

**UiO • Institutt for informatikk**

Det matematisk-naturvitenskapelige fakultet

**INF1300 - Introduksjon til databaser**

**ORM I**



## Dagens tema:

- Grunnbegrepene i ORM
- Sammenheng mellom ORM og naturlig språk
- Elementære setninger (fakta)
- Faktatyper og broer
- Entydighetsskranker og påkrevde roller

# ORM – Object Role Modelling

Tre viktige prinsipper:

1. **Ogdens trekant:** Sammenhengen mellom virkelighet og modell
2. **Naturlig språk:** Menneskespråk/naturlig språk er det mest fundamentale kommunikasjonsverktøyet vi har. Modellen må kunne uttrykkes i naturlig språk for å sikre at den kan forstås fullt ut av informerte brukere (de som kjenner virksomhets-området)
3. **100%-prinsippet:** Vi kan lage en nøyaktig nok modell av virkeligheten ved hjelp av naturlig språk

Universe of Discourse

UoD

domene

interesseområde

In the formal sciences, the domain of discourse, also called the universe of discourse, universal set, or simply universe, is the set of entities over which certain variables of interest in some formal treatment may range.

- For å lage et begrepsmessig skjema for UoD må vi velge hvilke *begreper* skjemaet skal inneholde

- For å lage et begrepsmessig skjema for UoD må vi velge hvilke *begreper* skjemaet skal inneholde

Fra kinobillettdomenet:

Kino  
Film  
Rad  
Forestilling  
Dato  
Billett  
Sete  
...

# Representasjon

- I tillegg må vi for hvert begrep bestemme oss for hvordan vi skal lagre informasjon om forekomster av dette begrepet

# Representasjon

- I tillegg må vi for hvert begrep bestemme oss for hvordan vi skal lagre informasjon om forekomster av dette begrepet

Fra kinobillettdomenet:

Kino – kinonavn f.eks. “Colloseum”

Film – filmnavn f.eks. '七人の侍'

Rad – et naturlig tall mellom 1 og 54

Forestilling – tidspunkt og navn på film og kino

Dato – dag, måned, år

Billett – billettnavn

Sete - et naturlig tall mellom 1 og 29

...



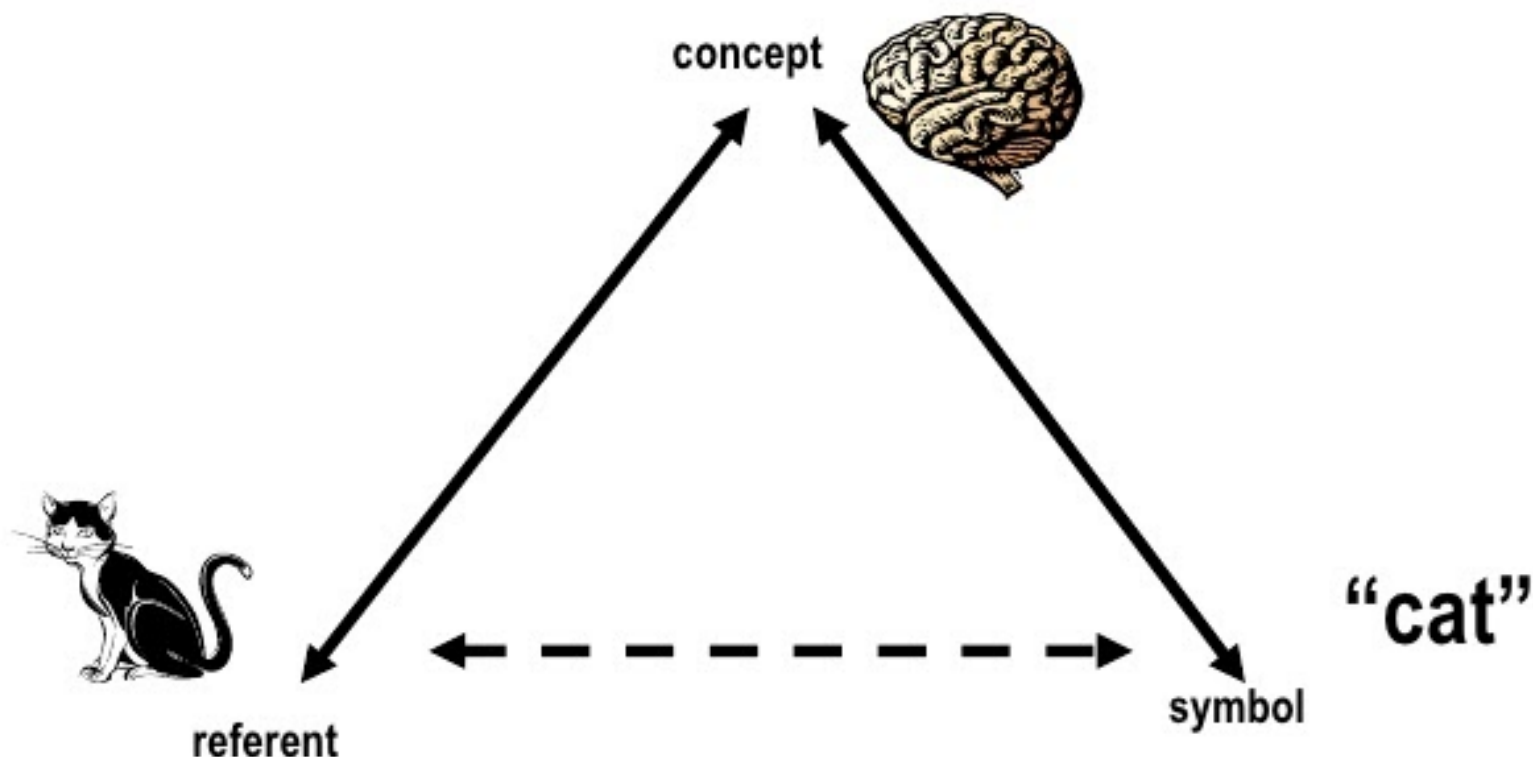
# Representasjon

- Vi kan ikke lagre kyr i databasen, så vi (eller bonden som skal registrere melkeproduksjon i en database) må finne en måte å lagre informasjon som identifiserer hver enkelt ku

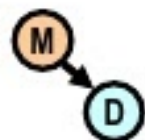
# Representasjon

- En slik identifikasjonsmåte kalles en **verditype** eller **representasjonstype** for begrepet

# Semantic Triangle



- Symbols can only link to referents through concepts
- You can not link directly from a symbol to a referent



# Ogdens trekant

Begrep

Representasjon

Fenomen i UoD

# Ogdens trekant

Begrep *Ku*

MT NO 12345678 2001

Representasjon



Fenomen i UoD

# Elementære setninger

- En setning som ikke kan deles opp uten å miste meningsinnhold, kalles *elementær*
- Eksempel:
  - Per liker is og brus

Denne setningen er *ikke* elementær fordi den kan erstattes av de to elementære setningene

- Per liker brus
- Per liker is

# Øvelse

## **Hvilke av disse setningene er elementære?**

- Dag er født 27. august 1996
- Hans og Grete er søsken
- DK41050 er en hvit Toyota Yaris
- Anne fikk B i INF1010
- Liv ble ansatt som rådgiver 1.8.2003

# Setninger og Ogdens trekant - 1

- Se på setningen «Hanne tar IN1000»
- Det er mye informasjon som er underforstått:
  - Hanne er navn på en student
  - IN1000 er en emnekode
- Det vi mener er at
  - studenten med navn Hanne
  - tar emnet med emnekode IN1000



# Setninger og Ogdens trekant - 1

- Se på setningen «Hanne tar IN1000»
- Det er mye informasjon som er underforstått:
  - Hanne er navn på en student
  - IN1000 er en emnekode
- Det vi mener er at
  - studenten med navn Hanne følger undervisning og er tatt opp i emnet med emnekode IN1000

