

Exempel-tentamen med lösning EIT:DB

Databasmetodik

11 januari 2012

Inga hjälpmedel tillåtna (kort syntaxsammanställning finns i slutet av denna tentamen)

Skriv bara på en sida av pappret
Skriv namn på varje papper
Skriv läsligt, annars rättas inte tentamen

Lycka till!

Uppgift 1 (svarar mot lärandemål 1 avseende databasteknik och informationsadministration)

Uppgift 1 består av 10 flervalsfrågor där rätt alternativ ska väljas (bara *ett* är rätt). Välj ett alternativ för var och en av flervalsfrågorna nedan. **Rätt lösningsrad kommer efter sista frågan!**

1) Vad innebär redundans i databasen?

- a) Att flera kolumner (i olika tabeller) heter samma sak.
- b) Att samma information lagras på flera ställen.
- c) Att en association/koppling går från och till samma tabell.

2) Vilket av följande påståenden är sant om entitetsintegritet?

- a) En sammansatt primärnyckels ingående kolumner måste alla var för sig vara dubblettfria.
- b) Om primärnyckeln består av flera kolumner får högst en av de kolumnerna vara NULL.
- c) Alla kandidatnycklar i en tabell kan väljas till att bli primärnyckeln i den tabellen.

3) Unionskompatibilitet krävs för de relationsalgebraiska operationerna:

- a) projektion och selektion
- b) differens och snitt
- c) theta-join och natural join

4) Vilket av följande måste alltid finnas i en tabell som följer relationsmodellen?

- a) En primärnyckel
- b) En surrogatnyckel
- c) En alternativnyckel

5) Vad av följande gäller för 2PL (Two-Phase Locking)?

- a) En transaktion kan endast begära ett nytt lås om den inte redan har släppt ett annat lås.
- b) En transaktion kan få ett nytt lås trots att den redan släppt ett lås, om den körs mot endast en tabell.
- c) En transaktion kan begära ett nytt lås om den först släpper ett lås den fått tidigare.

6) En relations/tabells grad är:

- a) antalet attribut/kolumner i relationen/tabellen.
- b) exakt samma sak som dess intension.
- c) antalet rader i relationen/tabellen

7) Vilket av följande är inte sant vad gäller nycklar?

- a) En kandidatnyckel som inte väljs till primärnyckel blir alternativnyckel.
- b) Främmande nyckel-kolumner kan ingå i en primärnyckel.
- c) En supernyckel kan alltid bli primärnyckel.

8) Enligt ANSI-SPARC 3-nivåersarkitektur, vilket av följande är sant?

- a) Logiskt dataoberoende innebär att man kan göra ändringar i interna nivån utan att konceptuella nivån påverkas.
- b) Logiskt dataoberoende innebär att man kan göra ändringar i konceptuella nivån utan att interna nivån påverkas.

c) Logiskt dataoberoende innebär att man kan göra ändringar i konceptuella nivån utan att externa nivån påverkas.

9) Vilket av följande är sant vad gäller referensintegritet?

- a) En främmande nyckel får aldrig vara NULL.
- b) En främmande nyckel får aldrig innehålla dubletter.
- c) En främmande nyckel som inte tillåts vara NULL, måste matcha ett exakt värde i primärnyckeln i den tabell den främmande nyckeln refererar till.

10) 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.
- c) Att en transaktion alltid utförs på snabbast möjliga sätt.

