

Datamodellering med E/R

- ☐ Fasene i systemutvikling og databasedesign
- ☐ E/R (Entity/Relationship)
- ☐ Entitet
- ☐ Attributt
- ☐ Identifikator
- ☐ Forhold og roller
- ☐ Kardinaliteter: 1:1, 1:M, M:N
- ☐ Oppløsning av mange-til-mange (M:N) forhold
- ☐ Svake entiteter
- ☐ Egenforhold
- ☐ Subtyper

Pensum: Kapittel 7

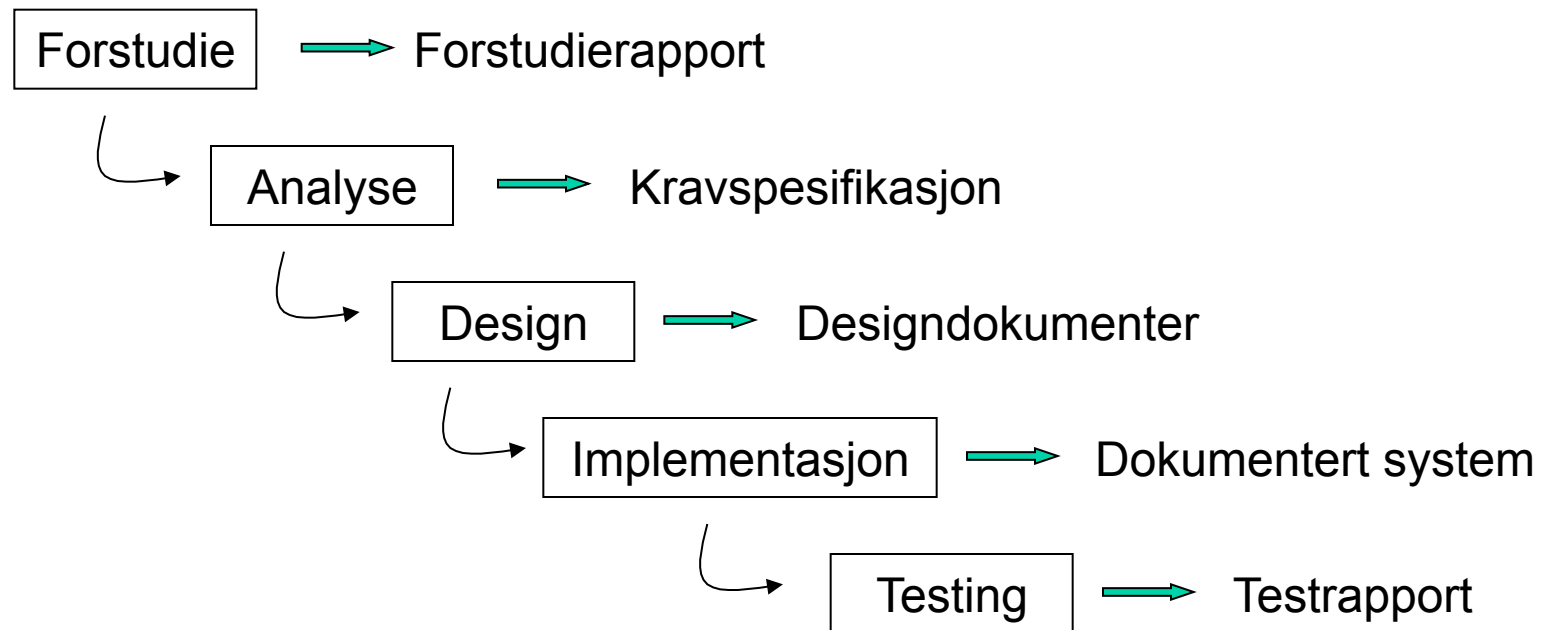
Modeller og perspektiver

- ❑ Når man skal konstruere noe som er komplisert er det nyttig å lage en modell først.
 - Hus-eksempel: Arkitekttegning, 3D-modell
- ❑ En modell er en representasjon av noe, der visse egenskaper, som er viktige for det formål representasjonen skal brukes til, er fremhevet, mens øvrige egenskaper utelates.
- ❑ Perspektiver:
 - Prosessorientert: Hva skal systemet gjøre?
 - Informasjonsorientert: Hva må systemet vite?
 - Objektorientert: Et system består av kommuniserende objekter.

