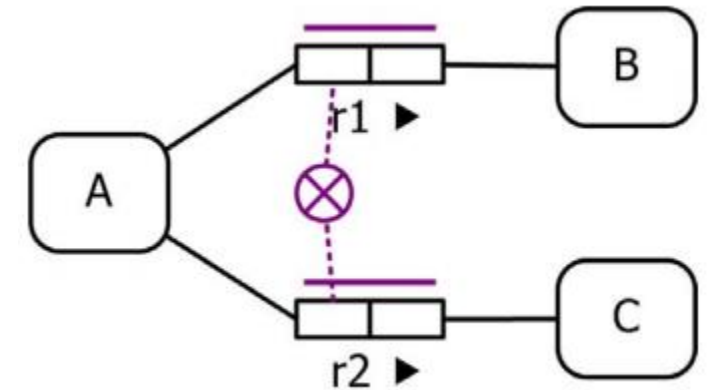
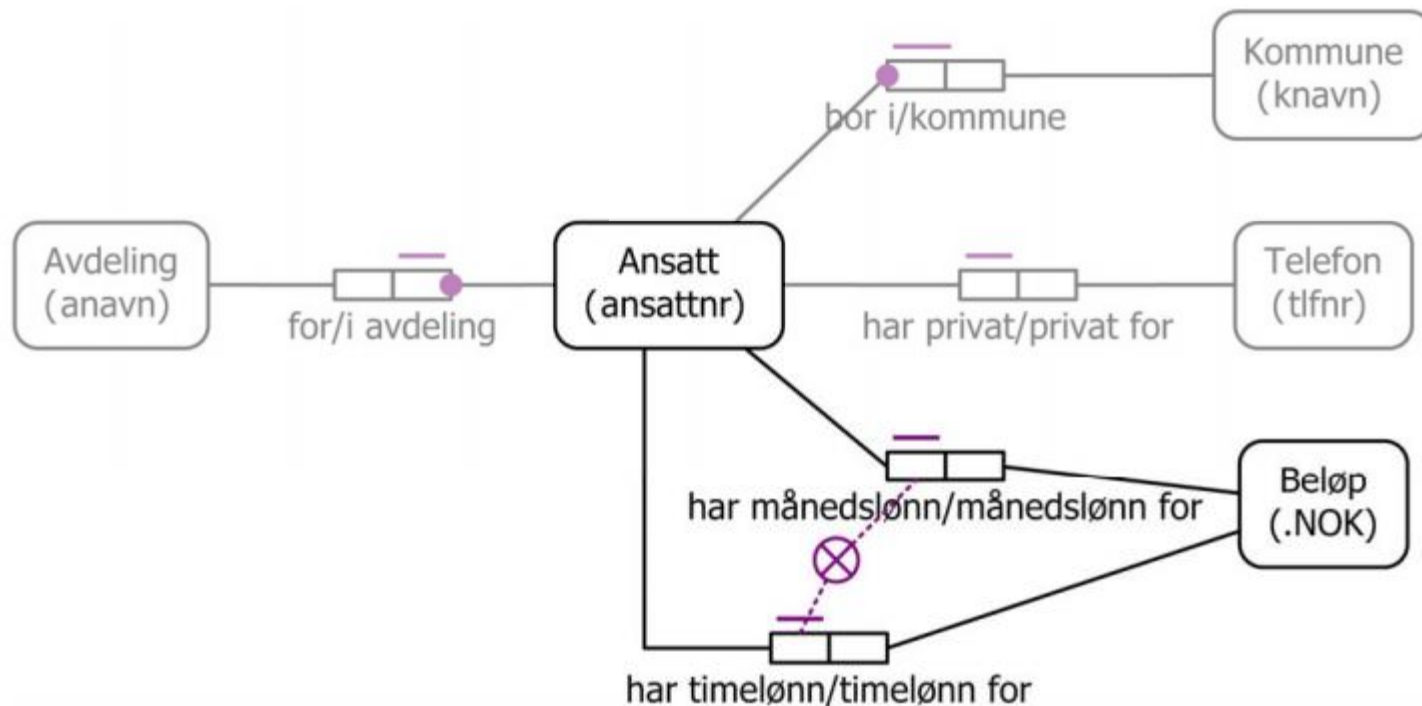