# Setninger og Ogdens trekant - 2

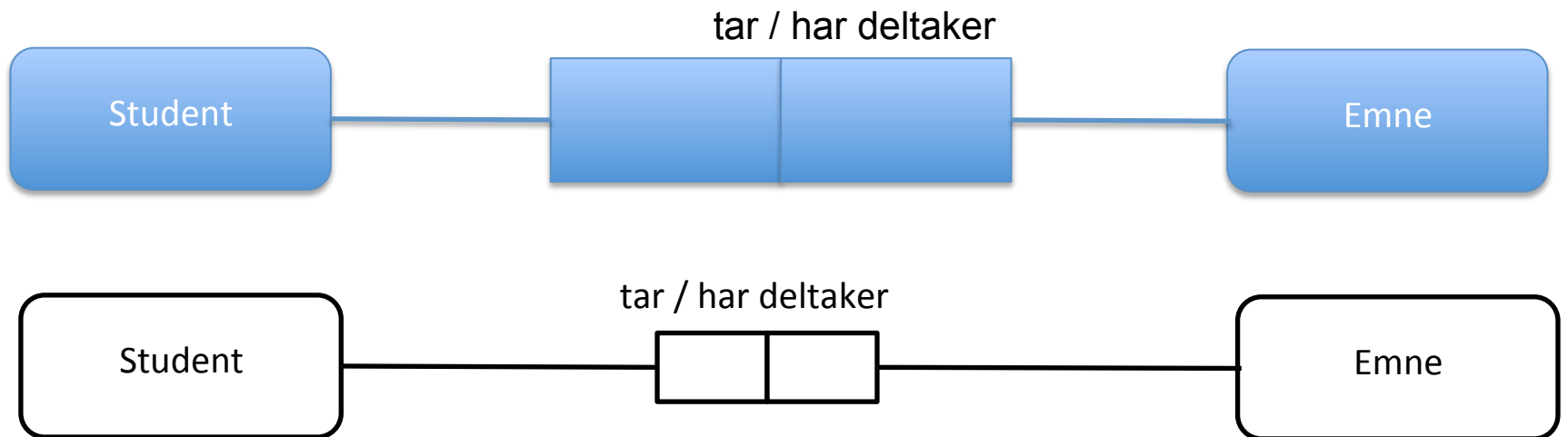
- La oss se nærmere på setningen:  
«Studenten med navn Hanne tar emnet med emnekode IN1000»
- «student» og «emne» er **begreper**
- «navn» og «emnekode» er deres **verdityper** (**representasjonstyper**)
- «Hanne» og «IN1000» er **forekomster** (data)

# Setninger og faktatyper

- De to setningene under har samme meningsinnhold:
  - «Studenten med navn Hanne tar emnet med emnekode IN1000»
  - «Emnet med emnekode IN1000 har som deltaker studenten med navn Hanne»
- Vi kan forme liknende **fakta** ved å bytte ut forekomstene:
  - «Studenten med navn Henrik tar emnet med emnekode IN1020» (eller: «Emnet med emnekode IN1020 har som deltaker studenten med navn Henrik»)
- I ORM tegner vi denne **faktatypen** slik:

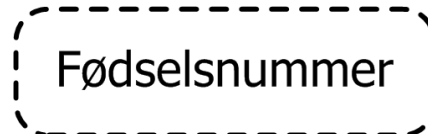


# faktatype



# Ogdens trekant og ORM

- I ORM tegner vi **begreper** og **verdityper** (**representasjonstyper**) som henholdsvis **heltrukne** og *stiplede* rektangler/sirkler/elipser...
- Eksempel:  
Begrepet **Person** og verditypen **Fødselsnummer** tegnes slik:



# Ogdens trekant og ORM

- I ORM brukes ordet **objekttype** som en felles betegnelse på begreper og verdityper

Person

Fødselsnummer

# Setningers aritet

- Den elementære setningen:  
*«Studenten med navn Anne fikk i emnet med emnekode INF1010 resultatet karakteren B»*  
inneholder tre begreper:  
«student», «emne» og «resultat»
- Antall begreper i en setning kalles setningens **aritet**
- Vårt eksempel har aritet 3

# Setningers aritet

- Setninger med aritet 1 kaller vi unære
- Setninger med aritet 2 kaller vi binære
- Setninger med aritet 3 kaller vi ternære
- Man kan konstruere elementære setninger med vilkårlig høj aritet
- Elementære setninger med aritet  $> 3$  er sjældnere, så vi gir dem ikke egne navn (n-ære setninger)



# Setning med aritet 4

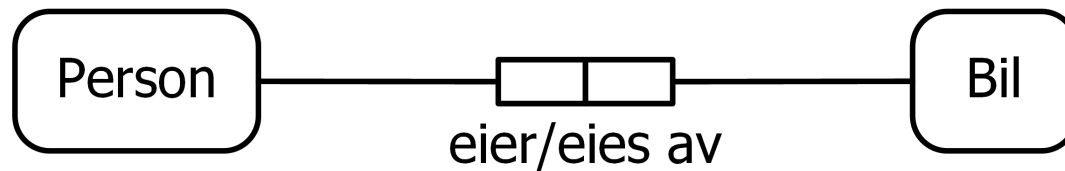
- Eksempel på en elementær setning med aritet 4:
  - 25.8.2008 lånte Per NOK 200 000 av Pål

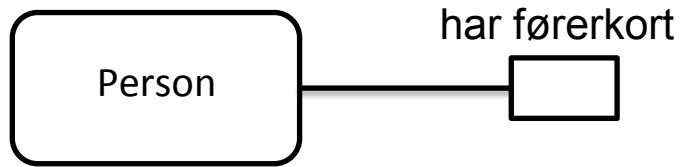
# Roller og faktatyper

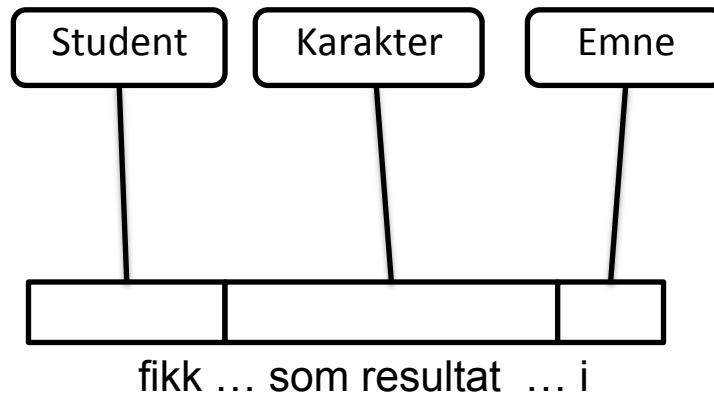
- Se på setningsparet:  
*«En person med navn Siri eier en bil med registreringsnummer DL12345»* og  
*«En bil med registreringsnummer DL12345 eies av en person med navn Siri»*
- Her kan vi åpenbart få lignende fakta ved å bytte ut forekomsten «Siri» med et annet navn og/eller forekomsten «DL12345» med et annet registreringsnummer
- Vi sier at begrepet «Person» **spiller rollen** «eier» overfor begrepet «Bil», og at «Bil» spiller rollen «eies av» overfor «Person»
- Et slikt rollepar mellom to begreper kalles en (**binær**) **faktatype**

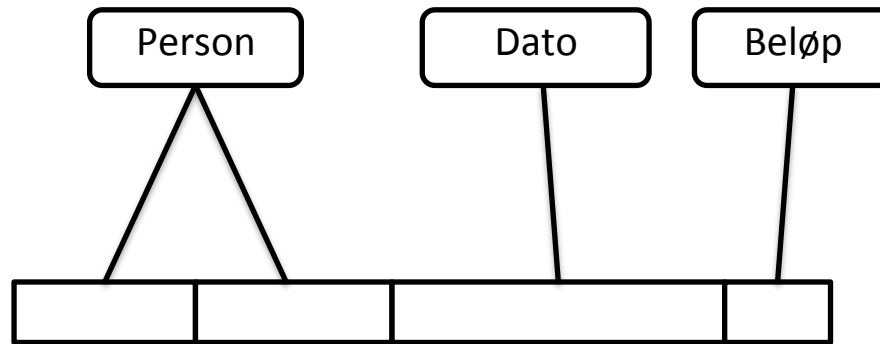
# Faktatyper i ORM

- Vårt eksempel på en faktatype mellom begrepene Person og Bil tegner vi slik:









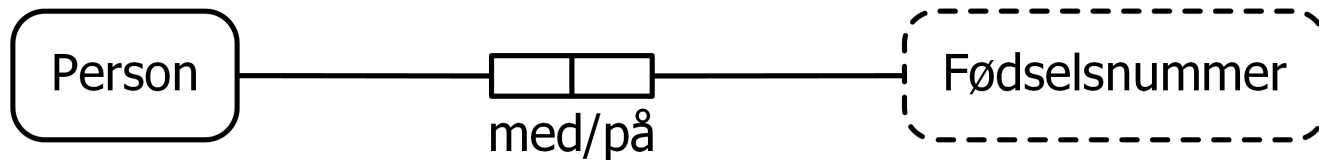
er låntaker ... er långiver ... er dato for lån ... er beløp for lån

# Broer

En **bro** er en forbindelse mellom et **begrep** og en **verditype** (representasjonstype)

