

Inst. för Data- och Systemvetenskap
SU
Maria Bergholtz

Tentamen

21/3 2015 kl. 14 – 18

Inga hjälpmedel är tillåtna (annat än ordbok).

Skall även användas som omtentamen för DB:EIT/DSK/FK samt äldre databaskurser.

Med lösningsförslag (påbörjat) – klart nu.

Skriv bara på en sida av pappret

Skriv läsligt, annars är det omöjligt att rätta tentamen

Studenter som kompletterar FX-uppgift: Skriv att du gör FX-uppgift

Lycka till!

Uppgift 1 (Lärandemål: databasteknik och informationsadministration)

Del 1 till 6 är flervalsfrågor där rätt alternativ ska väljas (bara *ett* är rätt). Välj *ett* alternativ för var och en av flervalsfrågorna nedan.

1) Vilket av följande påståenden är sant om vyer?

- a) En vy kan användas för att låta olika användar-grupper få tillgång till olika typer av information i databasen.
- b) En vy kan aldrig uppdateras.
- c) En vy kan inte användas för att förenkla SQL mot databasen eftersom vyn inte sparas när man lämnar databasen.

A.

2) Vad av följande gäller för en alternativnyckel?

- a) En alternativnyckel får inte innehålla NULL.
- b) En alternativnyckel kan innehålla dubletter.
- c) En alternativnyckel är en fd. kandidatnyckel som inte blivit vald till primärnyckel.

C.

3) Vilket av följande påståenden är sant om relationen mellan sub-klass och superklass i ett UML klassschema?

- a) En subklass kan vara subklass till flera olika super-klasser.
- b) En klass A kan vara subklass till en klass B som i sin tur är subklass till A.
- c) En superklass måste ha minst två sub-klasser.

A.

6) Tabellen RIDNING har en främmande nyckel mot HÄST (beskrivs nedan). Denna nyckel är deklarerad som ON DELETE RESTRICT. Vad betyder det?

HÄST		RIDNING		
<u>HästId</u>	HästNamn	<u>Person</u>	<u>HästId</u>	<u>Datum</u>
1	Man O' War	Lisa	1	120321
2	Ego Boy	Lisa	1	120101
3	Darley Arabian	Lily	4	121231
4	Byerley Turk	John	3	121231
		Lily	1	121231

RIDNING.HästId << HÄST.HästId

- a) Om jag tar bort en rad i tabellen HÄST så tas de rader i tabellen RIDNING som har en främmande nyckel mot den borttagna hästen också bort.
- b) Jag får inte ta bort rader i tabellen RIDNING så länge det finns rader i tabellen HÄST som motsvaras av HästId:n i RIDNING.
- c) Jag får inte ta bort rader i tabellen HÄST så länge det finns rader i tabellen RIDNING som har främmande nycklar mot de rader i HÄST som man vill ta bort.

C.

5) Vilket av följande påståenden gäller för översättning av ett UML klassdiagram till ett relationsschema?

- a) Om man översätter ett UML klass-diagram till ett relationsdatabasschema (via de översättningsregler som gäller för att översätta alla delar i klass-diagrammet till relationsscheman) så erhåller man ett antal relations-scheman som är i minst första normalform.
- b) Om man översätter ett UML klass-diagram till ett relationsdatabasschema (via de översättningsregler som gäller för att översätta alla delar i klass-diagrammet till relationsscheman) så erhåller man ett antal relations-scheman som är i minst andra normalform.
- c) Om man översätter ett UML klass-diagram till ett relationsdatabasschema (via de översättningsregler som gäller för att översätta alla delar i klass-diagrammet till relationsscheman) så erhåller man ett antal relations-scheman som är i minst tredje normalform.

A.

6) Tabellen RIDNING¹ satisfierar följande funktionella beroenden:

HästId → Hästnamn, Mankhöjd, Färg

Hästnamn → Mankhöjd

RIDNING ska nu uppdateras, vilken av satserna nedan kan utföras utan att tabellen bryter mot något av de funktionella beroendena?