Konsekvens: Ansatt **har bonus** bare hvis Ansatt **har månedslønn**

Mer avanserte skranker

Mengdeskranker

⊗ Ulikhetskranker

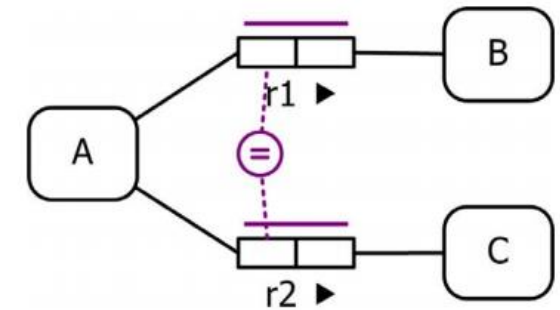
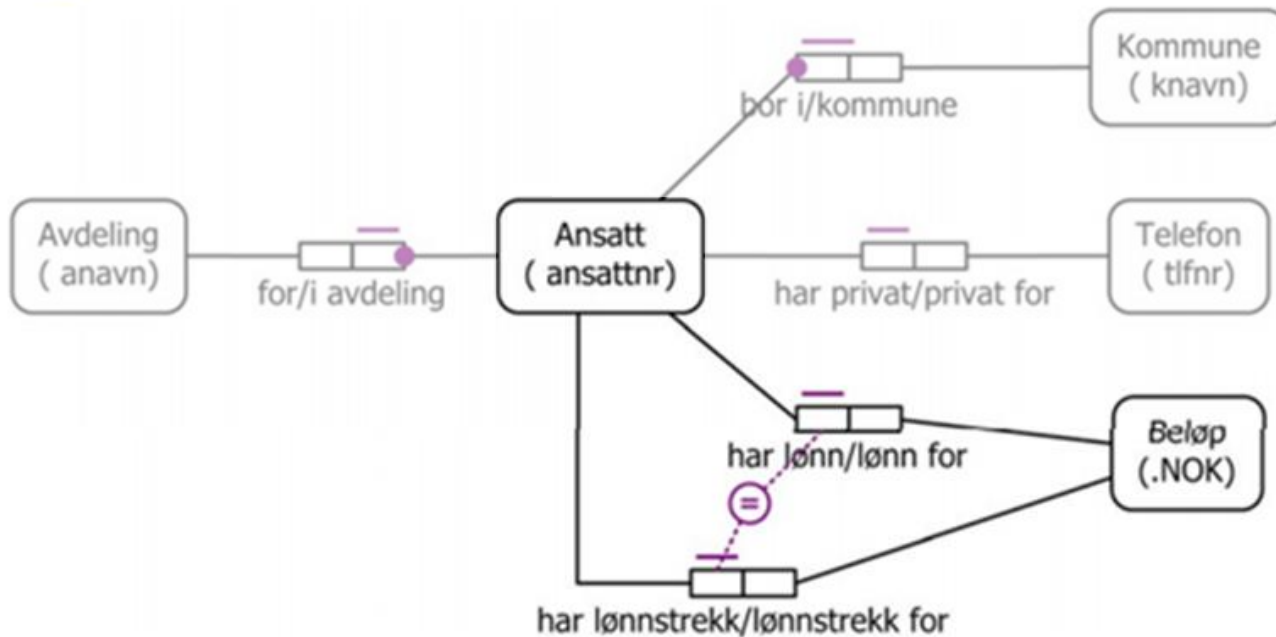


Konsekvens: En ansatt kan ikke ha både månedslønn og timelønn

Mer avanserte skranker

Mengdeskranker

⊜ Likhetskranker

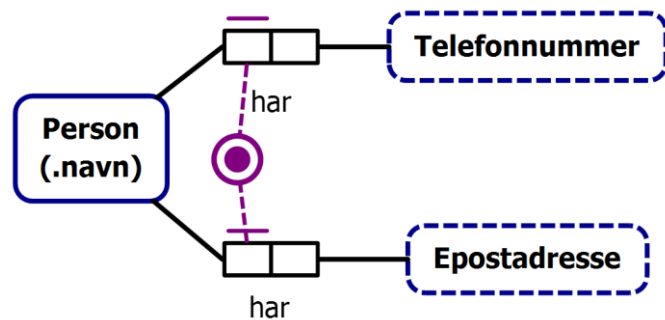


Konsekvens: En ansatt som **har lønn** må også **ha lønnstrekk** (og motsatt)

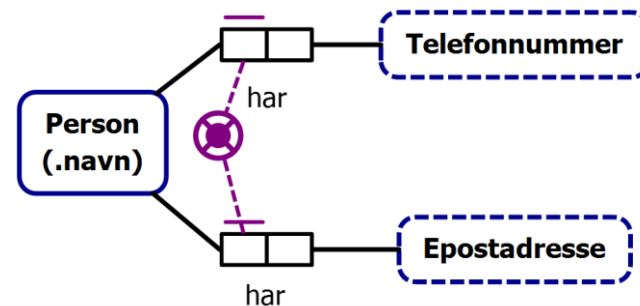
Mer avanserte skranker

Ekstern påkrevd rolle

- Gjør det samme som vanlig påkrevd rolle
- Kan oversettes til «minst en av disse»
- Kan stå over flere enn to
- Må stå over roller spilt av samme begrep for å gi mening



Ekstern påkrevd rolle

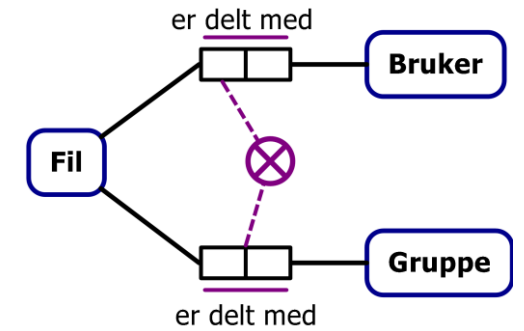
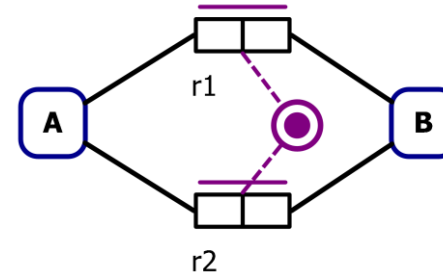
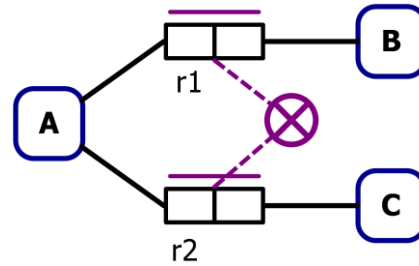
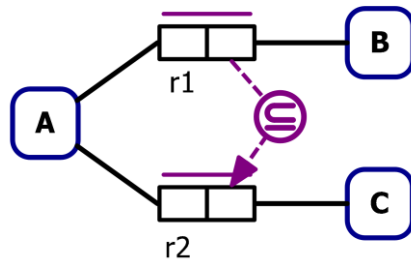


