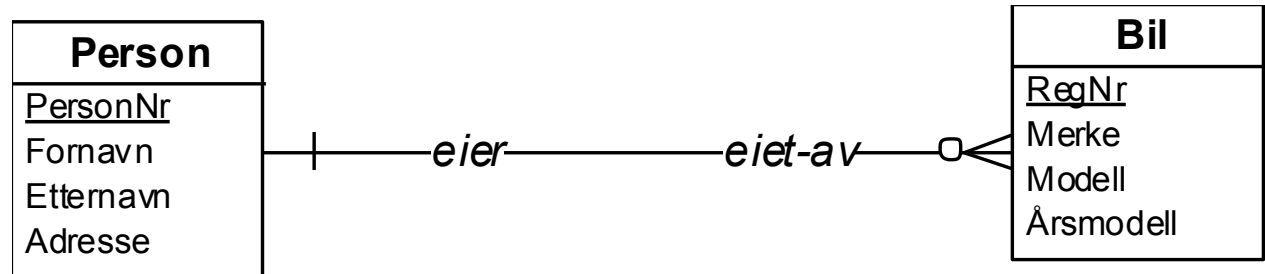


# Repetisjonsoppgaver

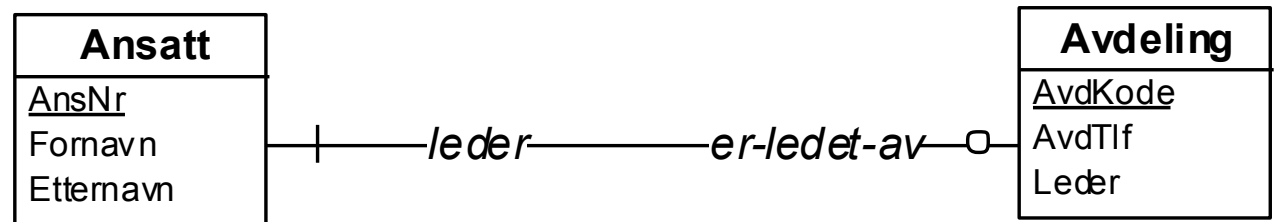
- ☐ Hva er en entitet? Gi eks.
- ☐ Hva er en attributt? Gi eks.
- ☐ Hva er en primærnøkkel? Gi eks.
- ☐ Hva er en fremmednøkkel? Gi eks.
- ☐ Hva er en kandidatnøkkel? Gi eks.
- ☐ Hva er en relasjon? Gi eks.
- ☐ Hva viser en ER-modell?
- ☐ Hva viser et logisk skjema?
- ☐ Hva er domenet til en attributt? Gi eks.
- ☐ Hva er en egenrelasjon? Gi eks.
- ☐ Hva menes med en relasjons kardinalitet? Gi eks.

# Kardinalitet

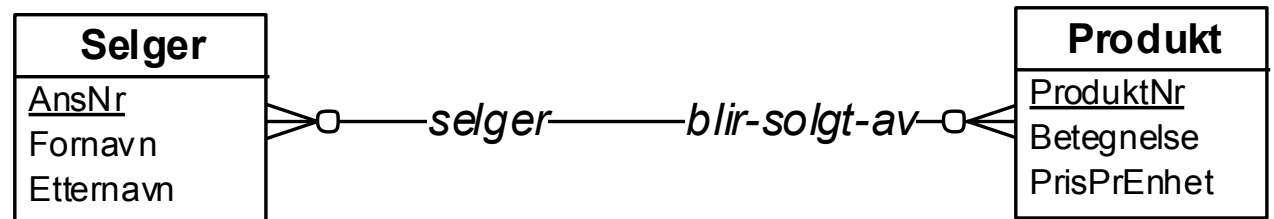
1:N  
(en-til-mange)



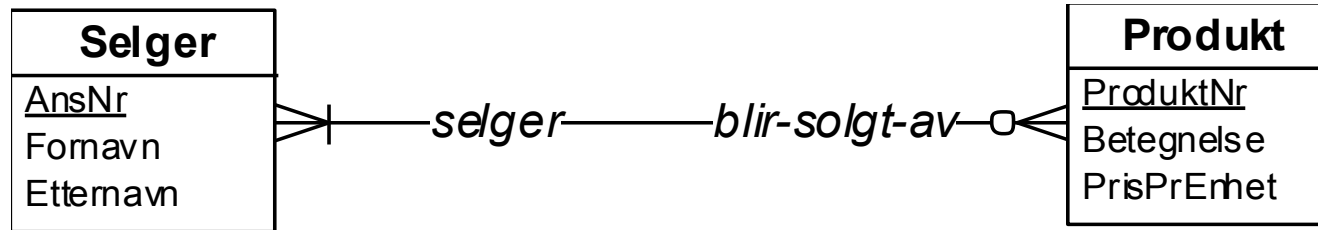
1:1  
(en-til-en)



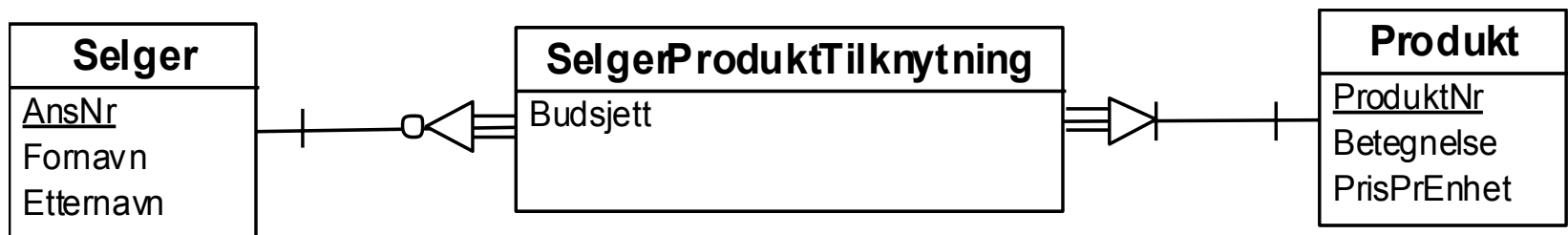
M:N  
(mange-til-mange)



