

Modeller for design av Web-Applikasjoner

→ **Kapittel 2: Data Modell**

→ **Kapittel 3: Hypertekst Modell**

Av Eskil Saatvedt og Arianna Kyriacou.

<http://www.ii.uib.no/~eskil/fag/>

<http://www.ii.uib.no/~arianna/fag/>

Kapittel 2:

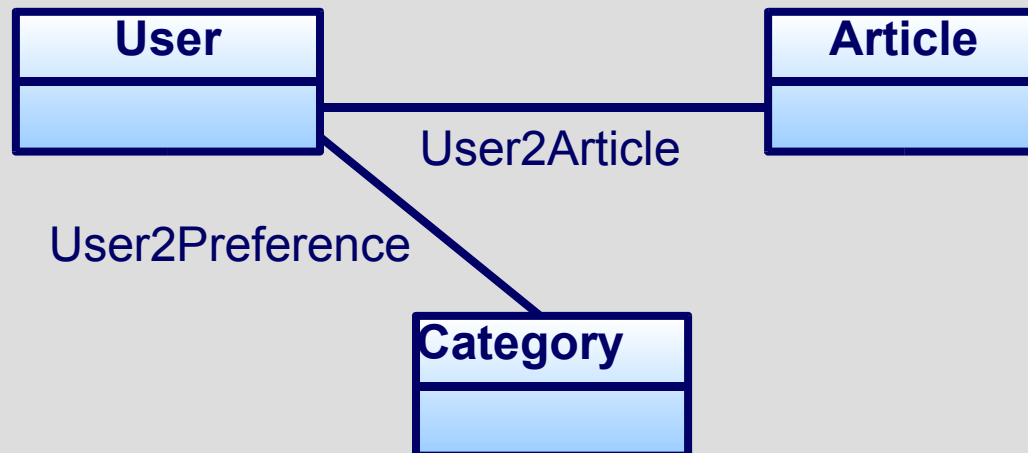
DATA MODELL

Innhold:

- Mål
- Resultat
- ER-Modellen
 - Basis Notasjon
 - Entiteter
 - Attributter
 - Attributt-typer
 - Identifikasjon og Primær Nøkkel
 - IS-A Hierarkier (Generaliserende Hierarkier)
 - Relasjoner
 - Binære Relasjoner og Relasjonsroller
 - Relasjonsroller og Kardinalitet
 - MultiVerdi-Attributter
 - Strukturerte Attributter
 - Relasjoner med Attributter
 - N'ære Relasjoner
 - UtleDET (Derivert) Informasjon

DATA MODELL

- **Mål:** Data som skal brukes av applikasjonen blir spesifisert på en formell og likevel intuitiv måte.
- **Resultat:** Vi får et konseptuelt skjema som på en lettfattelig måte viser tilgjengelig kunnskap om applikasjonsdataene.



DATA MODELL

•ER (Entity - Relationship) Modellen:

Essensielle ingredienser i ER-modellen:

- Entitet
- Attributt
- IS-A Hierarki
- Relasjon
- Relasjonsrolle

Essensielle ingredienser i ER-modellen:

Entitet(Sentralt konsept): Kontainer av strukturert data. En klasse av objekter i applikasjons-området som beskriver et sett elementer med lignende egenskaper. F.eks. klassene Lærer og Elev.

Attributt: Beskriver entiteter. Uttrykker en egenskap til en entitet. F.eks ForNavn og EtterNavn til en Lærer eller Elev.

Hvert attributt inneholder kun en enkelt verdi.

Essensielle ingredienser i ER-modellen:

IS-A Hierarki: Uttrykker en utledning av et spesifikt konsept fra et mer generelt. Brukes til å klassifisere og gruppere. (Som arv i Objektiv Programmering) F.eks. kan entitetene Lærer og Elev utledes fra den mer generelle entiteten Person.

