

# Tentamen för DD1370

## Databasteknik och informationssystem

24 Augusti 2015

**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 på max 75.

Notera att bonuspoäng från tidigare år (t.ex. HT14) inte får tillgodoräknas, vilket är anledningen att maxpoängen höjts från 65 till 75.

En poängsumma på 40 ger säkert godkänt.

(lösningsförslag kommer på kurswebben)

*Lycka till,*

*Petter*



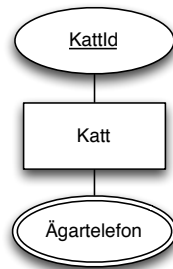
1. (Totalt: 18p)

- a) (3p) Rita en ER-modell innehållande entiteten *Katt* och ett *flervärt attribut*. Överför ER-modellen till en databasstruktur (du behöver inte motivera).

**Lösning:** Det finns många möjliga svar. Ett förslag på ER-modell finns i Figur 1 och motsvarande databasstrukturen blir då enligt nedan.

Katt(KattId)

Ägarteleson(KattId, TelefonNr)



Figur 1: En ER-modell som skulle ge upphov till den givna databasstrukturen.

- b) (3p) Rita en ER-modell innehållande entiteten *Katt* och ett *flervägssamband*. Överför ER-modellen till en databasstruktur (du behöver inte motivera).

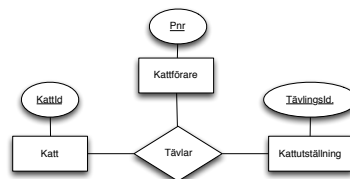
**Lösning:** Det finns många möjliga svar. Ett förslag på ER-modell finns i Figur 2 och motsvarande databasstrukturen blir då enligt nedan.

Katt(KattId)

Kattutställning(TävlingId)

Kattförare(Pnr)

Tävlar(KattId, TävlingId, Pnr)



Figur 2: En ER-modell som skulle ge upphov till den givna databasstrukturen.

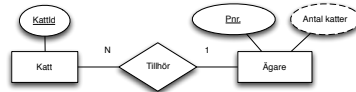
- c) (3p) Rita en ER-modell innehållande entiteten *Katt* och ett *härlett attribut*. Överför ER-modellen till en databasstruktur (du behöver inte motivera).

**Lösning:** Det finns många möjliga svar. Ett förslag på ER-modell finns i Figur 3 och motsvarande databasstrukturen blir då enligt nedan.

Katt(KattId, Pnr)

Ägare(Pnr)

Notera att det härledda attributet inte dyker upp i databasstrukturen, se regel 10 i kompendiet.

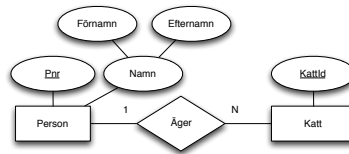


Figur 3: En ER-modell som skulle ge upphov till den givna databasstrukturen.

- d) (3p) Rita en ER-modell innehållande entiteten *Katt* och ett *sammansatt attribut*. Överför ER-modellen till en databasstruktur (du behöver inte motivera). **Lösning:** Det finns många möjliga svar. Ett förslag på ER-modell finns i Figur 4 och motsvarande databasstrukturen blir då enligt nedan.

Katt(KattId, Pnr)

Ägare(Pnr, Förnamn, Efternamn)



Figur 4: En ER-modell som skulle ge upphov till den givna databasstrukturen.

- e) (3p) Rita en ER-modell innehållande entiteten *Katt* och ett *fullständigt deltagande*. Överför ER-modellen till en databasstruktur (du behöver inte motivera).

**Lösning:** Det finns många möjliga svar. Ett förslag på ER-modell finns i Figur 5 och motsvarande databasstrukturen blir då enligt nedan.

Katt(KattId, Pnr)

Ägare(Pnr)

Notera att fullständigt deltagande behandlas som vanligt deltagande i databasstrukturen. Dock kan vi senare kräva att denna kolumn fylls i, i databasapplikationen.



Figur 5: En ER-modell som skulle ge upphov till den givna databasstrukturen.

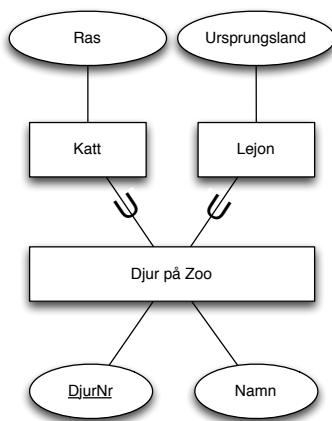
- f) (3p) Rita en ER-modell innehållande entiteten *Katt* och ett *arv*.  
Överför ER-modellen till en databasstruktur (du behöver inte motivera).

**Lösning:** Det finns många möjliga svar. Ett förslag på ER-modell finns i Figur 6 och motsvarande databasstrukturen blir då enligt nedan.

Djur på Zoo(DjurNr, Namn)

Katt(DjurNr, Ras)

Lejon(DjurNr, Ursprungsland)



Figur 6: En ER-modell som skulle ge upphov till den givna databasstrukturen.