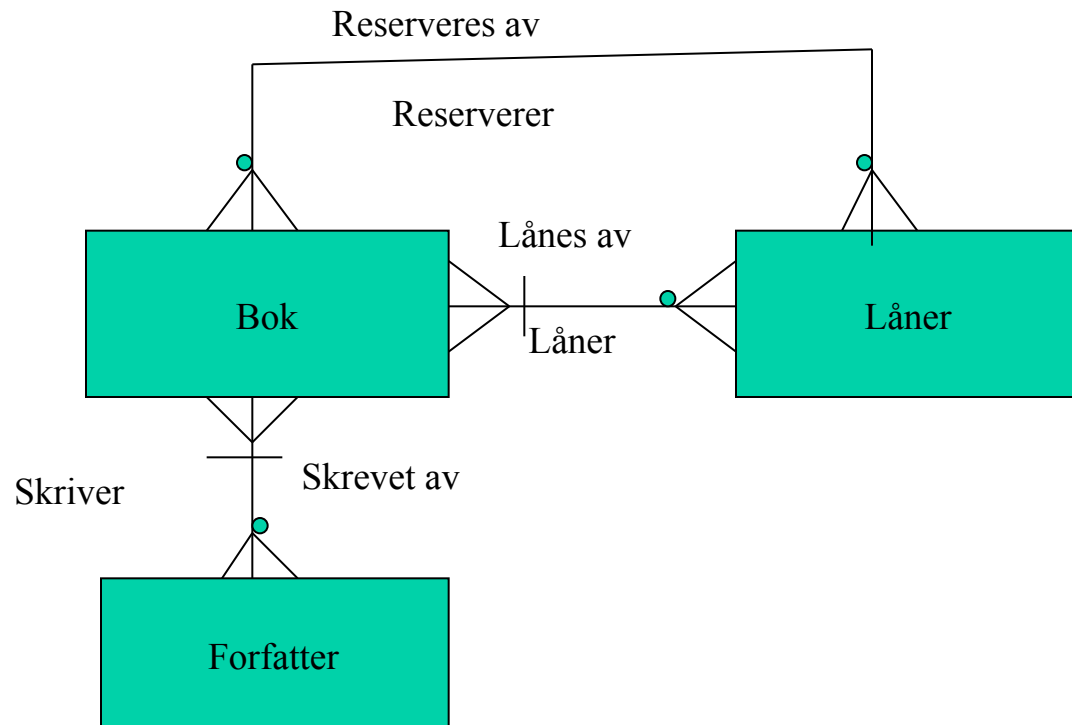
# Oppløsning av mange-til-mange forhold



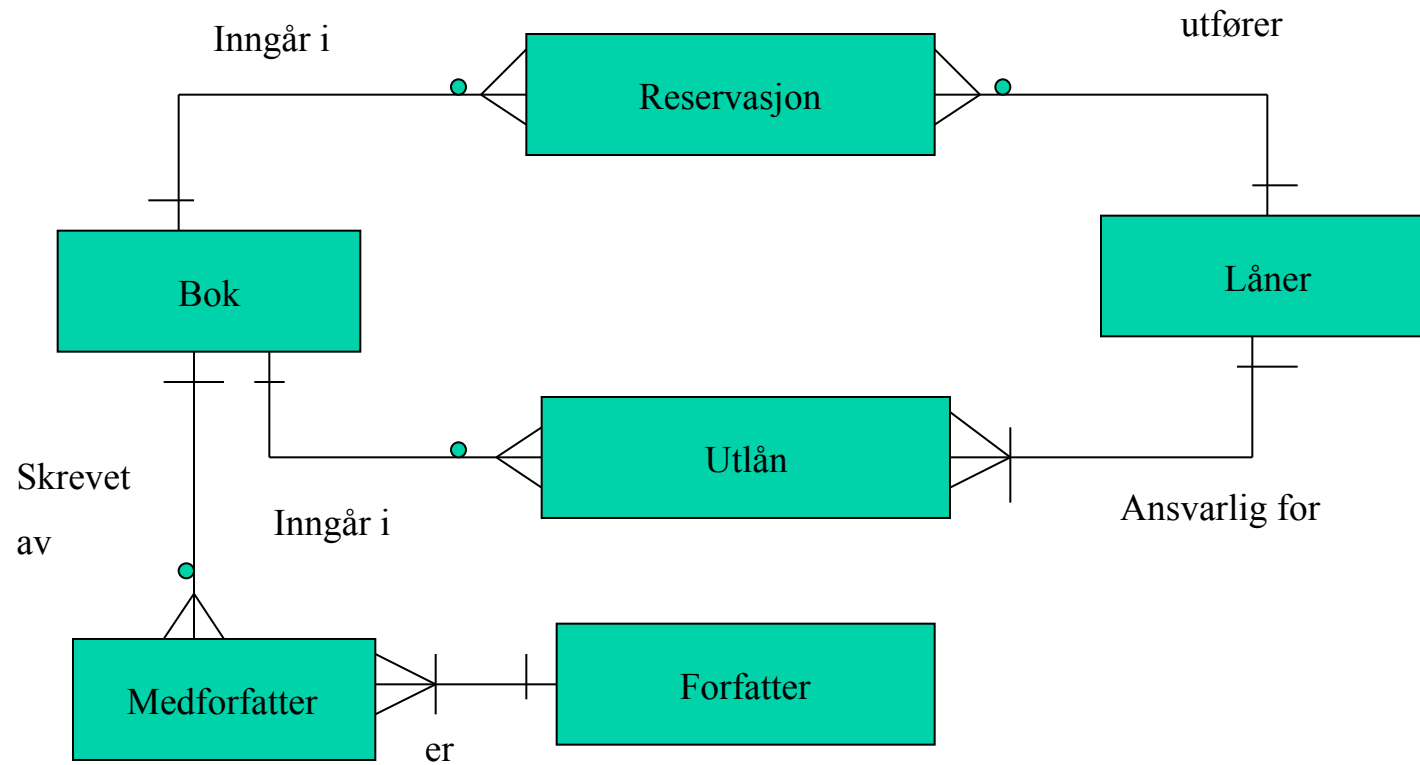
Er det interessante  
egenskaper ved forholdet?



# ER-modell



# Logisk skjema



## Lag ER-modell:

- ❑ En bedrift består av mange avdelinger. Hver av disse skal ha et avdelingsnummer og navn. Videre har bedriften mange ansatte med ansattnummer, navn og timelønn. Av modellen skal det fremgå hvem som leder hver avdeling(kun én ansatt), og hvilken avdelingen den enkelte er ansatt i. Det drives utstrakt prosjektvirksomhet i bedriften og hvert prosjekt har et nummer og et navn. Alle prosjekt har en «eier» som er en avdeling og dette skal fremgå i modellen. Ansatte kan delta på prosjekt og det skal fremgå hvilke prosjekter de arbeider/har arbeidet på og hvor mange timeverk det totalt har gått på hvert prosjekt for hver ansatt. Modellen skal også fortelle hvem av de ansatte som er prosjektleder for det enkelte prosjekt.

# Dagens tema

- ☐ En ER-modell
- ☐ Fra ER-modell til logisk skjema

# Logisk databasedesign

## ❑ Fra E/R til tabellstruktur:

- Hvilke tabeller får vi?
- Hvilke kolonner inneholder tabellene?
- Hva blir primærnøkler og fremmednøkler?

