

**Lösningsförslag till
Fiktiv tentamen för DD1370
Databasteknik och informationssystem**

Torsdag 4 dec 2008

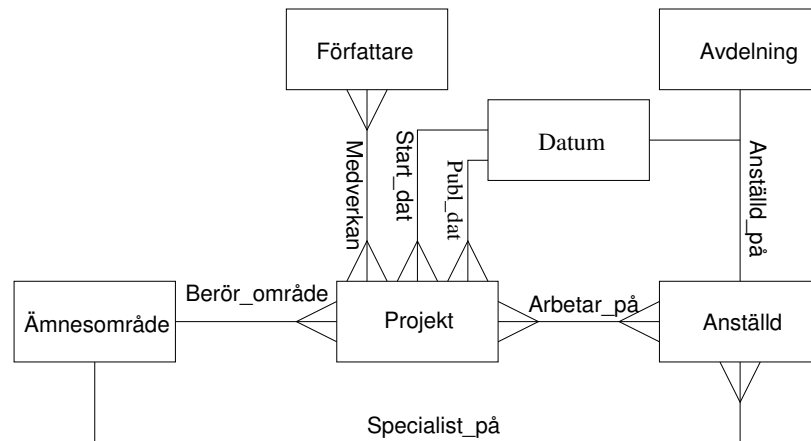
1. a) Jag använder kokbokens regler
 - a. En objektklass som innehåller e-term(-er) bildar en tabell

Ämnesområde	(<u>Änr</u> , Änamn)
Anställd	(<u>Anr</u> , Namn, Adress, Tel)
Projekt	(<u>Pnr</u> , Projektnamn)
 - b. En objektklass som inte har e-termer men finns på N-sidan av någon 1:N-sambandsklass bildar en tabell
Finns inga
 - c. En sambandsklass av högre ordning än 2 bildar en tabell
Finns inga
 - d. En M:N-sambandsklass bildar en tabell

Ämnesområde	(<u>Änr</u> , Änamn)
Anställd	(<u>Anr</u> , Namn, Adress, Tel)
Projekt	(<u>Pnr</u> , Projektnamn)
Medverkan	(<u>Fnamn</u> , <u>Pnr</u>)
Arbetar_på	(<u>Pnr</u> , <u>Anr</u>)
 - e. En 1:N-sambandsklass försvinner men 1-sidans objektklass i-term blir e-term i tabellen som bildas av n-sidans objektklass
Tillagda attribut är markerade med kursiv stil

Ämnesområde	(<u>Änr</u> , Änamn)
Anställd	(<u>Anr</u> , Namn, Adress, Tel, <i>Änr</i> , <i>Avdnamn</i>)
Projekt	(<u>Pnr</u> , Projektnamn, <i>Änr</i>)
Medverkan	(<u>Fnamn</u> , <u>Pnr</u>)
Arbetar_på	(<u>Pnr</u> , <u>Anr</u>)
 - f. En 1:1-sambandsklass hanteras antingen som en 1:N-sambandsklass där man "godtyckligt" väljer en sida som "N-sida" eller tar man bort den och slår samman objektklasserna som sambandsklassen binder samman
Finns inga 1:1-sambandsklasser

b)



med följande egenskapsmatris:

Typ	Namn	I-termen	E-termen
Obj	Författare	Fnamn	
	Projekt	Pnr	Projektnamn
	Ämnesområde	Änr	ÄNamn
	Anställd	Anr	Namn, Adress, Tel
	Avdelning	Avdnamn	
	Datum	Datum	
Samb	Medverkan	Fnamn, Pnr	
	Berör_område	Pnr, Änr	
	Arbetar_på	Pnr, Anr	
	Anställd_på	Anr, Avdnamn, Datum	SlutDatum
	Specialist_på	Anr, Änr	
	Start_dat	Pnr, Datum	
	Publ_dat	Pnr, Datum	

Effekten på DB-strukturen blir att två datum tas med som egenskapstermer i Projekt. Man kan ge dem samma namn som de sambandsklasser som enligt regel *e* försvinner, d.v.s. Start_dat och Publ_dat. Dessutom kommer Anställd_på att bli en tabell med tre termer i primärnyckeln och med egenskapstermen Slutdatum. Slutligen försvinner Avdnamn som extra egenskapsterm i Anställd

2. a) Enligt min bedömning finns följande beroenden:

- Enligt uppgiftstexten beror Adress, Postadress och Rabattsats av Bokhandlare
- Boktitel beror av BokISBN på ett naturligt sätt
- För att få fram försäljningen av en viss bok hos en viss handlare under en speciell månad måste man ha information om alla tre och alltså {Bokhandlare, BokISBN, Månad} -> Sål
- Slutligen kan en bok ha flera författare samtidigt som en författare kan ha varit med om författandet av flera böcker
- Alltså bör man dela upp strukturen på flera tabeller. T.ex. skulle databasstrukturen nedan duga

Handlare (Bokhandlare, Adress, Postadress, Rabattsats)
Författat (FörfattarNamn, BokISBN)
Bok (BokISBN, BokTitel)
Försäljning (Bokhandlare, BokISBN, Månad, Sål)

b) Om man behåller den ursprungliga strukturen kommer man få insättningsproblem eftersom man inte kan sätta in en bok utan att också ange alla andra uppgifter, t.ex. bokhandlare och dess adress eller sätta in nullvärden på många platser.

Man får också borttagningsproblem. Om man tar bort information om en bokhandlare måste man kanske ta bort många rader och om en viss bok bara finns till försäljning hos den bokhandlare man tar bort så försvinner all information om boken.

Man får uppdateringsproblem, om man ska ändra adressen på en bokhandel måste man gå igenom hela databasen och kanske ändra på många ställen.

3.
 - a) Vilken avdelning på andra våningen säljer störst antal olika varor?
 - b) Vilka tjänar mindre än den som tjänar minst på våning 2?
 - c) Vilka företag levererar någon vara som något annat företag också levererar?
4.
 - a) En kandidatnyckel är en minimal mängd attribut som duger för att unikt identifiera en rad i en tabell. Kandidatnycklar är viktiga eftersom en av dem väljs som primärnyckel och de övriga kandidatnycklarna måste tas hänsyn till vid design av en databas.
Inget attribut i en kandidatnyckel får vara null i någon rad.
 - b) Man har ett transitivt beroende i en tabell då något attribut beror av nyckeln via ett annat attribut som inte ingår i en kandidatnyckel. Det visar på en brist i databasstrukturen och bör åtgärdas genom att splittra tabellen i två.