Super- och subentitetstyper

Ett argument för att använda super- och subentitetstyper är att det finns attribut som är relevanta endast för några av entiteterna i en superentitetstyp eller att det finns sambandstyper där bara entiteter i en subentitetstyp kan delta. Detta är information eller restriktioner som kan vara viktig att tydligt visa i ett ER-diagram.

En superentitetstyp är en entitetstyp som innehåller distinkta subentitetstyper. En subentitetstyp är en distinkt entitetstyp som är knuten till en viss superentitetstyp. Både super- och subentitetstypen visas i ER-diagrammet och subentitetstypen ärver superentitetstypens attribut. Processen med att hitta super- och subentitetstyper kallas för generalisering och specialisering.

Generalisering/specialisering

Generalisering och specialisering går ut på att försöka finna entitetstyper som delar gemensamma attribut. När dessa entitetstyper hittats får vi en hierarki som består av super- och subentitetstyper som används för att representera det vi kan kalla för förälder/barn-samband mellan entitetstyper.

Figur 1. Enheter som inte är samma.

Specialisering är processen att försöka hitta skillnaderna mellan entitetstyper genom att identifiera särdrag. Detta är en toppen-ner-metod där vi först försöka identifiera superentitetstypen och därefter de tillhörande subentitetstyperna. Därefter försöker vi hitta eventuella samband mellan subentitetstyperna. Specialisering betonar skillnaderna.

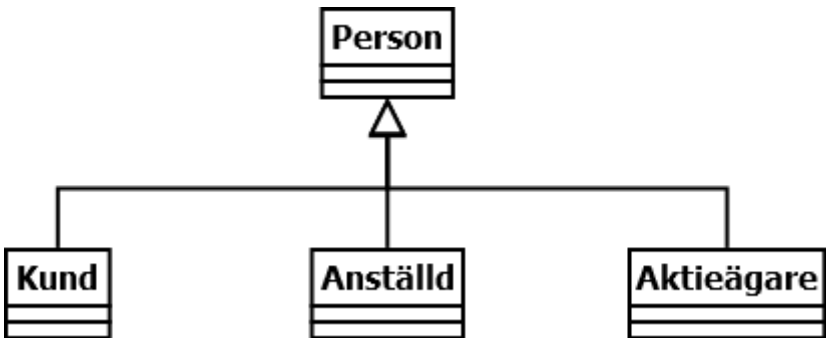
Generalisering är processen att försöka hitta skillnaderna mellan entitetstyper genom att identifiera gemensamma drag. Det här är ett botten-upp-metod. Generalisering fångar likheter mellan entitetstyper och drar nytta av likheter mellan entitetstyper på samma sätt som en entitetstyp utnyttjar likheter mellan entiteter. Den generella entitetstypen blir en superentitetstyp.

Figur 2. Enheter som har saker gemensamt.

Denna typ av hierarkier ger oss mer flexibilitet eftersom varje subklass ärver gemensamma attribut från superklassen. Ett exempel är personer med olika roller på ett företag.

Eftersom en person kan spela olika roller inom ett företag kan information om denna person spridas till olika delar av företaget. Vi ser ofta attribut som namn och adress på kunder, säljare, anställda och aktieägare. Om samma person är kund, anställd och aktieägare på samma gång skulle denna personliga information redundant registreras för varje roll personen spelar. Därför är det vanligt att modellera detta som en superentitetstyp med subentitetstyper. Vi skapar entitetstyperna Person, Kund, Anställd och Aktieägare. Vi låter Person vara en superentitetstyp och övriga entitetstyper blir subentitetstyper. Nu är det möjligt att definiera gemensamma attribut som namn, telefonnummer, adress och födelse-datum till superentitetstypen Person och sen låta dessa ärvas av subentitetstyperna. Genom att göra på det här sättet kan vi undvika att upprepa data om samma person flera gånger.

I figur 3 ser vi exempel på symbolen som används för att modellera arv mellan superentitetstyp och subentitetstyper. När UML används som notation i ER-diagrammet pratar man ofta om superklass/subklass i stället för superentitetstyp/subentitetstyp.