2. (Totalt: 19p)

- a) (2p) Vad är en Vy, hur skapar man en Vy i Base, och vad använder man den till?

**Lösning:** En vy är en SQL-fråga som fått ett eget namn, och som sedan kan användas i andra SQL-frågor på samma sätt som tabellerna i databasen. Man skapar den genom att klicka på Tabeller/Skapa Vy..., och sedan skriva in SQL-frågan. Den används till att samla bearbetad information från resten av databasen i ett format som ser ut som de andra tabellerna. (I standard-SQL skriver man Create View X As)

- b) (3p) Förklara skillnaden mellan 1:1, 1:N och N:M-samband. Hur avgör man vilken typ ett samband är? Ge exempel på alla 3 typerna.

**Lösning:** Som exempel noterar vi att en person kan *köra* en bil, *äga* en bil och *ha kört* en bil.

Förutom att dessa samband innebär olika saker i praktiken är de dessutom olika vad gäller antalet deltagande Entiteter på varje sida av sambandet.

*Kör* är ett så kallat 1:1-samband. 1 person kan bara styra 1 bil åt gången, och 1 bil kan bara styras av 1 person åt gången.

*Äger* är ett 1:N samband. 1 person kan äga en eller flera bilar, men en bil kan bara ägas av en person. Det senare kan tyckas lite märkligt, men enligt svensk lag kan man inte samäga bilar.

Har kört är slutligen ett N:M samband. Flera personer kan ha kört samma bil, och en person kan ha kört flera olika bilar.

Se kompendiet för en formell beskrivning av hur man avgör vilken typ ett samband är.

- c) (3p) 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. order by är ett SQL-uttryck för att sortera en tabell. sort by är inte ett SQL-uttryck.

- d) (3p) Ange 6 datatyper som kan användas i SQL-databaser. För varje datatyp, ange ett exempel på användning.

**Lösning:** Date (datum, t.ex. någons födelsedag), Time (klockslag, t.ex. när man brukar äta lunch), Timestamp (datum och klockslag, t.ex. tidpunkt för en aktienotering), Integer (heltal, t.ex. antal varor i lager), Float (decimaltal, t.ex. någons vikt), Varchar (text, t.ex. någons namn)

- e) (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.

- f) (2p) Antag att tabellerna A och B är givna enligt nedan.

MedlemsNr	Namn
1	Anna
2	Shakira
3	Kim

A=

TröjNr	Namn
10	Zlatan
8	Lotta

B=

Rita tabellen man får som resultat av:

```
SELECT * FROM A,B WHERE TröjNr='8'
```

**Lösning:** Resultatet blir följande

MedlemsNr	Namn	TröjNr	Namn
1	Anna	8	Lotta
2	Shakira	8	Lotta
3	Kim	8	Lotta

- g) (2p) Antag att tabellerna A och B är givna enligt ovan. Rita tabellen man får som resultat av

```
SELECT * FROM A JOIN B ON MedlemsNr=TröjNr.
```

**Lösning:** Resultatet blir en tom tabell.

- h) (2p) Antag att tabell C har 1000 rader och 5 kolumner, samt att tabell D också har 1000 rader och 5 kolumner. Hur stor blir tabellen man får som resultat av

```
SELECT * FROM C,D.
```

**Lösning:** Resultatet blir 1000000 rader och 10 kolumner.

3. (Totalt: 12p) Ett postorderföretag använder följande databasstruktur för att hålla ordning på verksamheten.

Kund(KundNr, Namn, Adress, Email)

Order(OrderNr, OrderDatum, Förfalldatum, Betald, KundId)

Produkt(ProdId, Beskrivning, Pris)

Innehåller(OrderNr,ProdId, Antal)

- a) (8p) Rita upp en ER-modell som skulle resultera i ovanstående Databasstruktur. Motivera varje steg genom att mkt kort beskriva den regel i "kokboken" som använts.

**Lösning:**

Se figur 7. Steg 1 i kokboken (Varje vanlig entitetstyp blir en tabell, attribut blir kolumner) verkar ha tillämpats på *Kund*, *Order*, och *Produkt*. Gör vi dem till egna entiteter med attribut så får vi ungefär rätt databasstruktur för dessa. Dock väntar vi lite med *Innehåller* vilket kanske är resultat av någon annan regel.

Steg 2 i kokboken (Varje 1:N-samband blir referensattribut i "många"-sidans tabell) verkar ha tillämpats på ett samband mellan Kund och Order, vi kallar sambanden Läger (kunden lägger en order).

Steg 3 handlar om 1:1-samband, något sådant verkar vi inte ha.

Steg 4 (Varje N:M-samband bildar egen tabell) verkar ha gett upphov till tabellen *Innehåller* (en Order innehåller en viss Produkt).

Steg 5 handlar om flervägssamband, något sådant verkar inte finnas.

Steg 6 handlar om attribut på samband, något sådant verkar inte finnas.

Steg 7 handlar om svaga entiteter, några sådana verkar inte finnas.

Steg 8 (sammansatta attribut blir som delarna), något sådant verkar inte finnas.