HÄST

HästId	HästNamn	Mankhöjd	Färg
1	Man O' War	1.60	Röd
2	Ego Boy	1.62	Grå
3	Darley Arabian	1.67	Röd
4	Byerley Turk	1.51	Brun

- a) INSERT INTO RIDNING VALUES(5, Byerley Turk, 1.61).
- b) INSERT INTO RIDNING VALUES(5, Ego Boy, 1.62).
- c) INSERT INTO RIDNING VALUES(4, Ego Boy, 1.62).

B.

7) Vad innebär att en databas-transaktion är 'atomär' ?

- a) Att en transaktion antingen utförs i sin helhet eller inte alls.
- b) Att en transaktion *inte* kan bestå av flera olika delar, tex inte både en INSERT följt av en UPDATE.

¹ OBS, skall vara tabellen HÄST (som den är benämnd i bilden).

c) Att en transaktion alltid utförs på snabbast möjliga sätt.

A.

I del 8 ska giltigheten i följande utsagor diskuteras. Om du tycker påståendet är sant så skriv det och motivera sen varför, om du tycker påståendet är falskt så skriv det och motivera varför. Motivera utförligt.

8) Varför behövs lås när man utför en transaktion mot en databas? Ge exempel på någon typ av princip för att tilldela lås till en transaktion och vad den innebär.

Här kan man fokusera på olika specifika problem eller bara skriva generellt om konkurrerande transaktioner, dvs om (minst) två olika transaktioner vill ha tillgång till (läsa eller skriva) resurser (hela tabeller, vissa rader, kolumner etc) i databasen uppstår konflikter om data ändras av någon av transaktionerna. Med konflikter menas här att databasen inte är konsistent med avseende på att båda transaktionerna hela tiden ser samma data och ändrar i samma data. För att omöjliggöra att transaktioner ser fel data eller att uppdateringar utförs fel eller förloras låser man olika typer av resurser som transaktionerna efterfrågar så att alla ändringar genomförs som skedde de i sekvens och inte parallellt. Alla lås-scheman går ut på att låsa så lite som möjligt med bibehållet krav på att data hela tiden är konsistent map konkurrerande användare. En typ av lås-schema är sk. Two-Phase-Locking där en trans antal begärda och erhållna lås först växer tills alla resurser som behövs för transen uppnåtts, därefter släpps låsen varefter de inte behövs.

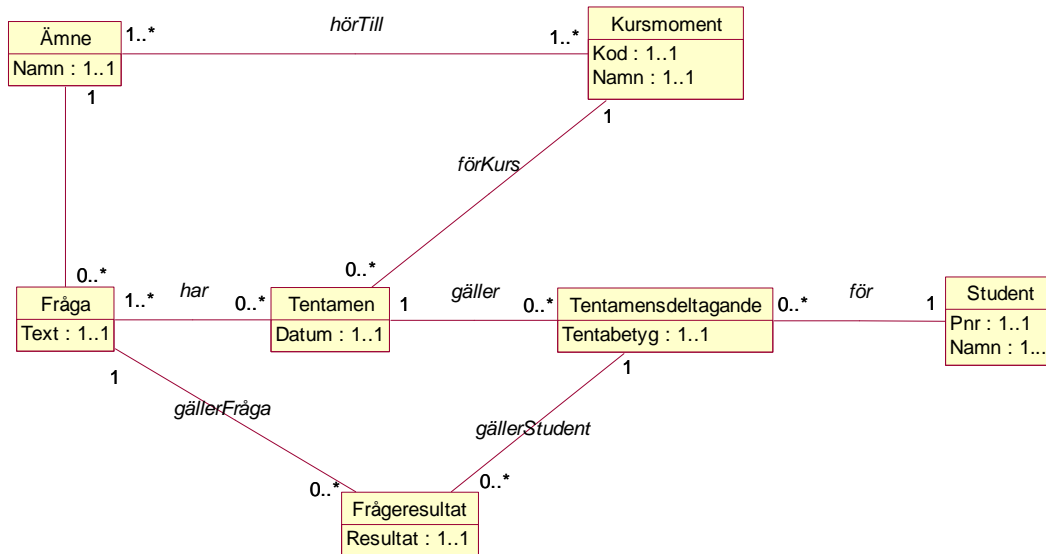
Uppgift 2 (Lärandemål: modellering)

