



# IRM INFORMATIONSMODELLERING

## Metodbeskrivning





# Innehåll

<b>1</b>	<b>INTRODUKTION</b>	<b>3</b>
1.1	Vad är en informationsmodell?	3
<b>2</b>	<b>BEGREPP INOM INFORMATIONSMODELLERING</b>	<b>5</b>
2.1	Entiteter och förekomster	5
2.2	Attribut	7
2.2.1	Uppräknade värden	8
2.2.2	Vilken entitet hör attributet till?	8
2.3	Relationer	9
2.3.1	Namnsättning av relationer	10
2.3.2	Relationstyper	10
2.3.3	Att etablera relationer mellan två entiteter	12
2.3.4	Parallella relationer	14
2.4	Ämnesområden	14
2.4.1	Varför gruppera i ämnesområden?	15
2.4.2	Så kan man tänka när man skapar ämnesområden	15
2.5	Verksamhetsregler	16
2.6	Kommentarer och noteringar	16
2.7	Ändringshistorik	16
<b>3</b>	<b>MODELLERINGSTIPS</b>	<b>17</b>
3.1	Att exemplifiera i tabellform	17
3.1.1	1:M-relation	17
3.1.2	M:M-relation	18
3.2	Entitet eller M:M-relation?	19
3.3	Visa existensregler i diagram	19
3.4	Generalisering och specialisering	20
<b>4</b>	<b>ARBETSSÄTT</b>	<b>22</b>
<b>5</b>	<b>LATHUND</b>	<b>23</b>

Version 1.0 september 2022



# 1 Introduktion

När man behöver beskriva en verksamhet i modellform finns det ett flertal sätt att uttrycka sig på. Man kan göra modeller över till exempel en verksamhets organisationsstruktur med ansvar och bemanning, en verksamhets processer som beskriver vilka aktiviteter som utförs och i vilken ordning, eller en förmågemodell som beskriver vad en verksamhet gör, det vill säga de funktioner som behöver utföras för att en verksamhet ska fungera.

Man kan också göra en modell över information. Till skillnad från till exempel en processmodell så finns inget flöde i en informationsmodell, utan informationsmodellen är statisk och beskriver en struktur. Den innehåller bland annat namnsättning på verksamhetens entiteter, hur entiteterna relaterar till varandra och vilka regler som gäller. Vi kan fånga mycket av verksamhetslogiken i informationsmodellen, men med det menas inte att en informationsmodell täcker alla behov av beskrivningar. Man behöver betrakta verkligheten ur flera synvinklar. Man kan till exempel behöva en funktions- eller förmågekarta för att navigera i verksamheten, hitta kontext för alla dialoger och se hur information används. I ett processflöde kan man titta på vilken information de olika aktiviteterna behöver för att utföras. I den här metodbeskrivningen är det dock informationsmodellen som är i fokus.

## 1.1 Vad är en informationsmodell?

En informationsmodell beskriver de företeelser som verksamheten behöver hålla reda på förekomsterna av, det vill säga information som kan behöva beskrivas i tabellform. Man kan till exempel behöva hålla reda på förekomster av **Kunder**, beskriva deras identitetsbegrepp (**Kundnummer**) och andra attribut (**Kundnamn**, **Adress** etcetera). Informationsmodellen beskriver med andra ord hur man strukturerar information för att hålla reda på förekomsterna.

En informationsmodell består ofta av ett eller flera ritade, sammanhållna diagram som har rutor (entiteter) och streck mellan rutorna (relationer). Men informationsmodellen gestaltas inte bara av diagrammen utan innehåller även strukturerade textuella beskrivningar. Texterna ses inte som kompletterande beskrivningar, utan det är viktigt att de ses som integrerade delar av modellen då de tillför mycket av kunskap till informationsområdet som modelleras. När man tar fram en informationsmodell jobbar man samtidigt med diagram och texter för att analysera och beskriva domänen i fråga.

En **modell** kan gestalta sig som flera **diagram**, kanske av flera olika typer, samt strukturerad **text**.

För att visualisera entiteter och relationer i en informationsmodell används Entity-relationship-diagram (ER-diagram). Det beskriver den logiska datastrukturen inom en domän, vilket till exempel kan vara ett verksamhetsområde. När vi i efterföljande kapitel pratar om diagram i en informationsmodell så avser vi just ER-diagram, och närmare bestämt, ER-diagram i syfte att

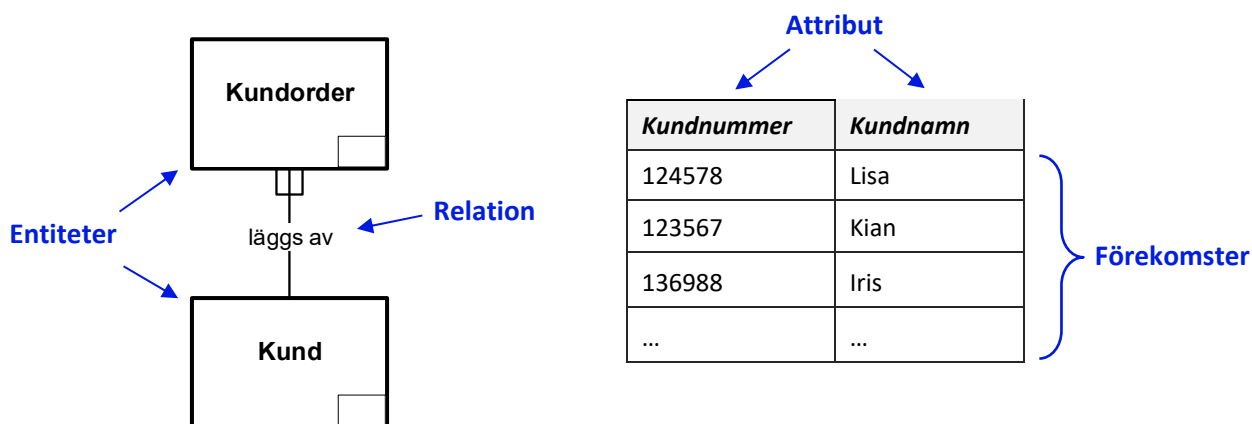


analysera och beskriva företeelser som hanteras inom en viss verksamhet, en viss tjänst eller ett visst system.