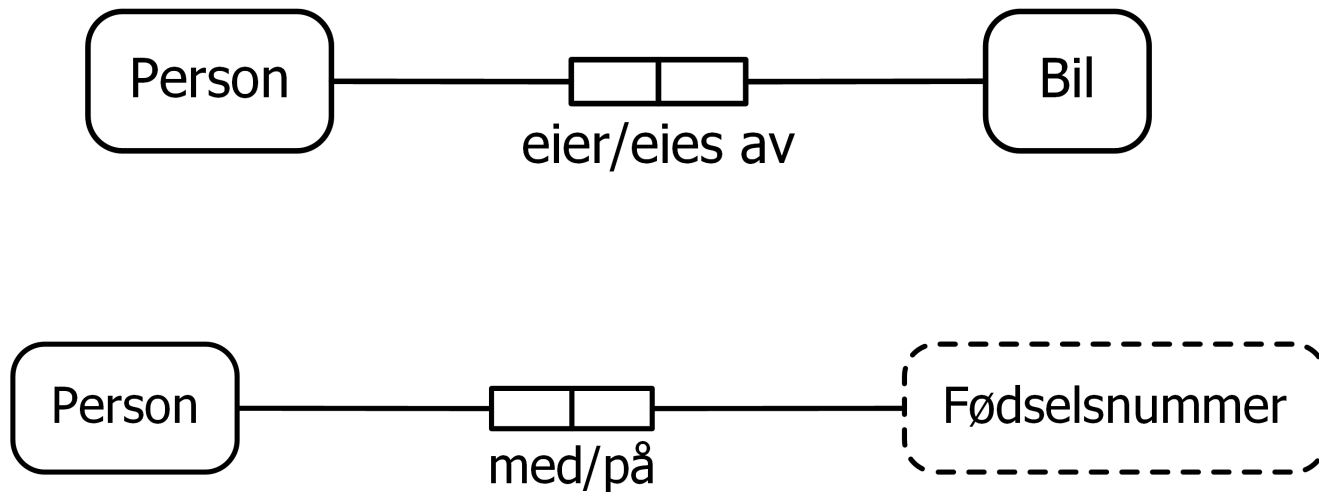
Eksempel:

Slik tegner vi broen mellom Person og Fødselsnummer i ORM:



# Setningstyper

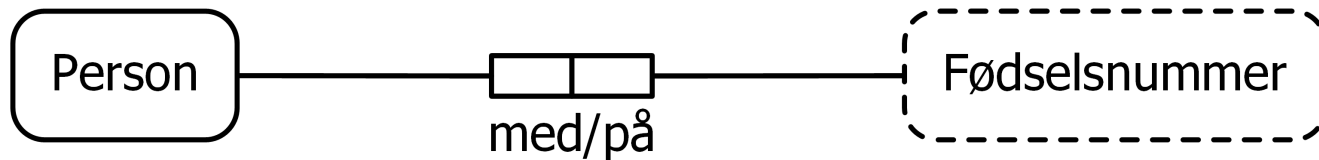
- Ordet **setningstype** er en felles betegnelse på faktatyper og broer





# Setningstyper

- Ordet **setningstype** er en felles betegnelse på faktatyper og broer
- Broer er alltid *binære* – de forbinder ett begrep og én verditype



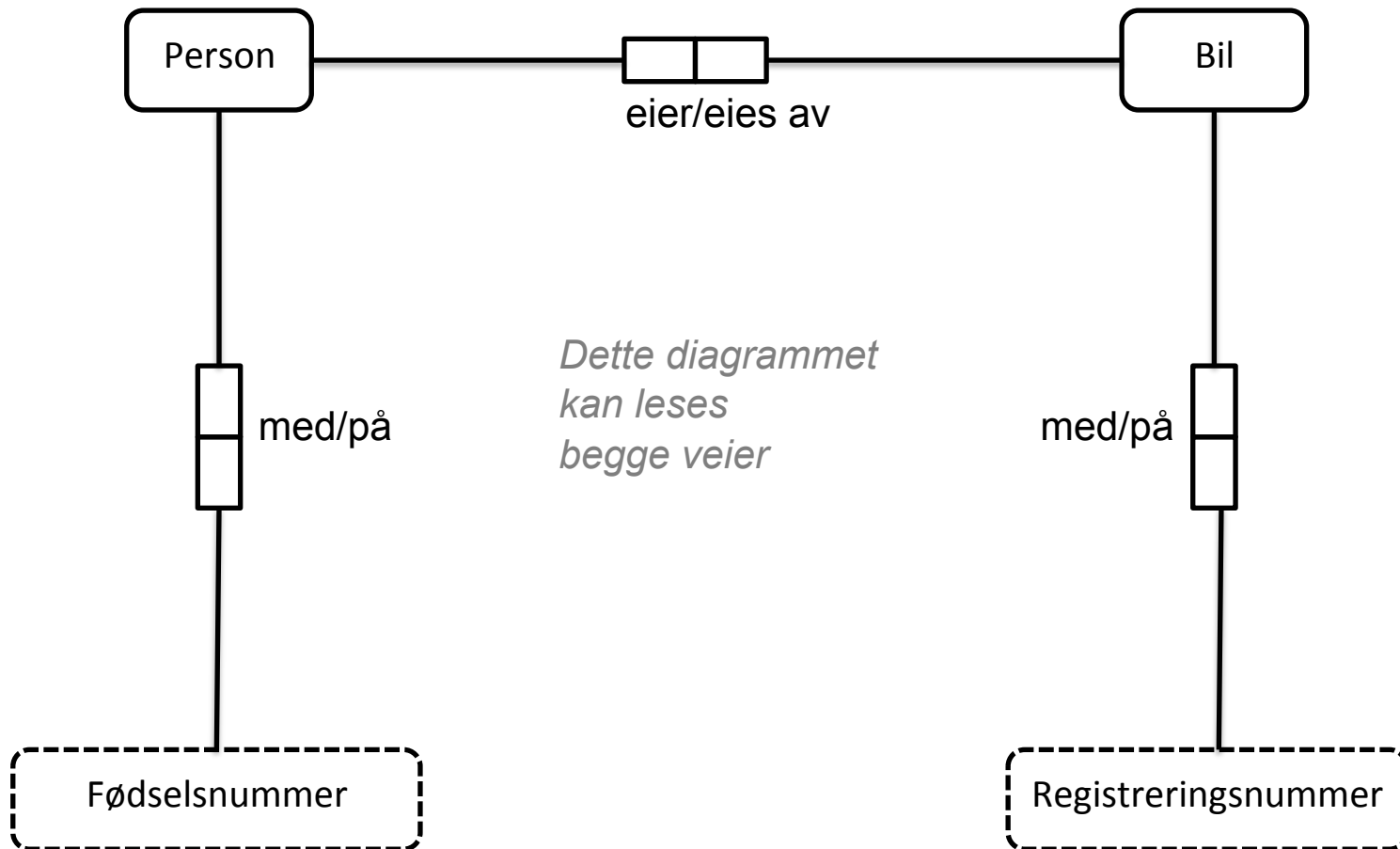
# Setningstyper

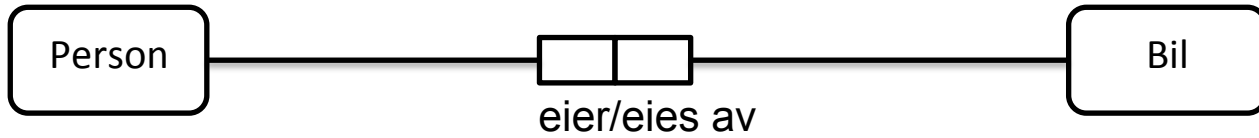
- Ordet **setningstype** er en felles betegnelse på faktatyper og broer
- Broer er alltid *binære* – de forbinder ett begrep og én verditype
- Faktatyper kan ha et vilkårlig antall roller (aritet)
  - hver rolle skal være knyttet til nøyaktig ett begrep
  - et begrep kan spille flere roller i samme faktatype (se f.eks. låneeksempelet ovenfor, der begrepet Person spiller rollene «er låntaker» og «er långiver»)

# Rollenavn

- I faktatyper bør alle rollenavn inneholde et verb (hvis ikke, er det grunn til å tro at rollenavnet er dårlig valgt)
- I broer er det vanlig med preposisjoner som rollenavn. De to vanligste rolleparene er
  - med/for
  - med/på