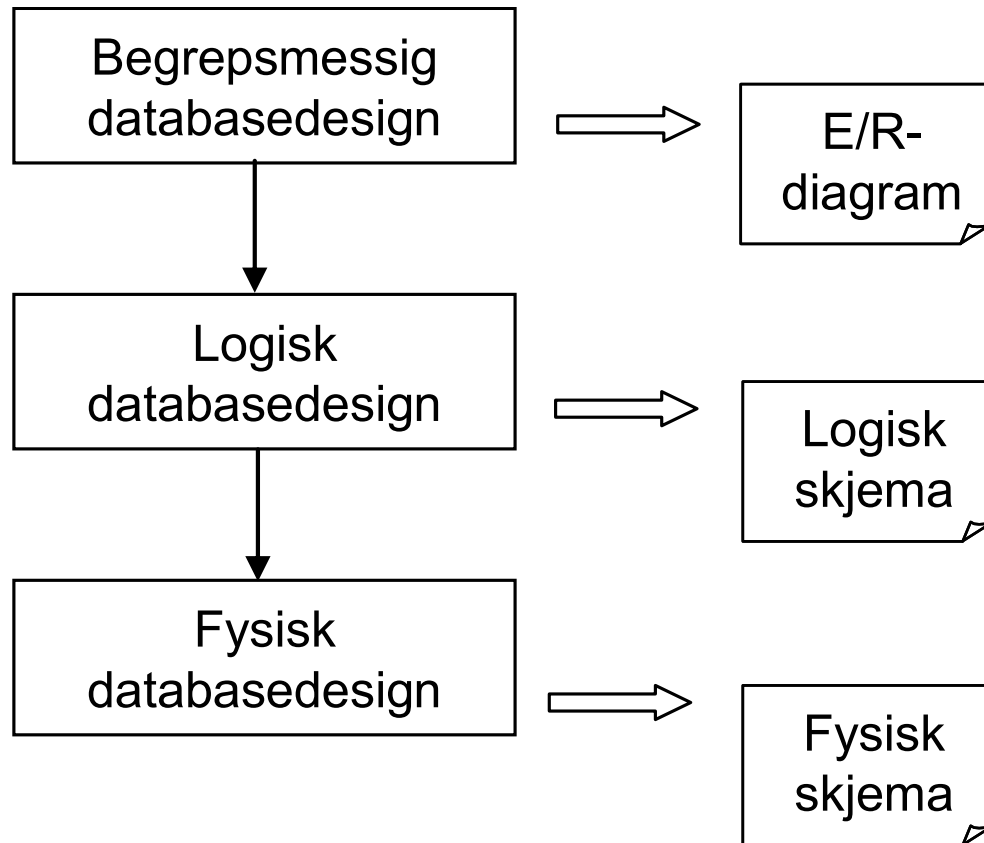
Livsløpet til et databasesystem

- ❑ Et databasesystem inngår som regel i et større (edb-basert) informasjonssystem.
- ❑ Systemutvikling gjøres i flere faser.
 - Forstudie: Skal vi starte utvikling?
 - Analyse: Hva skal systemet gjøre?
 - Design: Hvordan skal systemet bygges opp?
 - Implementasjon: Lag (programmer) systemet!
 - Test
 - Drift
 - Videreutvikling / avvikling

Systemutvikling etter fossefallsmodellen



Databasedesign



Fasene i databasedesign

❑ Begrepsmessig databasedesign

- Gir begrepsmessig skjema, dvs. en teknologiuavhengig beskrivelse av hva databasen skal inneholde.
- E/R (Entity/Relationship) mye brukt som "språk".

❑ Logisk databasedesign

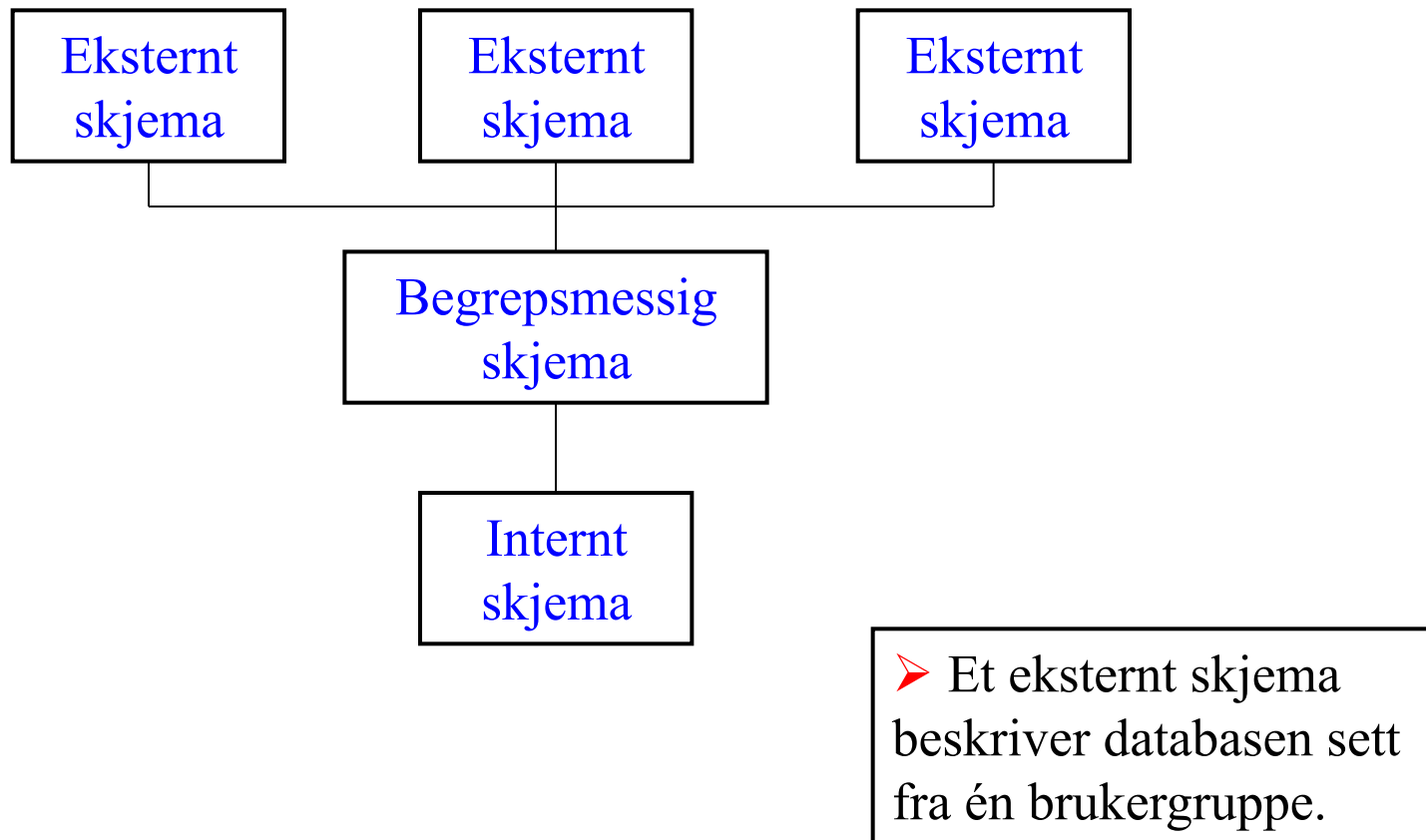
- Gir logisk skjema, dvs. tabellstruktur uavhengig av konkret DBMS.
- E/R-diagrammer kan oversettes til logisk skjema (delvis automatisk av modelleringsverktøyet).

