

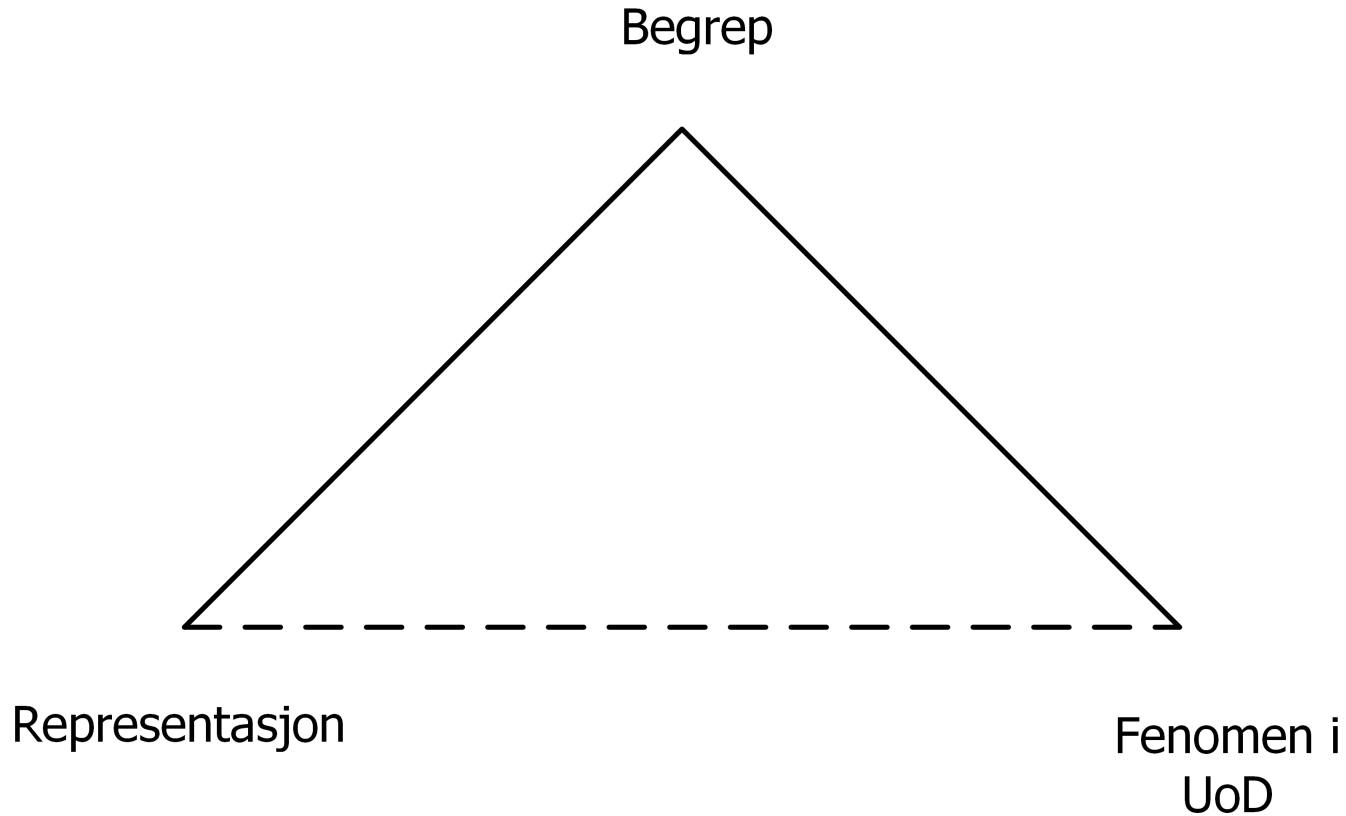


Introduksjon til databaser

Dagens tema:

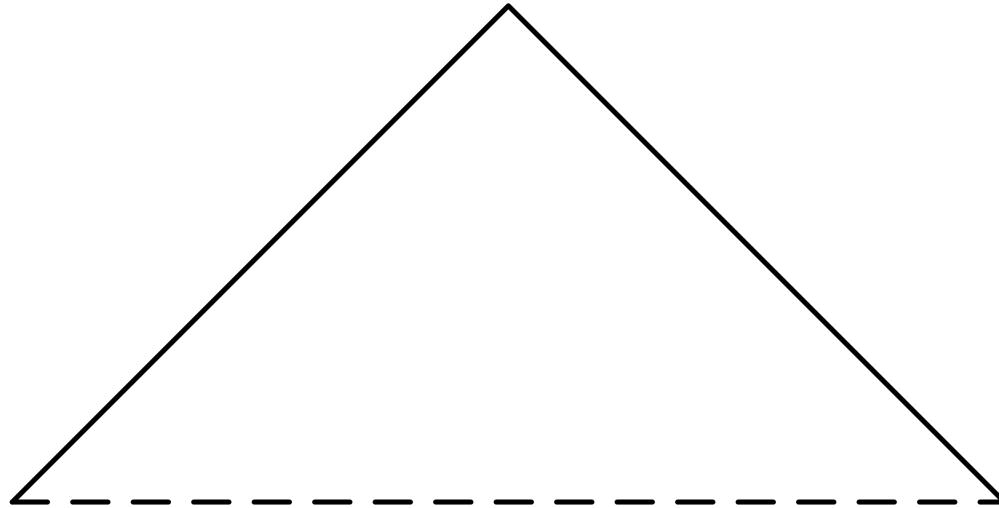
- Grunnbegrepene i ORM
- Sammenheng mellom ORM og naturlig språk
- Elementære setninger (fakta)
- Faktatyper og broer
- Entydighetsskranker og påkrevde roller

Ogdens trekant



Eksempel

Person



Fødselsnummer

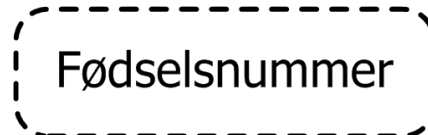


Ogdens trekant og ORM

- I ORM tegner vi **begreper** og **representasjonstyper** (**verdityper**) som henholdsvis heltrukne og stiplede rektangler med runde hjørner

- Eksempel:

Begrepet Person og representasjonstypen Fødselsnummer tegnes slik:



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

Setningers aritet - 1

- 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 - 2

- 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 høy aritet
- Elementære setninger med aritet > 3 er sjeldnere, så vi gir dem ikke egne navn (n-ære setninger)

Unære setninger

- Disse er vanlige i dagligtalen, men sjeldne i en informasjonsmodell
- Eksempel: En person med navn Siri har sertifikat
- En unær setning kan alltid erstattes av en binær setning hvor ett av begrepene har en boolsk representasjonstype
- Eksempel: En person med navn Siri har sertifikatstatus med boolsk verdi true

Binære setninger

- Disse er vanlige i dagligtalen og enda vanligere i en informasjonsmodell
- Eksempel: En person med navn Siri eier en bil med registreringsnummer DL12345
- Binære (og unære) setninger er alltid elementære
- En ORM-informasjonsmodell baserer seg nesten i sin helhet på binære setninger
- Elementære setninger med aritet $n > 2$ kan ved hjelp av et triks uttrykkes med binære setninger (vi kommer tilbake til dette i en senere ORM-forelesning)

Setning med aritet 4

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

Fakta og setningers dype struktur

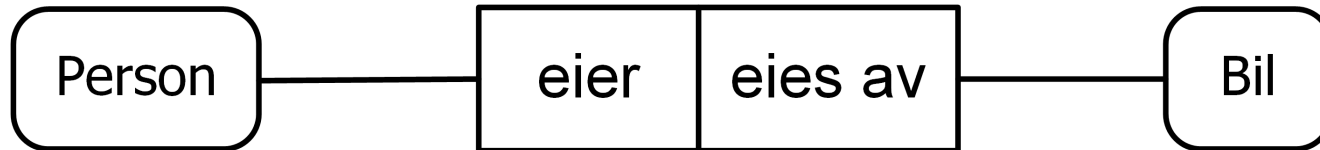
- Setningen: «En person med navn Siri eier en bil med registreringsnummer DL12345» kan også leses baklengs:
- «En bil med registreringsnummer DL12345 eies av en person med navn Siri»
- De to setningene har samme meningsinnhold
- Lingvistene kaller dette meningsinnholdet for **setningenes dype struktur**
- Vi informatikere kaller et slikt setningspar et **faktum**, og vi sier at ORM-metoden er **faktaorientert informasjonsmodellering**

Roller og faktatyper

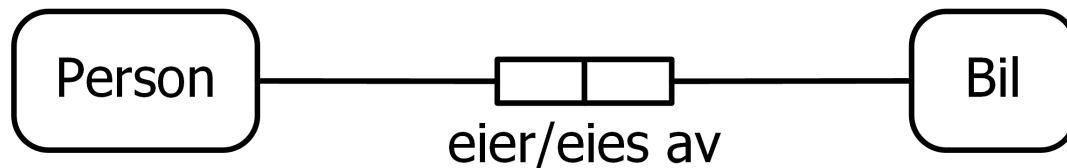
- Se på setningsparet: «En person med navn Siri eier en bil med registreringsnummer DL12345» og «En bil med registreringsnummer DL12345 eies av en person med navn Siri»
- Her kan vi åpenbart få lignende fakta ved å bytte ut forekomsten «Siri» med et annet navn og/eller forekomsten «DL12345» med et annet 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

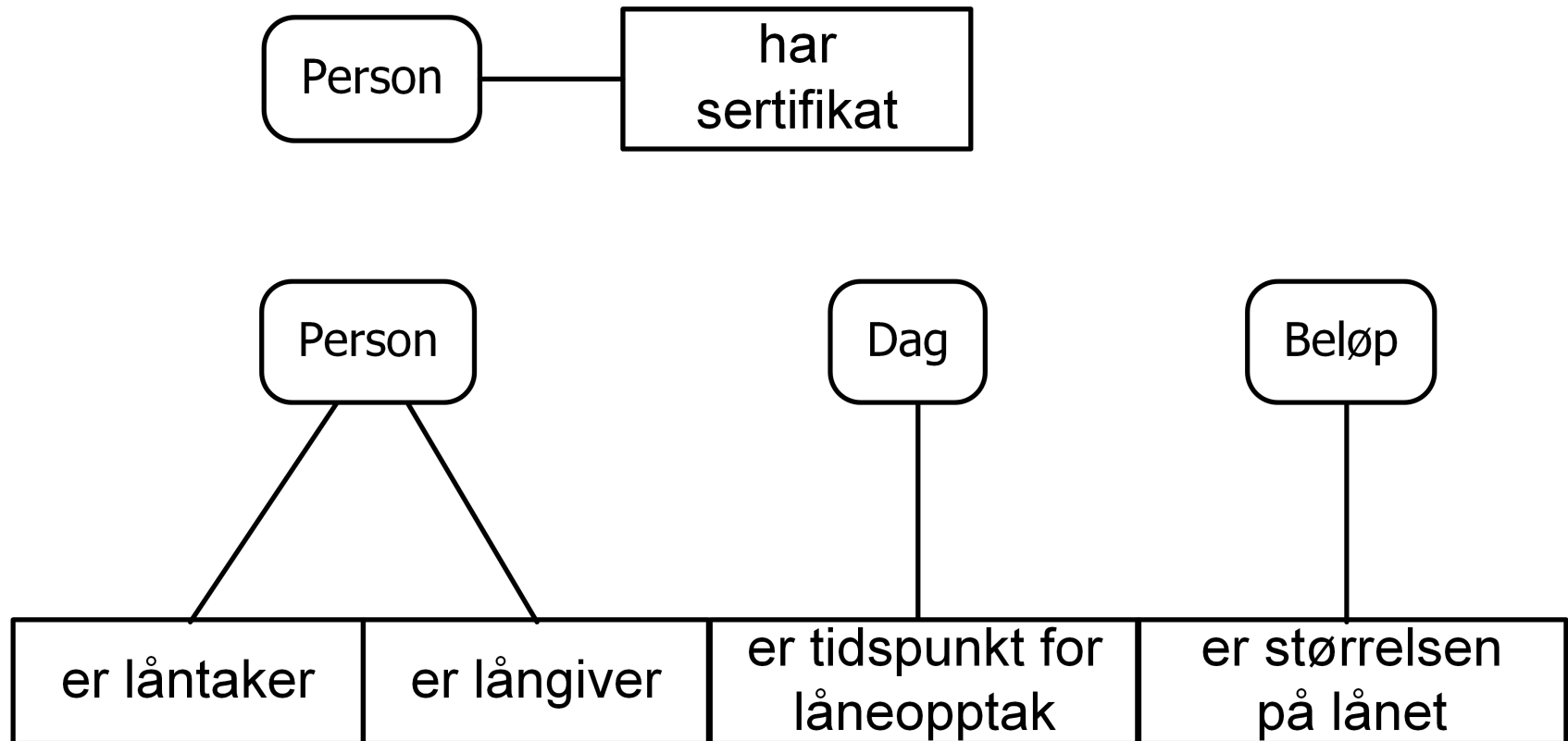
- Vårt eksempel på en faktatype mellom begrepene Person og Bil tegner vi slik:



- En mer kompakt tegnemåte er slik:

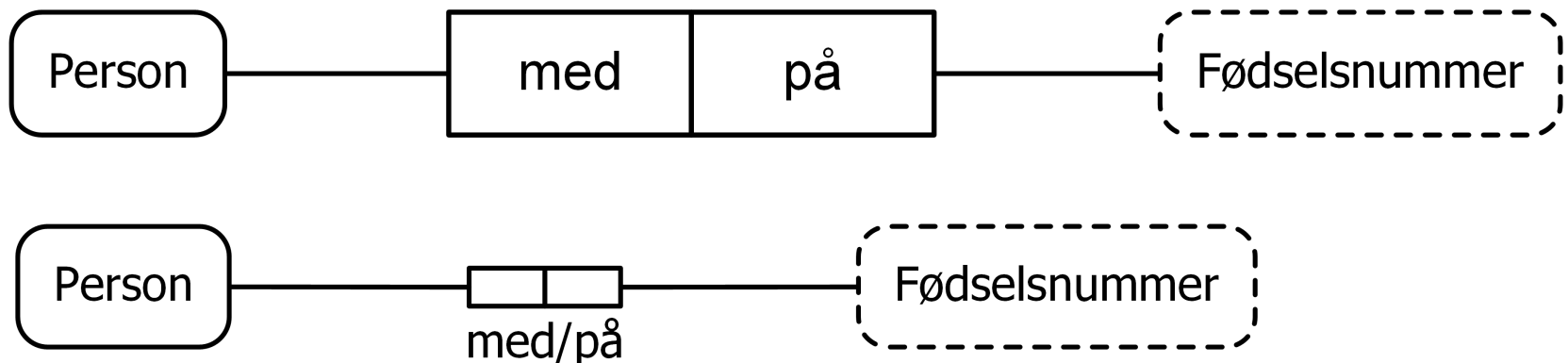


Faktatyper for eksemplene side 7 og 9



Broer

- En **bro** er en forbindelse mellom et begrep og en representasjonstype (verditype)
- Eksempel:
Her er de to måtene vi har i ORM for å tegne broen mellom Person og Fødselsnummer:



Setningstyper

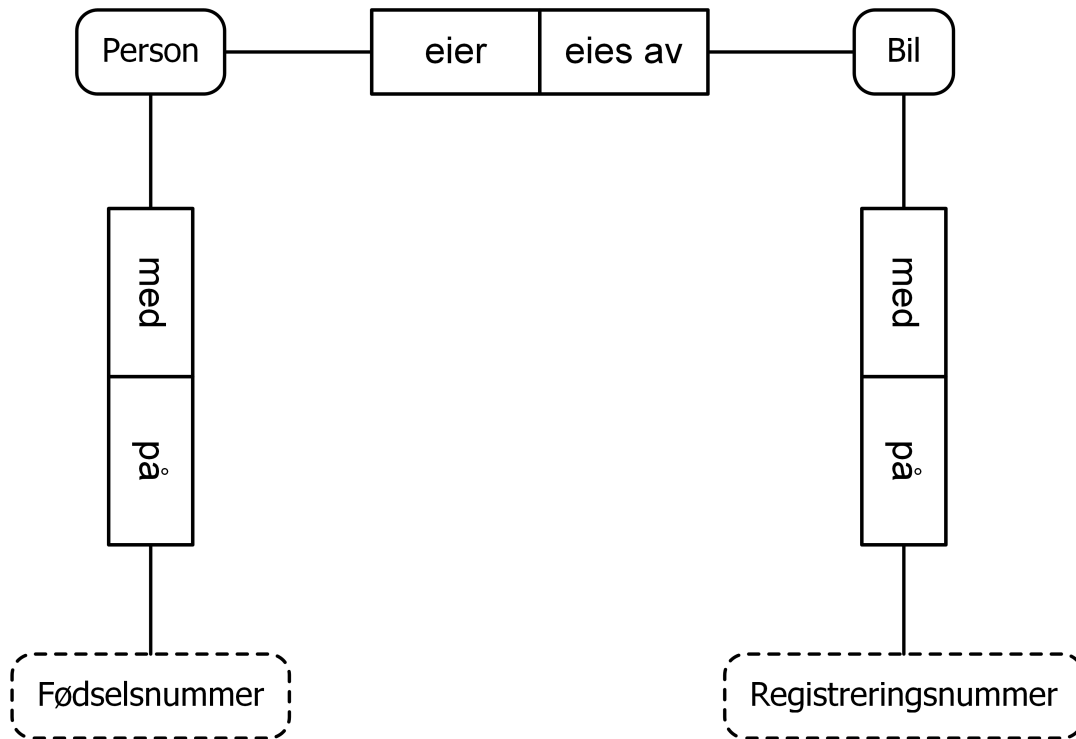
- Ordet **setningstype** er en felles betegnelse på faktatyper og broer
- Broer er alltid binære – de forbinder ett begrep og én representasjonstype
- Faktatyper kan ha et vilkårlig antall roller (aritet)
 - hver rolle skal være knyttet til nøyaktig ett begrep
 - et begrep kan spille flere roller i samme faktatype (se f.eks. låneeksempelet s.13, der begrepet Person spiller rollene «er låntaker» og «er långiver»)

