Tentamen för DD1370
Databasteknik och informationssystem

13 Mars 2014

Hjälpmedel:
Inga hjälpmedel utom papper och penna

Tänk på:
Skriv *högst* en uppgift på varje blad.
Använd *endast framsidan* på varje blad.
Skriv *namn och personnr* på varje blad.
Uppgifterna kommer inte i svårighetsordning.
Skriv tydligt, motivera svaren – endast begriplig och läsbar
lösning ger poäng.
Maximal poäng finns angiven inom parentes vid varje uppgift.
Totalt ger tentamen en poäng (max 65), som sedan läggs ihop med era bonuspoäng.
En summa på 40 ger säkert godkänt.
(lösningsförslag kommer på kurswebben)

Lycka till,

Petter

1. (*Totalt: 16p*)

   a) (*2p*) Förklara kopplingen mellan en vy (view), och härledda attribut.

   Lösning: En vy är påminner om en tabell, men har inga egna data. Istället beräknas vyn från data i andra tabeller genom en sql-fråga som uppdateras varje gång data uppdateras. Ett härlett attribut är ett attribut som beräknas utifrån andra data. Således kan en vy med fördel användas för att beräkna ett härlett attribut.

   b) (*2p*) Förklara skillnaden mellan group by, sort by, och order by?

   Lösning: group by är ett SQL-uttryck som används för att gruppera data för aggregerade funktioner så som sum eller avg. sort by är inte ett SQL-uttryck. order by är ett SQL-uttryck för att sortera en tabell.

   c) (*3p*) Vad är en svag entitetstyp? Förklara nyttan, ge exempel och visa/beskriv hur de ritas i ER-diagram.

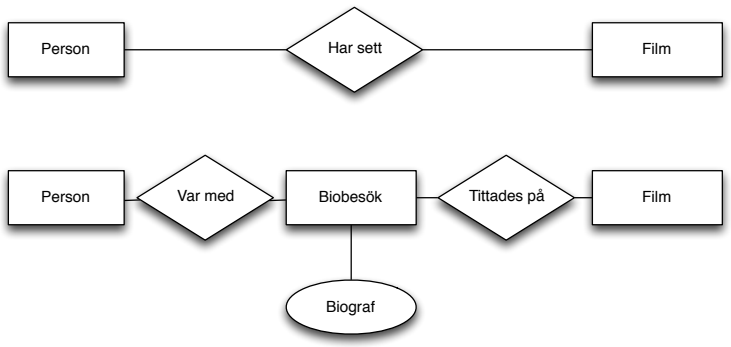
   Lösning: En svag entitet är en entitet som är beroende av en annan entitet. T.ex. kan de olika rummen i en lägenhet sägas vara beroende av själva lägenheten. Nyttan är att man då tvingas ta bort rummen, om man tar bort lägenheten. Svaga entiteter ritas med dubbelstreck-fyrkant. Se även boken s.37.

   d) (*2p*) Förklara förkortningarna, och beskriv skillnaden mellan OLAP och OLTP och ge exempel på användning.

   Lösning: OLAP - On Line Analytical Processing. OLTP - On Line Transaction Processing. OLAP används för beslutsstödssystem, mycket data som ändras sällan, komplexa frågor som skapas efter hand. Ex: aktiekurser eller fastighetspriser. OLTP används för driftdatabaser, data ändras ofta, återkommande frågor. Ex: varulager, kund- och order-hantering.

   e) (*2p*) Vad är en objektifiering av ett samband? Förklara, ge ett exempel, och rita ett ER-diagram hur det ser ut före och efter objektifieringen.

   Lösning:



Figur 1: Överst ett vanligt samband. Nederst ett exempel hur det kan se ut då sambandet objektifieras.

Se figur 1. Genom att objektifiera sambandet *har sett* till entiteten *Biobesök* kan vi t.ex. lagra flera olika tillfällen då samma person har sett samma film.

f) (*5p*) Antag att tabellerna X och Y är givna enligt nedan. Rita tabellen man får som resultat av `SELECT * FROM X,Y .`

X=

Namn	Nummer
Kalle	2
Lisa	5

Y=

Namn	Nummer
Adam	2
Anna	4

Lösning: Resultatet blir följande

Namn	Nummer	Namn	Nummer
Kalle	2	Adam	2
Kalle	2	Anna	4
Lisa	5	Adam	2
Lisa	5	Anna	4

Notera att antalet kolumner är 2+2, dvs 4, och antalet rader är 2*2, dvs 4.