I en informationsmodell kan det också ingå tillståndsdigram och/eller förekomstdiagram. Tillståndsdigram ger möjlighet att modellera livscyklar och händelser vilket är mycket viktiga företeelser i en informationsmodell. Förekomstdiagram (instansdiagram) ger exempel på förekomster. Det är användbart för att beskriva strukturer som är svåra att beskriva på annat sätt. Till exempel är beskrivningssättet användbart för dynamiska skeenden i form av scenarion. Tillståndsdigram och förekomstdiagram hjälper oss inte bara att förklara ett ER-diagram utan behövs lika mycket för att vi verkligen ska komma fram till hur företeelser kan bete sig och hur vi kan skapa begrepp och språk för det.

## 2 Begrepp inom informationsmodellering

Inom informationsmodellering pratar vi i huvudsak om entiteter, förekomster, relationer och attribut:



Figur 1: Entiteter, förekomster, relationer och attribut

### 2.1 Entiteter och förekomster

Entiteten är en konkret eller abstrakt företeelse som är väsentlig och intressant för verksamheten. Det kan vara en person, en organisation, en plats, ett fysiskt ting, en händelse eller en transaktion som man vill kunna hantera information om. Kort sagt, något man vill säga något om eller göra något med. Om en verksamhet jobbar med försäljning så kan **Kund** och **Order** vara exempel på entiteter.

#### Exempel på kategorier och entiteter

<b>Person/Organisation</b>	Kund, anställd, leverantör
<b>Plats</b>	Kontor, region, stad, land
<b>Ting</b>	Order, utrustning, byggnad
<b>Händelse</b>	Användare har klickat på länk
<b>Transaktion</b>	Beställning, arbetsorder, faktura

Märk att vi skiljer mellan händelse och transaktion. Transaktionsobjekt är saker som offerter, fakturor, kontrakt eller liknande. Saker som säger att en viss transaktion har skett. En sådan har en livscykel och kan ha en status. Händelser däremot är momentana, när de har inträffat så kan de aldrig förändras.

Entiteter är alltså något som är viktigt för verksamheten. Ett kännetecken för vissa av entiteterna (personer, platser och ting) är att de existerar självständigt; de är alltså inte beroende av annan information. De kan identifieras med ett unikt och stabilt id-begrepp, det vill säga en unik



identifierare (också kallad primärnyckel). Exempelvis kan en **Kund** identifieras med **Kundnummer** och en **Produkt** med en **Produktkod**.

Händelse- och transaktionsentiteter är inte lika självständiga eftersom de brukar ha beroenden till någon eller några resursentiteter. Därför kan de vara lite svårare att identifiera och definiera. Många transaktionsentitet är dock ganska tydliga och kan redan vara försedda med en unik identifierare; en **Kundorder** kan exempelvis identifieras av ett **Ordernummer** och ett **Projekt** av ett **Projektnummer**.

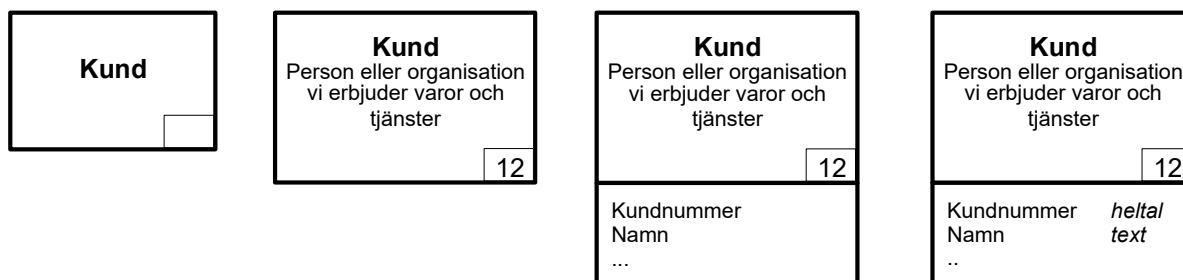
Entitet är ett övergripande begrepp för ett antal förekomster. Lite förenklat kan man säga att entitet är ett samlingsnamn för företeelser av samma slag. Entiteteten **Kund** kan exempelvis ha alla kunder inom aktuell verksamhet som förekomster. Den unika identifieraren gör att vi kan skilja en kund från en annan. Entiteten har dessutom ett antal attribut. Entiteten **Kund** kan till exempel ha attributen **Namn**, **Adress** och **Telefonnummer**.

#### Kännetecken för en entitet

- Är väsentlig och intressant för verksamheten
- Har förekomster
- Har en eller flera attribut
- Har en unik och stabil identifierare
- Existerar självständigt (dock inte alltid)

I ett diagram symboliseras en entitet av en rektangel. Inuti rektangeln skrivs entitetetens namn, med gemener och inledande versal, gärna fetstilt. Namnet är oftast ett substantiv (ibland består det av mer än ett ord) och skrivs alltid i singularis. Namnet bör anknyta till det namn som entiteteten har i verkligheten.

Till entiteteten hör en definition som klart och tydligt beskriver den, till exempel är en **Kund** en "Person eller organisation vi erbjuder varor och tjänster". En definition kännetecknas, till skillnad från andra beskrivningar, av att den kan ersätta namnet i en mening. Den måste alltså ringa in betydelsen och varken vara för vid eller för snäv. Ibland är det svårt att hitta en bra definition och då får man göra så gott det går.



Figur 2: Exempel där olika mycket information visas för en entitet

Beroende på syfte så kan man välja att ha med olika mycket information i diagrammet, se Figur 2. Förutom namnet på entiteten kan man också till exempel ha med entitetsdefinitionen, attribut, attributbeskrivningar och synonymer i diagrammet. Ibland kan det underlätta hanteringen av entiteterna om varje entitet ges ett unikt nummer, men det är inget måste.

Information som vi kan behöva beskriva om våra entiteter:

<b>Namn</b>	Namnet som vi bestämmer oss för att använda för entiteten	Alltid ha med
<b>Synonymer</b>	Andra namn som används för samma företeelse och i vilka sammanhang de används	Alltid ha med
<b>Definition</b>	Kortfattad förklaring av entiteten. Ska kunna ersätta namnet och man ska kunna förstå betydelsen.	Alltid ha med
<b>Beskrivning</b>	Ytterligare beskrivning utöver definitionen	Ha med om relevant
<b>Exempel på förekomster</b>	Exempel på förekomster för entiteten	Ha med om relevant
<b>Verksamhetsregler</b>	Regler som gäller entiteten, se kapitel 2.5	Ha med om relevant
<b>Kommentar/notering</b>	Kommentarer om entiteten, se kapitel 2.6	Ha med om relevant

## 2.2 Attribut

Attributen är de egenskaper som vi vill hålla information om för våra entiteter. För en **Order** kan det exempelvis vara **Ordernummer** och **Orderdatum**. För en **Kund** kan det vara **Kundnummer**, **Kundnamn** och **Kundadress**:

<b>Kundnummer</b>	<b>Kundnamn</b>	<b>Kundadress</b>
124578	Lisa	Industrigatan 1
123567	Kian	Storgatan 13
136988	Iris	Storgatan 13
...	...	...