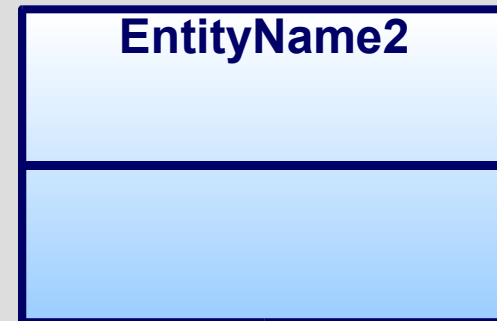
Essensielle ingredienser i ER-modellen:

Relasjon: Semantisk assosiasjon / forbindelse mellom entiteter. En beskrivelse av forholdet mellom to objekter. I forholdet mellom Lærer og elev kan relasjonen f.eks. få navnet «Undervisning».

Relasjonsrolle: Beskriver hvilken betydning forholdet har for den enkelte entiteten. For entiteten Elev kan f.eks. relasjonsrollen fra Elev til Lærer få navnet «Bli_undervist_av».

Basis Notasjon

Entiteter: Entiteter er firkantede figurer, merket med entitetsnavnet i øvre halvdel.



Basis Notasjon

Attributter(assosiert med entiteten): Attributter blir listet opp i nederste halvdel av entitets-firkanten.



Eksempler:



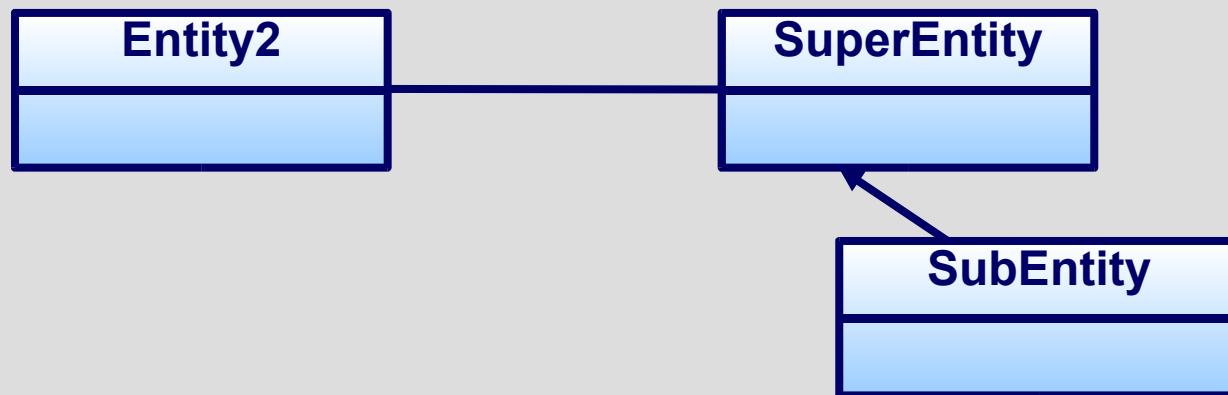
Basis Notasjon

Relasjon: Relasjoner representeres med en solid linje mellom entiteter.



Basis Notasjon

IS-A Hierarki: Notasjonen her er en pil som peker fra Sub-Entiteten (Mer spesifikt konsept) til Super-Entiteten (Generelt konsept).



PS:

En knagg å henge det på :

Sammenligner vi løst med objekt-orientert programmering, vil en entitet tilsvare en *klasse*, populasjonen til entiteten vil tilsvare *instanser* (opprettede objekter) av klassen, attributter tilsvare klassens *field*, primær-nøkkel vil tilsvare en *objektID*, deriverte attributter vil tilsvare klassens *funksjoner*, IS-A hierarkiet tilsvare *arv* (der multibel arv ikke er tillatt), en relasjon viser en *forbindelse* fra en klasse til en annen og kardinalitet for en relasjonsrolle vil løst tilsvare *hvor mange objekter av en annen klasse en klasse har lov til å opprette* .

Entiteter

Entiteter har en populasjon: et sett av objekter beskrevet av entiteten, dvs instansene til entiteten.

eks: populasjonen til Artist er et sett av artister.

Entiteter

Attributter(assosiert med entiteten):

- - de egenskapene til et objekt fra den virkelige verden som er relevante for hensikten til applikasjonen.

Eksempel: navn, adresse og telefonnummer til en person.

- Alle instanser av en entitet har samme sett med attributter.

- En attributt kan noengang ha en *null-verdi*, dvs at den spesifikke instansen ikke har noen verdi for attributten eller at verdien er ukjent.

Eksempel: en person uten telefon vil få en null-verdi på attributten telefonnummer.