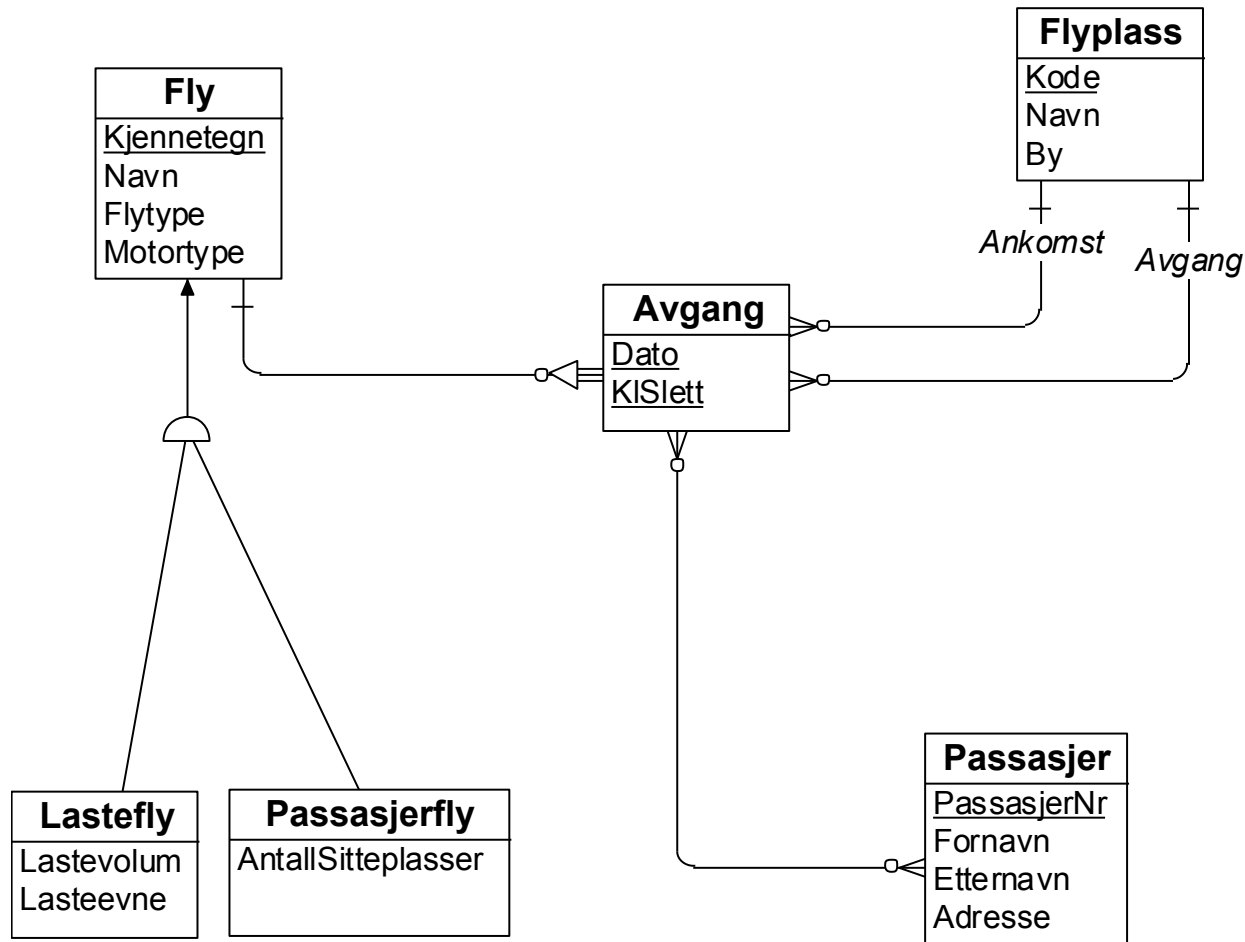
## ❑ Utsnitt (views):

- Hva brukes utsnitt til?
- Hvordan er de implementert?
- Oppdaterbarhet av utsnitt
- Forskjellen på spørringer og utsnitt

**Pensum: Kapittel 8.1 og 8.3**



# Eksempel: Flypark



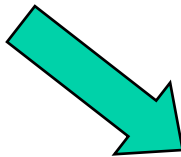
# Entiteter, attributter og identifikatorer

<b>Passasjer</b>
<u>PassasjerNr</u>
Fornavn
Etternavn
Adresse

Entiteter → tabeller

Attributter → kolonner.

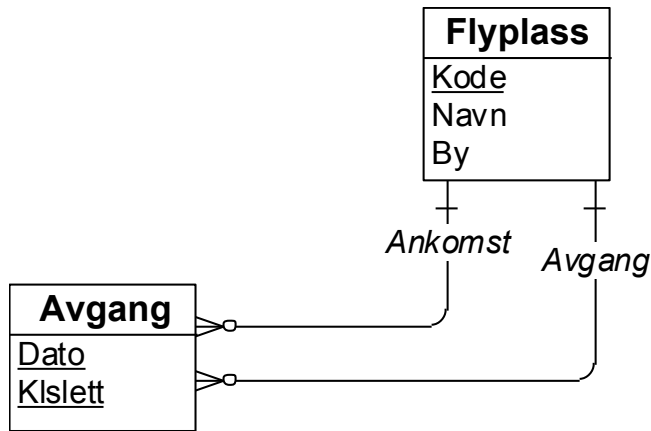
Identifikatorer → primærnøkler



---

<b>PASSASJERNR</b>	<b>FORNAVN</b>	<b>ETTERNAVN</b>	<b>ADRESSE</b>
1	Per	Hansen	Hansegata 3
2	Lise	Jensen	Jenseveien 12

# En-til-mange forhold



Flyplass( Kode, Navn, By )

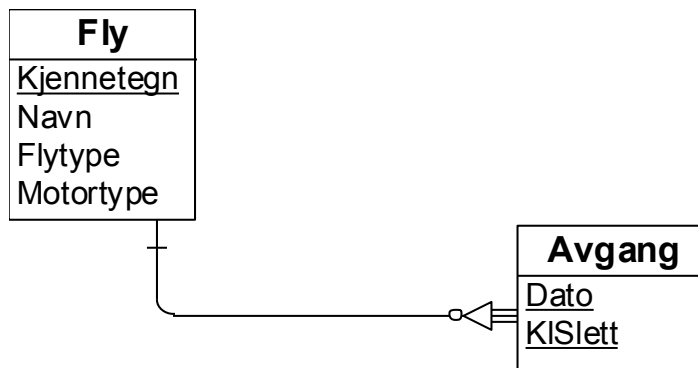
Avgang( ..., Dato, Klslett, *Ankomst*<sup>\*</sup>, *Avgang*<sup>\*</sup> )

➤ Identifikatorene på «en-siden» blir kolonner på «mange-siden»,

...