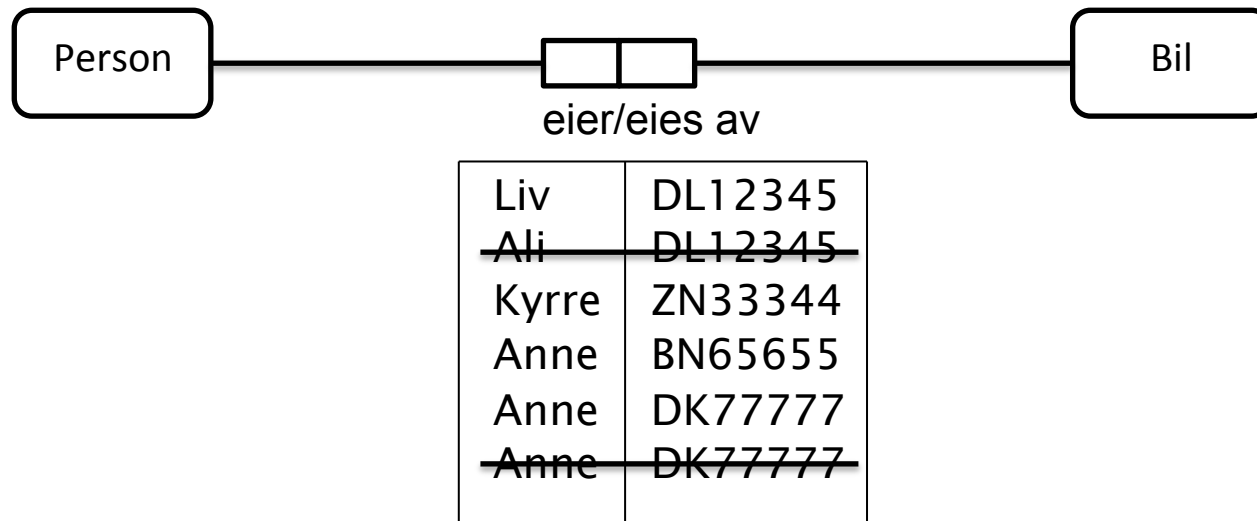
# Fakta – setningers dype struktur



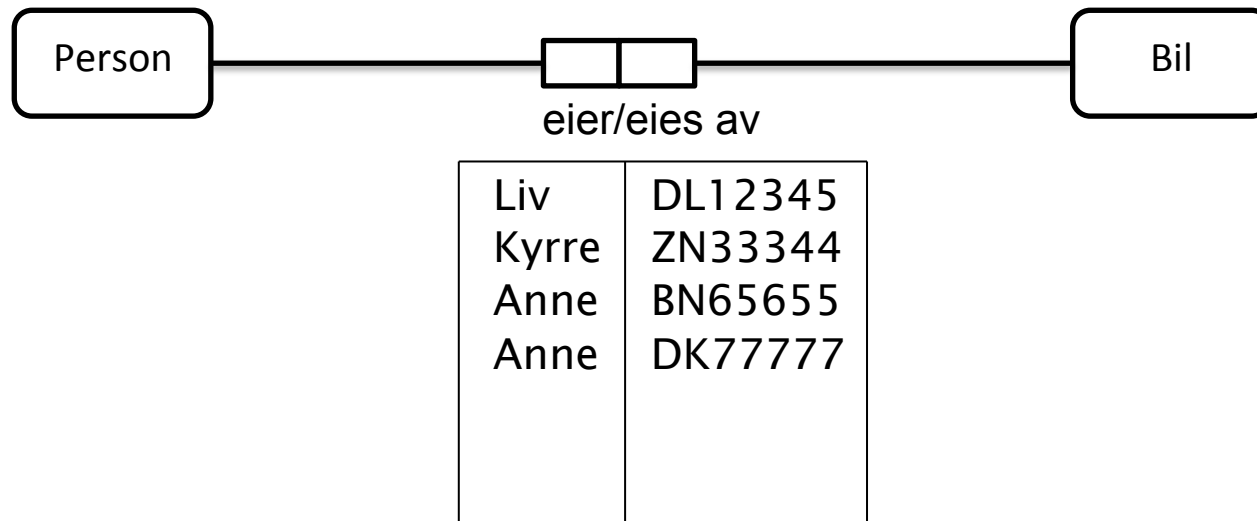


*forekomsttabell*

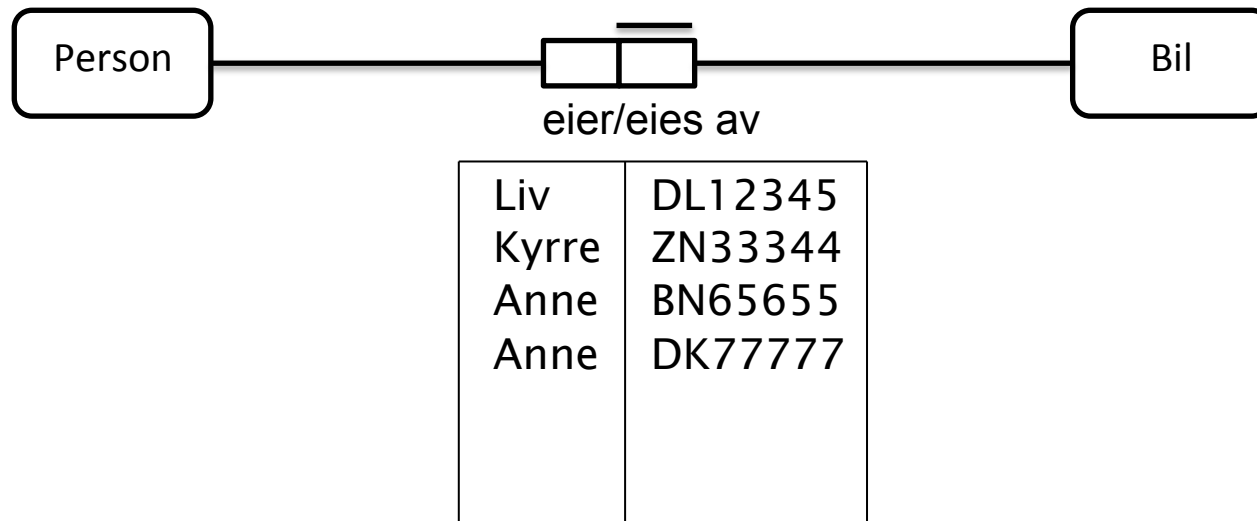
Liv	DL12345
Ali	DL12345
Kyrre	ZN33344
Anne	BN65655
Anne	DK77777
Anne	DK77777



Anta at vi har en forretningsregel som sier at en bil bare kan ha én eier (mens en person kan eie flere biler)

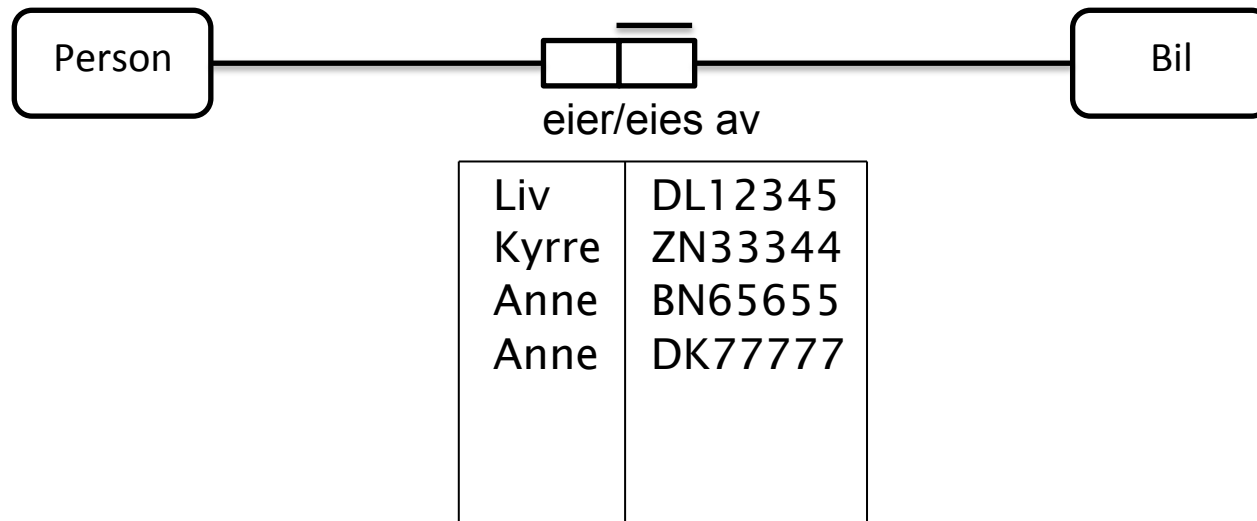


Anta at vi har en forretningsregel som sier at en bil bare kan ha én eier (mens en person kan eie flere biler)