Null-verdier kan gi ambiguitet-problemer. (Ikke entydig tolkning av data)

Entiteter

Attributt Typer:

- Attributter kan være typert, og det er god praksis å uttrykke attributt-typen i Data-Modellen.
- Eksempel på datatyper: String, Text, Integer, Float, Date, Time, Boolean, Enumeration, BLOB (Binary Large Object) og URL.
- Notasjon: **Attributtnavn:Datatype**
- Eksempler: **Fornavn:String**
 Tlf_Nummer:Integer
- Grafisk Eksempel:



Entiteter

Identifikasjon og Primær Nøkkel:

iii. Alle instanser av en entitet må kunne skilles fra hverandre med en **unik identifikator**; en **Primær Nøkkel**.

- En eller flere attributter kan bli definert som entitetens primærnøkkelen, f.eks. kan attributtene Fornavn og Etternavn sammen definere en unik identifikator for en person.
- Primærnøkkel-attributtene må tilfredstille to betingelser:
 - 1 . Den må være *ikke_null* og
 - 2 . den må være *unik* for hver instans som blir opprettet.

Entiteter

Identifikasjon og Primær Nøkkel :

- Det er god praksis å definere primærnøkkelen til en entitet ved å bruke en egen attributt spesielt beregnet på dette: en objekt-identifikator (OID).
- En OID har kun et formål: å tilegne hver instans av entiteten en unik identifikator.
- NB: Boken antar OID som implisitt definert for alle entiteter, og skriver ikke denne attributten opp i ER-Diagrammene. Det er likevel viktig å være klar over at den er der.

Entiteter

Identifikasjon og Primær Nøkkel:

- En kan i tillegg definere alternative nøkler: Disse nøkkelattributtene må også tilfredstille betingelsene om å være ikke_null og unik for hver instans av entiteten.

- Notasjon:

Attributtnavn nøkkelsymbol

- Eksempel:

Fornavn nøkkelsymbol

Etternavn nøkkelsymbol

Entiteter

IS-A Hierarkier:

- ER-modellen tillater designeren å organisere entiteter i et hierarki, der de deler felles egenskaper.
- Det grunnleggende(*basic*) generaliserte hierarki blir kalt IS-A hierarki. Et IS-A hierarki har en super-entitet og en eller flere sub-entiteter.
- Hierarkiet er ikke begrenset til to nivåer; en sub-entitet kan igjen spesialiseres i en eller flere sub-entiteter osv, så en kan designe så mange nivåer som en ønsker.

Entiteter

IS-A Hierarkier:

- En sub-entitet arver alle attributter og relasjoner som er definert i super-entiteten.
- En sub-entitet kan legge til lokalt definerte attributter og relasjoner.

Entiteter

IS-A Hierarkier:

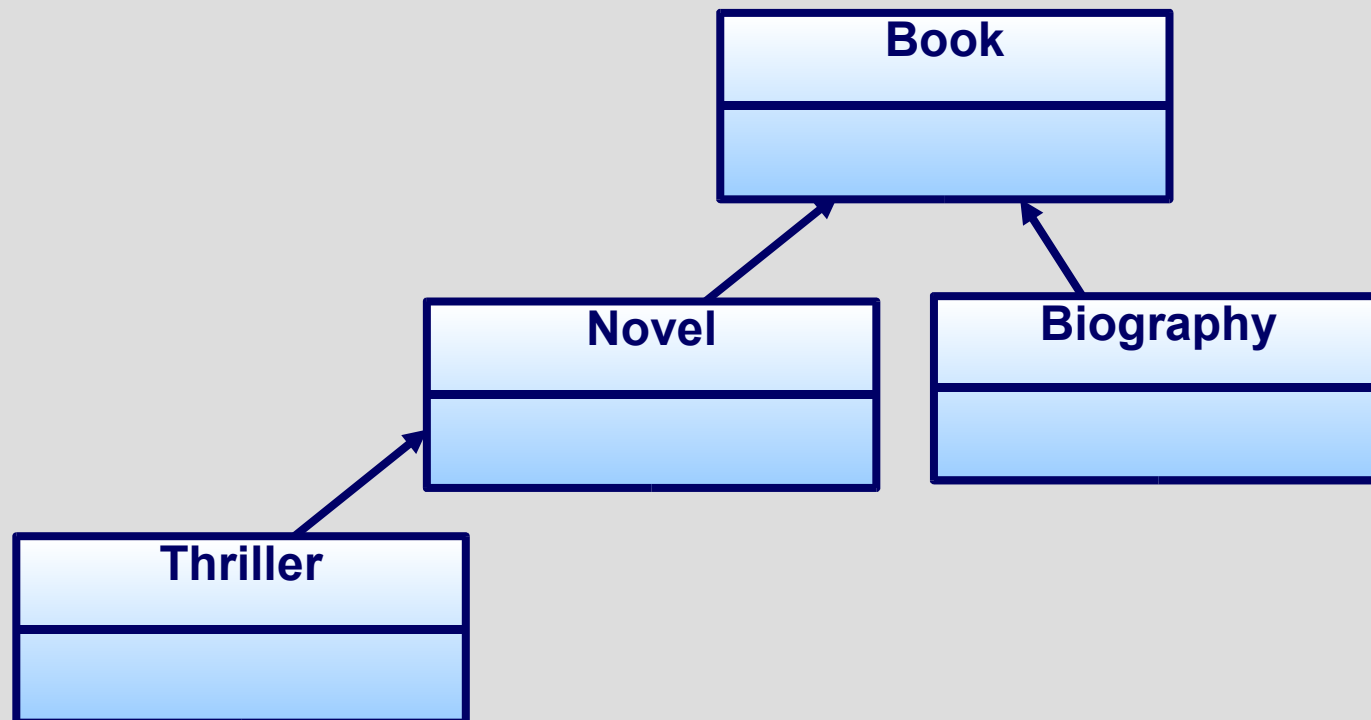
Følgende sikrer at et ER-skjema er lett å implementere ved å bruke vanlig database teknologi:

1. Hver entitet er definert som en spesialisering av max en super-entitet. En sub-entitet kan da arve fra kun en super-entitet; multippel arv unngåes.
2. Hver instans av en super-entitet blir spesialisert til kun en sub-entitet. Dvs. instansen kan ikke være to eller flere sub-entiteter samtidig.
3. Hver entitet opptrer i max et generaliserende hierarki.

Entitet

IS-A Hierarkier:

Eksempel på et IS-A Hierarki:



Relasjoner

- representerer semantiske forbindelser mellom entiteter.
En relasjon beskriver forholdet mellom entiteter.
- har et navn.
En relasjon blir gitt navn som assosieres med betydningen forbindelsen har.

Eksempel: Relasjonen mellom entitetene Artist og Album kan f.eks få navnet «Publikasjon».

Relasjoner

Binære Relasjoner og Relasjonsroller:

- Den enkleste formen for relasjon er den binære relasjonen. Denne forbinder to entiteter.
- N-ære relasjoner er tillatt men ikke anbefalt. Det anbefales heller å bruke multiple binære relasjoner, noe N-ære relasjoner lett kan oversettes til.
- En binær relasjon har to relasjonsroller. Hver av de to rollene uttrykker hvilken funksjon en av de to tilhørende entitetene har i relasjonen.

Relasjoner

Binære Relasjoner og Relasjonsroller:

- I eksempelet med Artist og Album har vi en relasjonsrolle fra Artist til Album(rolle1) og en fra Album til Artist (rolle2). Rolle1 kan gies rollenavnet *Publiserer*, og rolle2 navnet *Er_Publisert_Av*, for å få frem hvilken funksjon entitetene Artist og Album har i relasjonen.
- Notasjon for relasjonsnavn: Skrives over eller under relasjonslinjen.
- Notasjon for rolle: pil med retning fra en entitet til en annen, plassert over eller under relasjonslinjen.

Relasjoner

Binære Relasjoner og Relasjonsroller:

Eksempel på notasjon:



Relasjoner

Relasjonsroller og Kardinalitet:

- En kan definere minimum og maksimum kardinalitetsbegrensning for en relasjonsrolle.

Eksempel: En Artist kan publisere mellom 0 og N album: Vi kan dermed sette kardinalitetsbegrensning for rolle1 til: minKard=0, maksKard=N. Notasjon: **0:N**

Et Album må ha minimum en artist , maksimum N artister.