Information som vi kan behöva beskriva om våra attribut:

<b>Namn</b>	Namnet som vi bestämmer oss för att använda för attributet	Alltid ha med
<b>Synonymer</b>	Andra namn som används för samma företeelse och i vilka sammanhang de används	Alltid ha med
<b>Definition</b>	Kort bestämning/förklaring av attributet	Alltid ha med
<b>Beskrivning</b>	Ytterligare beskrivning utöver definitionen	Ha med om relevant
<b>Exempel på förekomster</b>	Exempel på förekomster för attributet	Ha med om relevant
<b>Format</b>	Verksamhetsmässigt format, till exempel datumformat	Ha med om relevant
<b>Värdeomän</b>	Blir ofta bli en lista med uppräknade värden och då måste varje värde dokumenteras, se kapitel 2.2.1	Ha med om relevant
<b>Obligatorisk</b>	Anger om värdet måste vara ifyllt eller ej	Ha med om relevant
<b>Verksamhetsregler</b>	Regler som gäller attributet, se kapitel 2.5	Ha med om relevant
<b>Kommentar/notering</b>	Kommentarer om attributet, se kapitel 2.6	Ha med om relevant

## 2.2.1 Uppräknade värden

Vissa attribut har ett värdeförråd som har ett begränsat och fast antal värden. Det kan vara exempelvis statuskoder, typkoder eller kategorier. Dessa värden är ofta centrala för verksamheten och behöver listas och beskrivas.

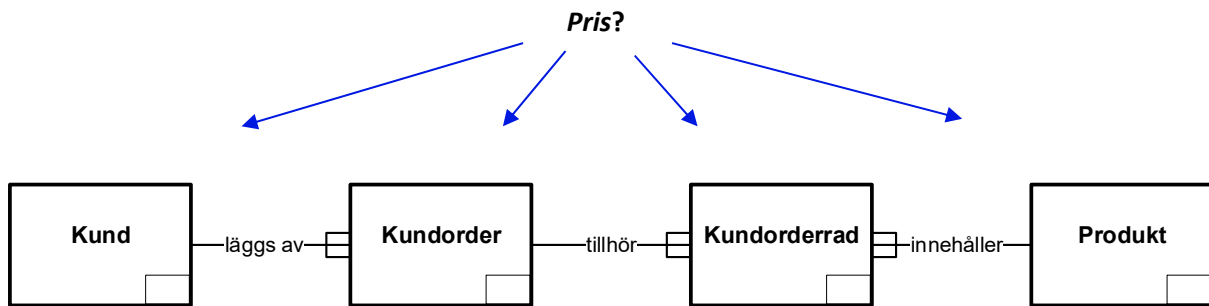
Information som vi kan behöva beskriva om våra uppräknade värden:

<b>Namn</b>	Namn på värdet	Alltid ha med
<b>Synonymer</b>	Andra namn som används för samma företeelse	Alltid ha med
<b>Definition</b>	Kort bestämning/förklaring av värdet	Alltid ha med
<b>Beskrivning</b>	Ytterligare beskrivning utöver definitionen	Ha med om relevant
<b>Verksamhetsregler</b>	Regler som gäller värdet, se kapitel 2.5	Ha med om relevant
<b>Kommentar/notering</b>	Kommentarer om värdet, se kapitel 2.6	Ha med om relevant

## 2.2.2 Vilken entitet hör attributet till?

För att förstå vilken entitet ett visst attribut tillhör måste vi förstå vad attributet är för något och vad det är beroende av. Ska till exempel **Pris** knytas till **Kund**, **Order**, **Orderrad** eller **Produkt**? Här måste vi ta reda på vad vi menar med pris och vad priset är beroende av.





Figur 3: Till vilken entitet hör attributet?

Hör **Pris** hemma på **Kund**? Nja, det låter väl inte så troligt att vi på en specifik **Kund** sätter ett specifikt pris. Hör **Pris** till **Produkt**? Ja, det låter rimligt att man skulle kunna sätta ett pris på en produkt, oberoende av vilken kund eller orderrad produkten knyts till. Hör **Pris** till **Orderrad**? Om priset på en produkt kan variera beroende på vilken orderrad den tillhör så knyts priset till orderraden. Det troliga kanske är att vi inte pratar om bara ett **Pris**, då priset kan vara många olika saker. Det kanske handlar om flera olika attribut som behöver höra till olika entiteter beroende på vad attributen är beroende av, som exempelvis **Produktpris** (ursprungspris som finns i prislistan för produkter) och **Rabatterat pris**.

## 2.3 Relationer

Mellan entiteter finns samband man vill kunna beskriva. Sambanden kallas relationer och kopplar ihop två entiteter med varandra.

Eftersom relationer är beroende av att ha entiteter att länka samman kan de till skillnad från entiteter inte existera självständigt och saknar både förekomster och egna unika identifierare. En relation illustreras med en linje mellan entiteterna.

### En relation

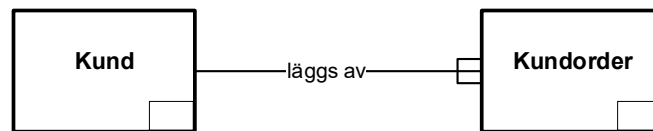
- Kopplar ihop entiteter
- Kan inte existera utan en entitet i varje ände

En relation är alltid en form av attribut hos en entitet, och den behöver beskrivas och hanteras på samma sätt. Läs mer om hur ett attribut beskrivs i kapitel 2.2.

### 2.3.1 Namnsättning av relationer

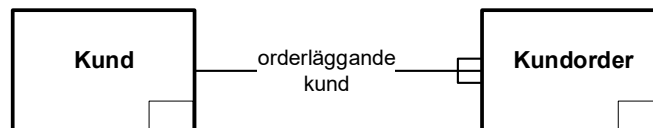
Relationen ges ett namn, som skrivs med gemener på eller utmed linjen. Namnet bör motsvara det kopplingen representerar. Det finns olika synsätt för hur man namnsätter relationer:

- Namnge relationen så att modellen kan "läsas" verbalt som ett naturligt språk.  
Exempel: En **Kundorder** läggs av en **Kund**.