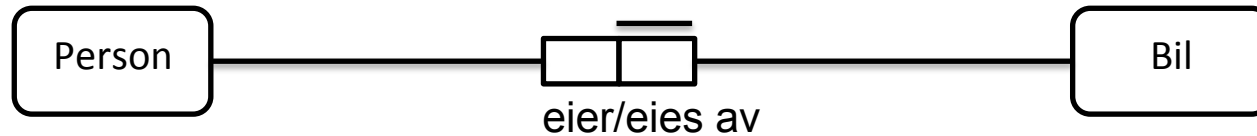


- I ORM-diagrammet plasserer vi *en strek* over rollen hvor samme forekomst ikke kan gjentas i forekomsttabellen





- I ORM-diagrammet plasserer vi en strek over rollen hvor samme forekomst ikke kan gjentas i forekomsttabellen
- Streken kalles en **entydighetsskranke**
- Entydighetsskranke kan gå over flere roller – da er det forekomstkombinasjonen i rollene som ikke kan gjentas



- I ORM-diagrammet plasserer vi en strek over rollen hvor samme forekomst ikke kan gjentas i forekomsttabellen
- Streken kalles en **entydighetskranke**
- Entydighetsskranger kan gå over flere roller – da er det forekomstkombinasjonen i rollene som ikke kan gjentas

interesseområde (UoD)

**begrep**

**verditype**/representasjonstype

elementær setning

forekomst

fakta

faktatype

objekttype

en setnings aritet

**rolle**

bro

setningstype

**entydighetsskranke**

forekomsttabell

påkrevd rolle

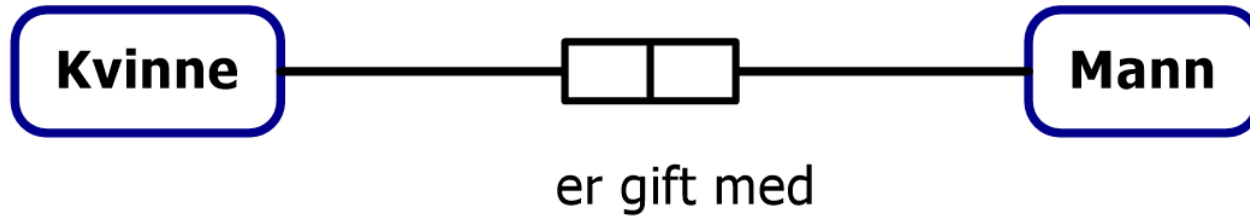
perfekt bro

skranke

**3** grense (l,1), hindring  
*bryte alle skranker / sette en  
skranke for noe*

# Ekteskap - 1

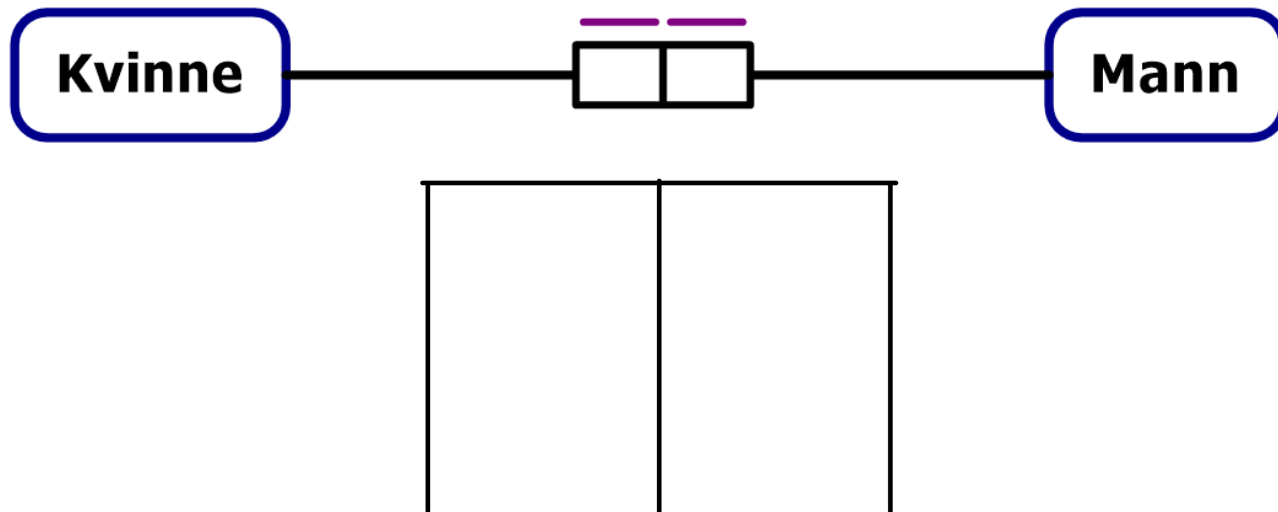
- Vi skal nå se på en faktatype mellom en kvinne og en mann kalt *ekteskap*
- Uten entydighetsskranke(r) ser modellen slik ut:



- Hvilke(n) entydighetsskranke(r) skal vi ha?
- Lag forekomststabell og sett på entydighetsskranke(r)!