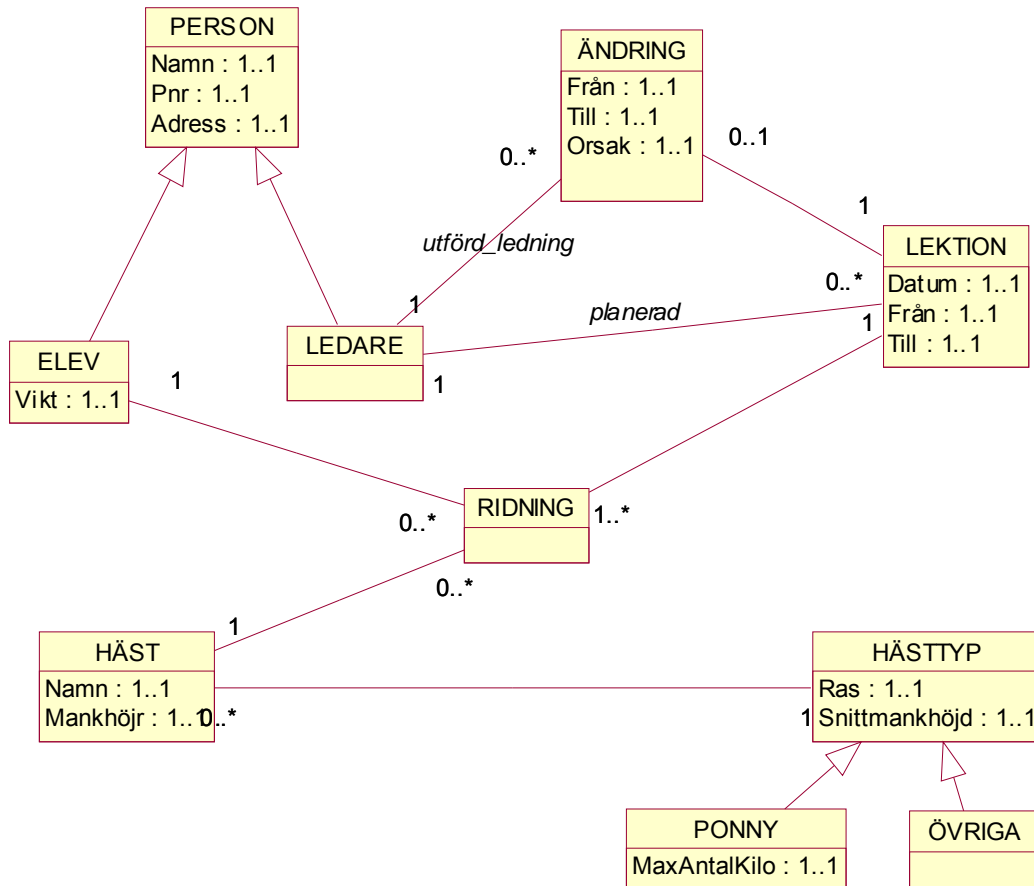
Rätt lösningsrad: b c b a a a c c c a

Uppgift 2(svarar mot lärandemål 2 avseende konceptuell modellering och databasmodellering)

KONSTRUERA ETT KONCEPTUELLT SCHEMA som modellerar verksamheten hos en ridskola. Schemat ska möjliggöra representation av samtliga typer av utsagor nedan:

- Ridskolan erbjuder ridlektioner till elever.
- Ridlektioner leds av en ledare.
- Lisa och Kalle arbetare som ledare på ridskolan.
- Anna och Olle är elever på ridskolan.
- Olle väger 61 kg, Lisa väger 55 kg.
- Ridlektionen 21 december kl. 13 – kl. 14 går av stapeln i sal A.
- Det går även en ridlektion i sal A kl. 16 – 17 21 december.
- Sal B används för lektionen 23 december.
- Lisa är bokad som ledare till ridlektionen som går 21 december kl. 13 – kl. 14.
- Hästen 'Darley Arabian' rids av Olle på ridlektionen 21 december.
- 'Darley Arabian' rids av Anna på ridlektionen 22 december.
- Anna rider 'Bayerly Turk' på lektionen 23 december.
- Anna rider 'Ego Boy' på lektionen 21 december.
- Kalle ledde ridlektionen som gick 21 december kl. 13 – kl. 14 (pga att Lisa blev sjuk)
- 'Darley Arabian' är ett arabiskt fullblod (mankhöjd 1.62).
- 'Ego Boy' är en ponny av rasen Connemara.
- 'Ego Boy' har en mankhöjd på 1.35 men Connemara-ponnyer i allmänhet har en mankhöjd som ligger 10 cm lägre.
- Arabiska fullblod kan ridas av alla oberoende av vikt.
- Ponnyer kan ridas av personer som väger mindre än 60 kg.