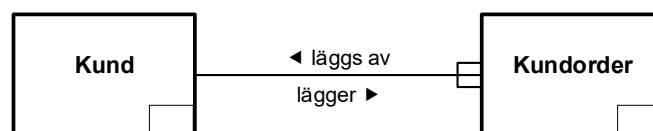
Figur 4: Relation namngiven så att den kan "läsas" verbalt

- Namnge relationen så att det attribut som relationen representerar avspeglas i namnet.  
Exempel: En **Kundorder** har en *orderläggande kund*.



Figur 5: Relation namngiven så att den avspeglar attributet som representeras

Relationerna ska alltid läsas i båda riktningarna, men det är inte alltid att båda riktningarna skrivs ut på relationen. Läsriktningen kan anges med en fylld pilspets vid namnet så att läsriktningen blir tydlig, se Figur 6. Om ingen pil anger läsriktningen så läses relationsnamnet från sidan med gaffeln.



Figur 6: Relation namngiven i båda riktningarna

Ibland när man ska namnsätta relationer finns inte ett naturligt begrepp. Då får man helt enkelt välja något som känns okej för alla inblandade. Det viktiga är att det är tillräckligt beskrivande för att förstå vad relationen uttrycker.

### 2.3.2 Relationstyper

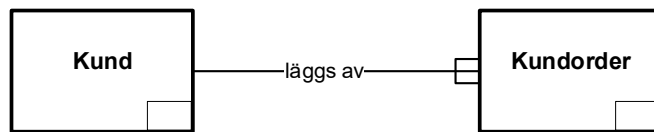
Relationerna representerar och illustrerar olika typer av samband. Det finns i huvudsak tre relationstyper där varje relationsände anger kardinaliteten på relationen.

### Relationstyper

En till många (1:M)	— —
En till en (1:1)	— —
Många till många (M:M)	≡ —

Relationstyperna talar om vilket slags samband som råder mellan entiteterna, det vill säga om det är en eller många förekomster som relationen kopplar samman.

**1:M** – Relationen *läggs av* mellan **Kundorder** och **Kund** i Figur 7 är ett exempel på en 1:M-relation.

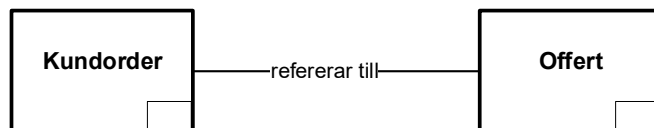


Figur 7: Exempel på en 1:M-relation

När man läser relationerna så utgår man alltid från *en* förekomst i den första entiteten som nämns, för att se hur många förekomster den relaterar till i den andra entiteten.

En **Kund** kan lägga flera **Kundorder**. En **Kundorder** läggs av endast en **Kund**. Till varje förekomst i en entitet relaterar en eller flera förekomster i en annan entitet.

**1:1** – Relationen *refererar till* mellan **Kundorder** och **Offert** i Figur 8 är ett exempel på en 1:1-relation. Här kan entitetens alla förekomster bara vara knutna till en förekomst i den relaterade entiteten.



Figur 8: Exempel på en 1:1-relation

I exemplet kan en **Offert** leda till högst en **Kundorder**. En **Kundorder** refererar bara till en **Offert**.

**M:M** – Relationen **Gångbar produkt** mellan **Produkt** och **Land** i Figur 9 är ett exempel på en M:M-relation. Dessa relationer saknar riktning. De namnsätts med ett substantiv och skrivs med gemener initierat av versal, gärna i fetstilt, på samma sätt som entiteter.



Figur 9: Exempel på en M:M-relation

Ett **Land** har flera **Produkter** som är gångbara i det landet. En **Produkt** kan var gångbar i mer än ett **Land**.

Tidigare nämndes att relationerna saknar både förekomster och egna unika identifierare. När det kommer till M:M-relationer så stämmer inte det helt. I en M:M-relation används kombinationer av förekomsterna för de relaterade entiteterna. Vi behöver med andra ord hålla reda på varje kombination, och för det krävs att varje kombination identifieras unikt. Istället för att ge kombinationen en egen unik identifierare så använder man sig av de unika identifierarna från de entiteter som de länkar samman. Mer om det i kapitel 3.1.

En M:M-relation kan också ha egna attribut. Det är attribut som inte hör hemma på entiteten **Land** och inte på **Produkt**, utan är beroende av kombinationen av **Land** och **Produkt**. Ett sådant exempel är **Lanseringsdatum** (från vilket datum en viss produkt är gångbar i ett visst land).

### 2.3.3 Att etablera relationer mellan två entiteter

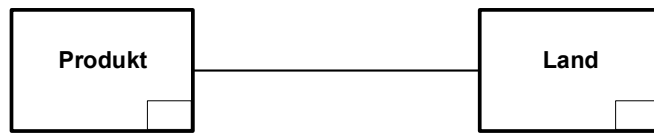
Medan man arbetar med att ta fram en modell så tar man kontinuerligt ställning till hur modellen ska uttrycka samband mellan entiteter. Nedan beskrivs en arbetsgång för att etablera en relation mellan två entiteter då vi har fått indikationer på att det kan vara intressant att hålla information om relationen mellan dessa.



Figur 10: Två entiteter mellan vilka det kan finnas en relation

#### 1. Finns relationen?

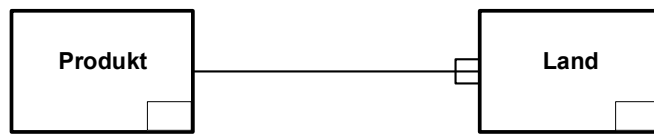
Är det intressant att hålla reda på sambandet mellan **Produkt** och **Land**? Om svaret är ja så blir frågan: Varför är det intressant? Svaret i det här fallet är att man vill hålla reda på vilka produkter som är gångbara (kan tänkas säljas) i vilka länder.