Ekstern påkrevd rolle og mengdeulikhhet
(= «exclusive or»)

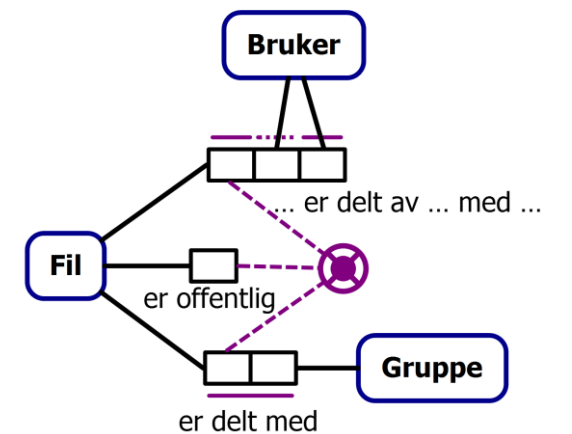
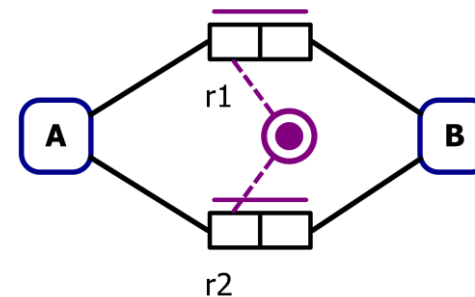
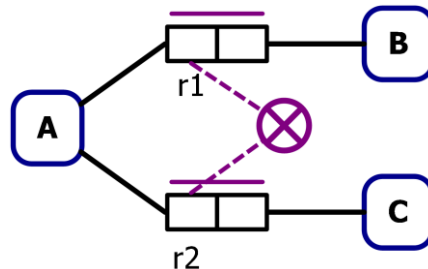
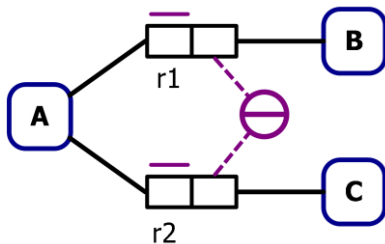
Mer avanserte skranker

Pass på hvor eksterne skranker kan settes!

Disse er **ikke lov:**



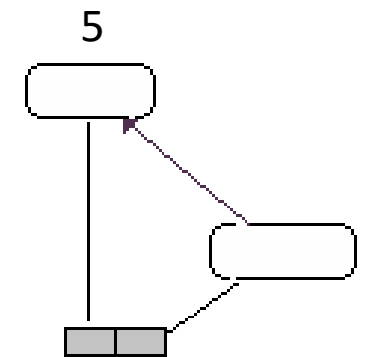
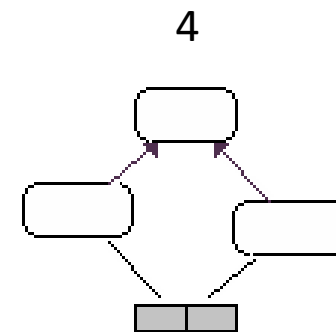
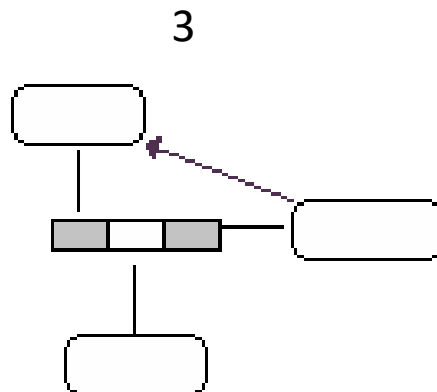
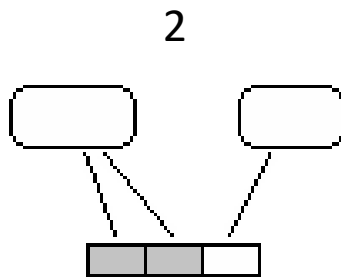
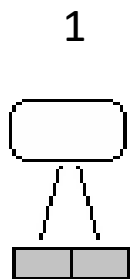
Men disse er **gyldige:**



Mer avanserte skranker









Ringskranker

- Benyttes over roller i samme setningstype spilt av samme objekt (eller underobjekt)



Mer avanserte skranker

Ringskranker

Positiv	(symbol)	Negativ (Skranke)	(Symbol)
Refleksiv		Irrefleksiv	
Transitiv		Intransitiv	
Symmetrisk		Antisymmetrisk	
		Asymmetrisk	
		Asyklisk	

Mer avanserte skranker

Ringskranker

Refleksiv og irrefleksiv



R1	R2	
A	B	
A	A	+
B	B	+

(ingen skranke)

R1	R2	
A	B	
A	A	+ -



R1	R2	
A	B	
A	A	-
B	B	-
C	C	-

Mer avanserte skranker

Ringskranker

Symmetrisk, antisymmetrisk og asymmetrisk



R1	R2	
A	B	
B	A	+
A	A	
A	A	-
A	C	
C	A	

Obs, entydighet!