➤ ... og fremmednøkler tilbake mot «en-siden».

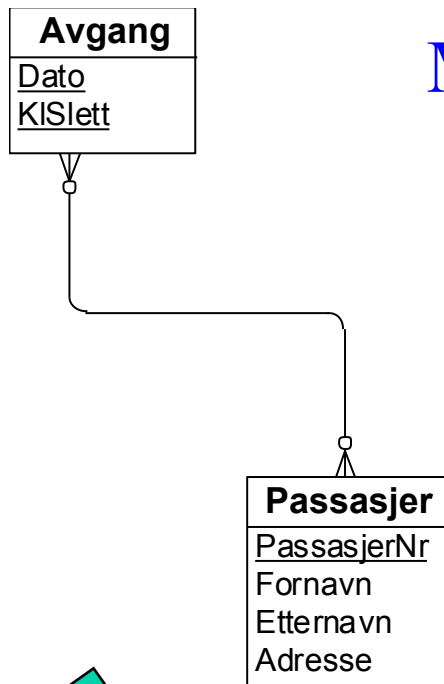
# Svake entiteter



- Svake entiteter arver primærnøkler fra entitetene de er avhengige av.
- Løpenumre er et alternativ til svake entiteter: AvgangsNr.

Fly( Kjennetegn, Navn, Flytype, Motortype )

Avgang( Kjennetegn<sup>\*</sup>, Dato, KlSlett, ...)



## Mange-til-mange forhold

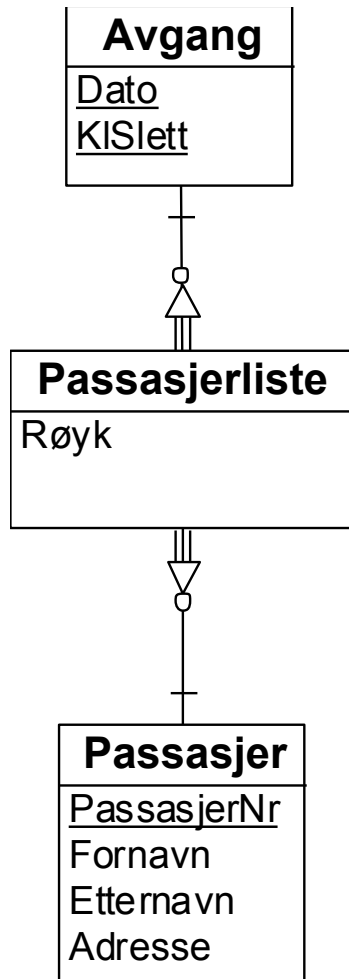
- Mange-til-mange forhold blir til «koblingstabeller».
- Identifikatorene for de involverte entitetene blir til en sammensatt primærnøkkel.
- Kan løse opp slike forhold i modellen, eller ved generering.

Avgang( Kjennetegn\*, Dato, KlSlett, ... )

**Passasjerliste**( Kjennetegn\*, Dato\*, KlSlett\*, PassasjerNr\* )

Passasjer( PassasjerNr, Fornavn, Etternavn, Adresse )

## Løse opp mange-til-mange forhold



- ☐ Løs opp mange-til-mange forhold hvis du ønsker å knytte attributter til forholdet.
- ☐ Koblingsentiteten vil inngå i to mange-til-en forhold.
- ☐ Power Designer danner en svak entitet.

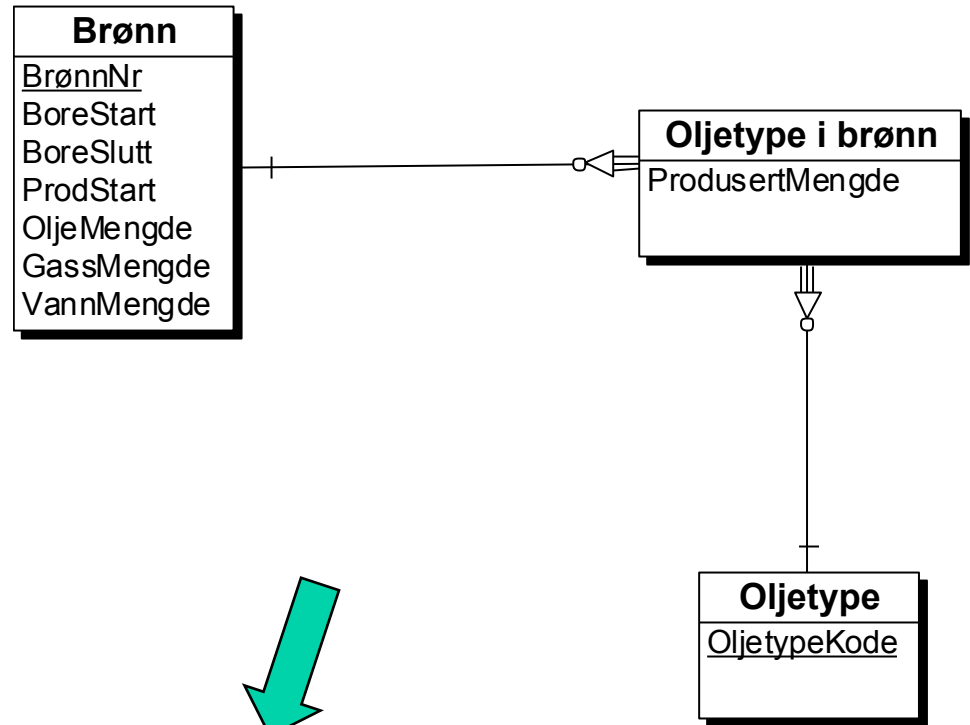
Passasjerliste(Kjennetegn<sup>\*</sup>, Dato, KISlett,  
PassasjerNr<sup>\*</sup>, Røyk)

# "Kodetabeller"

➤ Tabell for Oljetype?

➤ "Kodetabeller" gir oss kontroll med lovlige verdier (fremmednøkkel).

➤ Power Designer: Kan undertrykke tabeller ved å fjerne kryss i "Generate table".



OljetypeIBrønn(BrønnNr, OljeTypeCode\*, ProdusertMengde)

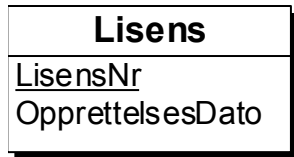
Oljetype(OljeTypeCode)

# Oppsummering av enkle regler

- ❑ Entiteter blir til tabeller
- ❑ Attributter blir til kolonner.
- ❑ Identifikatorer blir til primærnøkler.
  
- ❑ Mange-til-mange forhold blir til «koblingstabeller». Identifikatorene for de involverte entitetene blir til en sammensatt primærnøkkel.
  