Rollenavn

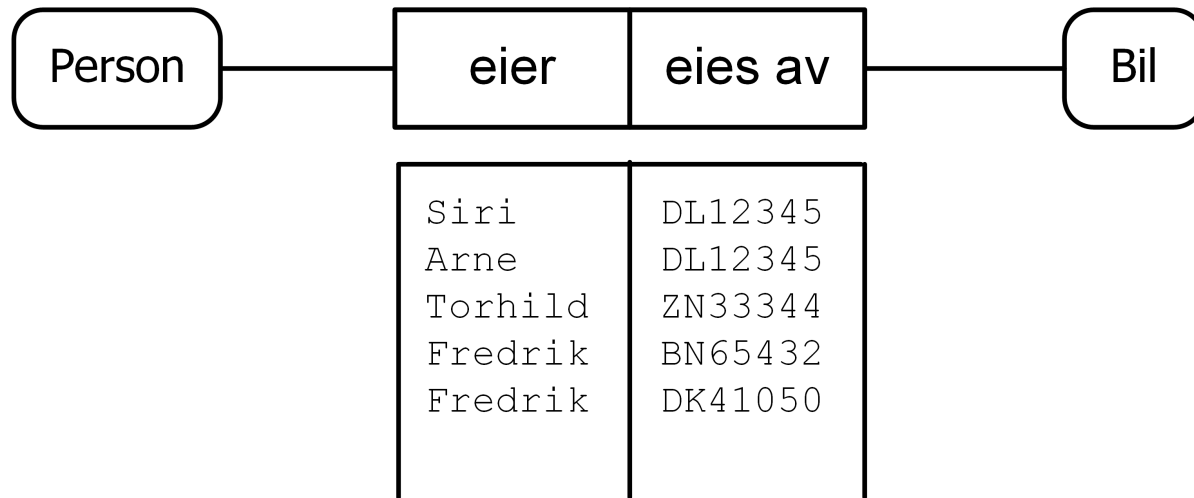
- I faktatyper bør alle rollenavn inneholde et verb (hvis ikke, er det grunn til å tro at rollenavnet er dårlig valgt)
- I broer er det vanlig med preposisjoner som rollenavn. De to vanligste rolleparene er
 - med/for
 - med/på

Fakta – setningers dype struktur

Dette diagrammet kan leses begge veier:

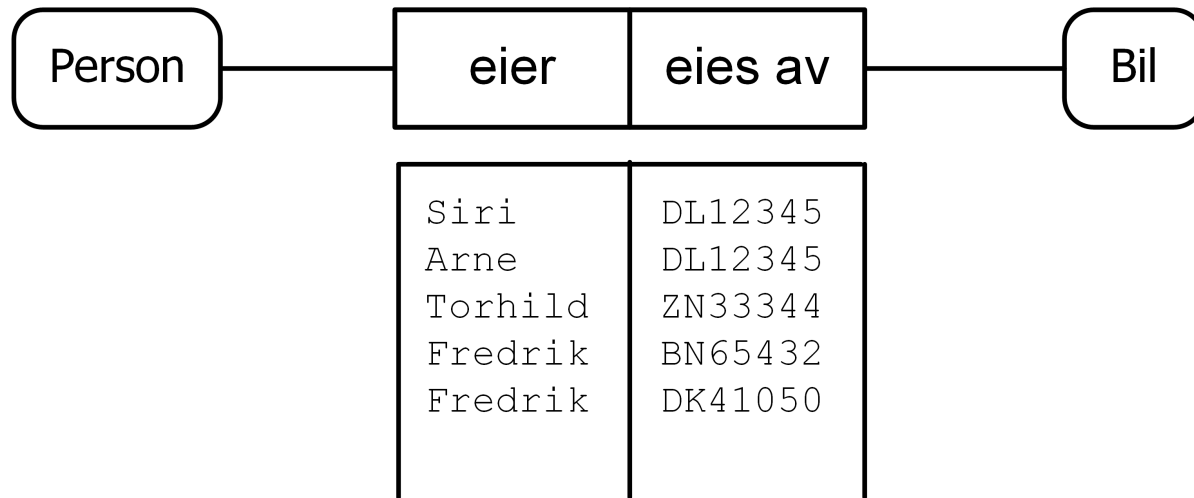


Faktatype med forekomststabell



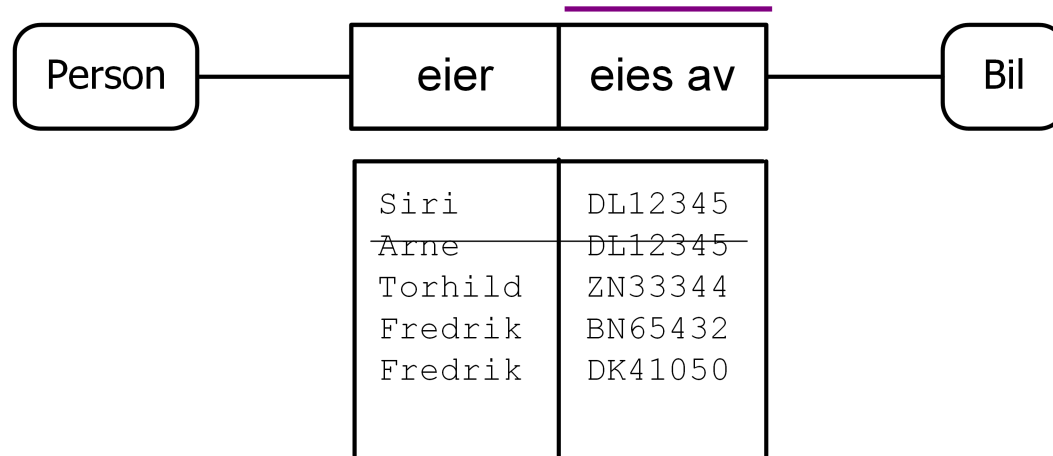
Legg merke til at en faktatype med forekomststabell gir oss like mange fakta som det er linjer i forekomsttabellen

Entydighetsstranker - 1



- Anta at vi har en forretningsregel som sier at en bil bare kan ha én eier (mens en person kan eie flere biler)
- Da kan ikke både Siri og Arne være eier av DL12345