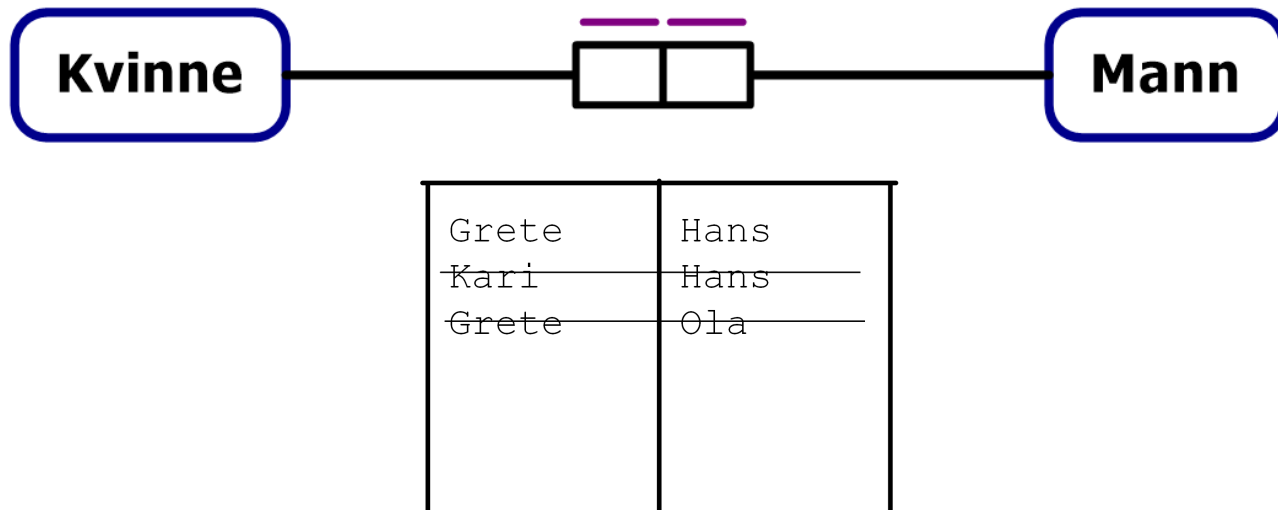
# Ekteskap - 2

- De fleste har vel foreslått modellen nedenfor
- Lag forekomststabell



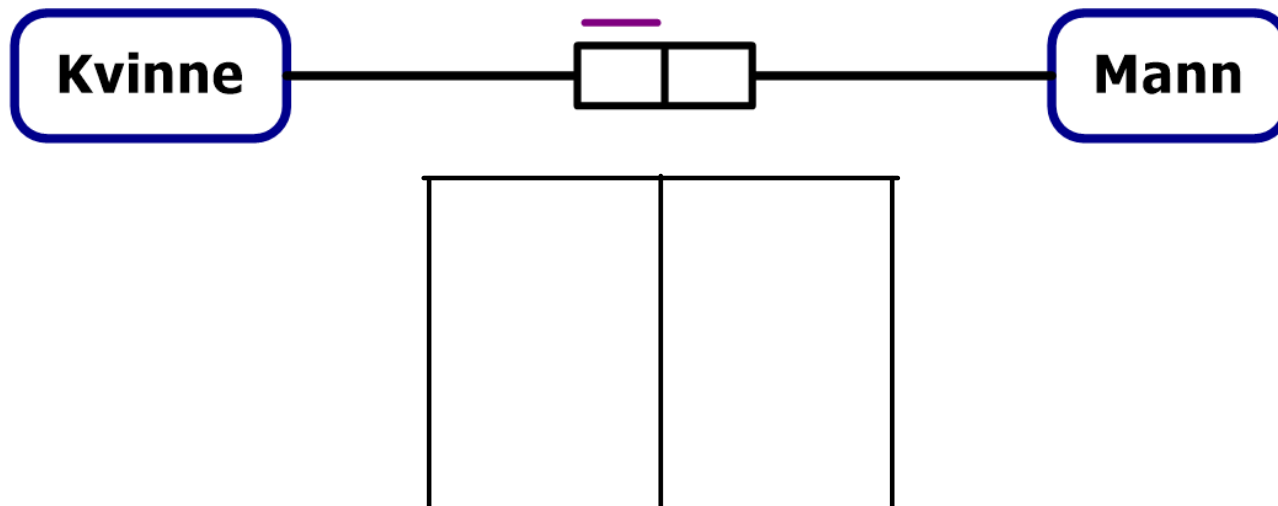
# Monogami

- Vi kaller dette en 1:1 (én-til-én) faktatype mellom (begrepene) Kvinne og Mann



# Ekteskap - 3

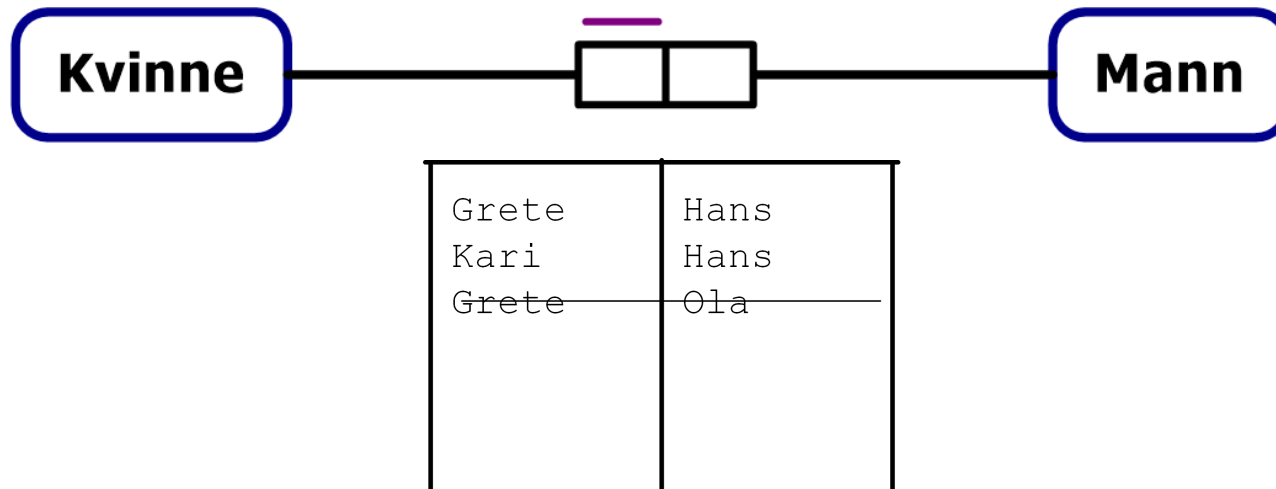
- En annen mulighet er nedenstående modell
- Lag forekomststabell





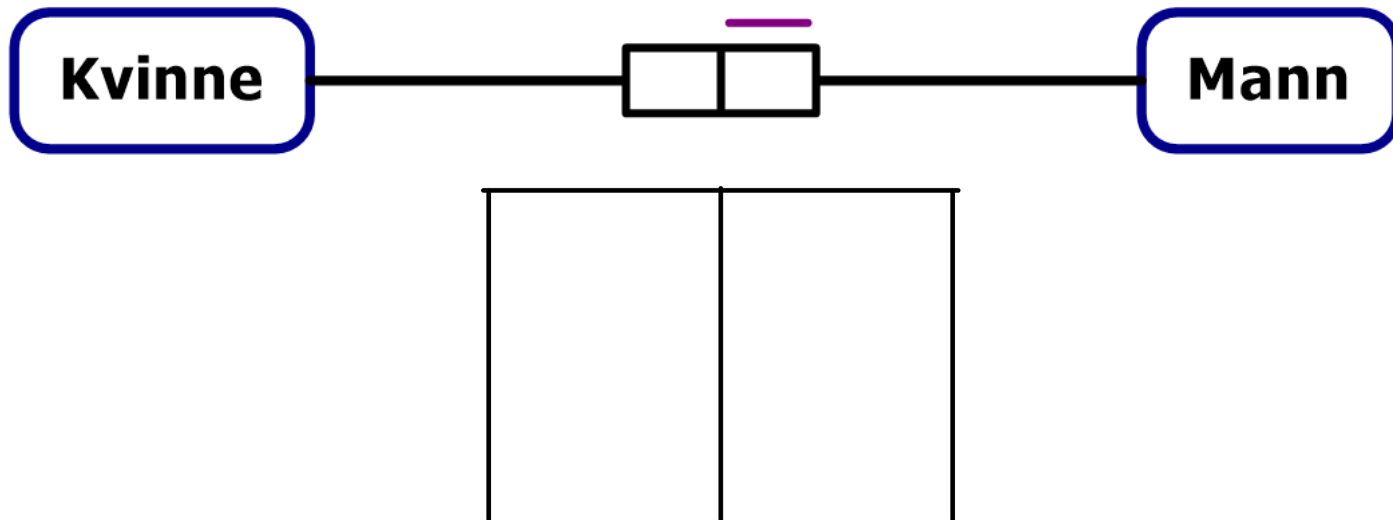
# Polygyni

- Polygyni (flerkoneri) er en ikke uvanlig ekteskapsform
- Dette er en n:1 (mange-til-én) faktatype fra Kvinne til Mann