(ingen skranke)

R1	R2	
A	B	
B	C	
B	A	+ -
A	A	+ -



R1	R2	
A	B	
A	A	
B	B	
B	A	-



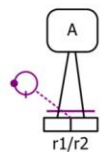
R1	R2	
A	B	
A	A	-
B	B	
B	A	-

Mer avanserte skranker

Ringskranker

Se forelesning fra 10. nov for full gjennomgang av alle ringskrankene

Irrefleksiv skranke

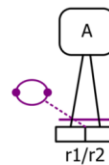


$$(x,x) \notin \text{pop}(r1,r2)$$

r1	r2
a	b
a	a
b	b

(a,a) og (b,b) skal aldri være med

Symmetriskranke

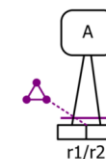


$$(x,y) \in \text{pop}(r1,r2) \Rightarrow (y,x) \in \text{pop}(r1,r2)$$

r1	r2
a	b
b	a

Hvis (a,b) er med, så skal (b,a) også være med

Transitiv skranke



$$(x,y) \in \text{pop}(r1,r2) \wedge (y,z) \in \text{pop}(r1,r2) \Rightarrow (x,z) \in \text{pop}(r1,r2)$$

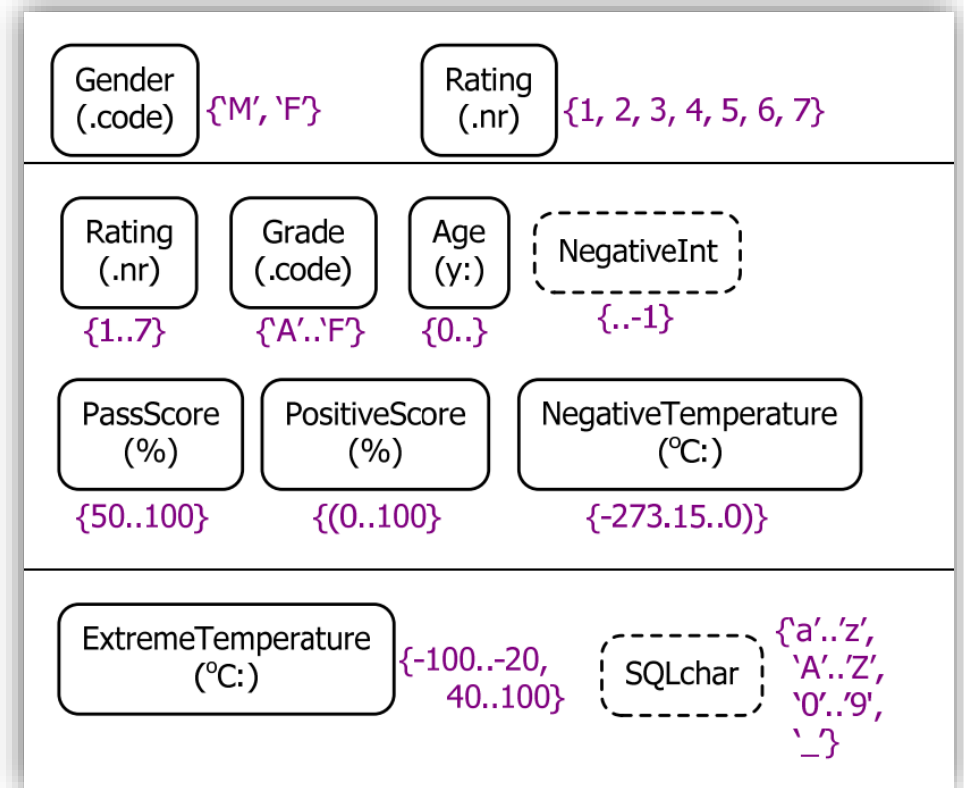
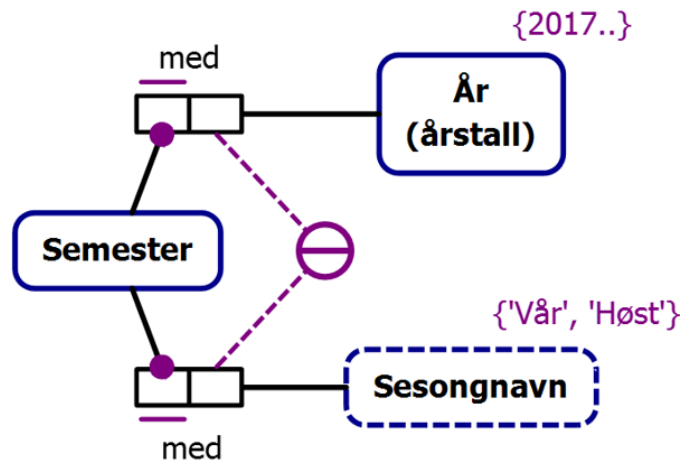
r1	r2
a	b
b	c
a	c

Hvis (a,b) og (b,c) er med, så skal (a,c) også være med

Mer avanserte skranker

Verdiskranker

- Verdiskranker
 - Definerer domenet av tillatte verdier
 - Kan settes på både verdityper og begreper