❑ Fysisk databasedesign

- Gir fysisk skjema, dvs. tabellstruktur tilpasset et konkret DBMS med beskrivelse av lagringsstrukturer og aksesteknikker.

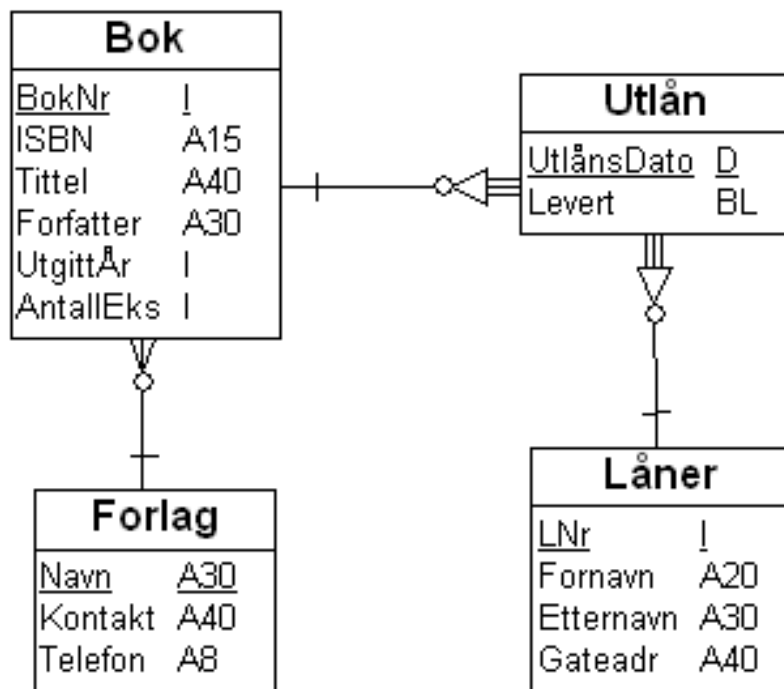
ANSI/SPARC 3-skjema arkitektur

- ❑ Et databasesystem bør/kan bli beskrevet på flere abstraksjonsnivåer.



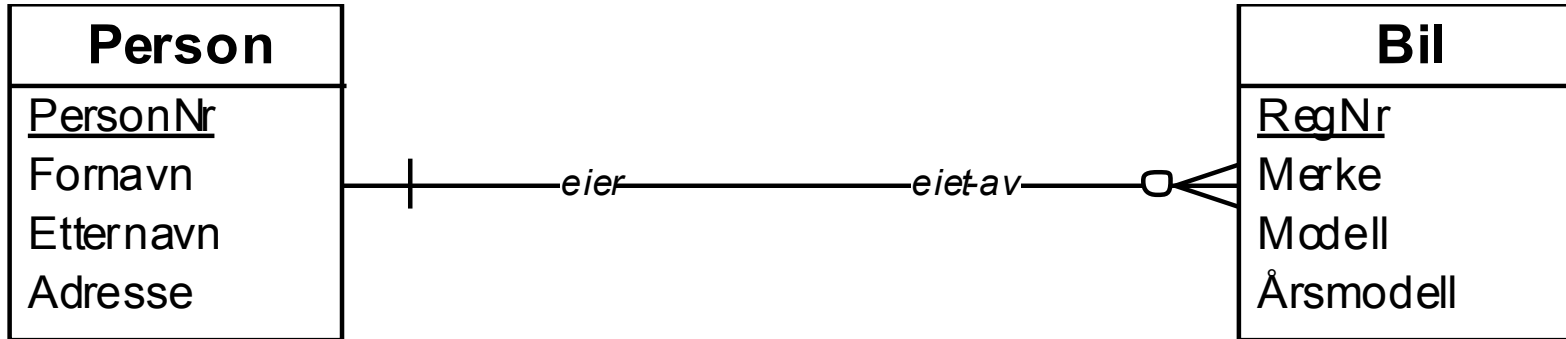
Introduksjon til E/R-diagrammer

- ❑ E/R-diagrammer er en visuell beskrivelse av strukturen til en database.
 - Hver entitet (boksene) svarer til en tabell.
 - Attributter (i nederste del av boksene) svarer til kolonner.
 - Identifikatorer er understreket og "svarer til" primærnøkler.
 - Forhold (linjene) "svarer til" fremmednøkler.



- E/R-diagrammer brukes når vi planlegger hva databasen skal inneholde.
- Gir god oversikt.
- Lesbare også for de som ikke er IT-eksperter.