Figur 3. Exempel på arv i ett ER-diagram.

Begränsningar vid Generalisering/specialisering

Det finns vissa begränsningar som är aktuella vid generalisering och specialisering. Det är restriktioner för hur deltagande av en entitet i superklassen är i subklasserna. Vi har en disjunkt restriktion när en entitet i superklassen bara kan delta i en av subklasserna. Utifrån

exemplet ovan kan en person antingen vara en kund, anställd eller aktieägare. Personen kan till exempel inte vara både en kund och anställd. Det kan finnas personer som inte är något av subklasserna.

Vi har en överlappande restriktion när en entitet i superklass kan delta i mer än en av subklasserna. En person kan då vara både en kund och anställd samtidigt. Det kan även finnas personer som inte är någonting eller enbart till exempel kund.

Vi har en fullständig restriktion när varje entitet i superklassen måste delta i en av subklasserna. Det får då inte finnas personer som inte är någonting av subklasserna utan en person måste minst vara en kund, anställd eller aktieägare.

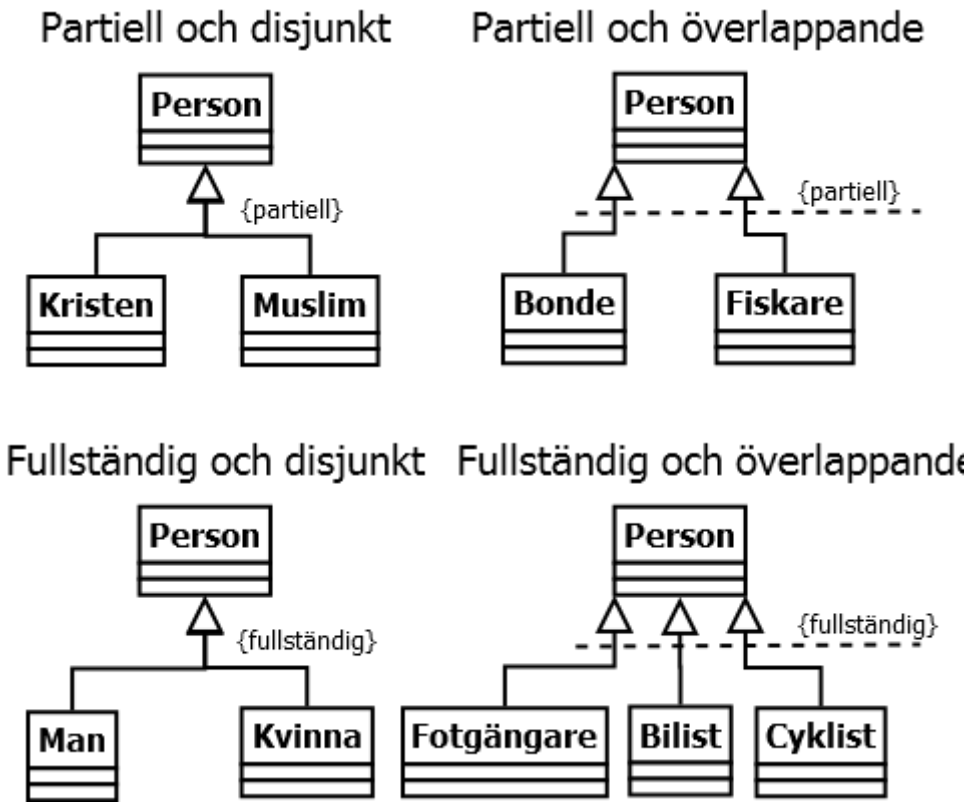
Vi har en partiell restriktion när en entitet i superklassen inte behöver delta i en subklass. Här behöver en person inte vara en kund, anställd eller aktieägare utan kan vara en person i allmänhet.

Tabell 1 sammanfattar ovan på ett överskådligt sätt.

Tabell 1. Restriktioner vid arv.