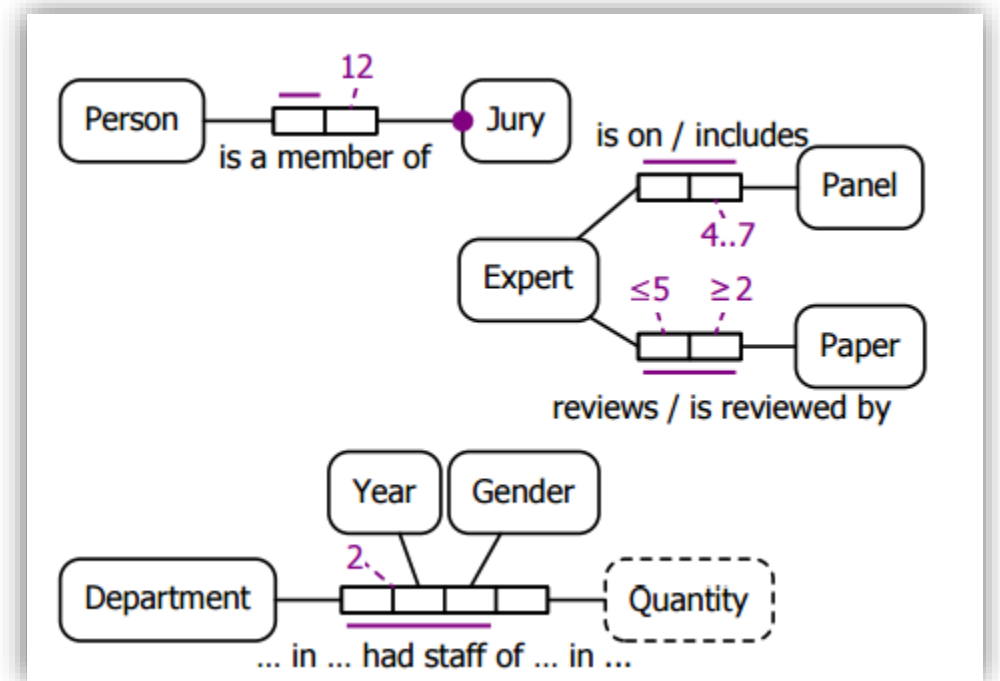


Fra ORM2 Graphical Notation, s. 3

Mer avanserte skranker

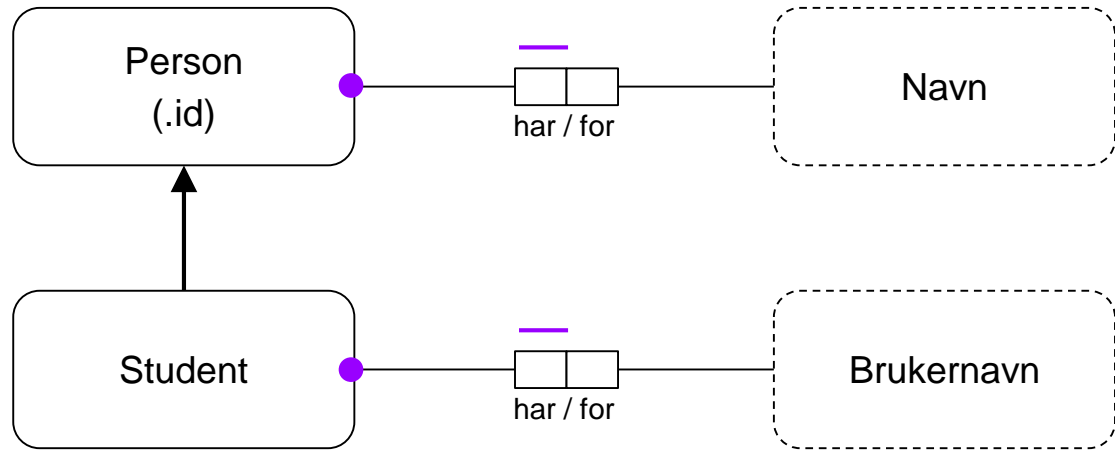
Frekvensskranker

- Bestemmer antall ganger en instans (forekomst) kan spille
- Altså antall ganger en verdi kan opptre i en forekomsttabell
 - Jury har 12 personer, en person er med i én jury
 - Panel har 4-7 eksperter, ekspert kan være i mange paneler
- Vi har også eksterne frekvensskranker



Fra ORM2 Graphical Notation, s. 5

Underbegrep



Person (personid, navn)

Student(personid, brukernavn)

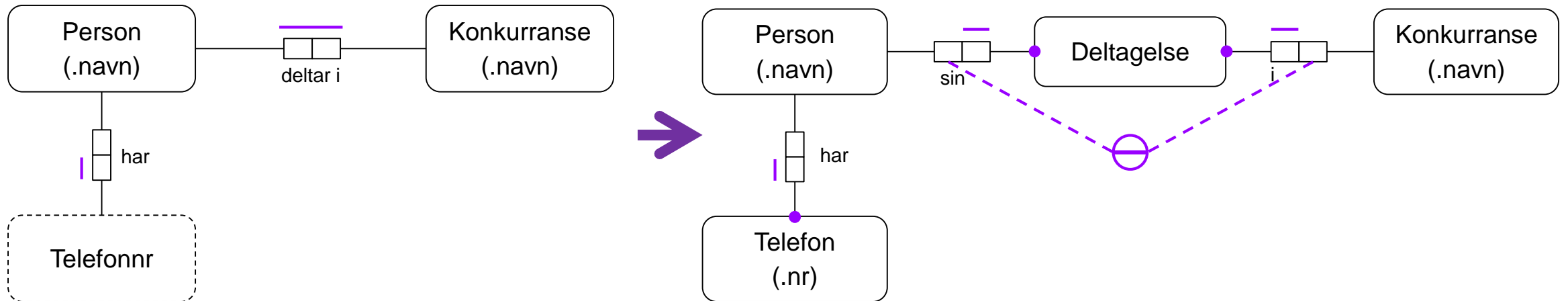
Student(personid) → Person(personid)