Grunnleggende begreper i E/R

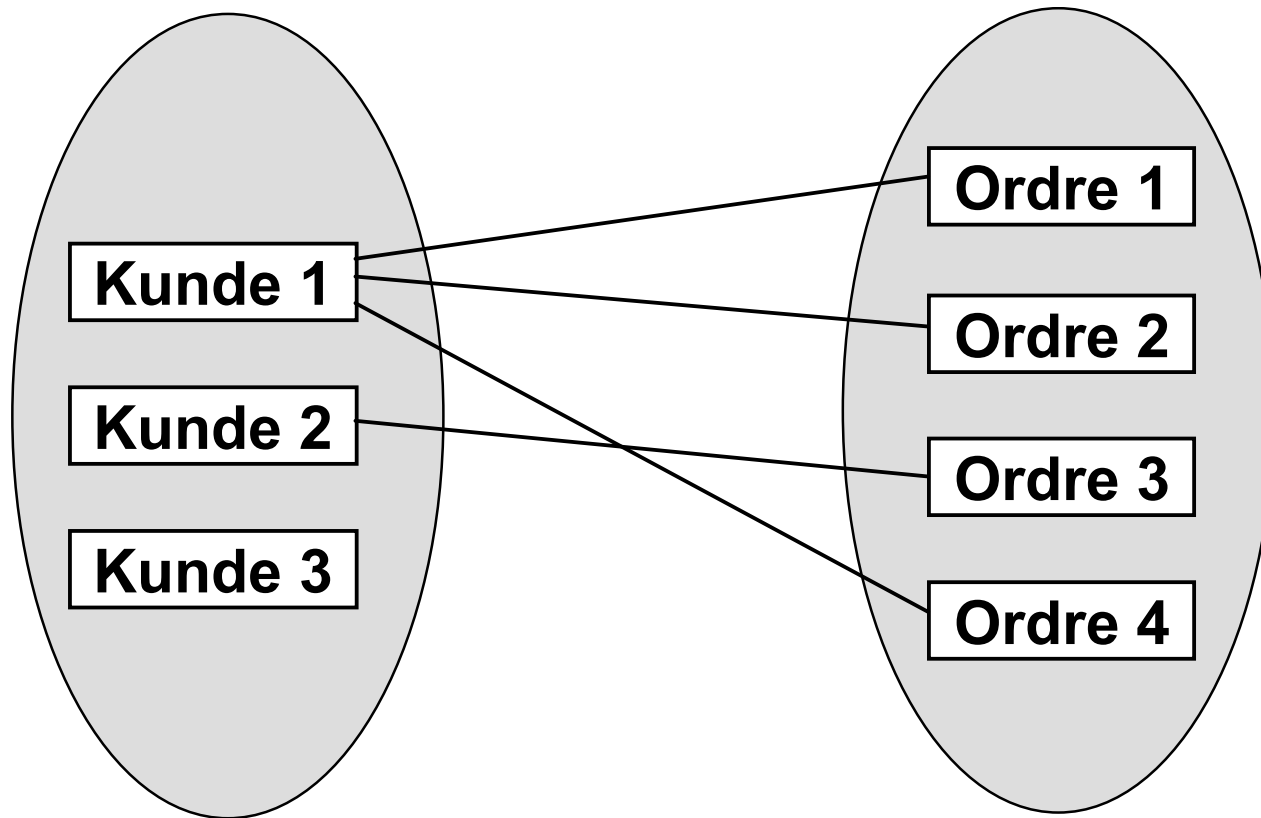


- ☐ Person og Bil er entiteter.
- ☐ Fornavn er et attributt til Person.
- ☐ RegNr er identifikator i Bil.
- ☐ En person kan spille rollen som eier i forholdet mellom Person og Bil.

Type og forekomst

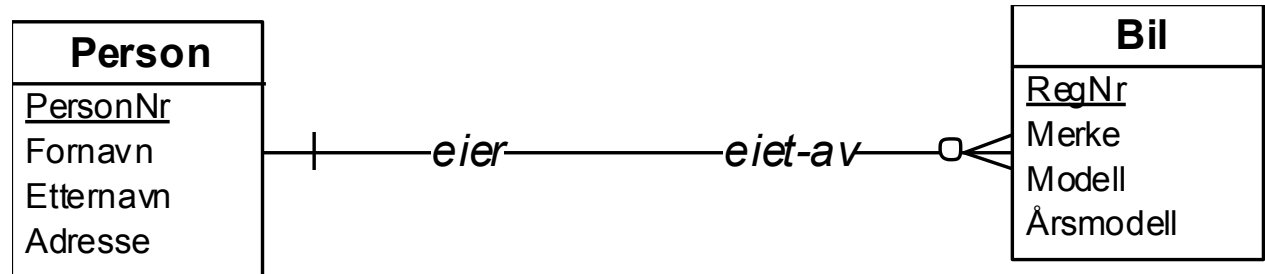
- ❑ Tabellen Ansatt inneholder mange rader.
- ❑ Hver rad beskriver én ansatt.
- ❑ En entitetsforekomst svarer til en rad i en tabell.
 - Eksempel: (AnsattNr=1, Fornavn="Hans", Etternavn="Hansen", Adresse="Hansegata 3")
- ❑ Entitetstypen representerer mengden av forekomster (og svarer dermed til hele tabellen).
- ❑ Tilsvarende skiller vi mellom type og forekomst av forhold.
 - Forekomst: Eierforholdet mellom Ola og hans Ascona.
 - Type: Samlingen av alle eierforhold.

Forholdstype og forholdsforekomster

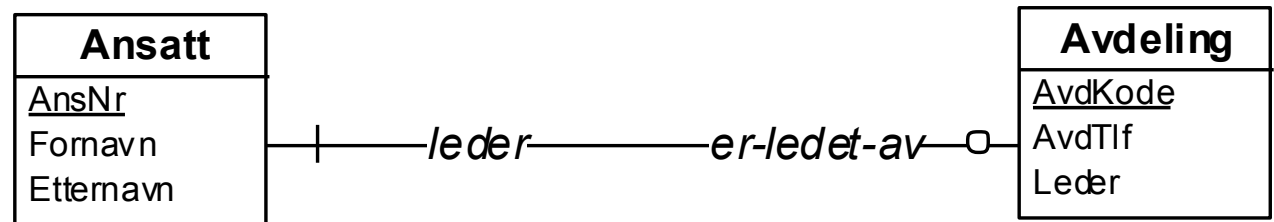


Kardinalitet

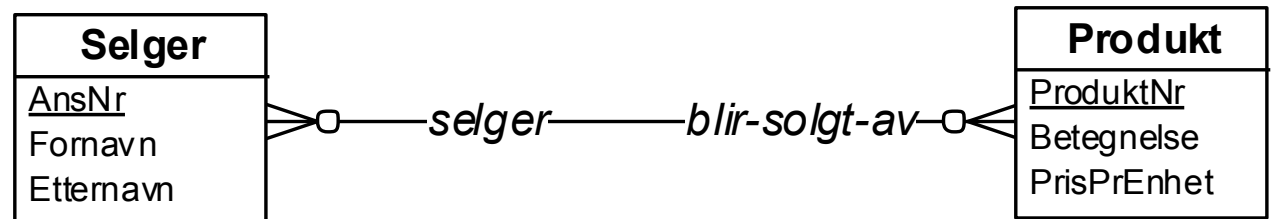
1:N
(en-til-mange)



1:1
(en-til-en)



