

Lite om databasdesign och modellering

Konceptuell databasdesign

Med *konceptuell databasdesign* avses processen att konstruera en datamodell för en verksamhet, oberoende av fysiska villkor. Modelleringen genomförs i steg:

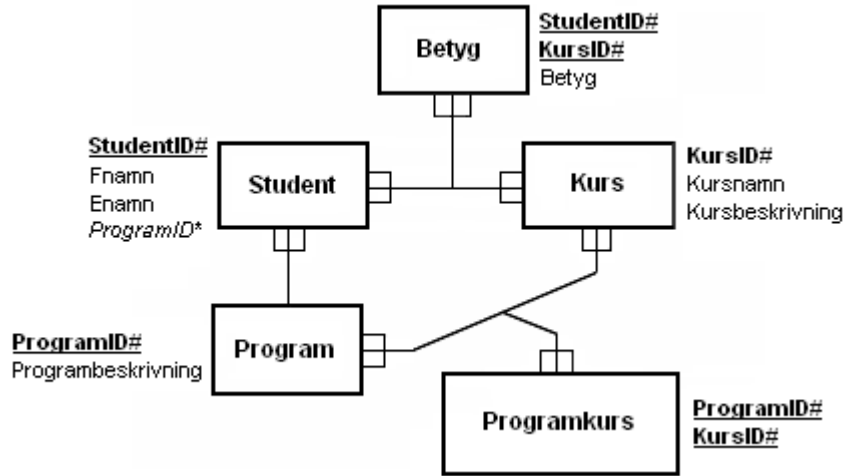
- Identifiera centrala entiteter (objekt) i målverksamheten. För ett säljande företag kan det vara frågan om kunder, beställningar eller leverantörer
- Identifiera relationer (kopplingar eller samband) mellan de identifierade objekten – Hur hänger de ihop, hur relaterar eller förhåller sig objekten till varandra?
- Identifiera objektens attribut - gemensamma egenskaper - t ex kundnummer, kundnamn och adress för ett objekt som Kund
- (Fundera över domäner för attributen, dvs vilka typer av data ska lagras – text, tal, etc?)
- Bestäm primärnyckelattribut och främmande nycklar
- Kontrollera att det inte finns omotiverad redundans i modellen, dvs att data inte dubbellagras i onödan
- Validera den konceptuella modellen mot de operationer (transaktioner) som ska kunna utföras mot databasen
- Utvärdera den konceptuella modellen med beställaren, t ex en kund, användare el dyl.

Vad är en konceptuell modell?

En konceptuell modell är ett resultat av modelleringsarbetet; en grafisk beskrivning som visar objekt, objekts attribut och relationer mellan objekt. Ett exempel på objekt (se modell 1a nedan) är **Student**. Exempel på objektsattribut är **StudentID#**, Fnamn, Enamn och **ProgramID***. **StudentID#** är en s k primärnyckel och **ProgramID*** en s k främmande nyckel. I det här fallet ska personnummer användas som värde för **StudentID#** i det kommande databassystemet. Det för att unikt identifiera varje student. Notera att det under vissa omständigheter kan vara olagligt eller olämpligt att lagra personnummer i en databas (se Personuppgiftslagen, PUL).