Underbegrepet arver primærnøkkel (eller preferert referanse) fra superbegrepet

Person		Student	
Personid	Navn	Personid	Brukernavn
0001	Odd	0001	oddtorrl
0002	Deniz	0002	denizaa
0003	Mathias	0003	mjstang
0004	Vegard	0004	vegarsaa

Realisering

- Hjelper oss med å gå fra modell til SQL-implementering
- Modell → realiserbar modell → relasjonsskjema → SQL: CREATE TABLE

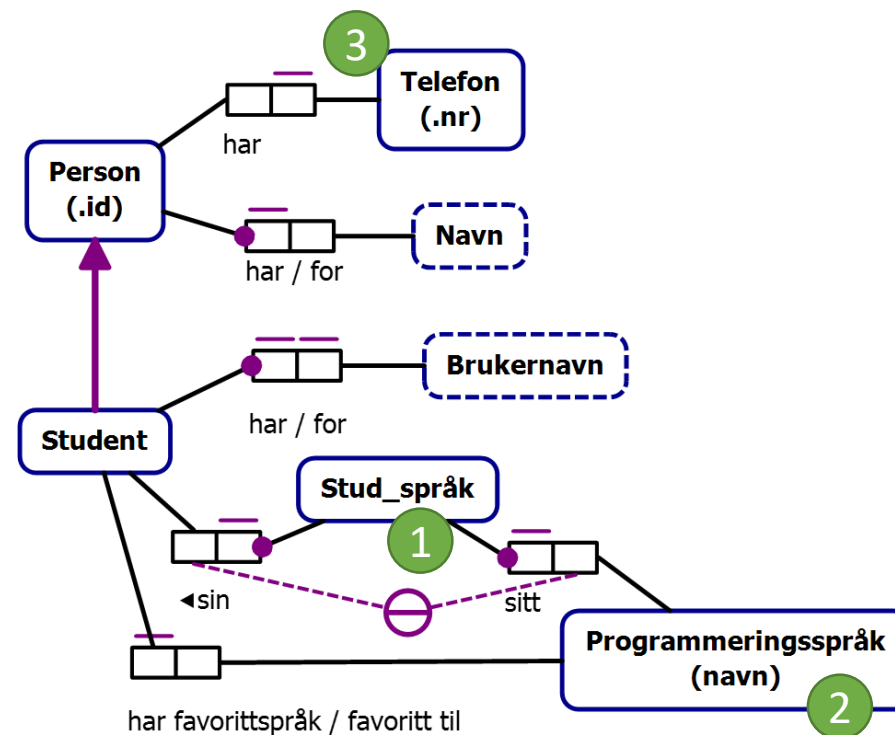
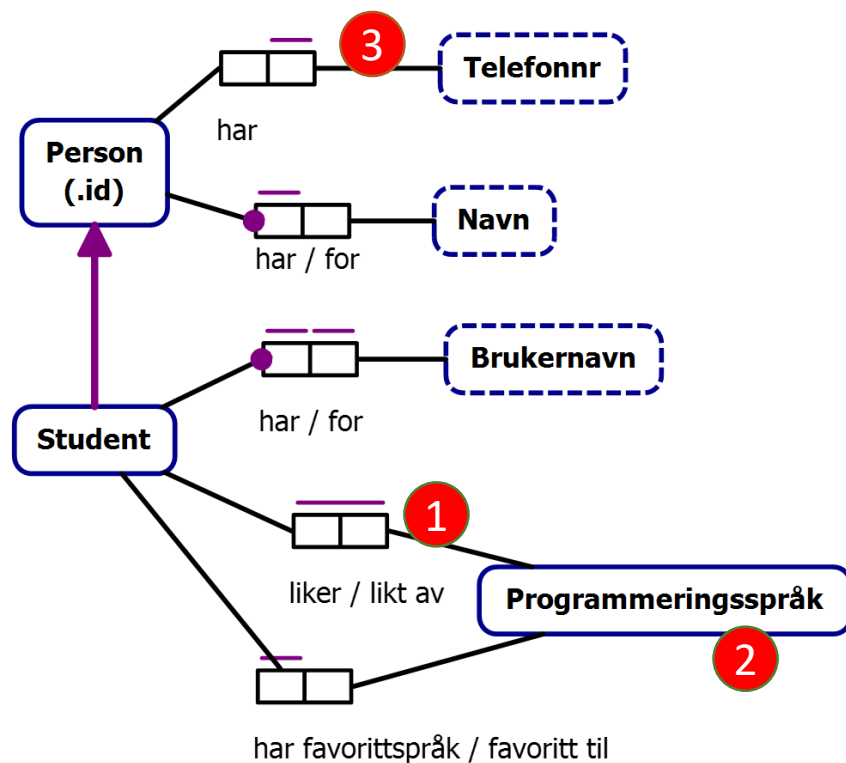


Person (personnavn)
Telefon (telefonnr, personnavn)
Deltagelse (personnavn, konkurransenavn)
Konkurranse (konkurransenavn)

```
create table person (  
    personnavn varchar primary key  
);  
...
```