Entydighetsskranker - 2



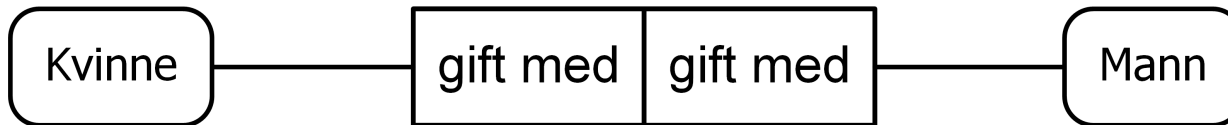
- I ORM-diagrammet plasserer vi en strek over rollen hvor samme forekomst ikke kan gjentas i forekomsttabellen
- Streken kalles en **entydighetsskranker**
- Entydighetsskranker kan gå over flere roller – da er det forekomstkombinasjonen i rollene som ikke kan gjentas

Bruk av forekomsttabeller

- Forekomsttabeller er ikke en del av ORM-modellen, men et nyttig, og ofte nødvendig, hjelpemiddel for å få riktige skranker i modellen.
- Derfor er det ikke så nøye hva slags verdier vi bruker i forekomsttabellene.
- For eksempel kan vi ha brukt et ansattnummer eller fødselsnummer for å representere personer, mens vi i forekomsttabellene bruker navn. I så fall antar vi for anledningen at alle personer har forskjellige navn.

Ekteskap - 1

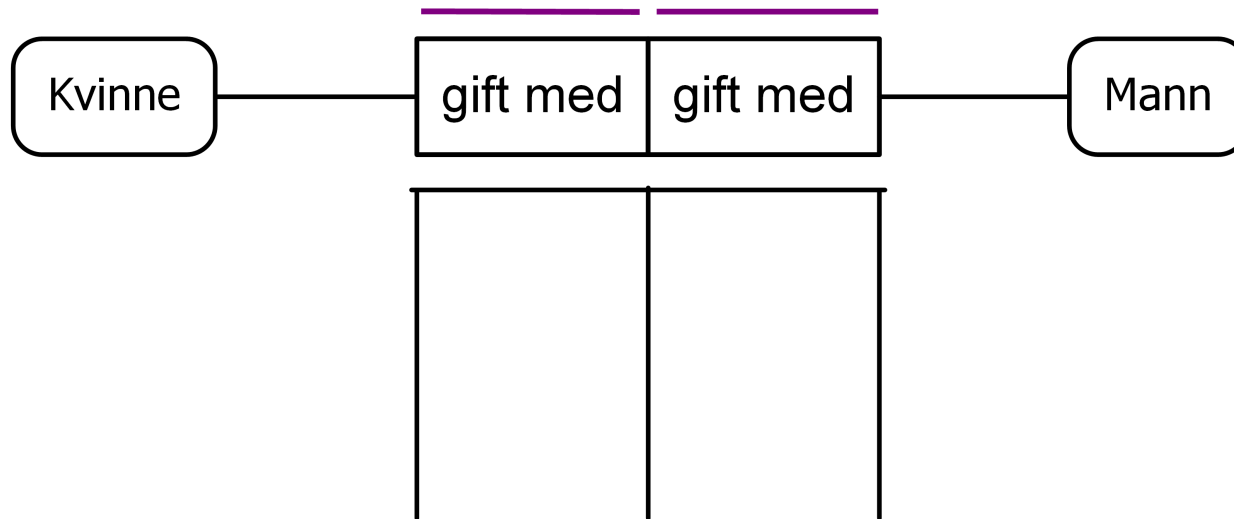
- Vi skal nå se på en faktatype mellom en kvinne og en mann kalt *ekteskap*
- Uten entydighetsskranke(r) ser modellen slik ut:



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

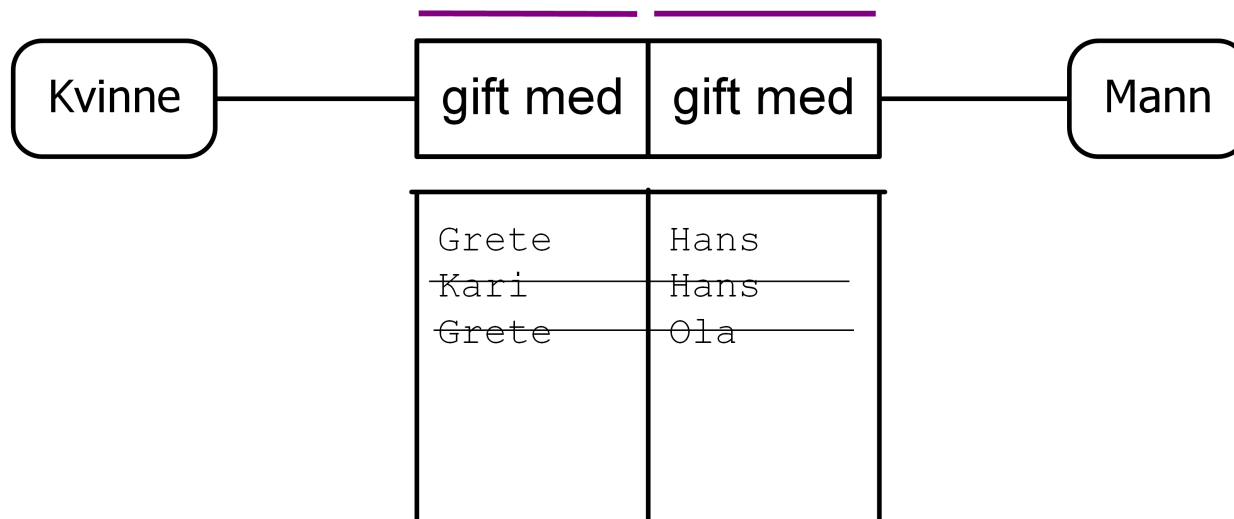
Ekteskap - 2

- De fleste har vel foreslått modellen nedenfor
- Lag forekomststabell



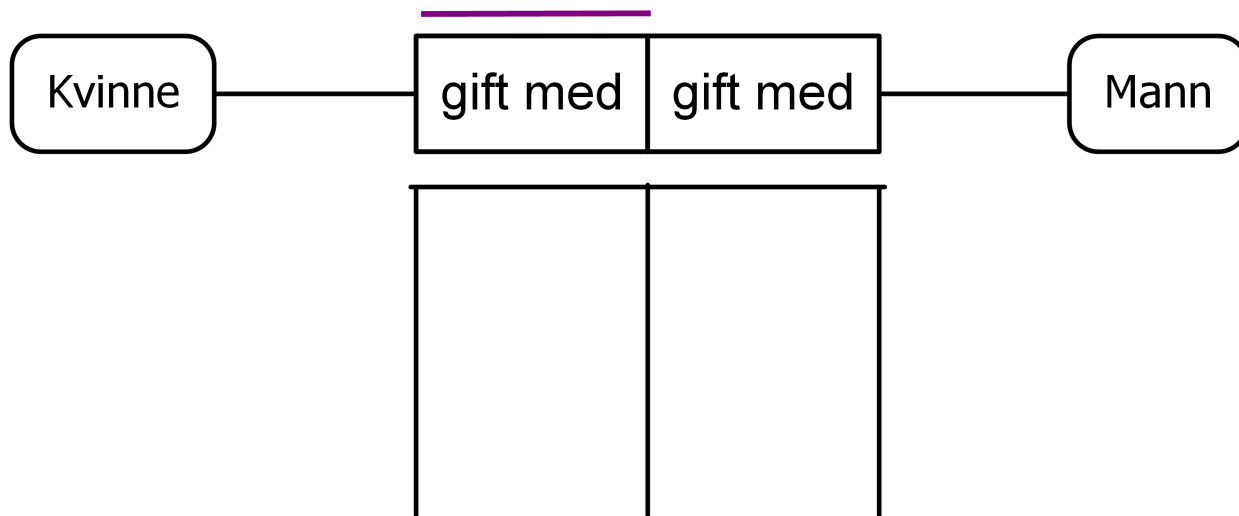
Monogami

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



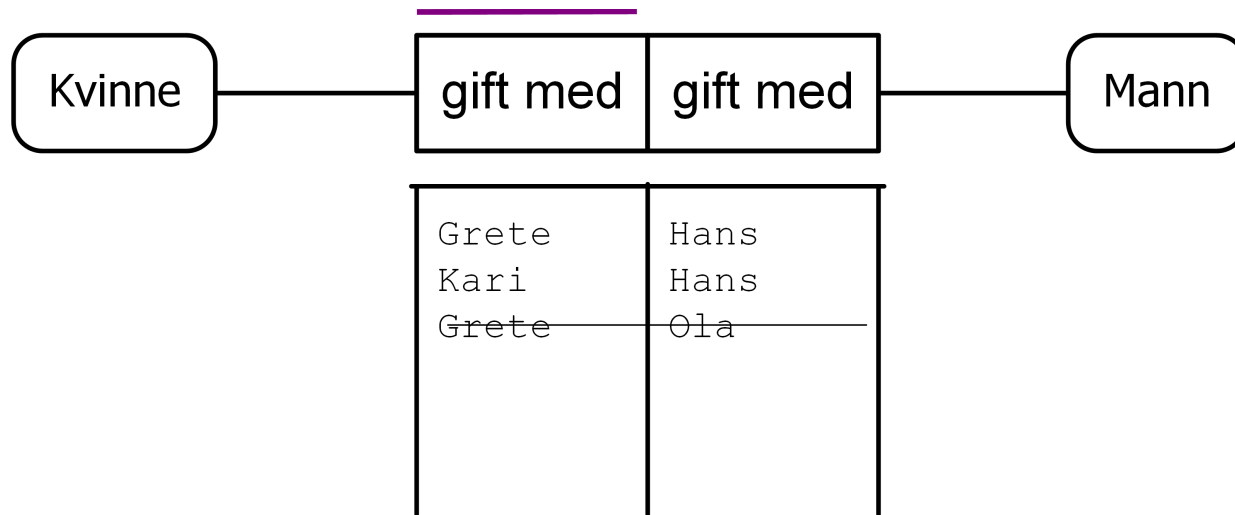
Ekteskap - 3

- En annen mulighet er nedenstående modell
- Lag forekomststabell



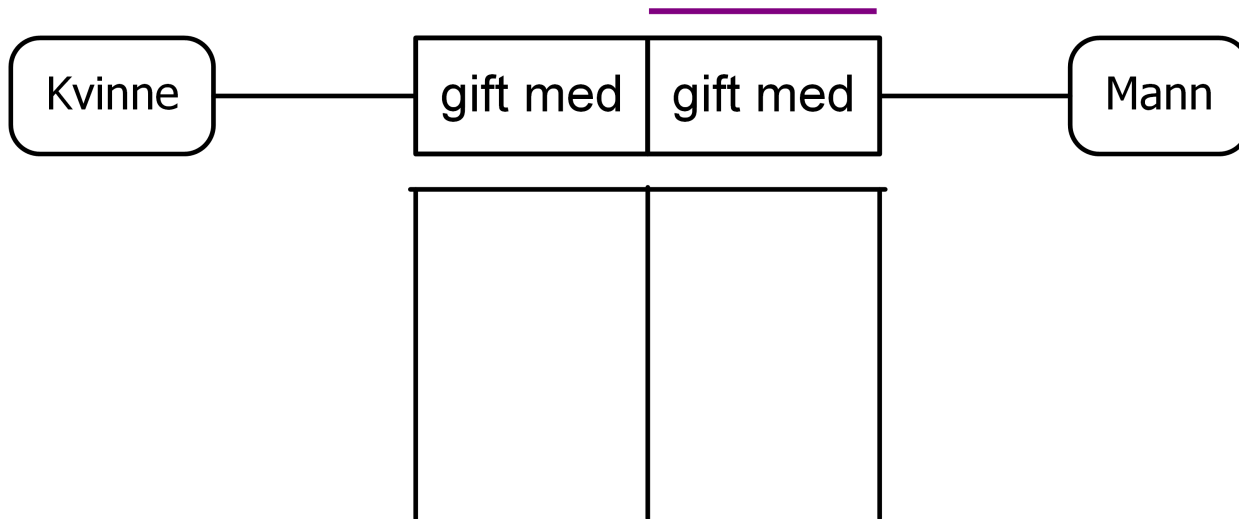
Polygyni

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



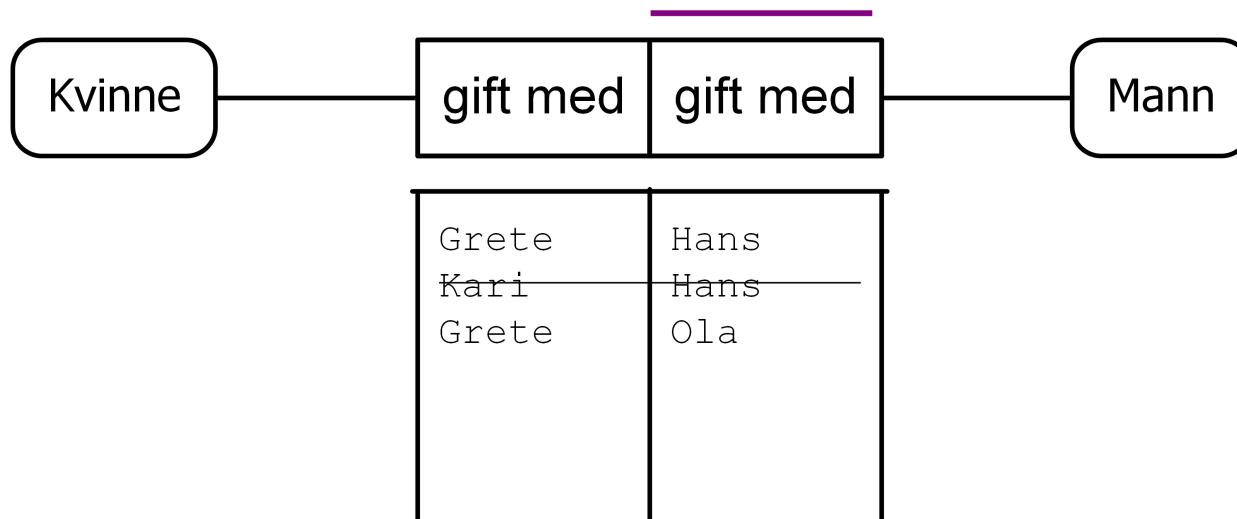
Ekteskap - 4

- En tredje mulighet er nedenstående modell
- Lag forekomststabell



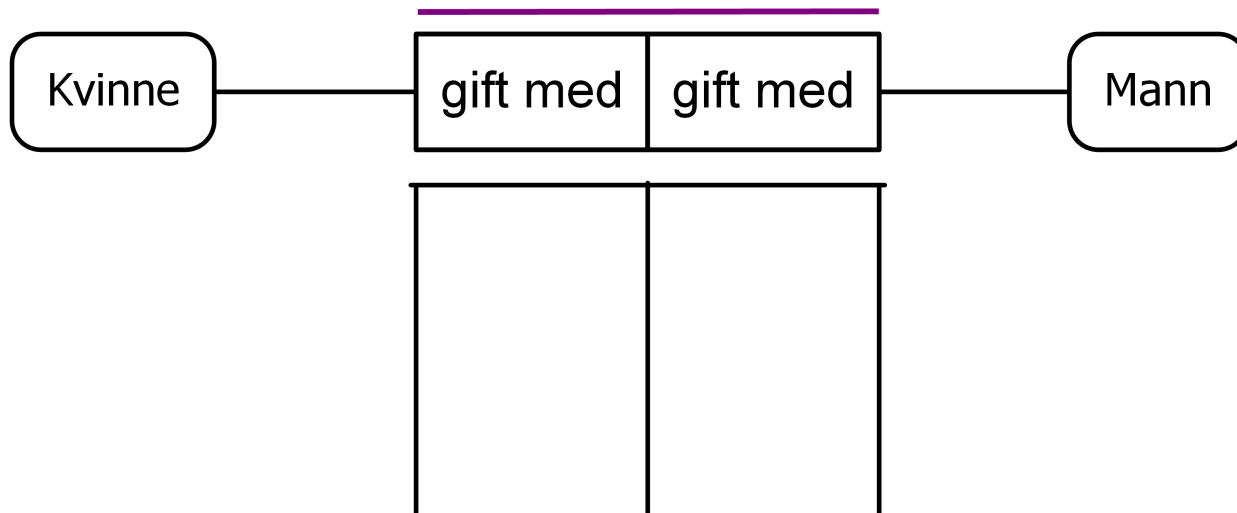
Polyandri

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



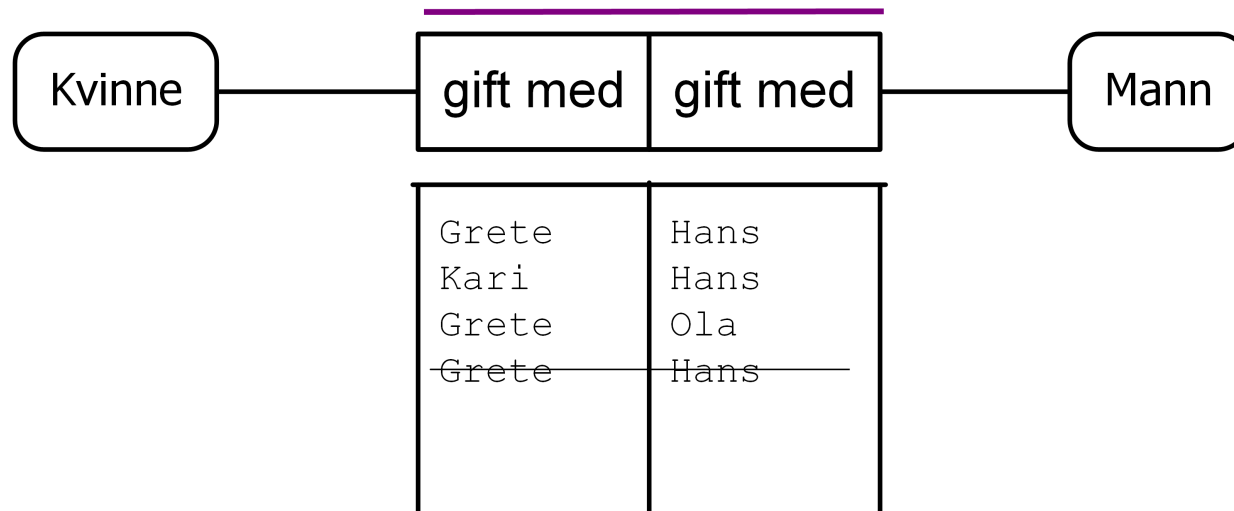