- ❑ Identifikatorene på «en-siden» i en-til-mange forhold blir kolonner på «mange-siden», og fremmednøkler mot «en-siden».
  
- ❑ Svake entiteter arver primærnøkler fra entitetene de er avhengige av.
  
- ❑ Hva gjelder: 1:1 forhold og subtyper.



# En-til-en forhold



- ☐ Identifikatoren for entitet A blir til fremmednøkkel i B eller motsatt (eller begge deler).
- ☐ Power Designer: Ved å gjøre A dominant får A fremmednøkkel fra B.
- ☐ Hva gir færrest nullmerker ?
  - Koblingstabell et alternativ hvis begge varianter gir mange nullmerker.
- ☐ Kan vi slå sammen entitetene ?



Lisens(LisensNr, OpprettelsesDato, BrønnNr\*)

Brønn(BrønnNr, Borestart, Boreslutt, Prodstart,  
OljeMengde, GassMengde, VannMengde)

# Koblingstabeller for en-til-en forhold

- ❑ Man kan også opprette koblingstabeller for en-til-en forhold.
  - Nyttig hvis få forekomster fra begge sider deltar.
- ❑ Eksempel: En-til-en forholdet "kontor plass" mellom Ansatt og Rom.
  - Bygningen har mange rom, noen få er kontorer.
  - Bedriften har mange ansatte, kun noen få har kontor (og ingen deler kontor).

Rom(RomNr, Tlf)

Ansatt(AnsNr, Fornavn, Etternavn)

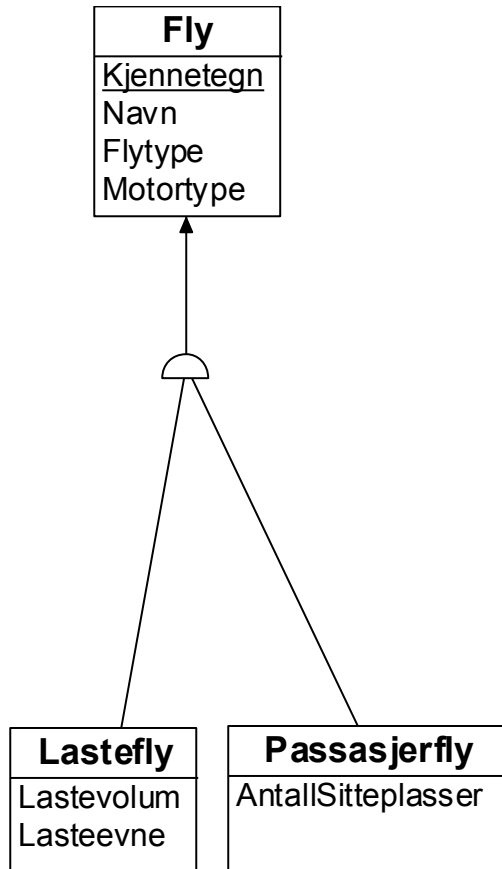
Kontor plass(RomNr\*, AnsNr\*)

- ❑ Forholdet blir altså håndtert på samme måte som et mange-til-mange forhold.

# Håndtering av subtyper

- ❑ Subtyper er en del av modelleringsspråket (E/R), men tradisjonelt støtter ikke relasjonsdatabaser dette.
- ❑ Dermed må vi "simulere" subtyper. Problemstillinger:
  - 1 eller flere tabeller ?
  - Hva skal "arves" ?
  - Nullmerker ?
  - Hvordan representere tilhørighet til subtyper ?
- ❑ Objektrelasjonelle databaser har støtte for subtyper.
  - Det er en fordel at "modelleringsspråket" bygger på de samme prinsippene som "implementasjonsspråket".

# Subtyper I: Separasjon



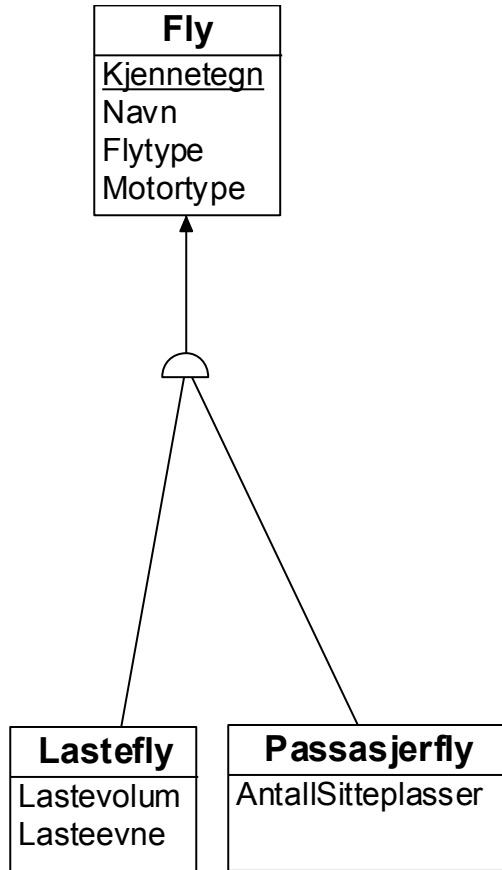
- ❑ Lager tabeller for både supertype og subtyper.
- ❑ Subtyper arver identifikatoren fra supertypen.
- ❑ Forekomster vil ligge i både "subtabeller" og "supertabell"!

Fly(Kjennetegn, Navn, Flytype, Motortype)

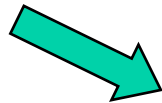
Lastefly(Kjennetegn\*, Lastevolum, Lasteevne)

Passasjerfly(Kjennetegn\*, AntallSitteplasser)

## Subtyper II: Absorpsjon

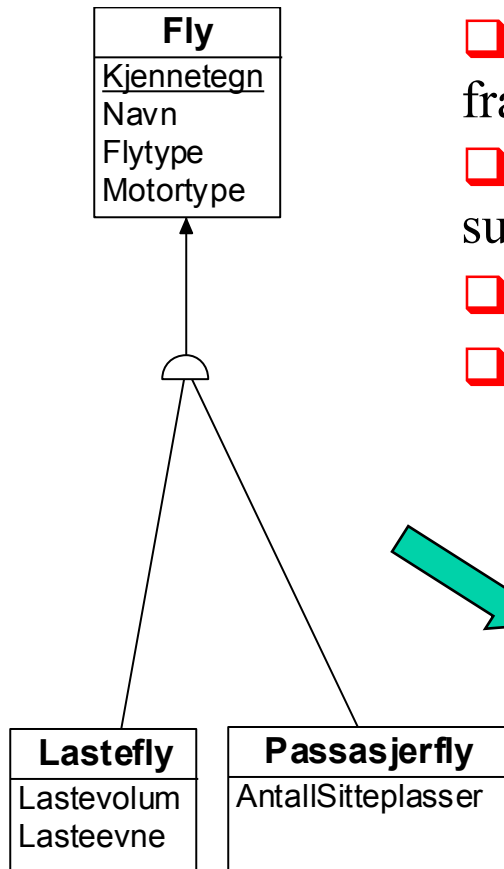


- ❑ Lager tabell kun for supertypen.
- ❑ Tabellen for supertypen får samtlige attributter fra subtypene + et «subtype»-attributt.
- ❑ Kan gi mange nullmerker.