2. (*Totalt: 10p*) En hästuppfödare/tränare i travbranchen använder en databas för att administrera sin verksamhet. Databasen har följande struktur:

Ägare(PNr, Namn, ÄgarAdress)

Travhäst(HästId, Födelseår, FarHästId, MorHästId, Ägare)

Tävlar(HästId,KuskId,TävlingsNr,Anmäld, Placering, Prispengar)

Kusk(KuskId, Knamn)

Tävling(TävlingsNr, Tnamn, Datum)

Email(Emailadress, Pnr)

a) (*5p*) Rita upp en ER-modell som skulle resultera i ovanstående Databasstruktur. Motivera varje steg genom angivande av vilka regler i ”kokboken” som använts.

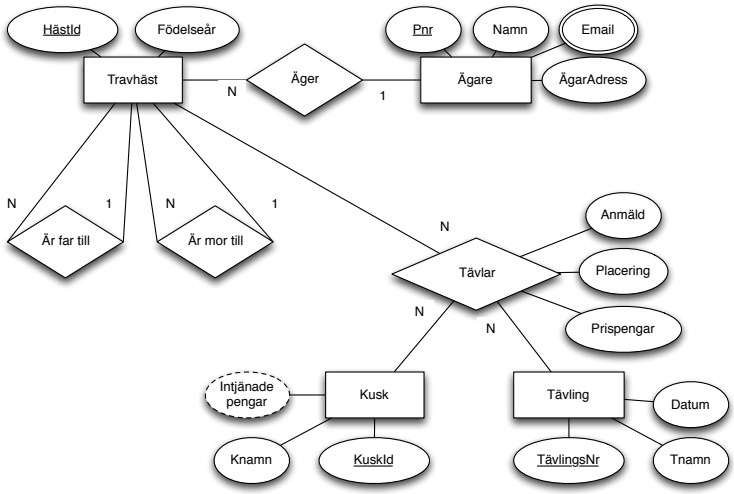
Lösning: Se figur 2. Steg 1 i kokboken (Varje vanlig entitetstyp blir en tabell, attribut blir kolumner) verkar ha tillämpats på *Ägare*, *Travhäst*, *Kusk* och *Tävling*. Gör vi dem till egna entiteter med attribut så får vi ungefär rätt databasstruktur för dessa fyra. Dock väntar vi lite med FarHästId, MorHästId och Ägare, vilka kanske är resultat av samband.

Steg 2 i kokboken (Varje 1:N-samband blir referensattribut i “många”-sidans tabell) verkar ha tillämpats på FarHästId, MorHästId och Ägare under Travhäst.

Tävlar ser ut som resultatet av steg 5 (Varje flervägssamband blir en egen tabell, med sammansatt primärnyckel bestående av primärnycklarna från de ingående tabellerna).

Email ser ut som resultatet av steg 9 (Varje flervärt attribut blir en egen tabell. Primärnyckeln består av entitetstypens primärnyckel, kombinerad med det flervärda attributet.)

Tillämpar vi reglerna i kokboken på ER-modellen i figur 2 kommer nr 1,2, 5 och 9 ge den givna databasstrukturen. De övriga reglerna resulterar inte i några förändringar av databasstrukturen. (Detta är en minimal förklaring. Det viktiga är att ni anger vilka regler som tillämpas och vad de leder till, samt vilka regler som inte leder till någon åtgärd. Det går även utmärkt att gå igenom reglerna lite mer strukturerat).



Figur 2: En ER-modell som skulle ge upphov till den givna databasstrukturen. Notera att *Intjänade pengar* tillkommer i deluppgift c

b) (*3p*) Ange för varje attribut i Databasstrukturen vilken datatyp som passar. Motivera dina svar.

Lösning:

Varchar (text): Namn, ÄgarAdress, Knamn, Tnamn, Emailadress, PNr

Integer (heltal): HästId, Födelseår, FarHästId, MorHästId, KuskId, TävlingsNr, Placering

Float (flyttal): Prispengar

Date: Datum

Boolean (sant/falskt): Anmäld

Man får även poäng om man har valt Prispengar som Integer och/eller Pnr som Integer. Notera dock att vissa system kan ha problem med att hantera heltal som är 10 siffror långa.

c) (*2p*) Hur skulle ER-modell och databasstruktur ändras om man lade till ett härlett attribut 'Intjänade pengar' till Kuskarna?

Lösning: I figuren skulle vi få en ny sträckad oval som det står "Intjänade pengar" i, med enkelstreck till Kusk-entiteten, se figur 2. Databasstrukturen ändras inte eftersom de härledda attributen inte innehåller någon egen data. Istället kan man lägga till en vy för att beräkna det härledda attributet.