KONSTRUERA ETT KONCEPTUELLT SCHEMA i form av ett UML klassschema som gör det möjligt att representera en tentamensdatabas (beskriven nedan):

Vi undervisar ju på flera kursmoment och varje kursmoment avser vanligen ett antal ämnen. Ett ämne, t ex konceptuell modellering ingår förstas i flera kurs moment (FK:DB, DSV1:DB, GES:OOS o s v). I databasen skall olika tentor lagras och för varje tenta skall man kunna få fram vilka frågor/uppgifter som den bestod av och vilket datum den gavs. En fråga kan ibland användas på flera olika tentor. Varje fråga i databasen är tänkt att examinera ett ämne. Skall studenterna vara med i databasen?? Jo, vi tror att det vore bra. Vi vill kunna se vilka studenter som har skrivit en viss tenta så att vi kan undvika att tråka ut dem med att ställa frågor de tidigare har stött på i tentasammanhang. Och det bör ju då framgå vilket kursmoment studenten tenterat. Skall vi ha in resultatet också? JA när vi ändå är på gång och lagrar vilka tentamina en student skrivit så är det väl lika bra att lagra hur det gick. Då vill vi förmodligen se hur det gick på en viss fråga för en student inom en tenta. Rita ett UML klassschema för ovanstående beskrivning. Jag har skissat lite själv och det blev inte så märkvärdigt. Max 8 klasser skulle jag tro.

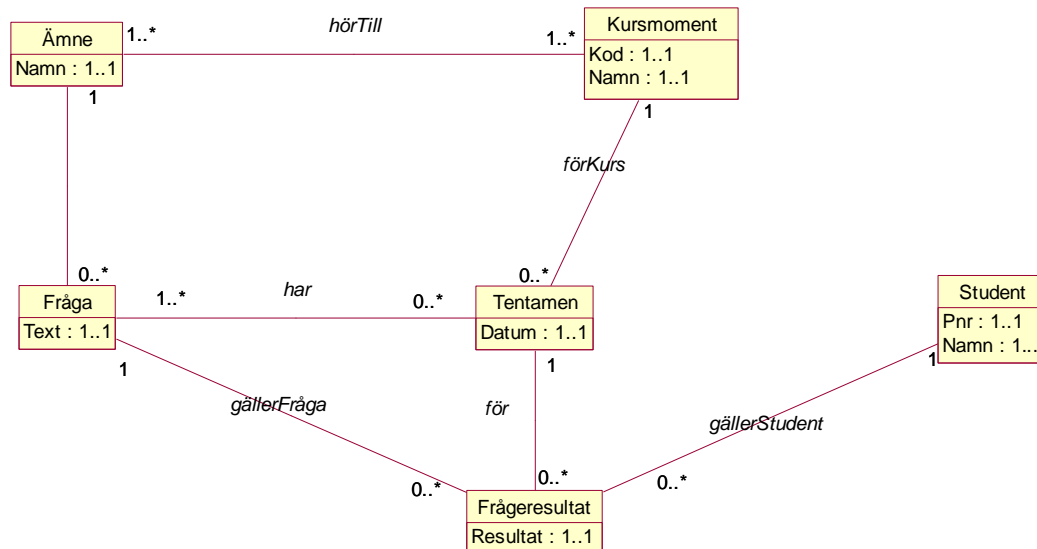
Modellera på ett sådant sätt att partiella attribut (attribut med min-värde = 0) inte förekommer! Ange avbildningsregler (min/max-värden) för alla roller i alla associationer och för alla attribut. Ange namn på alla associationer. Datatyper behöver INTE anges.