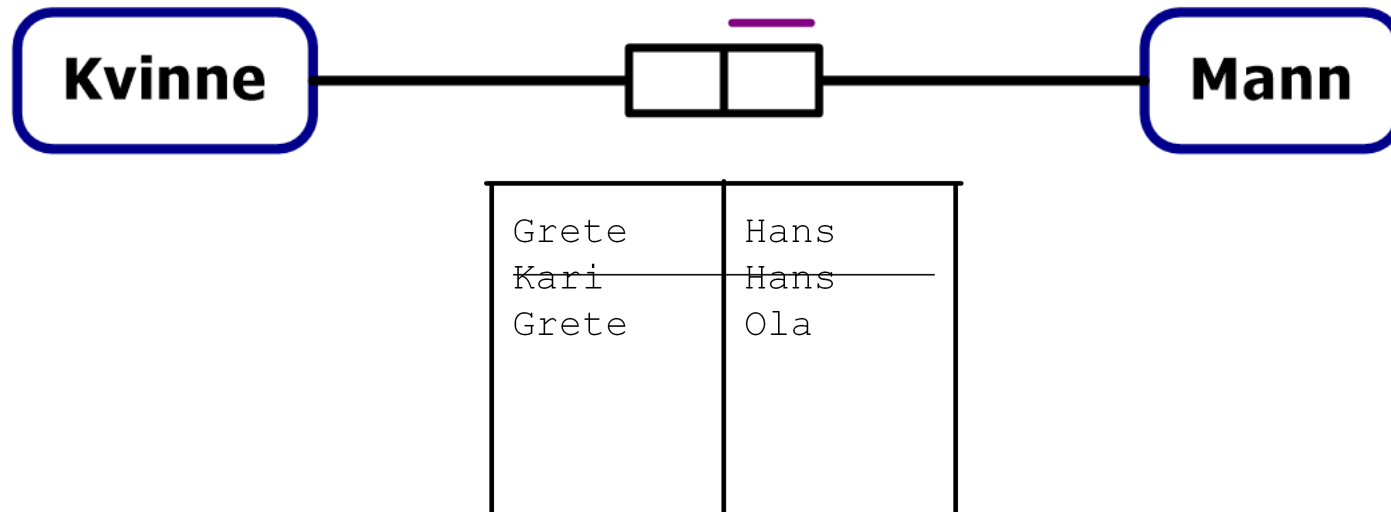
# Ekteskap - 4

- En tredje mulighet er nedenstående modell
- Lag forekomststabell



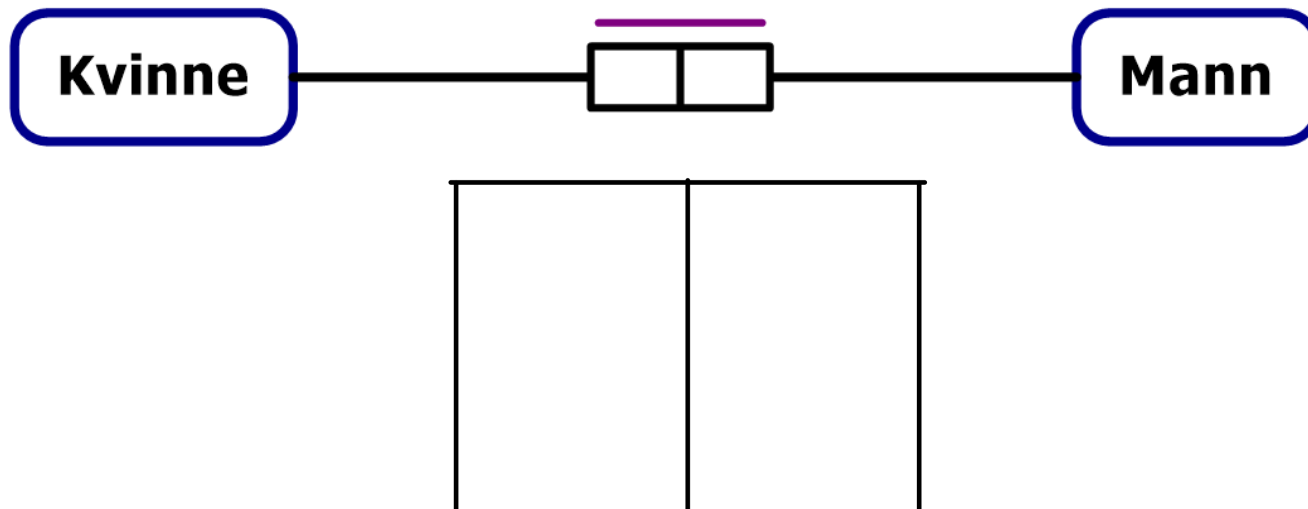
# Polyandri

- Polyandri (flermanneri) forekommer blant annet i Nepal og deler av India
- Dette er en 1:n (én-til-mange) faktatype fra Kvinne til Mann



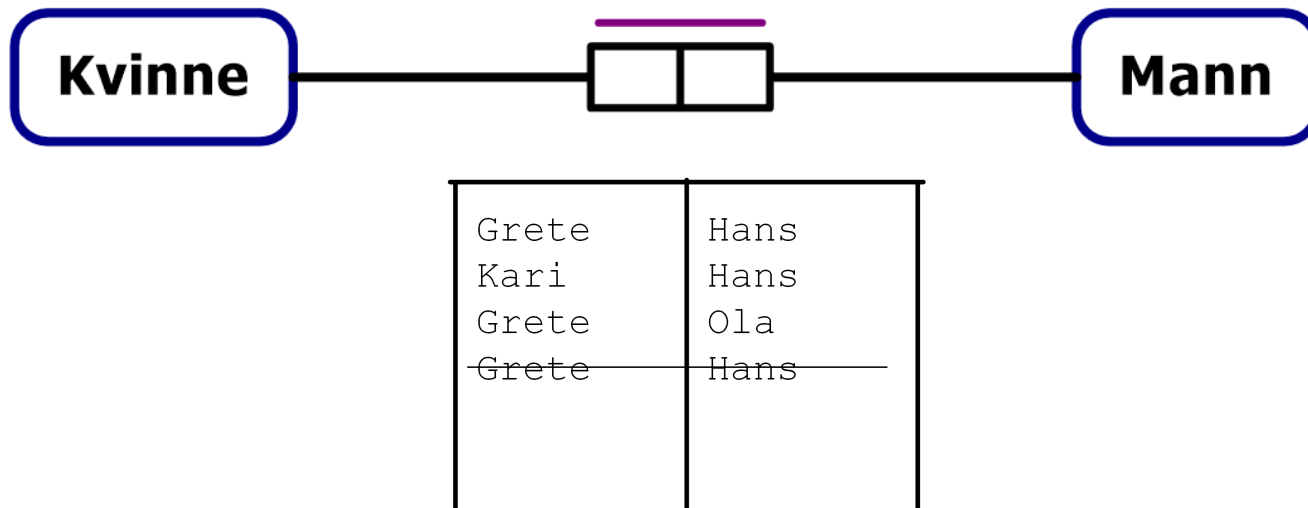
# Ekteskap - 5

- En siste mulighet er nedenstående modell
- Lag forekomststabell

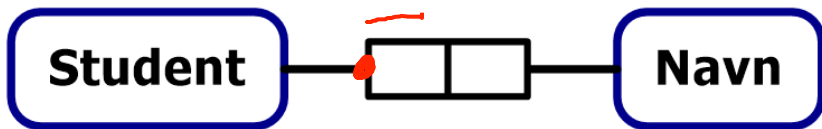


# Polygami

- *Ekte* polygami (flergifte, polyamori) forekommer nok ikke, i hvertfall ikke institusjonalisert
- Dette er en m:n (mange-til-mange) faktatype

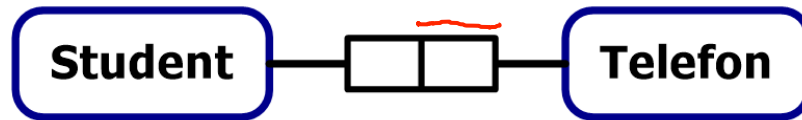


har / for n: 1



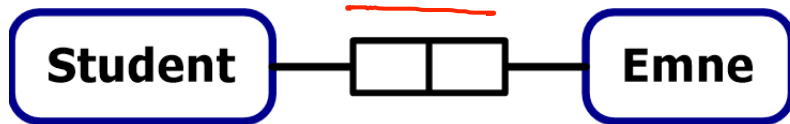
1 A  
 2 B  
~~1 B~~  
~~2 C~~  
 3 A

har / for 1: n



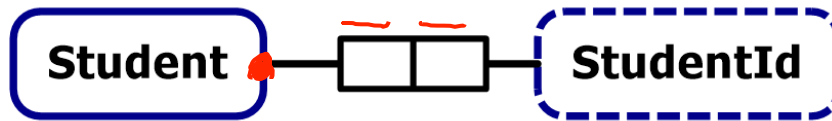
1 456  
 1 567  
~~2 456~~  
 2 891

tar / tas av m:n

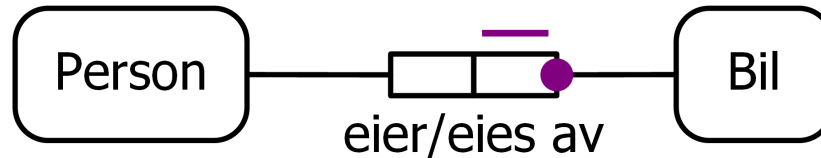


1 INF1300  
~~1 INF1300~~

har / for 1: 1



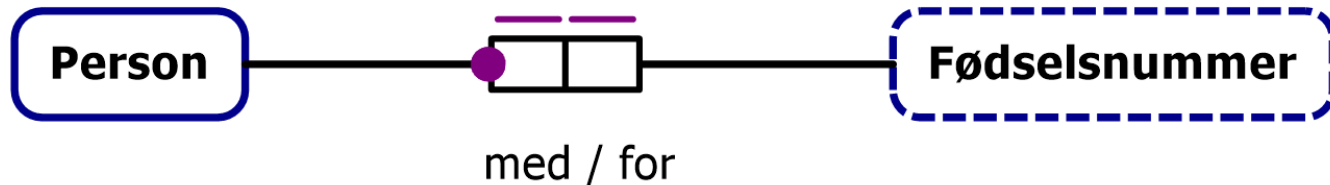
# Påkrevde roller



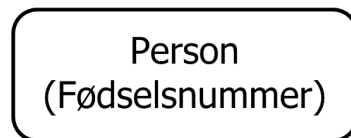
- Dersom alle biler **må** ha en eier, sier vi at rollen «eies av» er en **påkrevd rolle** for Bil og markerer det med en fet prikk på rollen
- **Merk:** Det at rollen er påkrevd, gjør at hver gang vi legger inn en bilforekomst i databasen, må vi **samtidig** registrere hvem som eier bilen
- Matematikerne sier at vi har en **total funksjon** fra Bil til Person (rollen «eies av» er definert for alle forekomster av Bil). Påkrevde roller kalles derfor også **totale roller**

# Perfekt bro - 1

- En 1:1 bro der begrepsrollen er påkrevd, kalles en **perfekt bro**



- Perfekte broer er så vanlige at vi har en egen kortform for dem (de implisitte rollenavnene er «med/på» eller «med/for» («with/of»)):