Figur 11: Entiteter som har en relation

2. Läs relationen åt ena hållet.

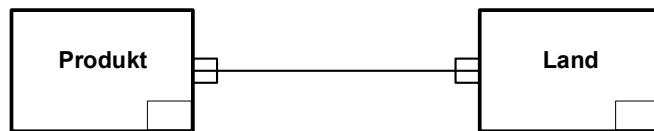
Kan en **Produkt** vara gångbar i bara ett **Land** eller i mer än ett **Land**? Här kan man med fördel ta konkreta exempel för att landa i frågan. En viss **Produkt**, till exempel en Röd herrcykel 28" med 18 växlar, är den gångbar i mer än ett **Land**? Om svaret är ja så ritas det ut en gaffel på landsidan. En **Produkt** kan vara gångbar i flera länder.



Figur 12: Entiteter med en 1:M-relation

3. Läs relationen åt andra hållet.

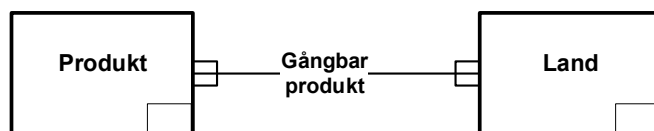
Kan ett **Land** ha bara en **Produkt** som är gångbar i landet eller kan flera olika **Produkter** vara gångbara i samma **Land**? Säg Belgien, har de fler produkter än den röda herrcykeln som är gångbara där? Om svaret är ja så ritas gaffeln in även på produktsidan av relationen. Ett **Land** har många **Produkter** som är gångbara i det landet.



Figur 13: Entiteter med en M:M-relation

4. Vad heter relationen?

Vad ska vi kalla relationen? I det här fallet finns ett naturligt begrepp som verksamheten använder sig av. De pratar om "gångbara produkter". Därmed namnsätter vi relationen enligt detta (fast i singularis). Läs mer om namnsättning i kapitel 2.3.1.

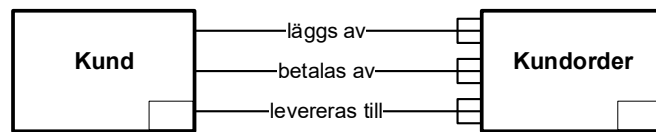


Figur 14: Entiteter med en namngiven M:M-relation

### 2.3.4 Parallella relationer

Ibland kan det finnas flera samtidigt, parallella, relationer mellan två entiteter. Parallella relationer kan vara beroende eller oberoende av varandra och kan ibland också utesluta varandra. Dessa regler beskriver vi oftast i textform vid sidan av diagrammet.

En **Kundorder** skulle till exempel kunna ha flera relationer till en **Kund**. En **Kundorder** kan *läggas av* en **Kund**, *betalas av* en annan **Kund** och *levereras till* ytterligare en annan **Kund**. Men relationerna skulle också kunna peka ut en och samma **Kund** för samtliga relationer. Här ges dock möjligheten att kunna peka ut olika kunder för olika syften.

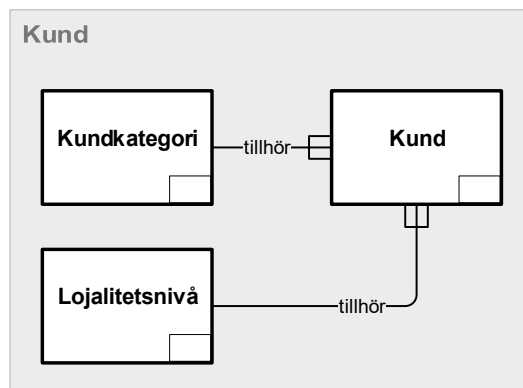


Figur 15: Entiteter med tre parallella relationer

## 2.4 Ämnesområden

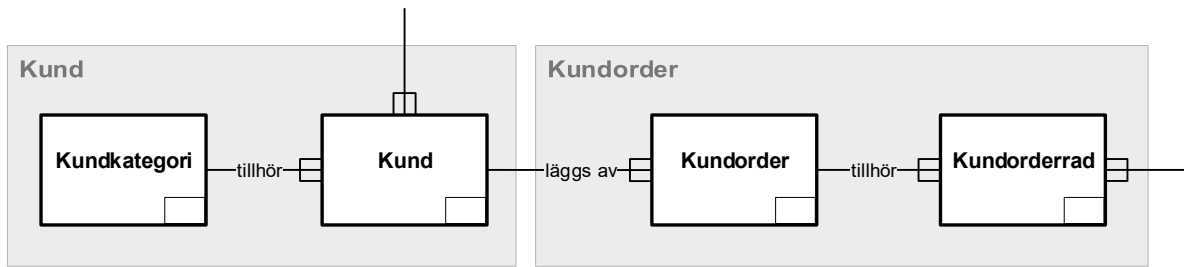
Om modellen är lite större kan det vara en bra idé att gruppera entiteterna i ämnesområden. Varje ämnesområde innehåller en eller flera entiteter som har en stark koppling till varandra.

Ämnesområdet Kund kan till exempel omfatta entiteterna **Kundkategori**, **Lojalitetsnivå** och **Kund**.



Figur 16: Ämnesområdet Kund

Entiteterna i en grupp har också relationer till andra entiteter som tillhör andra grupper.



Figur 17: Ämnesområdena Kund och Kundorder

Ämnesområden ritas som gråa rutor med ett litet mellanrum mellan rutorna ("korridorerna"). Använd korridorerna för att dra relationer som inte tillhör ämnesområdena.

### 2.4.1 Varför gruppera i ämnesområden?

Grupperingen kan göras av flera olika anledningar. Ibland vill vi få en bättre översikt av en stor modell, medan det i andra fall handlar om att till exempel identifiera tjänster. Ämnesområden kan därför omfatta olika många entiteter beroende på syftet med grupperingen.

Exempel på varför vi skapar ämnesområden:

- Högnivåindelning av vilken information som hanteras
  - Skapa övergripande förståelse för vilken information en verksamhet hanterar/bör hantera
  - Bra att göra en första gissning innan man börjar modellera för att få en idé om vilka områden som behöver fördjupas
  - Ge översikt, till exempel vid planering, prioritering och presentation
  - Bidra till att identifiera områden som ännu ej är modellerade
- Indelning utifrån informationsarkitekturella behov
  - Fördela modelleringsansvar och informationsägarskap
  - Beskriva gränssnitt och beroenden
  - Planering av applikationsportfölj
  - Skapa autonoma delar som kan utvecklas separat
  - Hjälpa oss att identifiera tjänster

### 2.4.2 Så kan man tänka när man skapar ämnesområden

Det kan ibland vara svårt att veta vilka entiteter som bör ligga i samma ämnesområde. Nedan beskrivs ett antal steg som underlättar när man skapar ämnesområden.