Lösningsförslag:



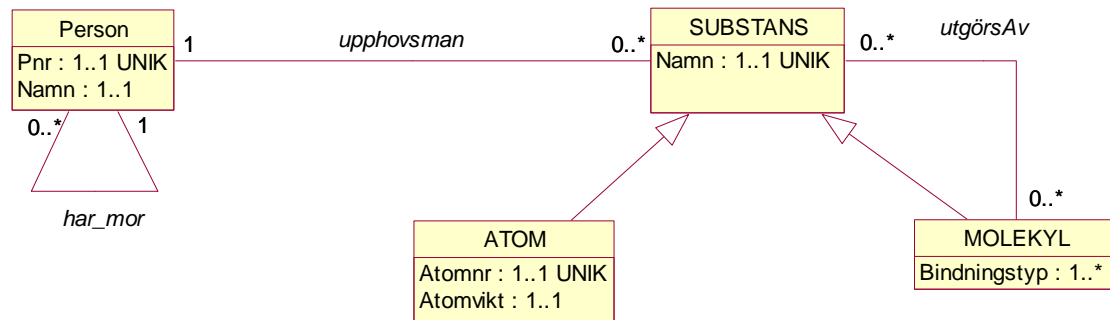
Man kan ha fler associationer, tex mellan student och Kursmoment och Tentadeltagande och Kursmodment, men dessa är redundanta (kan härledas ändå), inget större fel att ha med dem om man vill.

Tentadeltagande kan tas bort om man inte bryr man sig inte om att hålla reda på slutresultat för tentamen, vill man lagra detta samt kanske även andra egenskaper för tentadeltagandet bör klassen finnas. Så här blir det utan deltagandet (dvs här modelleras deltagandet som ett antal frågeresultat för en tenta och en student):



Uppgift 3 (Lärandemål: Syntetisk databasdesign (översättning från konceptuell modell till logiskt relationsdatabasschema))

Betrakta följande konceptuella schema.



ÖVERSÄTT DET KONCEPTUELLA SCHEMAT OVAN TILL ETT RELATIONS DATABASSCHEMA. Ange för varje relation vad som utgör primärnyckel samt vad som eventuellt utgör främmande nycklar (främmande nycklar måste specificeras med alla korrekta kolumner). I fallet främmande nycklar skall även specificeras mot vilken relation de utgör främmande nyckel. Översättningen *får ej* innebära att avsteg från den konceptuella modellen görs (annat än de avsteg som *måste* göras för att realisera relationsmodellen). **Surrogatnycklar får inte införas.**

Använd följande notation med understrykning av primärnycklar:

OMEGA(o1, o2, o3),

PI(p1, o1, o2, p2)

Tabellen PI innehåller två attribut o1 och o2 som utgör främmande nyckel mot tabellen OMEGA. Detta skrivs på följande sätt:

PI.(o1, o2) utgör FN mot OMEGA.(o1, o2)

Lösningsförslag:

PERSON(Pnr, Namn, Mor), PERSON.Mor utgör FN mot PERSON.Pnr

SUBSTANS(Namn, minupptäckare), SUBSTANS.minupptäckare är FN mot PERSON.Pnr

ATOM(atomnummer, atomvikt, substans), ATOM.substans är FN mot SUBSTANS.Namn,
ATOM.substans utgör även AN.

MOLEKYL(substans)

MOLEKYLBINDNING(bindning, molekyl), MOLEKYLBINDNING.molekyl är FN mot MOLEKYL.substans

UTGÖRSAV(molekyl, substans), UTGÖRSAV.molekyl är FN mot MOLEKYL.substans och
UTGÖRSAV.substans är FN mot SUBSTANS.Namn

Uppgift 4 (Lärandemål: Analytisk databasdesign (normalisering))

Betrakta följande relationsschema:

DELTAGARSKAP(Personnummer, Datum, Namn, Adress, Telefonnr, rumsnummer)

Följande funktionella beroenden gäller:

Personnummer, Datum → Namn, Adress, Telefonnr, rumsnummer
Personnummer → Namn, Adress, Telefonr

- a) Bestäm primärnyckel för tabellen DELTAGARSKAP
- b) DELTAGARSKAP förutsätts vara i 1NF. Normalisera DELTAGARSKAP till 3NF. Motivera dina svar. Gör det stegvis, om du anser att en tabell inte är i 2NF så överför den till 2NF först och sen till 3NF, anser du att tabellen redan är i 2NF så motivera varför och överför sen till 3NF. Hoppa inte över något steg.
- c) Motivera varför en högre normalform är att föredra framför en lägre. Finns det några skäl att inte normalisera?