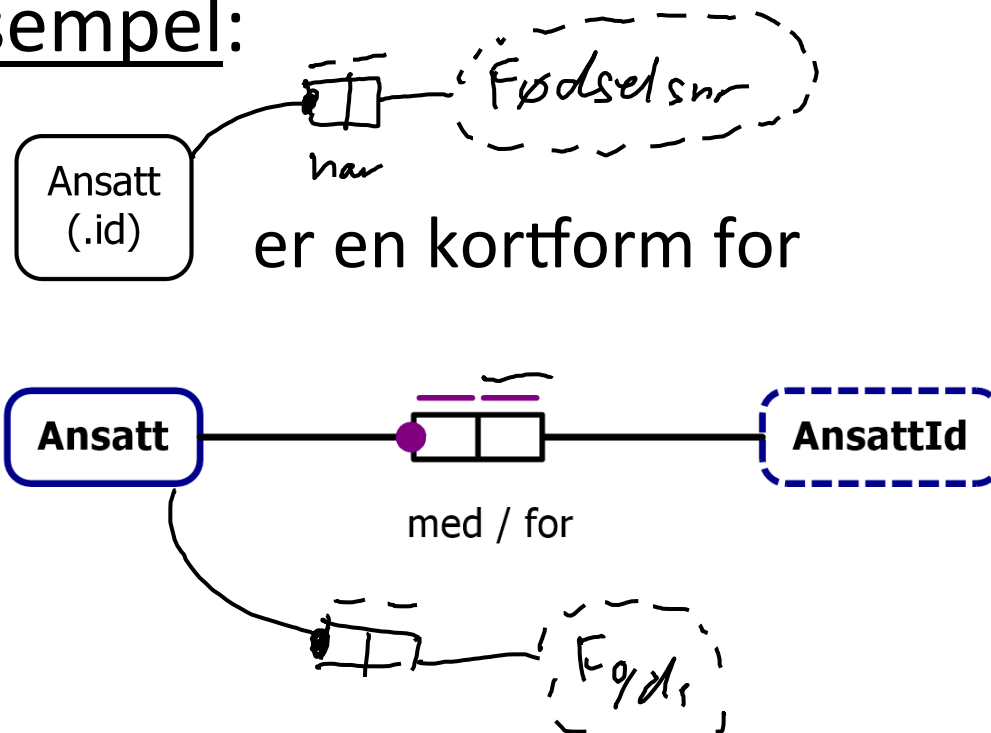


De to tegnemåtene er ekvivalente

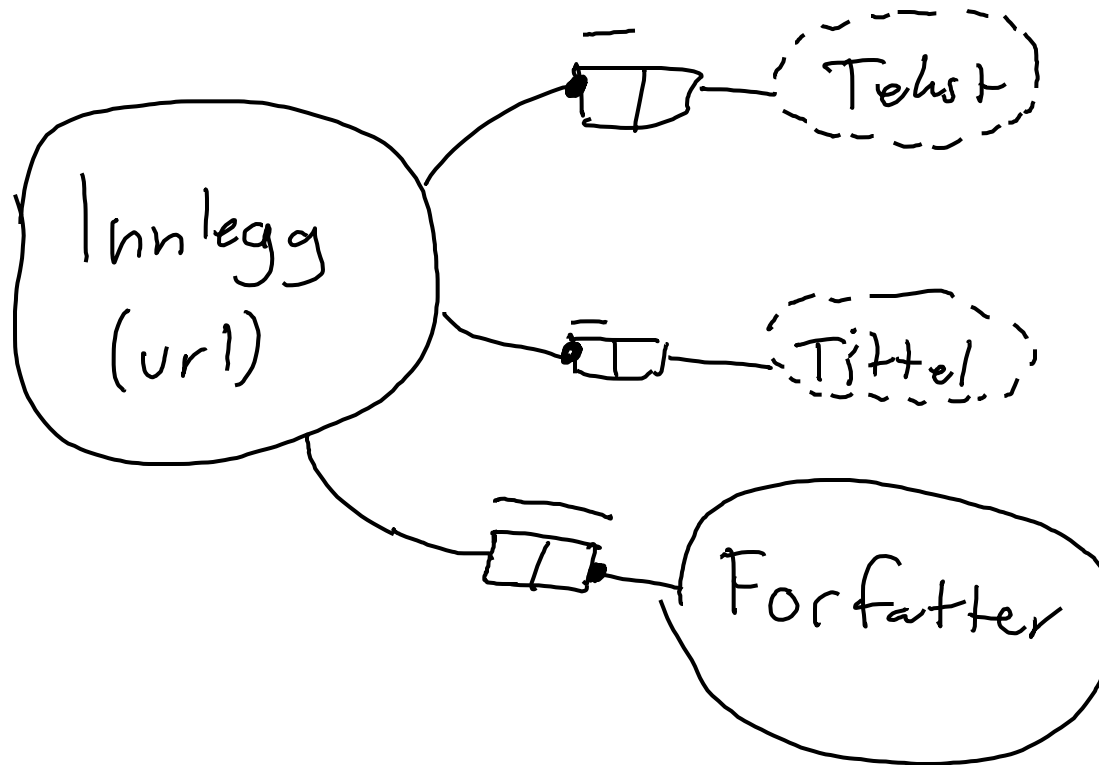


# Perfekt bro - 2

- Hvis vi har en perfekt bro hvor navnet på verditypen er lik begrepsnavnet med et suffiks, har vi en enda mer kompakt notasjon
- Eksempel:



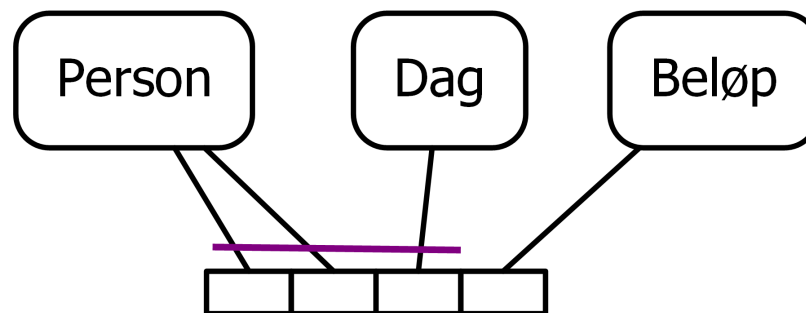
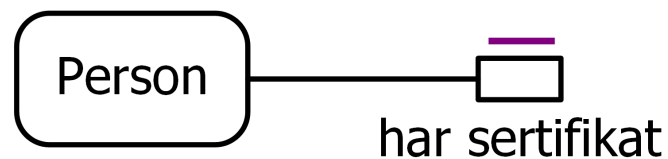
# Eksempel med blogg



# Mer om entydighetsskranke

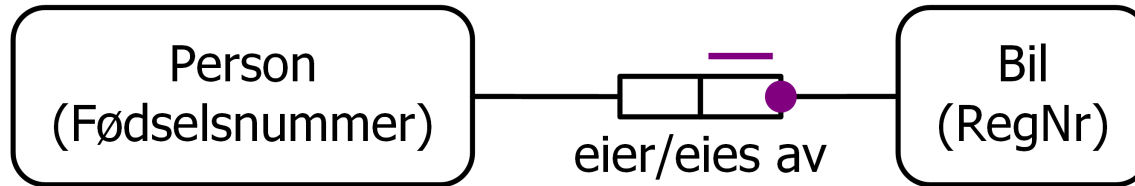
- Merk at hver setningstype (faktatype/bro) *alltid* skal ha minst én entydighetsskranke
- Hvis ikke, kunne samme faktum bli lagret vilkårlig mange ganger
- Merk også at en kort entydighetsskranke er strengere enn en lang
- Det er feil å la en lang entydighetsskranke dekke en kort

# Entydighetsskranker i ikke-binære setninger



er låntaker/er långiver/er tidspunkt for låneopptak/er størrelsen på lånet

# Funksjonelle avhengigheter



- Det at en bil alltid har én, og bare én, eier, gjør at hvis vi vet hvilken bil vi har (dvs. at vi kjenner bilens registreringsnummer), så vet vi også hvilken person som eier bilen (det er bare ett mulig tilhørende fødselsnummer)
- Faktatypen mellom Bil og Person definerer altså en funksjon fra forekomstene av Bil til forekomstene av Person som til en gitt bil gir oss eieren
- Vi sier at Person er **funksjonelt avhengig** av Bil, eller at vi har en FD (Functional Dependency) fra Bil til Person