Fly(Kjennetegn, Navn, Flytype, Motortype,  
LastEllerPassasjer, Lastevolum,  
Lasteevne, AntallSitteplasser)

# Subtyper III: Partisjonering

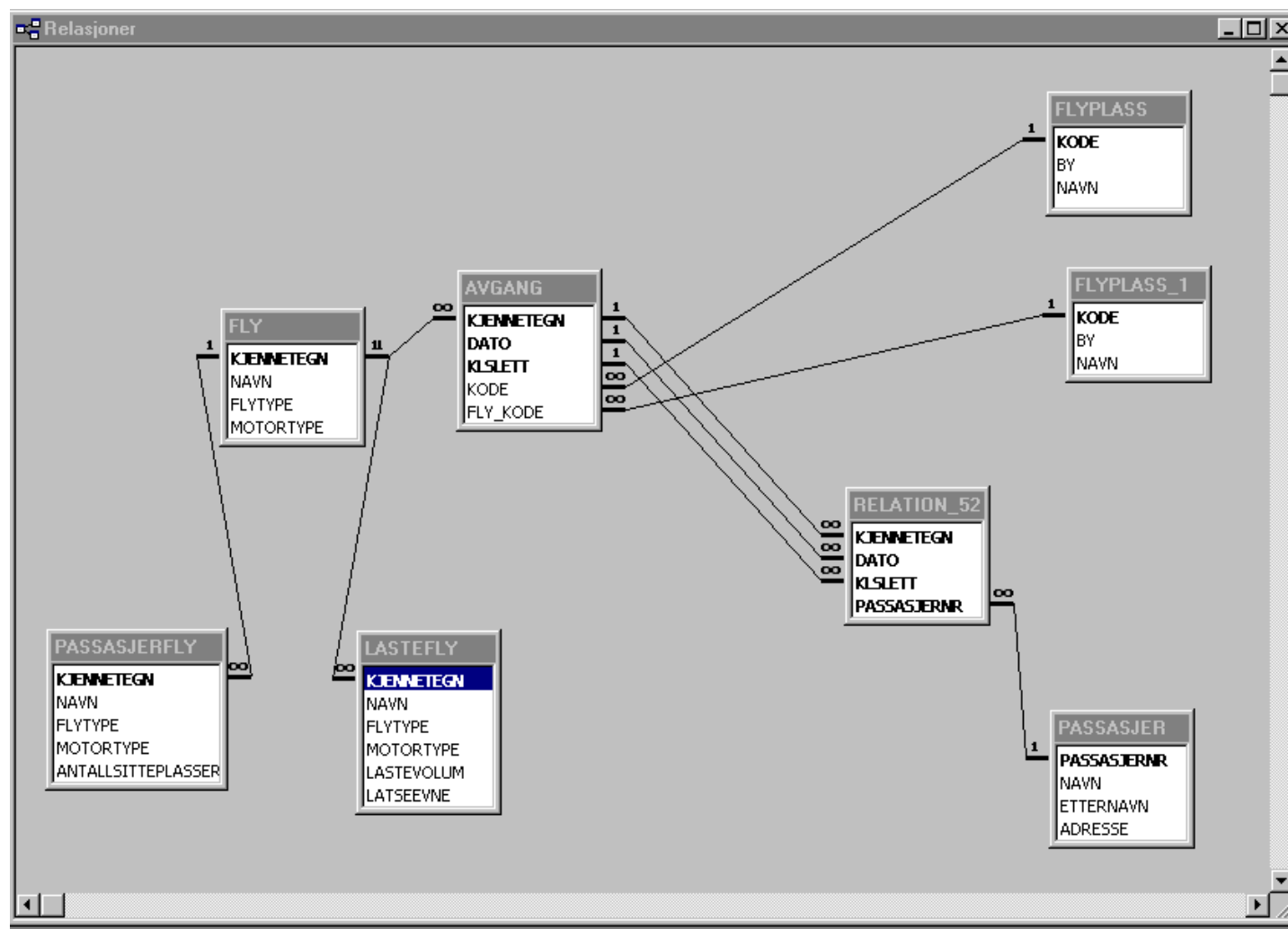


- ☐ Lager tabeller kun for subtypene.
- ☐ Tabellene for subtypene arver samtlige attributter fra supertypen.
- ☐ For å lage en liste over samtlige forekomster i supertypen må vi koble tabeller.
- ☐ Kan være problematisk å håndtere løpenumre.
- ☐ Alle må tilhøre en subtype!