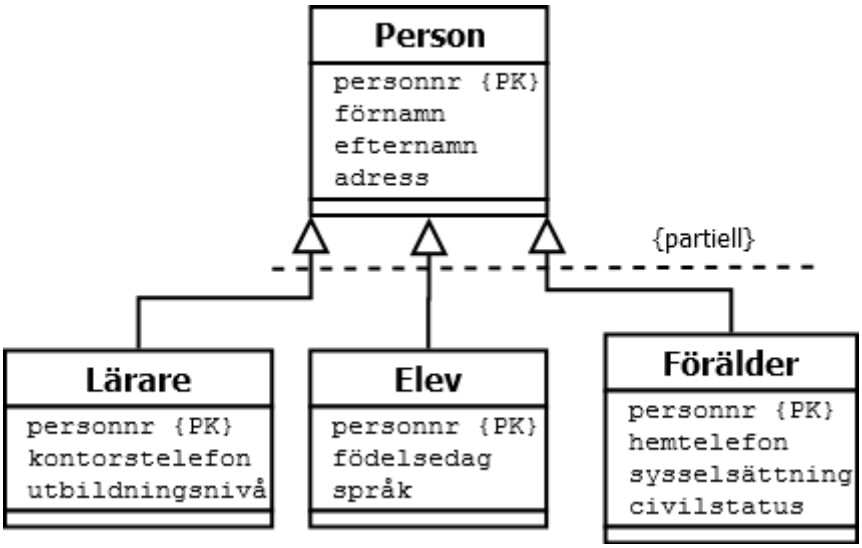
	DISJUNKT	ÖVERLAPPANDE
PARTIELL	Deltar i 0 eller 1 subklass	Deltar i 0 till många subklasser
FULLSTÄNDIG	Deltar i 1 subklass	Deltar i 1 till många subklasser

I figur 4 har vi översatt restriktionerna i tabell 1 till ett ER-diagram som visar de olika symbolerna som används.



Figur 4. Symboler för att representera restriktioner vid arv.

I figur 5 ser vi ett exempel på hur arv kan användas. Istället för att använda tre olika entitetstyper med en blandning av specifika och gemensamma attribut, har vi valt att samla de gemensamma attribut i en superklass som heter Person och sedan hålla de specifika attributen i tre subklasser: Lärare, Elev och Förälder. Restriktionerna är partiell och överlappande. Partiell eftersom det kan finnas personer som varken är en lärare, elev eller förälder. Överlappande eftersom en person både kan vara en lärare och en förälder.



Figur 5. Arvshierarki över lärare, elever och föräldrar.

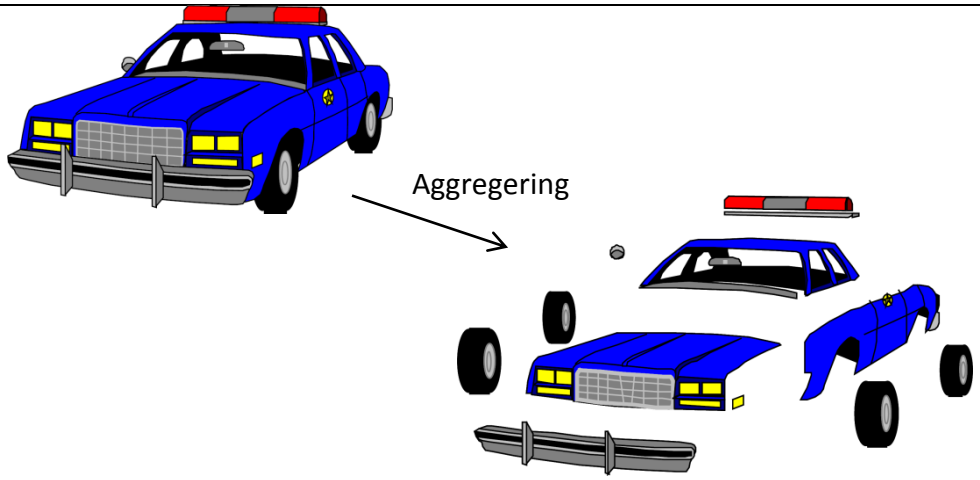
Sambandstypen mellan en superklass och subklass är alltid av typen en-till-en. En person är till exempel bara en lärare och kan aldrig vara två olika lärare. Primärnyckeln i superklassen kommer alltid även att vara primärnyckel i alla subklasser. Om vi översätter ER-diagrammet i figur 5 till relationsmodellen får vi följande tabeller:

```
Person (personnr, förnamn, efternamn, adress)
Lärare (personnr*, kontorstelefon, utbildningsnivå)
Elev (personnr*, födelsedag, språk)
Förälder (personnr*, hemtelefon, sysselsättning, civilstatus)
```