3. (*Totalt: 24p*) Givet samma databasstruktur som i frågan ovan, dvs

   Ägare(PNr, Namn, ÄgarAdress)
   Travhäst(HästId, Födelseår, FarHästId, MorHästId, Ägare)
   Tävlar(HästId,KuskId,TävlingsNr,Anmäld, Placering, Prispengar)
   Kusk(KuskId, Knamn)
   Tävling(TävlingsNr, Tnamn, Datum)
   Email(Emailadress, Pnr)

   Skriv SQL-frågor som löser följande uppgifter.

   a) (*1p*) Lista alla Tävlingar (namn) som genomförs under Juni 2014.
   Lösning:

```
SELECT "Tnamn" FROM "Tävling"
WHERE "Datum" <= '2014-06-30'
AND "Datum" >=  '2014-06-01'
```

   b) (*2p*) Lista alla hästar (HästId) som saknar både mor och far i databasen.
   Lösning:

```
SELECT "HästId" FROM "Travhäst"
WHERE "FarHästId" IS NULL
AND "MorHästId" IS NULL
```

   c) (*2p*) Skapa en sorterad tabell över hästarna i fallande ordning efter hur många andra hästar de är mor till. Låt de två kolumnerna i listan vara *MorHästId* och *Antal barn*.
   Lösning:

```
SELECT "MorHästId", COUNT( * ) AS "Antal barn"
FROM "Travhäst" GROUP BY "MorHästId"
ORDER BY "Antal barn"
```

   d) (*1p*) Lista namn på Tävlingar vars namn innehåller antingen *mästerskap* eller *cup*.
   Lösning:

```
SELECT Tnamn FROM Tävling
WHERE Tnamn LIKE '%cup%'
OR Tnamn LIKE '%mästerskap%'
```

   e) (*1p*) Ta fram TävlingsNr för Elitloppet.
   Lösning:

```
SELECT "TävlingsNr" FROM "Tävling"
WHERE "Tnamn" = 'Elitloppet'
```

   f) (*3p*) Ta fram KuskId och placering på alla kuskar som deltog i Elitloppet.
   Lösning:

```
SELECT "KuskId", "Placering"
FROM "Tävlar"
WHERE "TävlingsNr" IN
( SELECT "TävlingsNr" FROM "Tävling"
WHERE "Tnamn" = 'Elitloppet' )
```

g) (*5p*) Ta fram namn och placering på alla kuskar som deltog i Elitloppet. Lös uppgiften genom att använda JOIN på minst ett ställe.

Lösning:

```
SELECT "Knamn", "Placering"
FROM "Tävlar" JOIN "Kusk"
ON "Tävlar"."KuskId" = "Kusk"."KuskId"
WHERE "TävlingsNr" IN
(SELECT "TävlingsNr" FROM "Tävling"
WHERE "Tnamn" = 'Elitloppet' )
```

h) (*3p*) Ta fram namn och placering på alla kuskar som deltog i Elitloppet. Lös uppgiften helt utan att använda JOIN.

Lösning:

```
SELECT "Knamn", "Placering"
FROM "Tävlar", "Kusk"
WHERE "Tävlar"."KuskId" = "Kusk"."KuskId"
AND "TävlingsNr" IN
(SELECT "TävlingsNr" FROM "Tävling"
WHERE "Tnamn" = 'Elitloppet' )
```

i) (*1p*) Lista namn och Pnr på de ägare som har en adress som slutar på '1', byt namn på namnkolumnen till *Ettagluttare*.

Lösning:

```
SELECT "Namn" AS "Ettagluttare", "Pnr"
FROM "Ägare"
WHERE "ÄgarAdress" LIKE '%1'
```

j) (*2p*) Skapa en lista på alla Hästar (HästId), sorterad med avseende på hur mycket prispengar hästen dragit in.

Lösning:

```
SELECT "HästId", SUM( "Prispengar" ) AS "Inkomst"
FROM "Tävlar"
GROUP BY "HästId"
ORDER BY "Inkomst"
```

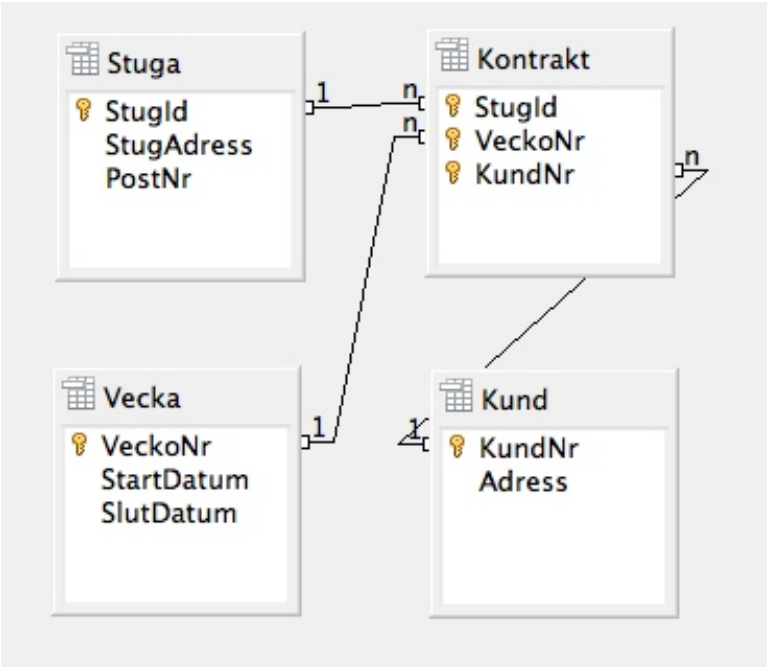
k) (*3p*) Skapa en lista på alla Hästar (HästId), sorterad med avseende på antalet tävlingar repektive häst har deltagit i under 2014.