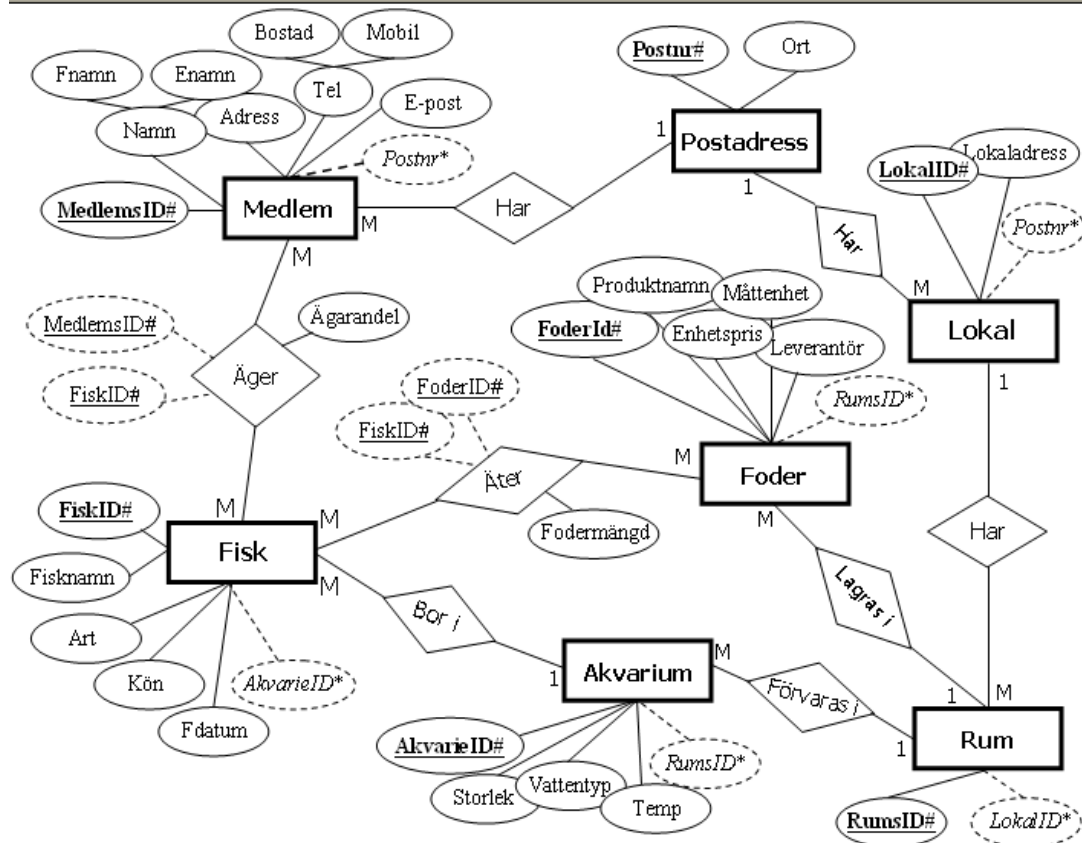
Modell 1a – enkel konceptuell modell

Notationen är ett exempel på en av de tre olika notationerna i detta häfte.



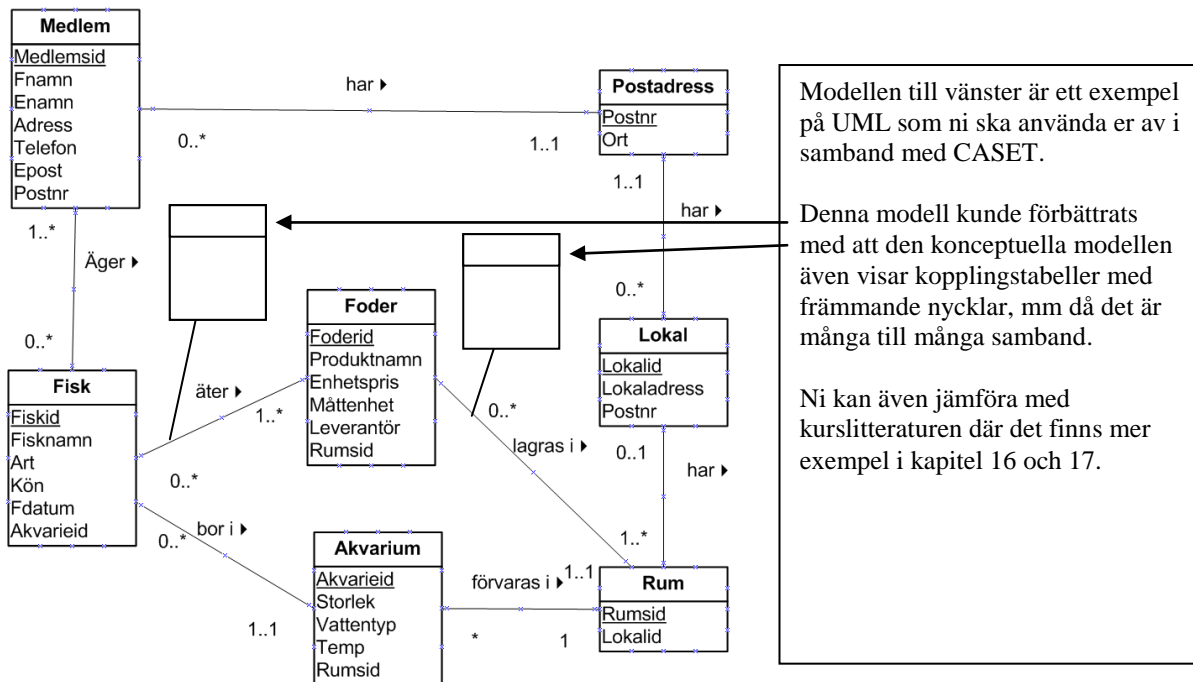
Modell 1b – E/R-graf

Ett annat exempel på en notation.



Modell 1c – E/R-graf (UML-notation)

Detta är den vanligaste notationen som ni bör använda, UML, som är de facto standard.



Modelleringsregler mm

Hur identifierar man rätt objekt och objekts attribut eller egenskaper?

- Analysera verksamheten och identifiera de objekt databasen ska innehålla data om. Handlar det om utbildningsverksamhet kan t ex Kursdeltagare, Kurser eller Program vara centrala objekt (jfr. substantiv)
- Undersök vilka gemensamma kvaliteter eller egenskaper ett objekt beskrivs med (jfr. vilka data som ska lagras). Objektet Student har t ex gemensamma egenskaper som personnummer, namn, adress och tel. Instanser av objektet Kund har gemensamma egenskaper som kundnummer, namn, adress, osv

Hur upptäcker man "relationer" eller samband?