Partiella attribut (attribut som har minvärde 0) ska undvikas!



Schemat ovan får väl sägas vara en minimal lösning, man måste dock inte ha med alla klasser för att bli godkänd, se betygskriterier i Daisy för EIT:DB HT12.. Rättning: En viktig klass är klassen RIDNING (också benämnd lektionstillfälle, närvaro etc.). Det räcker alltså inte att relatera hästar, elever etcetera till en och samma ridlektion. Man måste ju också hålla reda på vem som red vilken häst på denna lektion. Klassen RIDNING är egentligen ett relationsobjekt mellan tre klasser: LEKTION, ELEV och HÄST. Sen skulle man även hålla reda på vem som varit ledare på en lektion. Det kunde bli så att den som var planerad att leda inte gjorde det och i så fall vill man se det också. Här finns flera möjliga lösningar som alla har det gemensamt att klassen LEKTION modelleras via två klasser. Dels LEKTION, dels någon annan klass som antingen modellerar den bokade lektionen eller en klass som modellerar den fullbordade lektionen (det är inte så hankigt om man väljer det ena eller andra). Så någon typ av klass (kanske kallad bokad_lektion, fullbordad_lektion eller helt kort och gott ändring) behövs förutom LEKTION. HÄST och HÄSTTYP är exempel på en konkret typ och dess powertype. Man bör här skilja på egenskaper hos arten, t ex *snitt*-mankhöjd och egenskaper hos den enskilda hästen, t ex mankhöjd. Eventuellt kan man dela upp HÄSTTYP:en i PONNY:er och ÖVRIGA. Eftersom just ponnyer hade en maxvikt som de klarar av. Man kan även lägga den egenskapen i HÄSTTYP och strunta i ponnyer och övriga.

Uppgift 3 (svarar mot lärandemål 4 avseende analytisk databasdesign)

- a) Konstruera en tabell som är i 2NF men inte är i 3NF. Ange namn på tabellen, primärnyckel, vilka funktionella beroenden som rådet och lista några exempel-rader.

PERSON

Pnr	Namn	Gata	Postnr	PostOrt
11111	Eva	Byv.3	12345	Säfte
22222	Lisa	Byv. 5	12345	Säfte
33333	Olle	Solstigen 3	34111	Tofsoda
44444	Maria	Solstigen 4	34222	Tofsoda
55555	Klas	Solsigen 5	34111	Tofsoda

Pnr är primärnyckel i PERSON.

Följande funktionella beroenden råder mellan kolumnerna i PERSON:

Pnr → Namn, Gata, Postnr, PostOrt
Postnr → PostOrt

Då är alla attribut beroende av hela nyckeln (den är ju inte sammansatt), dvs vi är i 2NF. Sen har vi ett transitivt beroende mellan primärnyckeln och kolumnen PostOrt. Detta eftersom Pnr bestämmer Postnr funktionellt och Postnr i sin tur bestämmer PostOrt funktionellt (som ses ovan). Eftersom vi har ett transitivt beroende mellan nyckeln och ett icke-nyckel-attribut så är vi inte i 3NF.

- b) Överför, med utgångspunkt från svaret i a) tabellen till 3NF. Lista de nya raderna för de nya tabellerna. Ange nya tabellnamn, primärnycklar och främmande nycklar.

Två nya tabeller i 3NF:

PERSON

Pnr	Namn	Gata	Postnr
11111	Eva	Byv.3	12345
22222	Lisa	Byv. 5	12345
33333	Olle	Solstigen 3	34111
44444	Maria	Solstigen 4	34222
55555	Klas	Solsigen 5	34111

POST

Postnr	PostOrt
12345	Säfte
34111	Tofsoda
34222	Tofsoda

där PERSON.Postnr utgör främmande nyckel mot tabellen POST.

- c) Med utgångspunkt från svaren i a), b), motivera varför man normaliserar! Exemplet bör innefatta det ni gjort i uppgift a-b. Finns det något som talar mot att normalisera?