M:N
(mange-til-mange)



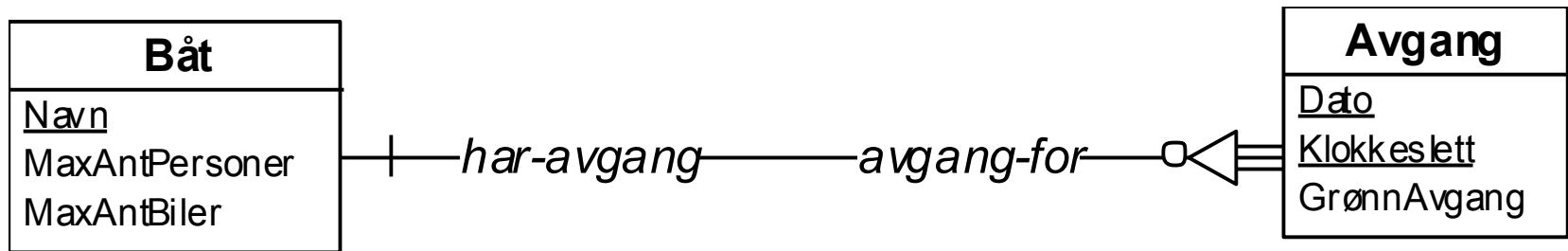
Minimums- og maksimumskardinalitet

- ❑ En gitt person kan eie minimum 0 og maksimum mange biler.
 - Symbol nærmest entitetene angir maksimumskardinalitet.
 - Kråkefot betyr "mange", mens | betyr 1.
 - Innerste symbol angir minimumskardinalitet.
 - Sirkel betyr 0 og | betyr 1.

- ❑ Forenklet notasjon (brukes i PowerDesigner):
 - En sirkel alene betyr maksimum=minimum=0.
 - En | alene betyr maksimum=minimum=1.

- ❑ Generelt er det mulig å bruke andre "tall" enn 0, 1 og "mange".
 - Eksempel: 3..7
 - Forskjellen på 0, 1 og "mange" avgjør tabellstruktur.

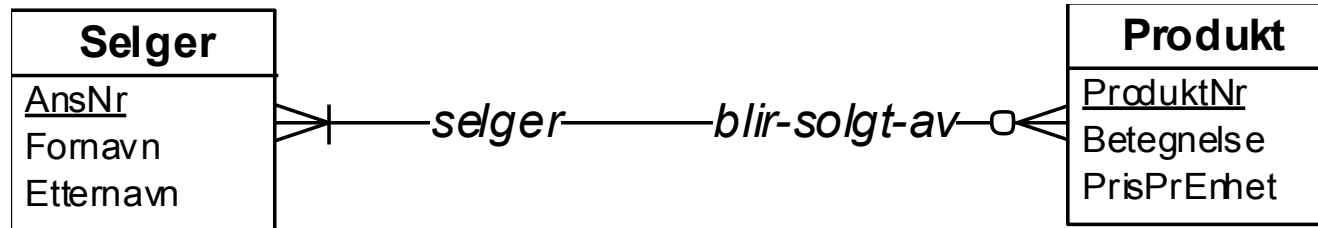
Svake entiteter



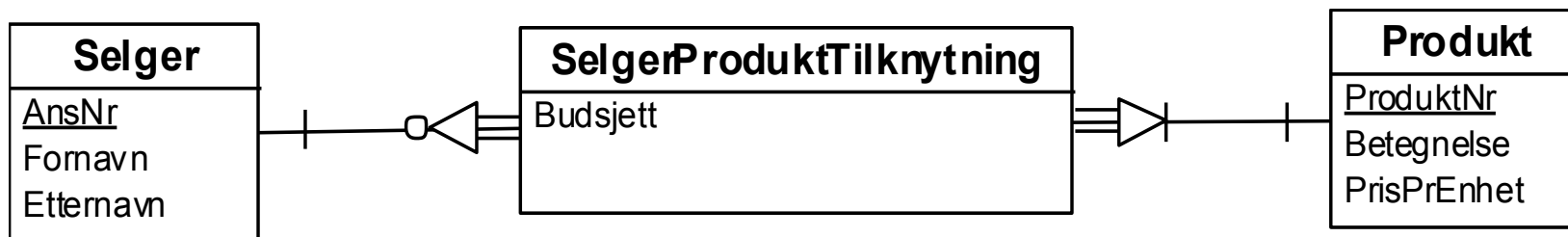
❑ En svak entitet arver deler av sin identifikator fra en annen og kan ikke eksistere uten denne.

- En Avgang kan ikke eksistere uten en tilhørende Båt.
- Identifikatoren for Avgang er (delvis) avledet fra identifikatoren for Båt.