1. Hitta de mest centrala entiteterna och ringa in dessa.
  - En stark, central entitet som motiverar/definierar ämnesområdet.



2. Till vilket ämnesområde hör övriga entiteter?

- Fördefinierade värdeförråd (till exempel kategoriserande entiteter och statusentiteter) tillhör ofta samma ämnesområde som den entitet som de har en 1:M-relation till.
- Kan entiteterna inte leva utan varandra? Måste de hänga ihop i en transaktion eller en händelse? Har de samma livscykel? I så fall ska de troligtvis ligga i samma område.

3. Till vilket ämnesområde hör relationerna?

- En 1:M-relation tillhör samma område som den entitet som har "gaffeln".
- En M:M-relation tillhör ofta samma område som den entitet som föds sist. Ibland finns ingen tidsmässig koppling mellan entiteterna och då får man reda ut vilken entitet den har starkast koppling till.

## 2.5 Verksamhetsregler

När det kommer till verksamhetsregler så är informationsmodellen ett naturligt ställe att beskriva dessa. Verksamhetsregler kan höra hemma i flera delar av informationsmodellen, till exempel en entitet, en gruppering av entiteter, ett ämnesområde, en relation, ett visst attribut, ett visst attributvärde, mellan attributvärden eller attribut. Beskrivningar av verksamhetsregler kan därför förekomma på olika ställen.

## 2.6 Kommentarer och noteringar

Under modelleringarna är det viktigt att dokumentera kommentarer kring beslut, problem eller annat som man behöver veta för att förstå modellens innehåll och sammanhang. Kommentarererna kan behöva göras för såväl hela modellen som för enskilda entiteter, attribut, relationer etcetera. Man kan med andra ord behöva kunna placera ut kommentarer exakt där de behövs. Använder man ett mer avancerat modelleringsverktyg så är man ofta styrd till att sätta kommentarer på vissa specifika företeelser och kan inte alltid styra var man kan dokumentera dem. Det underlättar att ha ett verktyg där det är enkelt att lägga till kommentarer där de hör hemma. Bra att tänka på är att skriva kommentarerna i avvikande typsnitt eller färg så att de lätt kan särskiljas från de faktiska beskrivningarna.

## 2.7 Ändringshistorik

Det är viktigt att ha med en ändringslogg så det är tydligt vem som har ändrat vad i diagram och beskrivningar samt när ändringen skedde.

Om man har en beskrivning av varje ämnesområde så blir det en lämplig omfattning för ändringshistoriken. Varje gång man gör en ändring i ämnesområdet loggar man det där. Man skriver datum, vad man gjort och varför, samt vem som gjort det. På så sätt får vi historiken där den behövs. Ändringar som handlar om stavfel eller dylikt behöver inte loggas.



## 3 Modelleringsstips

### 3.1 Att exemplifiera i tabellform

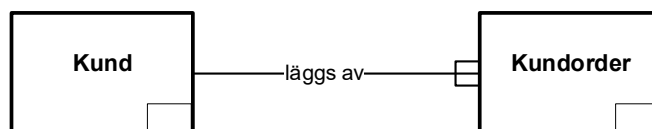
Att ge exempel på vilka attribut och förekomster som kan finnas för en entitet skapar en bättre förståelse för vad entiteten representerar. Det kan göras med hjälp av tabeller. Genom att sätta in verkliga exempel i tabellerna kan man se om tabeller och informationsmodell verkar stämma överens med den tänkta verksamheten.

#### Tabell exempel

- Illustrerar exempel på attribut och förekomster för en entitet eller en M:M-relation
- Består av kolumner med attribut och rader med förekomster
- Visar exempel på attribut och förekomster för en entitet (ersätter dock inte de rikare beskrivningarna av attribut eller förekomstdiagram)

#### 3.1.1 1:M-relation

I exemplen nedan har tabeller gjorts för entiteterna **Kund** och **Kundorder**.



Figur 18: Entiteter med 1:M-relation för vilka tabell exempel tagits fram

Tabell för entiteten **Kund**:

<b>Kundnummer</b>	<b>Namn</b>
124578	Lisa
123567	Kian
136988	Iris
...	...

Tabell för entiteten **Kundorder**:

<b>Kundordernummer</b>	<b>Läggs av Kundnummer</b>	<b>Kundorderdatum</b>
0003467	123567	2022-06-22
0003479	123567	2022-07-18
0003498	136988	2022-07-18
...	...	...

De första kolumnerna i varje tabell innehåller unika identifierare och har ett heldraget "tak" över kolumnen. I det här fallet är det **Kundnummer** och **Kundordernummer**. Att en relation existerar mellan två entiteter visar vi genom att låna in identifieraren från 1-änden på relationen till entiteten som har M-änden på relationen. Den inlånade identifieraren kallar vi "relationsattribut" och den har ett "streckat tak" över sin kolumn. I exemplet ovan pekar relationsattributet **Kundnummer** ut vilken

**Kund** som en viss **Kundorder** tillhör. Det är också tydligt att en **Kundorder** bara kan tillhöra en **Kund**. Däremot ser man att samma **Kund** kan ha lagt mer än en **Kundorder**. En enkel regel för i vilken tabell vi lägger relationsegenskapen är att man "trycker in den unika identifieraren med gaffeln". Med andra ord så trycker vi in den unika identifieraren för **Kund** med gaffeln in i **Kundordern**. Då ser man ju också tydligt gaffelperforeringen i "taket" över relationsattributet.

Märk att relationsattributet i exemplet ovan bara dyker upp i en entitets tabell, den entitet som har M-sidan av relationen kopplad till sig.

Efterföljande kolumner, utan heldraget streck eller streckat tak, innehåller andra attribut tillhörande entiteten. **Namn** och **Kundorderdatum** är sådana exempel.

### 3.1.2 M:M-relation

I exemplet nedan har tabeller gjorts för entiteterna **Produkt** och **Land** samt för relationen **Gångbar produkt**.



Figur 19: Entiteter med M:M-relation för vilka tabell exempel tagits fram

Tabell för entiteten **Produkt**:

<b>Produktkod</b>	<b>Namn</b>	<b>Vikt</b>	<b>Färg</b>
345945703	Vega	10 kg	Blå
484793934	Trason	8 kg	Röd
349723847	Beenie	12 kg	Svart
...	...	...	...

Tabell för entiteten **Land**:

<b>Landskod</b>	<b>Namn</b>
SE	Sverige
FR	Frankrike
PL	Polen
...	...