Man normaliserar först och främst för att *undvika redundans*. Varje faktum på en säger en plats enbart. Redundans kan leda till så kallade uppdateringsanomalier. Vill man *tex ändra* på

vilken postOrt ett visst postnummer hör till så måste man ändra på alla ställen som detta postnummer är nämnt se fråga a) – risk för inkonsistens om ngn rad blir glömd (då kan ju ett och samma postnummer råka peka ut OLIKA postorter).

Dessutom *vinner man i plats*. Databasen blir vanligen mindre. Det kan man t ex se i svaret på b) där man bara behöver ange på ETT ställe att postnumret 34111 verkligen pekar ut Tofsboda. I a) var man tvungen att upprepa detta faktum på varje rad i tabellen PERSON som råkade ha det postnumret. I vissa undantagsfall kan man få en större databas (om främmande nycklarna som införs i och med normaliseringen är längre än de attribut som inte upprepas lika många gånger och/eller om antalet rader i databasen är få.

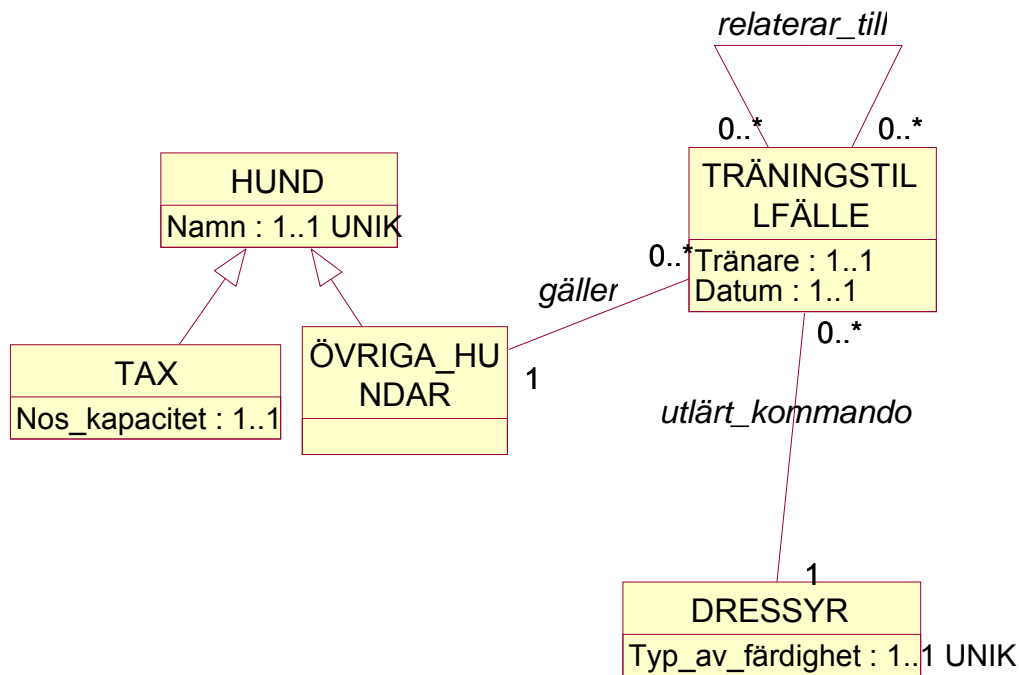
Så vad *förlorar man då, jo oftast i tid*. En godtycklig transaktion mot databasen kommer att ta längre tid eftersom den förmodligen involverar fler tabeller, läs fler joins behöver göras.

Uppgift 4 (svarar mot lärandemål 3 avseende syntetisk databasdesign)

Översätt det konceptuella schemat nedan till ett relationsdatabasschema. Ange alla primärnycklar och främmande nycklar. Vid översättningen får inga förändringar av det konceptuella schemat införas (annat än de som innebär att en översättningsregel tillämpas). Surrogatnycklar får inte införas (om de inte redan finns i schemat). Använd följande notation:

TABELL1(Prim1, Prim2, Kolumn1), där (Prim1, Prim2) är primärnyckel för TABELL1
TABELL2(Prim, For1, For2), där Prim är primärnyckel för TABELL2 och där (For1, For2) utgör främmande nyckel mot TABELL1.