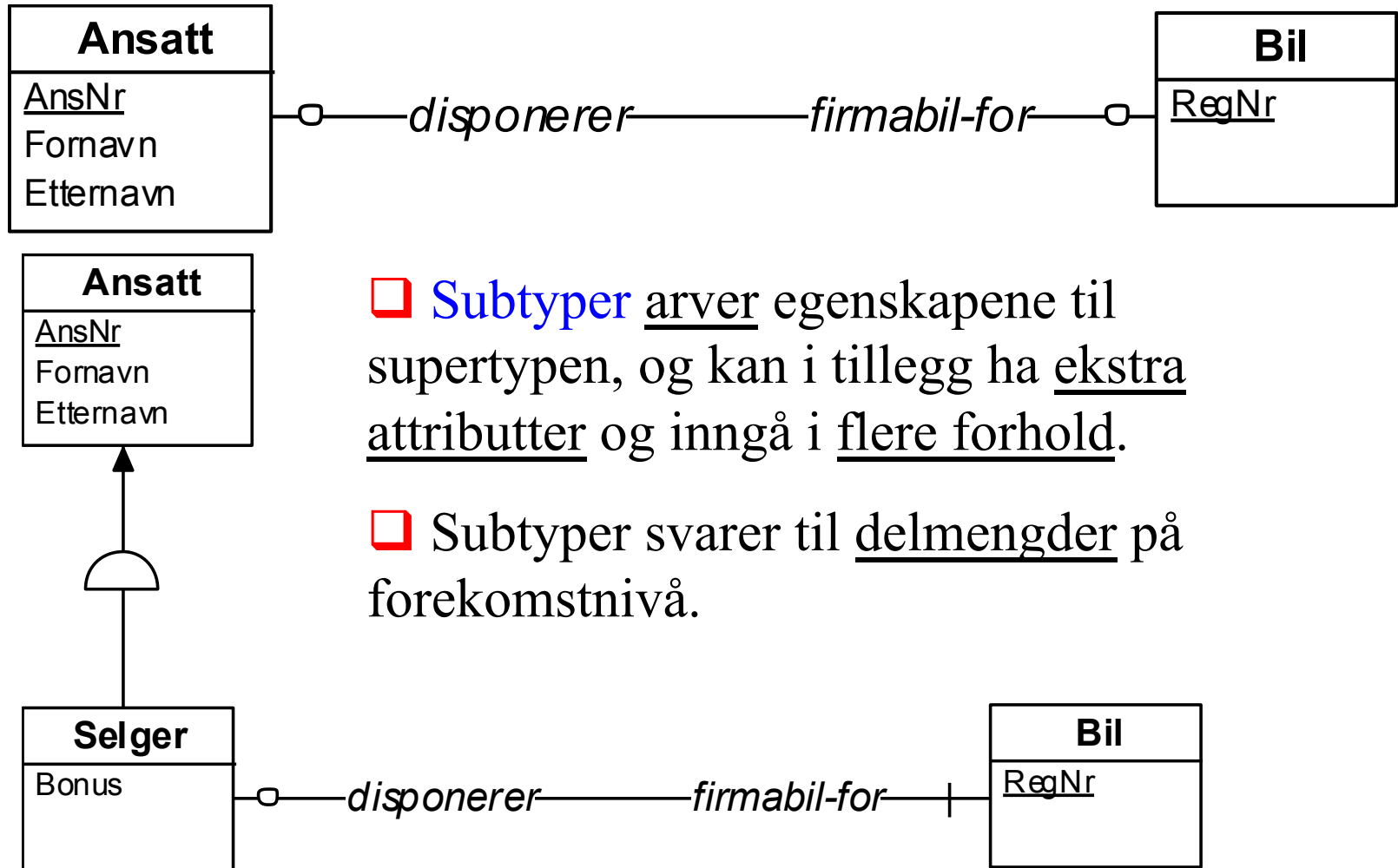
Oppløsning av mange-til-mange forhold



Er det interessante
egenskaper ved forholdet?



Subtyper gir spesialisering



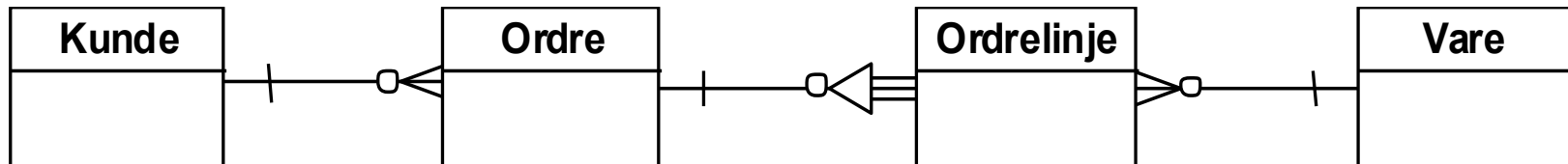
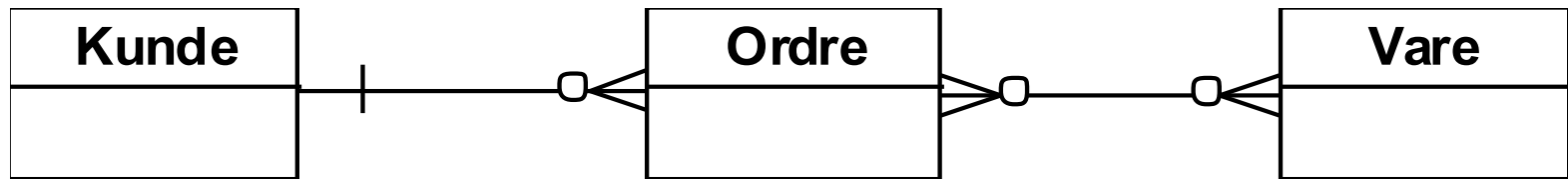
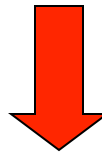
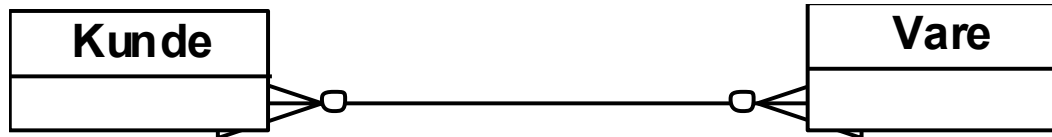
❑ Subtyper arver egenskapene til supertypen, og kan i tillegg ha ekstra attributter og inngå i flere forhold.

❑ Subtyper svarer til delmengder på forekomstnivå.

Modelleringsklisjeer

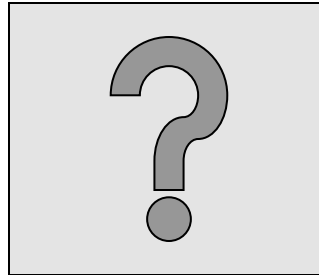
- ❑ Mange datamodelleringsproblemer er eksempler på noen få "klassiske problemer".
- ❑ Eksempler:
 - " Et papirskjema er bygget opp fra et hode og et antall linjer.
 - Skal man kun lagre nå-situasjonen, eller også historikken?

Hode/linje-mønsteret (master/detail)



Skjema eller forekomst?