Lägg märke till att superklassens primärnyckel blir både primärnyckel och främmande nyckel i subklasserna.

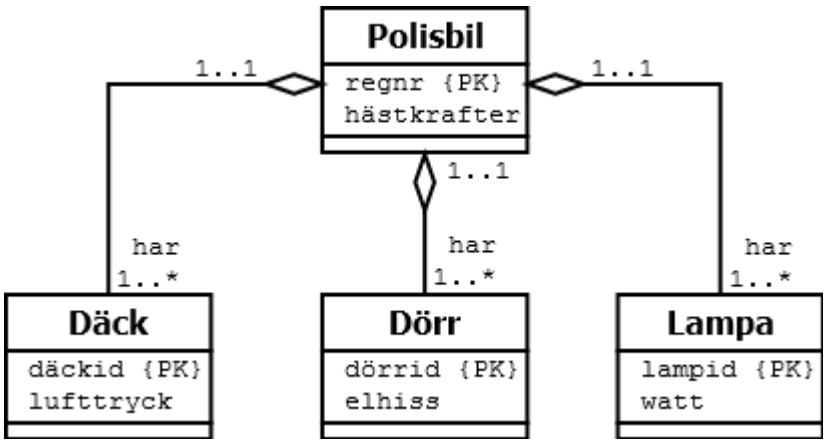
Aggregering

Aggregering representerar en "har-en" eller "är-del-av" samband mellan entiteter i olika entitetstyper. Här representerar en entitetstyp "helheten" och en annan entitetstyp representerar "delen". En aggregering är baserad på det faktum att en entitetstyp kan vara sammansatt av flera komponenter som i exemplet i figur 6.



Figur 6. En bil består av bilkomponenter.

Figur 6 kan översättas till ett ER-diagram så som visas i figur 7.



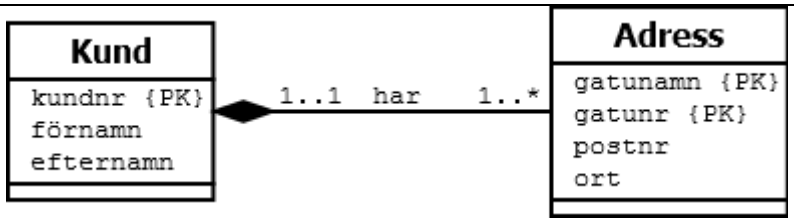
Figur 7. ER-diagram med aggregering.

Symbolen som används för aggregering är en diamant (romb) som placeras vid den entitetstyp som utgör helheten. En översättning av ER-diagrammet i figur 7 till relationsmodellen ser ut på följande sätt:

```
Polisbil (regnr, hästkrafter)
Däck (däckid, lufttryck, regnr*)
Dörr (dörrid, elhiss, regnr*)
Lampa (lampid, watt, regnr*)
```

Komposition

Komposition är en särskild form av aggregering som representerar ett samband mellan entiteter där det finns en stark äganderätt mellan "helheten" och "delen". Om helheten tas bort kommer även tillhörande delen att tas bort. Figur 18 visar en komposition.



Figur 8. ER-diagram med komposition.

Deltagandet vid helheten vid en komposition måste vara 0..1 eller 1..1. En del kan nämligen inte vara del i mer än en helhet vid komposition.

Översatt till relationsmodellen får vi följande tabeller:

```
Kund (kundnr, förnamn, efternamn)
Adress (gatunamn, gatunr, postnr, ort, kundnr*)
```