Kardinalitetsbegrensning for rolle2 kan bli: minKard=1, maksKard=N. **1:N**

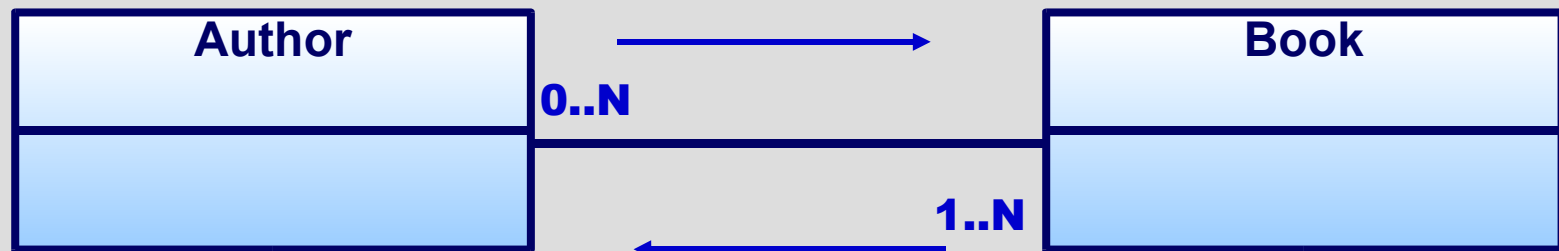
- Relevante verdier for minimum kardinalitet er 0 eller 1, og for maksimum kardinalitet 1 eller N.
- Hvis minimum kardinaliteten = 0 er relasjonen valgfri, ellers er den obligatorisk.

Relasjoner

Relasjonsroller og Kardinalitet:

- Basert på maksimumsverdiene blir relasjonene kalt:
One-To-One hvis kardinaliteten er 1:1
One-To-Many hvis kardinaliteten er 1:N: og
Many-To-Many hvis kardinaliteten er N:N

Eksempel: Many-To-Many



MultiVerdi-Attributter

- Et multiverdi-attributt er et attributt der objektet kan ha et sett av verdier. F.eks. kan en person ha en hel rekke med telefonnumre.
- Et multiverdi-attributt blir representert ved hjelp av en entitet og en relasjon.

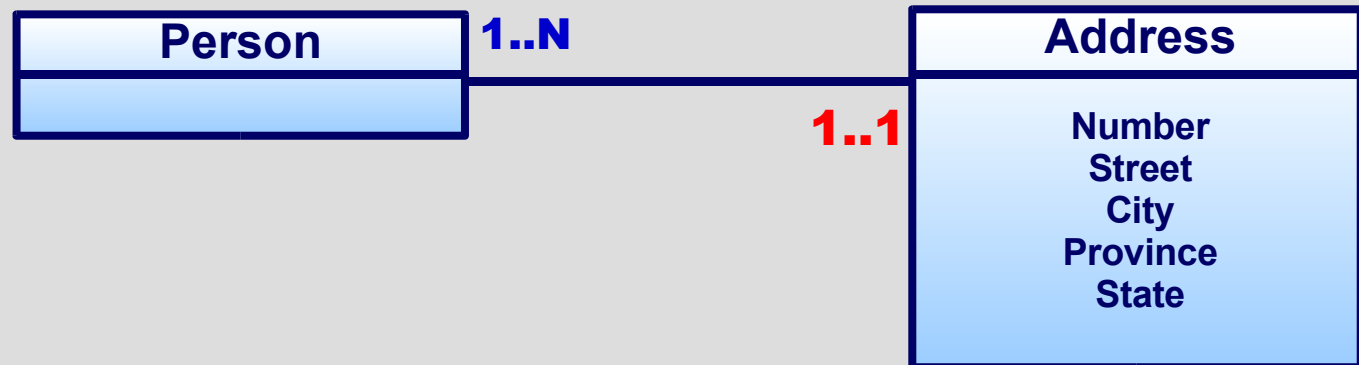
Eksempel:



Strukturerte attributter

- Attributter med en indre struktur. F.eks. attributten Adresse, der hver adresse består av gatenavn, gateNr, postnummer, poststed/by, land.
- Et strukturert-attributt blir også representert ved hjelp av en entitet og en relasjon.