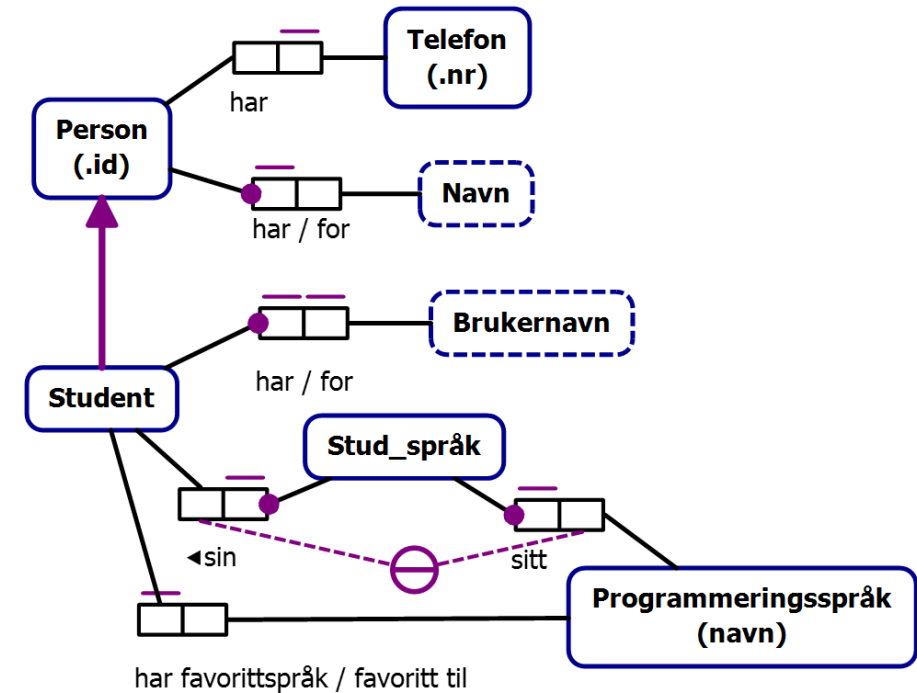
Realisering: Gjør modellen realiserbar

1. Alle lange skranker må gjøres om til begreper og gis et navn
2. Alle begreper må være refererbare
3. Diagrammet må ikke inneholde synonyme broer



Realiseringsalgoritmen

1. For hvert begrep, lag en relasjon
2. Velg referansemåte for alle begreper. Referansemåtene blir primærnøkler
3. Grupper resterende broer til sine respektive begreper. Hver bro gir ett attributt
4. Grupper resterende faktatyper. Hver faktatype blir en fremmednøkkel
5. Overfør skrankene
6. Bestem hvilke relasjoner som skal fjernes («undertrykkes»)



For detaljert gjennomgang, se plenum 26. sep og 10. okt.

Relasjonsskjema

Person (personid, navn)

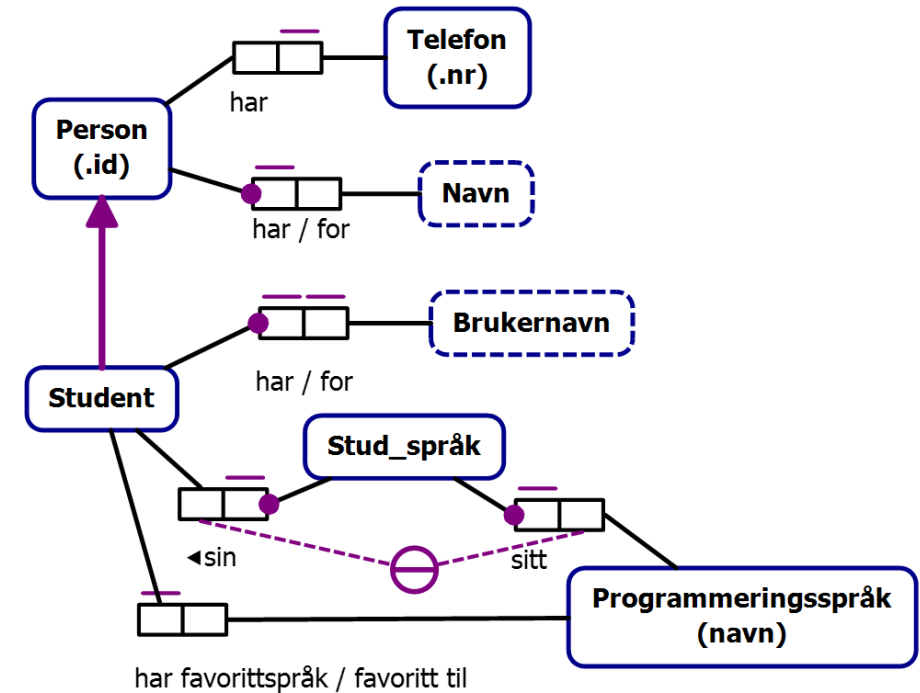
Telefon (telefonnummer, personid)

Student (personid, brukernavn, [favorittspråk])

Student_språk (personid, språk)

Programmeringsspråk (navn)

- Telefon(personid) → Person(personid)
- Student(personid) → Person(personid)
- Student(favorittspråk) → Programmeringsspråk(navn)
- Stud_språk(personid) → Student(personid)
- Stud_språk(språk) → Programmeringsspråk(navn)



Hvilke relasjoner kan undertrykkes?