Ekteskap - 5

- En siste mulighet er nedenstående modell
- Lag forekomststabell

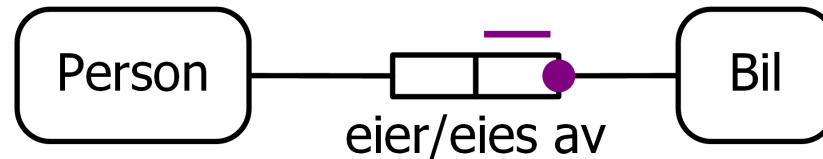


Polygami

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



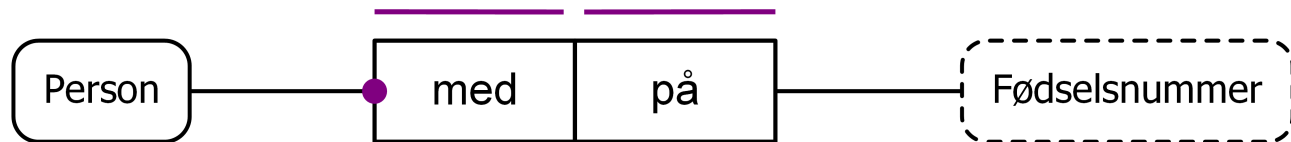
Påkrevde roller



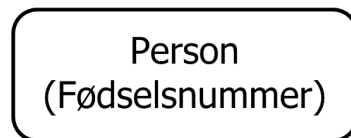
- Dersom alle biler **må** ha en eier, sier vi at rollen «eies av» er en **påkrevd rolle** for Bil og markerer det med en fet prikk på rollen
- **Merk:** Det at rollen er påkrevd, gjør at hver gang vi legger inn en bilforekomst i databasen, må vi **samtidig** registrere hvem som eier bilen
- Matematikerne 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**



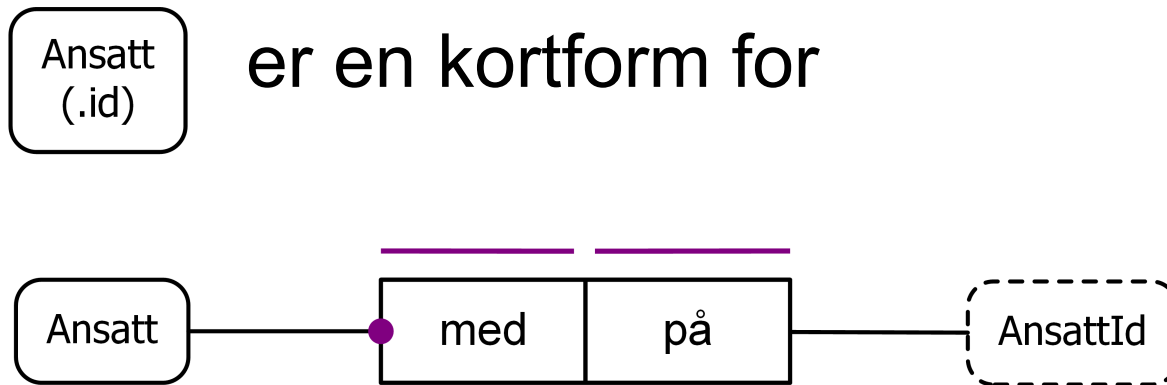
- Perfekte broer er så vanlige at vi har en egen kortform for dem (de implisitte rollenavnene er «med/på» eller «med/for» («with/of»)):