Eksempel:



Relasjoner Med attributter

- Relasjoner med attributter beskriver en egenskap som refererer til et **par** av objekter (eller et sett av objekter).
- Relasjoner med attributter representeres av en entitet og to (eller flere) relasjoner

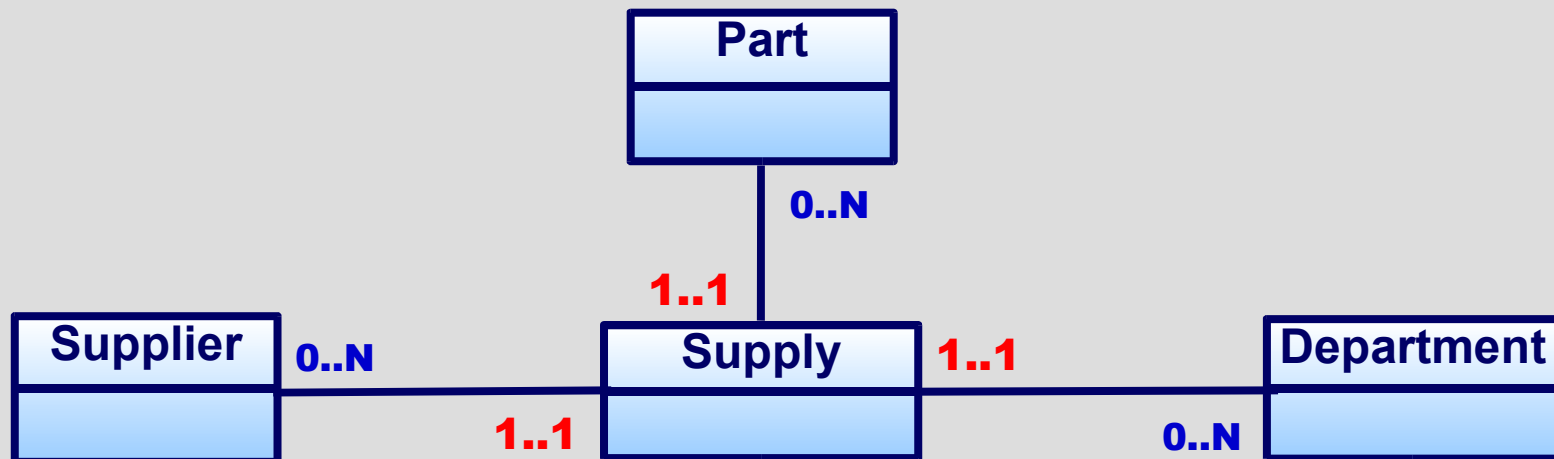
Eksempel: Karakteren en student får for avlagt eksamen i et fag.



N-ære Relasjoner

- En N-ær relasjon er en relasjon som har flere enn to entiteter ($N > 2$).
- Representeres av en entitet pluss N binære relasjoner.

Eksempel: «A Supplier Supplies a Part to a Department»



Utledet (Derivert) Informasjon

- Verdien til noen attributter eller relasjoner kan kalkuleres ut fra verdiene til andre elementer i skjemaet. Disse attributtene/relasjonene kalles for *utledede(deriverte)*.
- ER har ikke standard notasjon for dette, men spesifikasjonen for en attributt eller relasjon kan lett utvides til å støtte derivert informasjon.
 - En legger til en skråstrek foran attributt eller relasjons navnet.

notasjon: **/attributtnavn**

Utlede (Derivert) Informasjon

- Utlednings(Derivasjons)regelen som definerer den deriverte verdien blir definert som er uttrykk(expression). Uttrykket blir så lagt til bak attributt eller relasjonsnavnet.

Eksempler:

/Rabattert_Prisk {Prisk*Rabatt}

/Antall_Album {Count(Artist.ArtistToAlbum)}

Utledningsregelen er her uttrykt som et sti-uttrykk.

Mer informasjon og eksempler finnes
på boken "Designing Data-intensive
Web Applications" sine
hjemmesider:

<http://www.webml.org/webml/page2.do>

Modeller for design av Web-Applikasjoner

Kapittel 3 er tilgjengelig på:

<http://www.ii.uib.no/~eskil/fag>

og

<http://www.ii.uib.no/~arianna/fag>