Tabell för M:M-relationen **Gångbar produkt**:

<b>Produktkod</b>	<b>Landskod</b>
345945703	SE
484793934	FR
349723847	SE
345945703	PL
484793934	PL
...	...

I kapitel 2.3.2 nämndes att en M:M-relation visar kombinationer av förekomsterna för de relaterade entiteterna. Vi behöver hålla reda på varje kombination, och för det krävs att varje kombination identifieras unikt. Istället för att ge kombinationen en egen unik identifierare så använder man sig av id-begreppen från de entiteter som de länkar samman. Märk väl att samma **Produktkod** och **Landskod** inte får förekomma två gånger, då ju varje förekomst måste kunna identifieras unikt.

### 3.2 Entitet eller M:M-relation?

Det går alltid att göra en entitet av en M:M-relation. Det som skiljer en M:M-relation från en entitet är att M:M-relationen skapar sin unika identifierare genom att låna identifierare från entiteterna som den relaterar till. Som i exemplet i kapitel 3.1.2, där varje **Gångbar produkt** identifieras unikt av kombinationen av **Produktkod** och **Landskod**.

Om vi skulle göra en entitet av **Gångbar produkt** så skulle diagrammet se ut som i Figur 20, där varje **Gångbar produkt** gäller för en **Produkt** och är gångbar i ett **Land**. I det här fallet får varje **Gångbar produkt** en egen unik identifierare, **Gångbar produkt ID**. Här kan därmed samma kombination av **Produkt** och **Land** förekomma flera gånger. **Gångbar produkt** lånar också in de unika identifierarna från **Produkt** och **Land** som relationsattribut för att visa relationerna till dem.



Figur 20: En entitet istället för en M:M-relation

Tabell för entiteten **Gångbar produkt**:

<b>Gångbar produkt ID</b>	<b>Produktkod</b>	<b>Landskod</b>
2367	345945703	SE
2456	484793934	FR
2458	349723847	SE
2787	345945703	PL
3466	484793934	PL
...	...	...

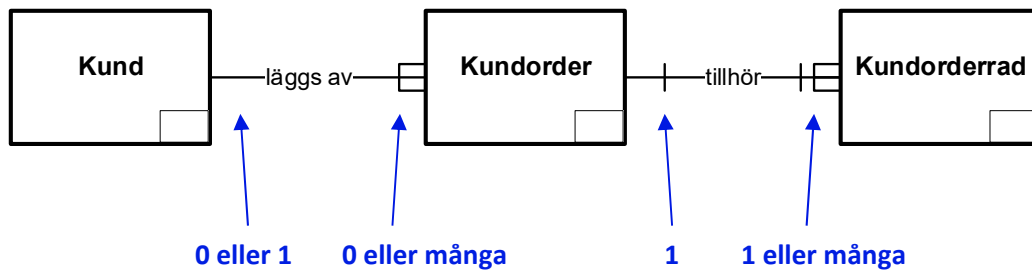
### 3.3 Visa existensregler i diagram

Genom att visa existensregler i diagram kan verksamhetsregler noggrannare preciseras. Om behovet finns kan det vara relevant att införa.

Relationer kan vara tvingande eller frivilliga. En tvingande relation innebär att en entitet bara kan finnas om den entitet den är relaterad till finns. I exemplet nedan är relationen *tillhör* mellan entiteten **Kundorder** och **Kundorderrad** tvingande; det måste finnas minst en förekomst av

**Kundorder** för att entiteten **Kundorderrad** ska kunna existera. En **Kundorderrad** kan med andra ord inte existera utan att vara kopplad till en **Kundorder**. Samtidigt måste en **Kundorder** alltid ha minst en **Kundorderrad**.

En tvingande relation markeras med ett tvärgående streck över relationslinjen på den oberoende entitetens sida. Frivilliga relationer markeras inte uttryckligen.



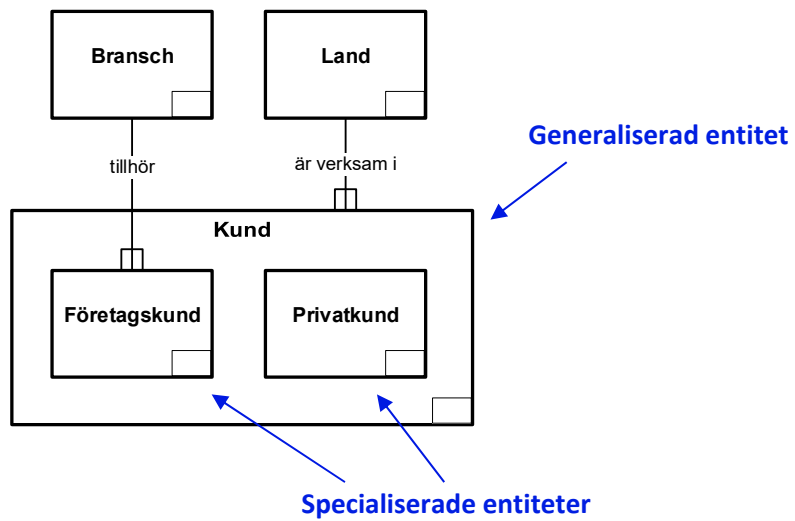
Figur 21: Exempel på frivilliga och tvingande relationer

Det är en god idé att först ta fram strukturen i informationsmodellen samt definitioner för entiteterna innan existensregler och andra detaljer förs in i modellen. Om existensregler införs i diagram, lägg då in det för samtliga relationer. Annars finns en risk att relationer utan tvärgående streck kan antas vara frivilliga, när de egentligen kanske inte visar en existensregel.

### 3.4 Generalisering och specialisering

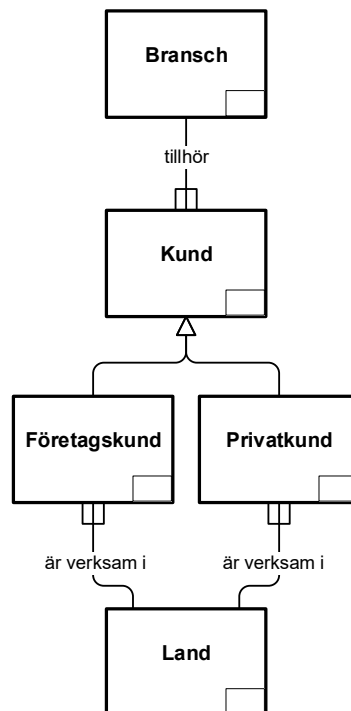
En del entiteter har flera attribut och relationer gemensamt. Man pratar om generalisering (där de gemensamma attributen och relationerna återfinns) samt specialisering (där de icke gemensamma attributen och relationerna återfinns). De attribut och relationer som knyts till den generella entiteten är automatiskt av de specialiserade entiteterna. Om de specialiserade entiteterna har egna attribut eller relationer som går enbart till dem så knyts dessa endast till den specialiserade entiteten. Specialiserade och generella entiteter benämns även "subtyper" respektive "supertyper".