De to tegnemåtene er ekvivalente

Perfekt bro - 2

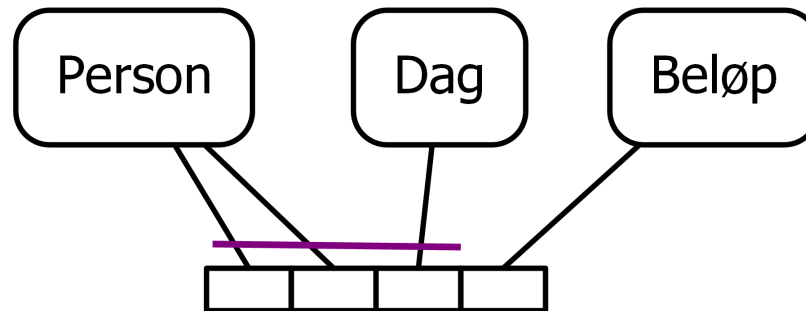
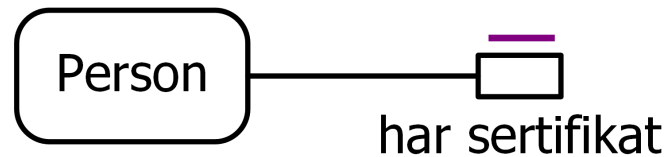
- Hvis vi har en perfekt bro hvor navnet på representasjonstypen er lik begrepsnavnet med et suffiks, har vi en enda mer kompakt notasjon
- Eksempel:



Mer om entydighetsskranke

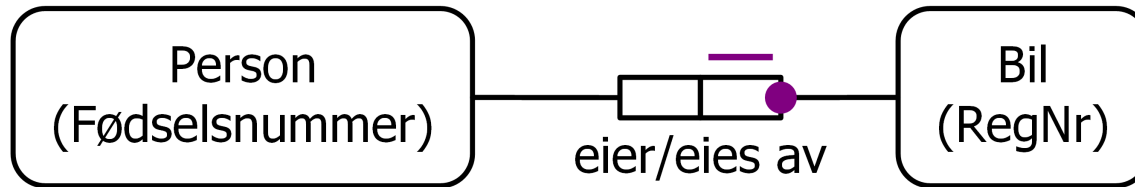
- **Merk at hver setningstype (faktatype/bro) alltid skal ha minst én entydighetsskranke**
- Hvis ikke, kunne samme faktum bli lagret vilkårlig mange ganger
- Merk også at en kort entydighetsskranke er strengere enn en lang
- Det er feil å la en lang entydighetsskranke dekke en kort

Entydighetsskranker for eksemplene side 7 og 9



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