Avdeling
<u>Kode</u>
Kvartal1
Kvartal2
Kvartal3
Kvartal4



Avdeling
<u>Kode</u>
<u>Kvartal</u>
Omsetning

☐ Kvartalsvis omsetning pr. avdeling kan lagres i 4 attributter. Til høyre blir kvartalsnummer lagret som data (fra skjema til forekomst).

➤ Kan vi bruke samme identifikator?

☐ Hva om vi skal lagre omsetning pr. måned/dag?

Typer relasjoner

- **Indikerer antall assosierte entiteter**
- **Enkel – en entitet involvert**
 - Relasjon knyttet til en enkelt entitet
 - Rekursiv relasjon/egenrelasjon
 - Eksisterer mellom forekomster i samme entitet.
 - Eks. Et land grenser til 0 eller mange land

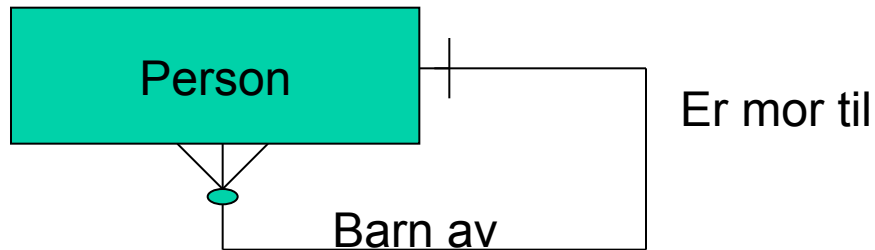
Binær relasjon –

- Relasjon mellom to entiteter
- **Tertiær relasjon**
 - Relasjon mellom 3 entiteter

Egenrelasjoner

- ☐ Tegn: En Person kan være mor til mange Personer
- ☐ Tegn Et land kan grense til mange land
- ☐ Hvilke tabeller vil dette gi?

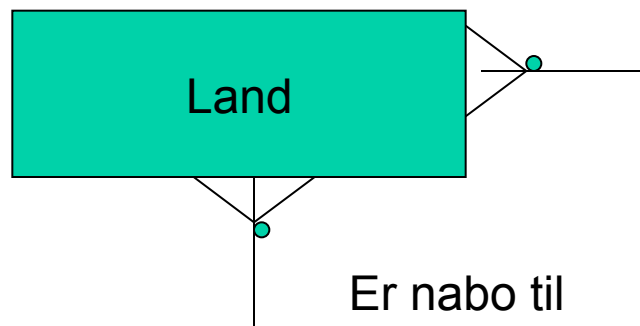
Eksempel på egenrelasjon



I persontabellen vil vi få et felt for
personnr til mor

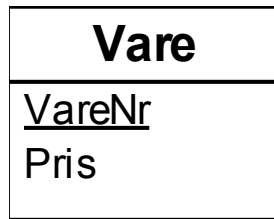
Eksempel på egenrelasjon

Her må vi løse opp mange-til-mange relasjonen og vi får en egen grensetabell med 2 felt for navn til land som har felles grense

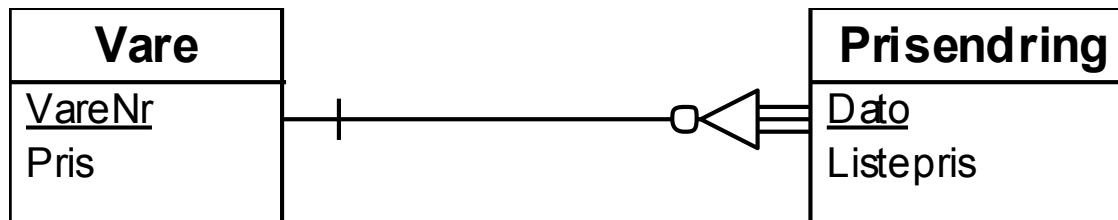


Grenser til

Tid og rom



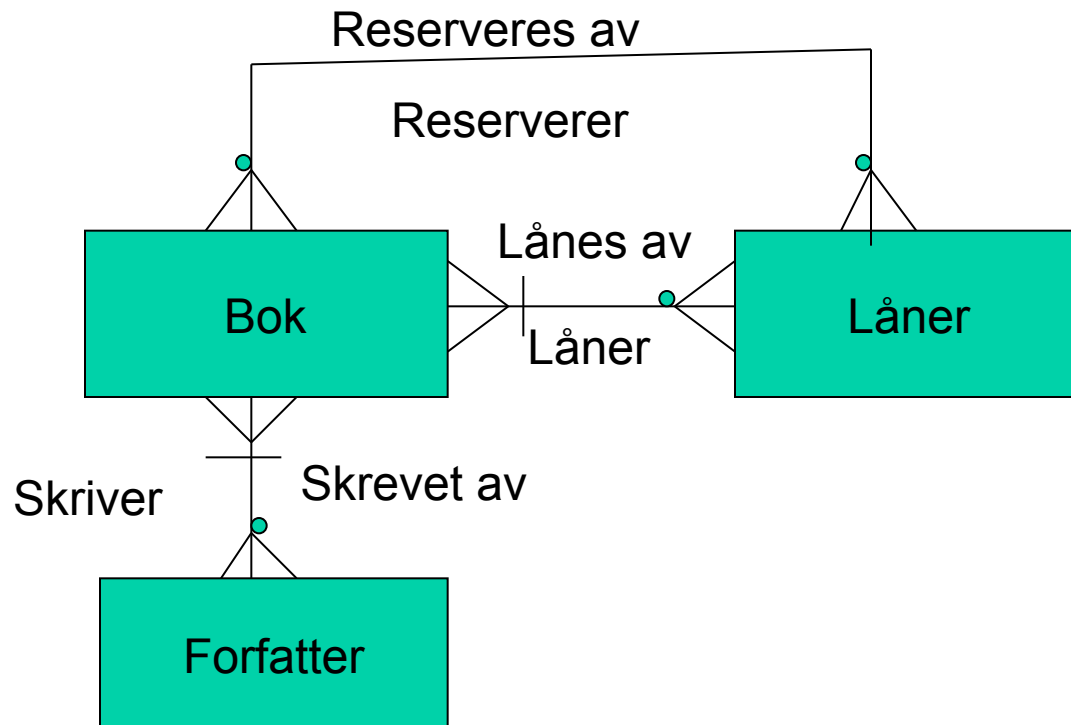
- ❑ Kun lagre nå-pris? Hva kostet varen tidligere?
 - Lagre prisendringer, eller pris for hver dag?
 - Håndtering av tidsperioder.
- ❑ Tidspunkt/sted som (del)identifikator i entiteter som representerer "hendelser" (vielse).