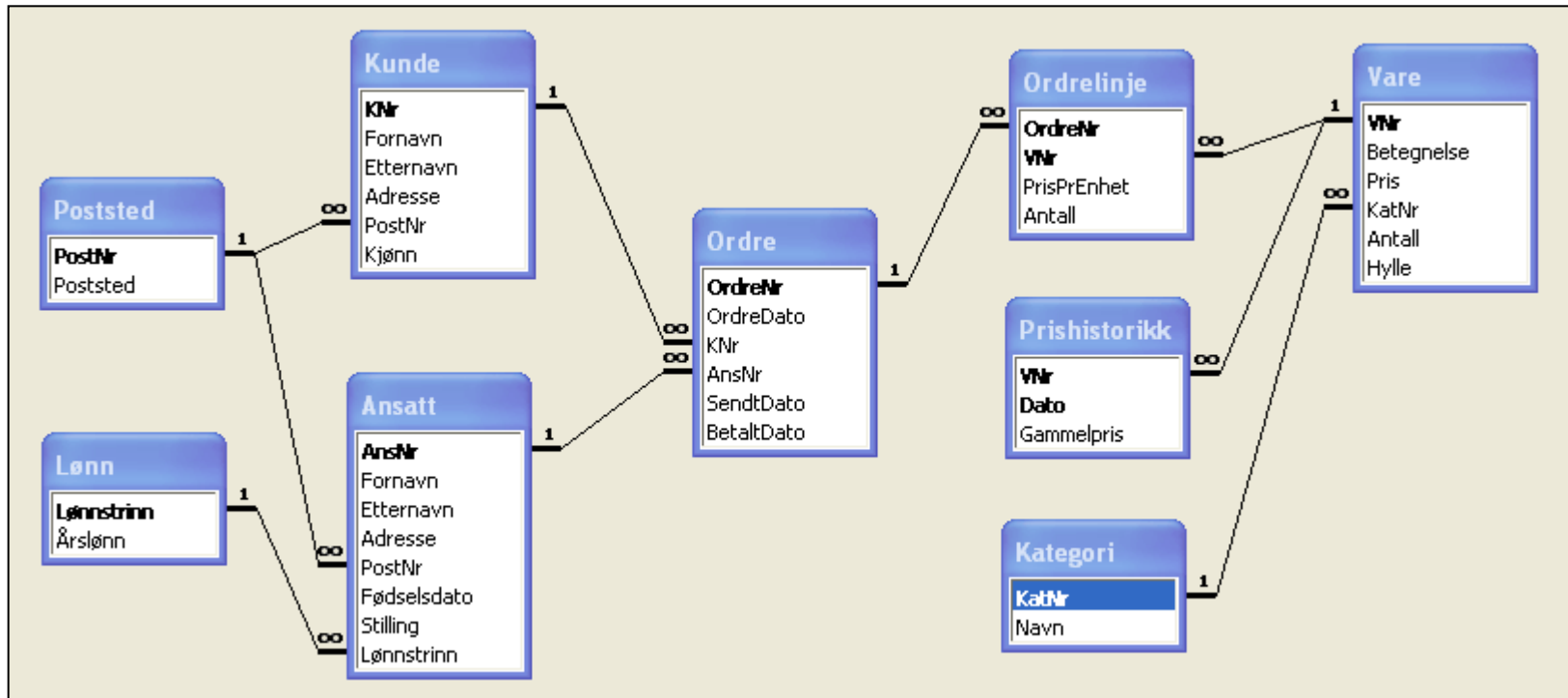
**Lastefly**(Kjennetegn\*, Navn, Flytype, Motortype, Lastevolum, Lasteevne)

**Passasjerfly**(Kjennetegn\*, Navn, Flytype, Motortype, AntallSitteplasser)

# Logisk skjema for Fly-databasen



# Her er logisk skjema, lag ER-modell





# Utsnitt (views)

- ❑ Et utsnitt (view) er en «virtuell» tabell.
- ❑ Anvendelser:
  - Sikkerhet
  - Presentasjon
  - Representasjonsuavhengighet
- ❑ Støtte for utsnitt:
  - Utsnitt er definert i SQL-92.
  - Oracle støtter utsnitt.
  - Vanlig oppsett av Access støtter ikke utsnitt.
- ❑ Access-spørringer og utsnitt er ikke helt det samme!

# Generelt om utsnitt

- ❑ Et utsnitt blir definert slik:

```
CREATE VIEW <Utsnittsnavn> ( <kolonner> ) AS  
  <Utvalgsspørring uten ORDER BY>
```

- ❑ Utsnitt skal kunne betraktes som tabeller og tabeller er ikke sortert. Et utsnitt er en «brille» mot tabellene.
- ❑ Et utsnitt er ikke en kopi! Oppdatering i et utsnitt medfører oppdatering i den underliggende tabellen.
  - Ikke alle utsnitt er oppdaterbare !
- ❑ Et utsnitt blir representert ved sin definerende spørring.
  - DBHS lagrer definisjonen av utsnitt i en systemtabell.

# Eksempel

- ❑ Et utsnitt som viser noen egenskaper om noen varer (vi bruker altså projeksjon og seleksjon) :

```
CREATE VIEW Keramikk( Varekode, Navn ) AS
SELECT VNr, Varenavn
FROM Vare
WHERE KategoriNr = 5
```

- ❑ Hvis vi sløyfer kolonnenavnene etter utsnittsnavnet "arver" utsnittet kolonnenavn fra spørringen.
- ❑ Sikkerhet: Vi kan gi brukere innsyn i utsnittet Keramikk, men ikke i hele Vare-tabellen.
- ❑ Utsnitt gjør sikkerhetssystemet mer "finmasket".

# Utsnitt fra flere tabeller

- ❑ Utsnitt som grunnlag for salgsrapporter:

```
CREATE VIEW salg AS
  SELECT OL.*, V.Varenavn, K.navn,
         O.Ordredato, O.AnsattNr, O.KundeNr
  FROM Ordre AS O, Ordrelinje AS OL,
       Vare AS V, Kategori AS K
 WHERE OL.OrdreNr = O.OrdreNr
        AND OL.VNr = V.VNr
        AND V.KatNr = K.KatNr
```

- ❑ Presentasjon: Brukere av utsnittet trenger ikke å koble tabeller ! Vi presenterer en "forenklet" database til brukerne.

# Spørringer mot utsnitt

- ❑ Salg siste måned fordelt på varer:

```
SELECT VNr, SUM(Antall*Pris) AS Totalt  
FROM Salg  
WHERE Year( Ordredato )= Year( Date() )  
      AND Month( Ordredato ) = Month( Date() )  
GROUP BY VNr
```

- ❑ Spørringen over blir utført ved å «sette inn» definisjonen av utsnittet **Salg** i spørringen.

- Hvordan ser resultatspørringen ut ?
- Hvorfor er denne teknikken effektiv ?
- Kan du tenke deg en alternativ framgangsmåte ?

# Oppdatering "i" utsnitt

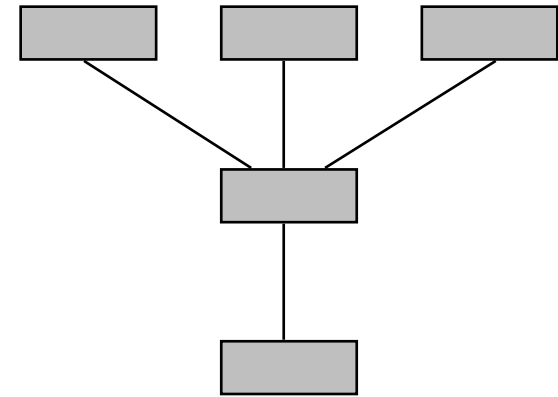
- ❑ Antall varer pr. kategori:

```
CREATE VIEW AntallVarerPrKategori AS  
SELECT KatNr, COUNT(*) AS AntallVarer  
FROM Vare  
GROUP BY KatNr
```

- ❑ Vi kan skrive INSERT, UPDATE, DELETE mot (noen) utsnitt.
- ❑ Hva om vi endrer antall varer i kategorien meierivarer fra 5 til 2? Hva skal effekten for den underliggende tabellen være?
- ❑ Hvilke utsnitt er det mulig å oppdatere? Seleksjoner? Projeksjoner? Grupperinger? Koblinger?
- ❑ SQL-92 har regler for hvilke utsnitt som er oppdaterbare.