Lösning:

```
SELECT "HästId", COUNT( * ) AS "Tävlingar2014"
FROM "Tävlar"
WHERE "TävlingsNr" IN
```

```
( SELECT "TävlingsNr" FROM "Tävling"
WHERE "Datum" <= '2014-12-31'
AND "Datum" >= '2014-01-01' )
GROUP BY "HästId"
ORDER BY "Tävlingar2014"
```

4. (*Totalt: 15p*) Betrakta Skärmdumpen från Open Office Base i figur 3.



Figur 3: En grafisk beskrivning av en databas i Base.

a) (*2p*) Det finns en liten otydlig symbol framför bl.a. orden *StugId*, *VeckoNr* och *KundNr* i bilden. Vad betyder denna symbol? Gör det något om man tar bort den?

Lösning: Symbolerna skall föreställa nycklar. De sitter framför de kolumner i tabellerna som tillsammans utgör primärnyckel, och måste fyllas i. Varje rad i tabellen måste sedan ha ett unikt värde i denna kolumn (dessa kolumner tillsammans).

b) (*2p*) Det finns fyra stora rutor i figuren, *Stuga, Kontrakt, Vecka* och *Kund*. Vad symboliserar dessa?

Lösning: Rutorna är namn på tabeller i databasen.

c) (*2p*) Det finns tre streck mellan de rutorna. Vad symboliserar dessa?

Lösning: Strecken symboliserar att nyckelattributen i Kontrakt är kopplade till nyckelattributen i de tre entiteterna på så sätt att om man lägger till en ny rad i Kontrakt, så måste de nyckelattribut man fyller i redan finnas i de andra tabellerna. Det går alltså inte att lägga till ett kontrakt på en stuga som inte finns i tabellen Stuga, eller med en Kund som inte finns i tabellen Kund.

d) (*2p*) Antag att man tänker köra ett antal SQL-frågor. Vad förändras om man tar bort strecken och sedan kör frågorna, jämfört med om man inte tagit bort strecken?

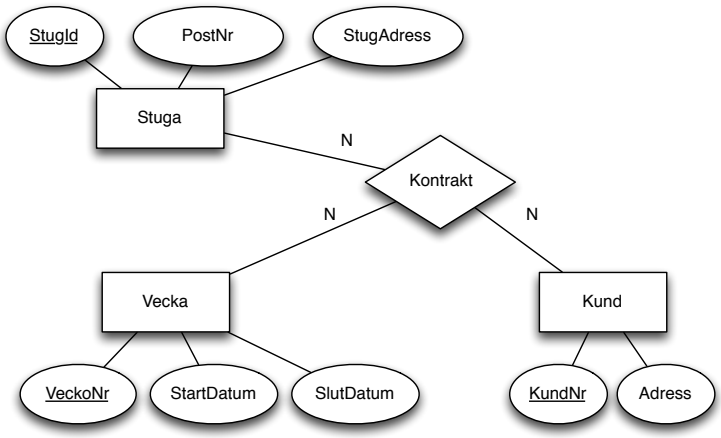
Lösning: SQL-frågorna påverkas inte alls av strecken. De funkar precis som vanligt oavsett om man tar bort, eller har kvar strecken.

e) (*2p*) Antag att man tänker lägga till data i databasen. Vad förändras om man tar bort strecken och sedan lägger till sitt data, jämfört med om man inte tagit bort strecken?

Lösning: Om strecken är borta kan inte Base hjälpa till och kolla att den nya datan är korrekt. Som beskrivs ovan så går det utan strecken t.ex. att lägga till nya kontrakt på stugor som inte finns i databasen.

f) (*3p*) Rita upp en ER-modell som motsvarar databasen i figur 3.

Lösning: Se figur 4.



Figur 4: En ER-modell som svarar mot figur 3.

g) (*2p*) Ta fram databasstrukturen som svarar mot figur 3.

Lösning: Antingen kan man gå via ER-modellen i uppgift 4f. Eller så kan man helt enkelt skriva ned de tabeller som står i de vita rutorna. Svaret blir

Stuga(StugId, StugAdress, PostNr)

Kontrakt(StugId,VeckoNr,KundNr)

Vecka(VeckoNr,StartDatum,SlutDatum)

Kund(KundNr, Adress)