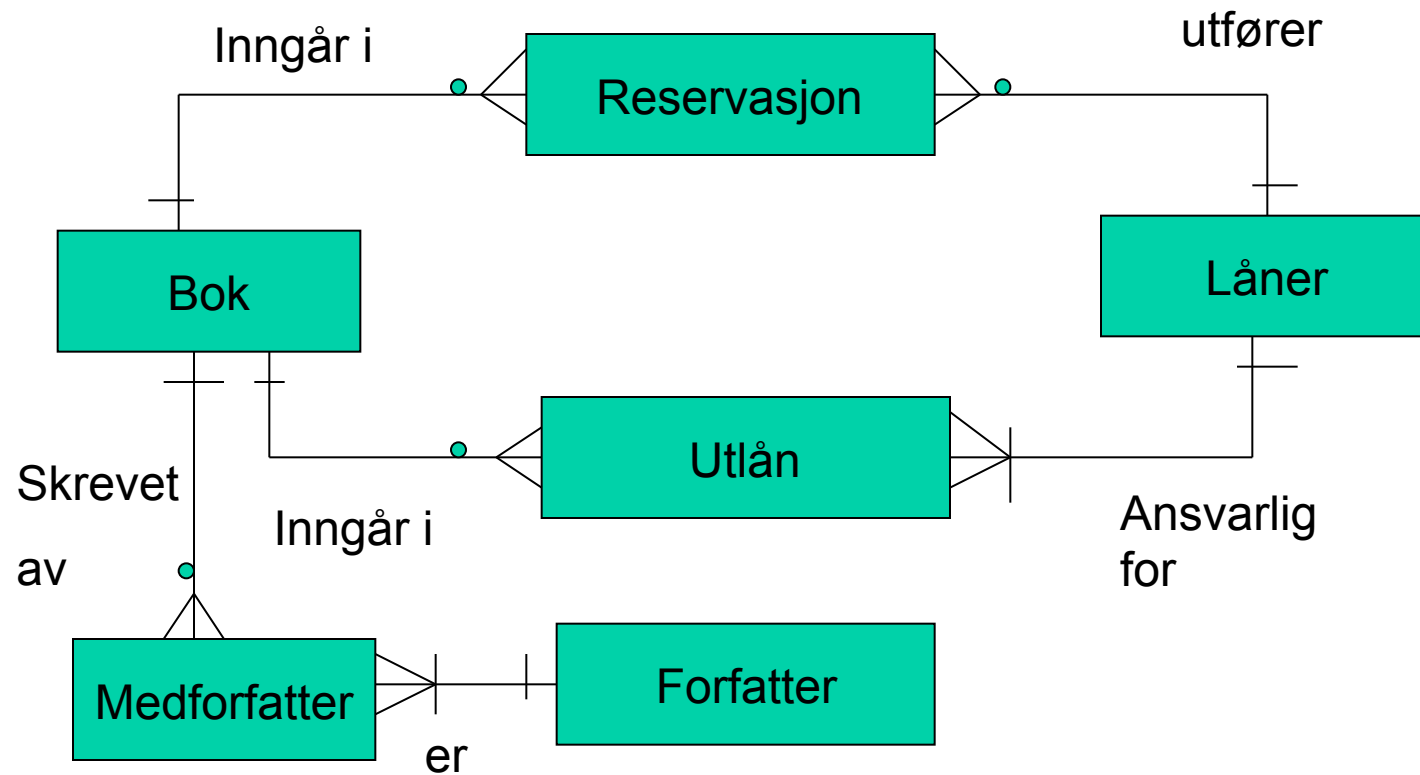
Et eksempel

- ❑ Det skal lages et bibliotekssystem som holder rede på hvilke bøker og utlån. Systemet skal også holde rede på hvilke forfattere som har skrevet de ulike bøkene. En låner kan reservere bøker. Når en reservert bok kommer inn skrives det ut et brev til kunden som har reservert boka. Det skal også kunne tas ut purringer dersom bøkene i leveres i tide.

ER-modell



Logisk skjema



Design

- ❑ Lager datadictionary/datakatalog
- ❑ Oversikt over alle entiteter, attributter, datatyper, domener, primær og sekundærnøkler

Bok:

ISBN-nummer PK, tekststreng 11 lang

Eksemplarnr PK, heltall >0

Tittel, tekststreng 40 tegn

Utgivelsesår, heltall 4 tegn >0

Resultatet:

- ☐ En fullstendig beskrivelse som kan gis til den som skal lage systemet
- ☐ Resultatet blir en relasjonsdatabase
- ☐ En relasjonsdatabase er tabeller som henger sammen ved at primærnøkler i en tabell lagres som sekundærnøkler i en annen tabell
- ☐ Nøkkelvandring ALLTID fra en til mange

OPPGAVE 1

- ❑ Studenter skal i løpet av studiet opp til flere eksamener. Hver eksamen består av én eller flere oppgaver. Om studenten lagres studentnummer, navn og adresse og om eksamen lagres informasjon om hvilket fag det gjelder og dato. Det lagres også informasjon om hvilke oppgaver som inngår i eksamen og oppgaveteksten til hver oppgave. Det skal også fremgå av modellen hvilke karakter studenten har fått i faget på en bestemt dato (studenten kan ha gått opp til eksamen flere ganger).