Lösningförslag:

- a) Primärnyckel för DELTAGARSKAP blir Personnummer + Datum. Detta eftersom Personnummer + Datum bestämmer övriga attribut funktionellt.
- b) I uppgiftstexten anges att tabellen är i 1NF. Vi ska nu överföra tabellen till högre normalform. Vi har ett partiellt beroende mellan alla personuppgifter och PN, dvs personuppgifterna bestäms enbart av Personnummer – således är vi ej i 2NF. Detta leder till att vi inte heller är i 3NF eftersom 3NF förutsätter att tabellen ifråga är i 2NF.

Vi överför tabellen till 2NF:

PERSON(Personnr, Namn, Adress, Telefon) där Pnr är primärnyckel

Kvar i den gamla tabellen blir:

DELTAGARSKAP(Personnr, Datum rumsnummer) där DELTAGARSKAP.Personnummer är FN mot PERSON.Personnr

Båda tabellerna är nu i 2NF. Och även i 3NF eftersom vi inte har något funktionellt beroende mellan icke nyckel-attribut som skulle kunna ge upphov till ett transitivt funktionellt beroende mellan ett icke nyckel-attribut och PN

c)

Man normaliserar för att undvika redundans, varje faktum ska lagras på ett ställe, inte spridas över fler tabeller (med undantag av kontrollerad redundans i form av främmande nycklar). Detta för att undvika uppdateringsanomalier men också för att spara plats då en normaliserad databas tar oftast mindre plats (pga att redundansen minskar). Bryr man sig bara om att minska redundansen så är en högre normalform bättre än en lägre. Ett normaliserat databasschema kommer dock att innehålla fler tabeller än ett onormaliserat, det blir alltså mer komplext och eventuellt svåröverskådligt. Hög normaliseringsgrad kan också leda till att transaktioner tar längre tid att genomföra pga att fler joins mellan de många nya tabellerna måste göras. Är det en ofta förekommande transaktion (-styp) som tar längre tid är detta en stor nackdel och man kan i optimeringssyfte välja att de-normalisera.

Uppgift 5 (Lärandemål: frågespråk (SQL och relationsalgebra))

Betrakta följande tabeller (Primärnyckel-kolumner understrukna och i fetstil):

Hotell

<u>Hotnr</u>	Hnamn	Stad	Telefon
10	Royal	Mariestad	021-456785
32	Skandia	Lidköping	032-876543
44	Royal	Strömstad	0123-98765
56	Cecil	Göteborg	031-300000
64	Viktoria	Umeå	0643-65432

Rum

<u>Rumsnr</u>	<u>Hotnr</u>	Anmärkning
20	10	Ej rökare
21	10	Rökare, husdjur
22	10	
25	10	Ej rökare
32	10	Allergisanerat
113	32	Rökare
117	32	
211	32	Husdjur
20	44	
22	44	Rökare
3219	56	Ej rökare
4321	56	Ljudisolerat
233	64	
433	64	Rökare
633	64	

RUM.Hotnr är FN mot HOTELL.Hotnr

Utrustningstyp

Benämning	Anmärkning	Ukod
Dator	WinXP, Internet, DVD	PC
CD-spelare		CD
TV 28"	text-tv	TV28
Kaffebryggare	8 koppar	KB
XBOX 360	Växlare med 100 spel	XB
Jacuzzi		JZ
TV 32"	platt, text-tv, PiP	TV32
DVD-spelare		DVD
Massagestol		MS

Rumsutrustning

Rumsnr	Hotnr	Ukod
21	10	TV28
22	10	XB
22	10	TV32
113	32	CD
117	32	PC
211	32	TV32
1234	44	KB
1664	44	DVD
1664	44	TV32
3219	56	TV28
3219	56	MS
4321	56	PC
233	64	KB
433	64	PC
633	64	JZ

RUMSUTRUSNING.(Rumsnr, Hotnr) är FN mot RUM.(Rumsnr, Hotnr)
 RUMSUTRUSNING.Ukod är FN mot UTRUSTNINGSTYP.Ukod

Relationalsgebraiska operatörer:

Selektion σ , Projektion π , Theta-join θ , Natural join \bowtie , Kartesisk produkt \square

Division \div , Snitt \cap , Union \cup , Differens $-$, Aggregering/gruppering G_f

Tilldelning \leftarrow , Namnändring ρ

SQL-syntax:

```
SELECT [DISTINCT] <attributlista>
FROM <tabellista>
[WHERE <villkorsuttryck>]
[GROUP BY <kolumnlista>
      [HAVING <villkorsuttryck>]]
[ORDER BY <kolumnlista>]
```

Formulera svar i SQL:

- a) Ta fram namn på hotell som har rum med Jacuzzi.
- b) Hur många rum med Kaffebryggare finns på de olika hotellen?
Ordna resultatet i stigande ordning på hotellnamn.