- Genom associationer t ex att *kunder* gör *beställningar*
- Genom att söka strukturer t ex att beställningar innehåller artiklar

Hur detaljerad ska man vara?

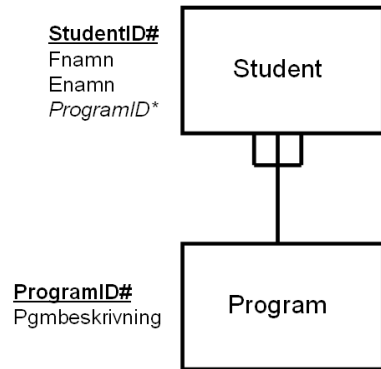
- Tillräckligt för kunna beskriva den aktuella verksamheten på den nivå som krävs - modellen ska gå att omvandla till en relationsdatabas i lägst 3NF

Modellering av 1:M-relationen

I en 1:M-relation "lånar" objektet på "många-sidan" det relaterade objektets identifierande egenskap (en primärnyckel). Den "lånade" nyckeln kallas främmande nyckel och utgör en referens till den relaterade tabellen. Relationen utläses: *Många studenter kan läsa ett program, t ex VIP eller IT-ekonom, men en*

student får bara läsa ett program åt gången (en verksamhetsregel). I den enkla modellen markeras en 1:M-relation genom att en "gaffel" sätts i objektet på "många-sidan".

Modelleringsregel 1:M-relation: exempel Student - Program



Tabell exempel: Student - Program

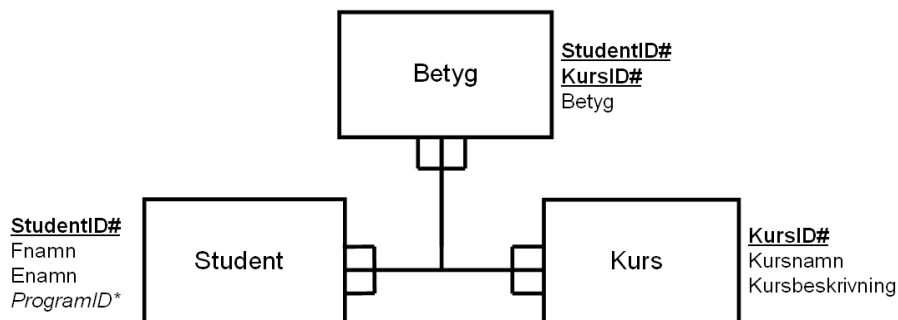
StudentID#	Fnamn	Enamn	ProgramID*
8201139901	Anna	Ek	VIP07-10
8611221113	Hanna	Ström	IT106-09
8403200278	Bo	Ahlberg	VIP07-10

ProgramID#	Programbeskrivning
VIP07-10	Programmet syftar till.....
IT106-09	Programmet syftar.....
WEB05-08	Programmet.....

Modellering av M:M-relationen

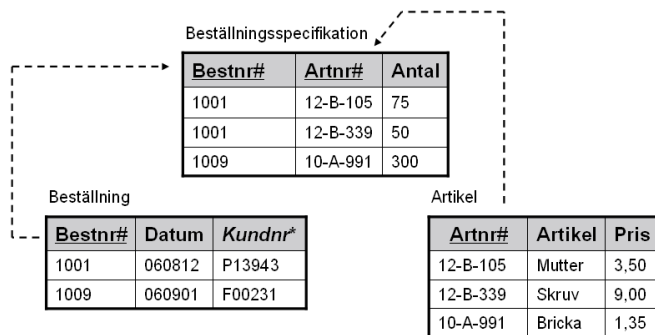
En M:M-relation kan ses som två 1:M-relationer som förutsätter ett tredje objekt; ett s k relationsobjekt eller en "kopplingstabell". Relationen i exemplet nedan kan utläsas: *En student kan läsa och få betyg på många kurser. En kurs kan läsas av många studenter*, osv. För att lagra data om betyg skapar vi ett helt nytt objekt – relationsobjektet Betyg.

Exempel - relationsobjektet Betyg



Relationen i tabellexemplet nedan kan utläsas: En beställning kan innehålla många artiklar (eg. artikelnummer, inte artiklar i sig), och en artikel kan ingå i många beställningar. För att lösa detta skapar vi ett helt nytt objekt – relationsobjektet Beställningsspecifikation.

Tabellexempel – relationsobjektet Beställningsspecifikation



Relationsobjekt är inte alltid lika självklara eller enkla att hitta som t ex Student, Kurs eller Artikel; ibland krävs lite tankemöda innan man finner dem...