Det sista kan även skrivas TABELL2.(For1, For2) << TABELL1.(Prim1, Prim2)



HUND identifieras av sitt Namn

DRESSYR identifieras av vilken typ av färdighet dressyren gäller

TRÄNING identifieras av sitt Datum plus sin association mot ÖVRIG_HUNDAR och sin association mot DRESSYR.

Övriga klassers identifierare framgår av schemat.

HUND(Namn)

TAX(Namn, Noskapacitet), TAX.Namn << HUND.Namn

ÖVRIGA_HUNDAR(Namn), ÖVRIGA_HUNDAR.Namn << HUND.Namn

TRÄNINGSTILLFÄLLE(Hund, Typ, Datum, Tränare), TRÄNINGSTILLFÄLLE.Hund << HUND.Namn,
TRÄNINGSTILLFÄLLE.Tränare << DRESSYR.Typ

DRESSYR(Typ_av_färdighet)

RELATERAR_TILL(Hund1, Typ1, Datum1, Hund2, Typ2, Datum2), RELATERAR_TILL.(Hund1, Typ1, Datum1) << TRÄNINGSTILLFÄLLE.(Hund, Typ, Datum), RELATERAR_TILL(Hund2, Typ2, Datum2) << TRÄNINGSTILLFÄLLE.(Hund, Typ, Datum)

Uppgift 5 (svarar mot lärandemål 5, frågespråk mot relationsmodellen)

Databasschemat nedan modellerar det faktum att hästar har färger och mankhöjder, personer har adresser och personer rider på olika hästar. Följande främmande nyckel-regler gäller:

RIDING.Horse << HORSE.HorseName

RIDING.Person << PERSON.Name

(Primärnycklarna i tabellerna nedan är understrukna och skuggade).

a) Översätt följande till SQL: Vilka personer aldrig ridit hästen 'Darley Arabian'?

```
SELECT Name
FROM PERSON
WHERE Name NOT IN (SELECT Person
FROM RIDING
WHERE Horse = 'Darley Arabian')
```

b) Översätt följande till SQL: Vilken person har ridit flest hästar men aldrig ridit 'Darley Arabian'?

Först en hjälpsam vy:

```
CREATE VIEW ridna (Name, Antal) AS
(SELECT Person, count(Horse)
FROM RIDING
WHERE Horse <> 'Darley Arabian'
AND Person NOT IN (SELECT Person FROM RIDING WHERE Horse='Darley Arabian')
GROUP BY Person)
```

```
SELECT Name
FROM PERSON
WHERE Name IN (SELECT R.Name
FROM ridna R
WHERE R.Antal = (SELECT max(Antal) FROM ridna))
```

I exemplen ovan ska koden vara generell och inte basera sig på att exempel-databasen nedan ser ut på ett visst sätt (just nu).

c) Vad blir resultatet (rita tabell med rader) om följande relationsalgebraiska uttryck exekveras (basera svaret på exempel-relationerna nedan)?

$\Pi_{\text{height}} (\sigma_{\text{height} < 161} (\text{HORSE}))$

Height

160

149

d) Vad blir resultatet (rita tabell med rader) om följande relationsalgebraiska uttryck exekveras (basera svaret på exempel-relationerna redan)?

$$\text{TEMP}(\text{Horse}) \leftarrow \pi_{\text{Horsename}} (\text{HORSE})$$

$$\pi_{\text{Person}} (\text{RIDING DIVISION TEMP})$$
PERSON

Lisa

HORSE

Horsename	Colour	Height
Ego Boy	Grey	160
Darley Arabian	Red	167
Bayerly Turk	White	149
Dart Hannover	Black	165
Lime Rodney	Black	160

PERSON

Name	Adress
Lisa	Byv. 3
Eva	Byv. 8
Kalle	Solstigen 4

RIDING

Horse	Person
Ego Boy	Lisa
Ego Boy	Kalle
Lime Rodney	Lisa
Bayerly Turk	Lisa
Dart Hannover	Lisa
Darley Arabian	Lisa

Relationsalgebraiska operatorer:

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

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>]