Relasjonsskjema

Person (personid, navn)

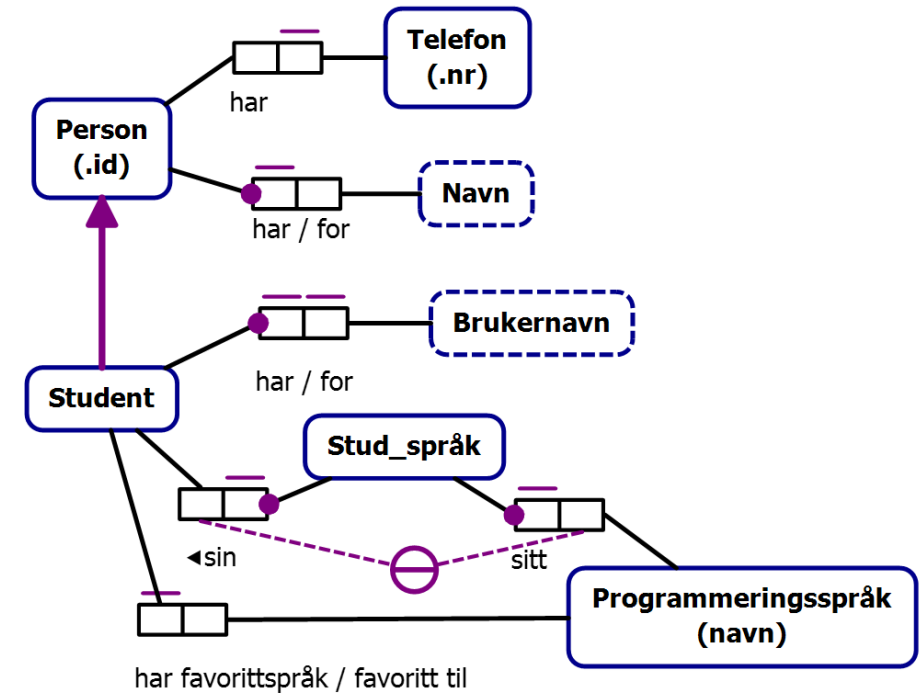
Telefon (telefonnummer, personid)

Student (personid, brukernavn, [favorittspråk])

Student_språk (personid, språk)

~~Programmeringsspråk (navn)~~

- Telefon(personid) → Person(personid)
- Student(personid) → Person(personid)
- ~~• Student(favorittspråk) → Programmeringsspråk(navn)~~
- Stud_språk(personid) → Student(personid)
- ~~• Stud_språk(språk) → Programmeringsspråk(navn)~~



Hvilke relasjoner kan undertrykkes?

Pass på dette

Dette er gjengangerfeil som burde være enkelt å unngå

- Alle setningstyper skal ha intern entydighet
- Vit forskjellen mellom begrep og verdityper
- Modellen skal være refererbar
- Relasjonsskjemaet skal gjenspeile modellen

Pass på dette

Refererbarhet

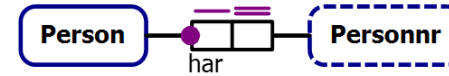
- Unik representasjon (identifisering) av hvert begrep
- Er nødvendig for å definere primærnøkkel i relasjonen
- «Hvordan skiller vi de forskjellige instansene av objektene fra hverandre?» «Hva gjør dette objektet unikt fra de andre objektene fra samme klasse?»
- Bursdag – Dato består av dag, måned år.
- Person – Unikt identifiserbart via fornavn og etternavn, eller personnr

Pass på dette

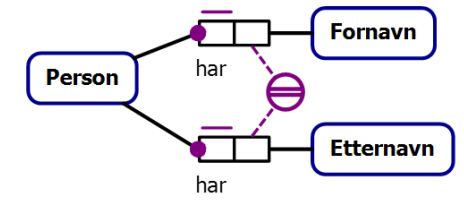
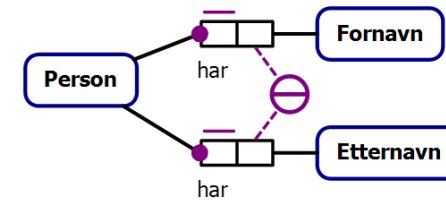
Refererbarhet

Skjer alltid på en av de fire følgende måtene:

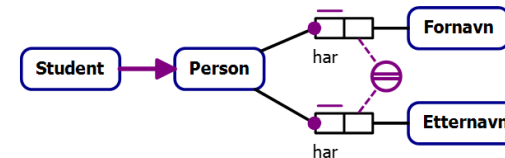
- Perfekt bro



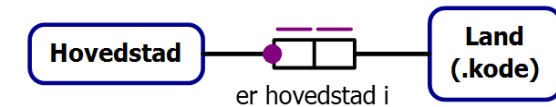
- Ekstern entydighet (og påkrevde roller)



- Via sitt overbegrep



- Én-til-én mot annet begrep med påkrevd rolle



Pass på dette

Relasjonsskjemaet skal gjenspeile modellen

- Relasjonsskjemaet inneholder alltid den samme informasjonen, eller litt mindre av informasjonen, som modellen har
- Relasjonsskjemaet har aldri annen informasjon enn modellen
- Dersom relasjonsskjemaet ser galt ut, er enten modellen realisert galt, eller så er modellen i seg selv feil
- Om du ender opp med å endre på noe når du realiserer, sørg for også å endre i modellen!

ORM-dokument på nett

«ORM – Tips og triks»

- Skrevet på bakgrunn av alle gjengangerfeil på obliger de siste årene
- Tar ikke for seg alt – da må man lese boka og se i foiler
- Men – tar for seg mye av det grunnleggende og de vanligste forvirringene

Link:

<http://www.uio.no/studier/emner/matnat/ifi/INF1300/h16/forelesningsmateriale/tips.html>