Generalisering och specialisering kan ritas genom att lägga de specialiserade entiteterna inuti den generella entiteten, se Figur 22.



Figur 22: Specialiserade entiteter inuti en generell

Alternativt kan det ritas som en linje med ofylld pilspets vid den generella entiteten.



Figur 23: Generell entitet med ofylld pilspets

Oavsett vilket sätt man ritat på så är **Företagskund** och **Privatkund** specialiseringar av entiteten **Kund**, och **Kund** är en generalisering av entiteterna **Företagskund** och **Privatkund**. Den unika identifieraren för **Kund** (**Kundnummer**) blir också unik identifierare för **Företagskund** och **Privatkund**. Även andra attribut som ligger på **Kund** (till exempel **Namn**) ärvs ner till **Företagskund** och **Privatkund**.



## 4 Arbetsätt

Modellering kan ske i flera olika arbetsformer. Man kan till exempel jobba fram modellen i seminarieform, riktat arbetsmöte, intervjuform, studie av dokumentation eller analys av datastrukturer och data. Varje arbetsform har sina speciella styrkor och svagheter. De behöver kunna kombineras på olika sätt, över tid och också beroende på vad syftet med uppgiften är, hur avgränsningen ser ut och vilka resurser man har att tillgå.

Oavsett vald arbetsform så kan följande steg vara något att hålla sig till när man jobbar fram en informationsmodell:

1. Skapa en förståelse över uppdraget. Vad är det vi ska leverera? I vilket syfte? Vem ska använda resultatet och till vad?
2. Skapa en förståelse för området.

Använd befintligt material, till exempel:

- Verksamhetsbeskrivningar (processbeskrivningar, rutinbeskrivningar, ordlistor, tidigare framtagna informationsmodeller etcetera)
- System och systemdokumentation (få tillgång till system, databaser, XML-scheman, Excellösningar, databasmodeller etcetera)

Vi får i det här steget ett hum om verksamhetsbegrepp och avgränsning och börjar skissa på modell och beskrivningar utifrån det vi hittills förstått. När vi läser och skissar så bygger vi successivt upp en första tes av modellen.

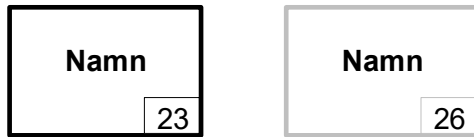
Bygg upp modellen steg för steg:

1. Identifiera några begrepp som kan vara entiteter.
2. Plocka ut två av dem som du tror har en relation till varandra, dra relationen och sätt kardinalitet och namn på relationen.
3. Lägg sedan till entiteter efter hand.

Fyll kontinuerligt på med attribut på entiteterna, exempel på förekomster, övriga beskrivningar och frågetecken som är relevanta. Man kan också under modelleringens gång ha behov av att titta närmare på specifika entiteters tillstånd (Tillståndsdigram), förekomstets beteenden (Förekomstdiagram) eller att rita bilder, göra tabell exempel eller annat som gör att vi successivt förstår mer och bygger på modellen med mer kunskap.

3. Vidareutveckla modellen med kunskap från andra. Att göra en modell enbart utifrån befintliga beskrivningar etcetera skapar något att utgå ifrån vi börjar bygga på oss frågor som vi vill ta vidare. Än så länge innehåller modellen många missförstånd och frågetecken. Dessa hanteras steg för steg genom att träffa personer som är väl insatta i verksamhetsområdet. Vilka arbetsformer man väljer är som sagt olika beroende på syfte och situation. Men fundera på vad det är för olika typer av frågor och problem som ska lösas och vilka arbetsformer som lämpar sig bäst de olika problemen, samt vad som är möjligt utifrån tid och resurser.

## 5 Lathund



**Entiteter** ritas som rektanglar. Entitetens namn är ett substantiv som skrivs med gemener och med inledande versal, gärna fetstilt.

”Släckt” entitet eller relation används om entiteten ligger utanför avgränsningen eller även finns med på annan plats i modellen.

———— motsvarar ————

———— tillhör ————

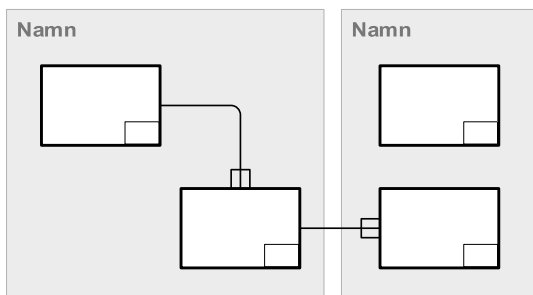
≡ — Lagersaldo — ≡

**Relationer** ska alltid namnsättas.

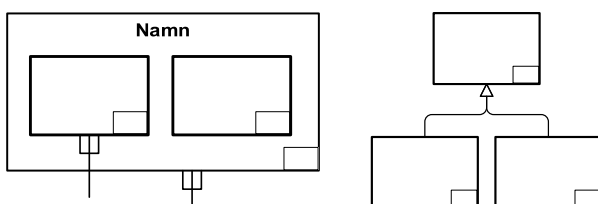
**1:1** – Substantiv eller verb skrivet med gemener.

**1:M** – Substantiv eller verb skrivet med gemener (läsriktning från gaffeln mot 1-sidan).

**M:M** – Substantiv i singular form, skrivet med gemener och med inledande versal, gärna fetstilt.



**Ämnesområden** ritas som gråa rutor med ett litet mellanrum mellan rutorna (”korridorerna”). Använd korridorerna för att dra relationer som inte tillhör ämnesområdena.



**Generalisering och specialisering** används när specialiserade entiteter ärver egenskaper från en generell entitet. Ritas genom att lägga de specialiserade entiteterna inuti den generella entiteten, alternativt som en linje med ofylld pilspets vid den generella entiteten.