# ANSI/SPARC 3-skjema arkitektur

- ❑ En database skal beskrives på tre abstraksjonsnivåer:
  - Eksterne skjema: En brukers syn på databasen. Består ofte av utsnitt.
  - Begrepsmessig skjema: Virksomhetens syn på data. Beskriver hva slags data som er lagret.
  - Internt skjema: Beskriver hvordan data er lagret (for eksempel hvilke indekser som er definert).

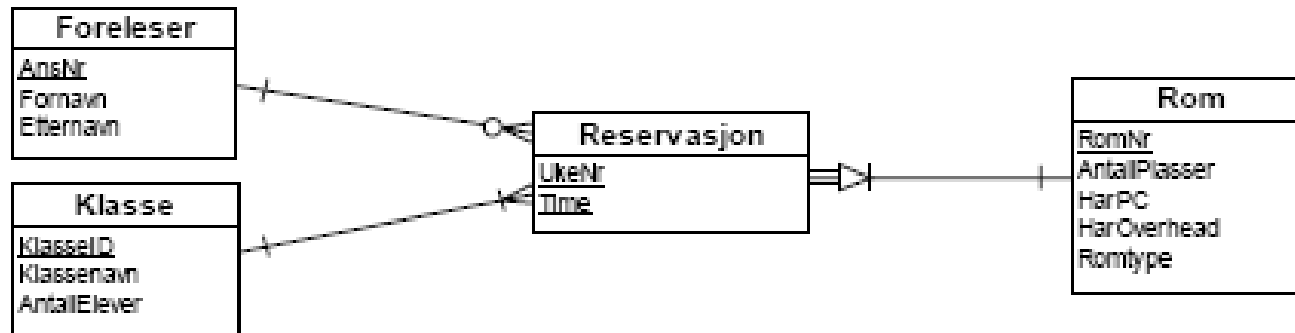


Uavhengighet av  
representasjon:

Må vi skrive om  
spørringer ved  
endringer i  
tabellstrukturen ?

# Oppgave

2. E/R-diagrammet under beskriver et romreservasjonssystem.Attributtene HarPC og HarOverhead forteller om et rom har PC med framviser og prosjektør for lysark (overheads). Attributtet Romtype er enten «aud» for auditorium, eller «flatt» for rom med flatt gulv. Oversett diagrammet til logisk tabellstruktur.





# Løsning

- ❑ Romreservasjonssystem.
- ❑ Foreleser(AnsNr, Fornavn, Etternavn)
- ❑ Klasse(KlasseID, Klassenavn, AntallElever)
- ❑ Rom(RomNr, AntallPlasser, HarPC, HarOverhead, Romtype)
- ❑ Reservasjon(RomNr\*, UkeNr, Time, AnsNr\*, KlasseID\*)

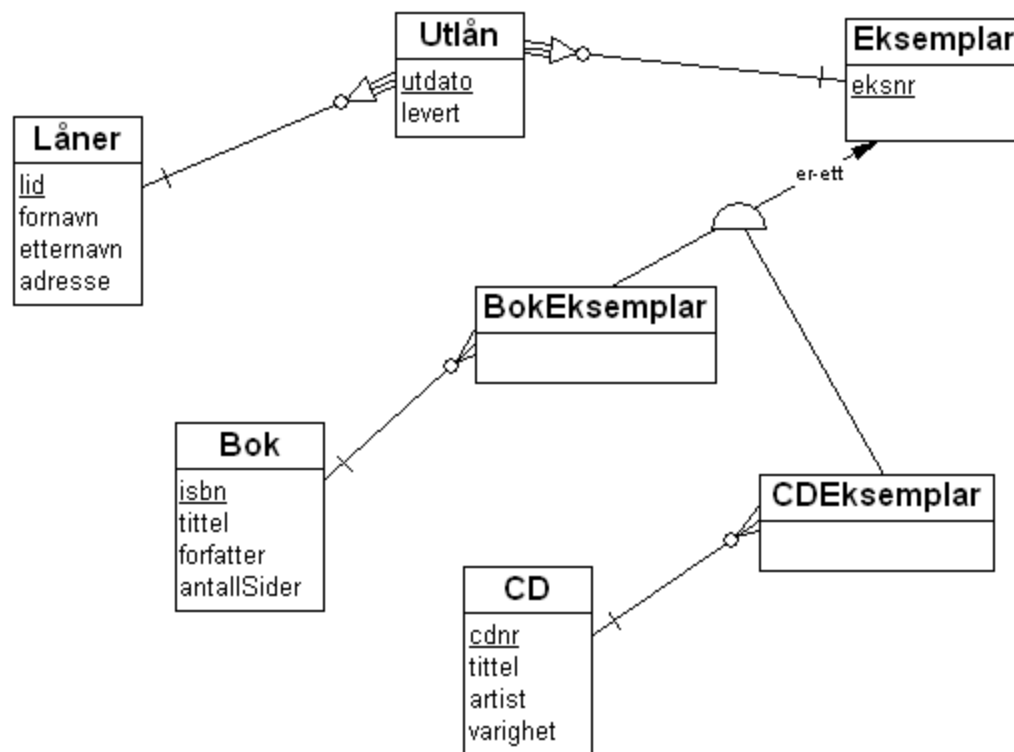
# Oppgave

3. Betrakt følgende database:

- Låner(LID, Fornavn, Etternavn, Adresse)
- Utlån(LID, EksNr, Utdato, Levert)
- Bokeksemplar(EksNr, ISBN\*)
- CDEksemplar(EksNr, CDNr\*)
- Bok(ISBN, Tittel, Forfatter, AntallSider)
- CD(CDNr, Tittel, Artist, Varighet)

Foreslå et E/R-diagram som beskriver databasen. Dette innebærer å lage en begrepsmessig modell ved såkalt «reverse engineering». Gjør nødvendige forutsetninger hvis noe er uklart.

# Løsning



# Oppgave

4. Lag et støttesystem for nettbasert undervisning. Det skal være mulig å opprette virtuelle klasserom. Hvert klasserom har en kode og et navn, og en eller flere lærere. Brukere av systemet er studenter og lærere. Brukere deles inn i grupper. En gruppe kan gis adgang («nøkkel») til ett eller flere klasserom. I et klasserom kan studentene lese beskjeder fra læreren, og både lærere og studenter kan delta i diskusjonsgrupper. En diskusjonsgruppe består av en eller flere diskusjonstråder. Hver tråd starter med et innlegg. Til et innlegg kan det komme flere svarinnlegg, som det igjen kan komme svar på. Lag en begrepsmessig datamodell for systemet og oversett deretter til logisk tabellstruktur. Både beskjeder og innlegg har en avsender, en dato, en overskrift og et innhold (tekst).