Formulera svar i relationalsgebra:

- c) Ta fram namn på hotell som har rum med Jacuzzi.
- d) I vilka rum finns alla typer av utrustning?

Lösningsförslag:

```
a)
SELECT Hnamn
FROM Hotell H, Rumsutrustning RU, Utrustningstyp U
WHERE H.Hotnr=RU.Hotnr
AND RU.Ukod=U.Ukod
AND Benämning='Jacuzzi'
```

Kommentar: Man behöver inte ta med RUM i joinen eftersom RUMSUTRUSTNING har en främmande nyckel mot RUM där även hotellets PN ingår. Om man tar med tabellen och gör en extra onödig join så är det ok ändå.

b)

Här kan man göra en stor join först för att sedan gruppera och räkna.

```
SELECT Hnamn, COUNT(*) AS Antal
FROM Hotell H, Rumsutrustning RU, Utrustningstyp UT
WHERE RU.Ukod = UT.Ukod AND H.Hotnr = RU.Hotnr
AND Benämning = 'Kaffebryggare'
GROUP BY H.Hotnr, Hnamn
ORDER BY Hnamn
```

I den här lösningen får man med endast hotell som har minst ett rum med Kaffebryggare. Om man vill få med alla hotell kan man tänka sig flera olika varianter. En variant skulle vara att ändra lösningen ovan så att den gör en outer join och sedan göra COUNT på en av de kolumner som joinen sätter NULL. En annan (kanske enklare) lösning är följande, där vi går igenom samtliga hotell och räknar antal rum i en egen SELECT-sats:

```
SELECT Hnamn, (SELECT COUNT(*)
                FROM Rumsutrustning RU, Utrustningstyp UT
                WHERE RU.Ukod = UT.Ukod AND H.Hotnr = RU.Hotnr
                AND Benämning = 'Kaffebryggare') AS Antal
FROM Hotell H
ORDER BY Hnamn
```

Även lösningar som endast tar med hotell med minst ett rum fått fullpoäng.

c) Vi kan börja med det givna villkoret och stega fram till det vi vill ha i resultatet. Vi börjar med selektionsvillkoret för att hitta rätt utrustningstyp och sedan joinar vi till vi kan komma åt hotellnamnen:

$$\Pi_{Hnamn} (\text{Hotell} \bowtie \text{Rumsutrustning} \bowtie_{\sigma_{\text{Benämning}='Jacuzzi'}} \text{Utrustningstyp})$$

Observera att vi inte behöver använda tabellen Rum eftersom vi redan har hotellets nummer från tabellen Rumsutrustning. Projektionen i slutet tar bort automatiskt eventuella dubletter. Man kan också tänka sig en lösning med flera projektioner så att man inte bär med oss onödiga kolumner. Att vänta med selektionen tills efter alla joins är också möjligt.

d)

$$\text{HotellMedUtrustning} \leftarrow \Pi_{\text{Hotnr, Ukod}} (\text{Hotell} \bowtie \text{Rumsutrustning} \bowtie \text{Utrustningstyp})$$

$AllUtrustning \leftarrow \Pi_{Ukod} Utrustning$

$SVAR \leftarrow HotellMedUtrustning \text{ KVOT } AllUtrustning$

Nu har vi ju i och för sig bara Hotellens surrogatnyckel så vi gör en sista join med Hotell för att få fram hotellegenskaper, tex hotellets namn (vi kunde även projicerat fram namn direkt i täljaren)

$Svarmedhotellnamn \leftarrow \Pi_{HName}(Hotell \bowtie SVAR)$