Steg 9 (Varje flervärd attribut blir en egen tabell. Primärnyckeln består av entitetstypens primärnyckel, kombinerad med det flervärda attributet.) Något sådant verkar inte finnas.

Steg 10 handlar om härledda attribut. Dessa syns inte i databasstrukturen.

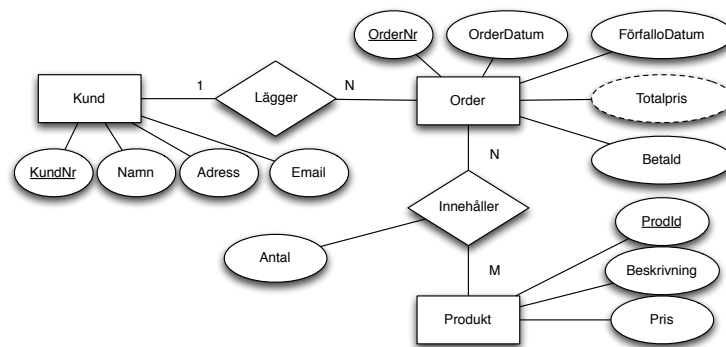
Steg 11 (Varje subclass blir en egen tabell) Något sådant verkar inte finnas.

Tillämpar vi reglerna i kokboken på ER-modellen i figur 7 kommer vi således få den givna databasstrukturen.

- b) (2p) Hur skulle ER-modell och databasstruktur ändras om man lade till ett härlett attribut 'Totalpris' till Ordern?

**Lösning:** I figuren skulle vi få en ny sträckad oval som det står "Totalpris" i, med enkelstreck till Order-entiteten, se figur 7.

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.



Figur 7: En ER-modell som skulle ge upphov till den givna databasstrukturen. Notera att *Totalpris* tillkommer i deluppgift 3b

c) (2p) Föreslå datatyper för alla kolumner i databasstrukturen.

**Lösning:**

Varchar: Namn, Adress, Email, Betald (kan också vara Boolean),  
Beskrivning

Integer: KundNr, OrderNr, ProdId

Date: OrderDatum, Förfalldatum

Float: Pris (kan också vara Integer)

4. (Totalt: 26p) Givet samma databasstruktur som i frågan ovan, dvs

Kund(KundNr, Namn, Adress, Email)

Order(OrderNr, OrderDatum, Förfalldatum, Betald, KundId)

Produkt(ProdId, Beskrivning, Pris)

Innehåller(OrderNr,ProdId, Antal)

Skriv SQL-frågor som löser följande uppgifter.

a) (2p) Lista alla Kunder vars namn slutar på *sson*.

**Lösning:**

```
SELECT * FROM "Kund"
WHERE "Namn" LIKE '%sson'
```

b) (2p) Lista alla Produkter som kostar mer än 1000kr.

**Lösning:**

```
SELECT * FROM "Produkt"
WHERE "Pris" > 1000
```

c) (2p) Lista ProdId på alla produkter som ingår i order nummer 3.

**Lösning:**



```
SELECT "ProdId"
FROM "Innehåller"
WHERE "OrderNr" = 3
```

d) (2p) Lista totalkostnaden för order nummer 3.

**Lösning:**

```
SELECT SUM( "Antal" * "Pris" ) AS "Ordersumma"
FROM "Innehåller", "Produkt"
WHERE "Innehåller"."ProdId" = "Produkt"."ProdId"
AND "Innehåller"."OrderNr" = 3
```

e) (2p) Lista totalkostnaderna för alla ordrar (dvs en totalkostnad för varje order).

**Lösning:**

```
SELECT "Innehåller"."OrderNr",
SUM( "Antal" * "Pris" ) AS "Ordersumma"
FROM "Innehåller", "Produkt"
WHERE "Innehåller"."ProdId" = "Produkt"."ProdId"
GROUP BY "Innehåller"."OrderNr"
```

f) (2p) Lista alla ordrar (OrderNr) som Kund 122 har lagt.

**Lösning:**

```
SELECT "OrderNr" FROM "Order"
WHERE "KundNr" = 122
```

g) (2p) Lista alla ordrar (OrderNr) som innehåller ProdId 11.

**Lösning:**

```
SELECT "OrderNr" FROM "Innehåller"
WHERE "ProdId" = 11
```

h) (2p) Lista alla ordrar som inte är betalda trots att förfalldatum har passerat (ligger tidigare än 2015-08-24). Antag att kolumnen Betald är tom om ordern inte är betald.

**Lösning:**

```
SELECT "OrderNr" FROM "Order"
WHERE "Betald" IS NULL
AND "Förfalldatum" < '2015-08-24'
```

i) (2p) Lista alla produkter, sorterade efter hur många exemplar som har sålts i fallande ordning.

**Lösning:**

```
SELECT "ProdId", SUM( "Antal" ) AS "Antal Sålda"
FROM "Innehåller"
GROUP BY "ProdId"
ORDER BY "Antal Sålda" DESC
```

- j) (2p) Lista snittpriset och totalpriset per Kund (Namn) på alla Ordrar hen har lagt. Du får anta att du löst uppgift 4e och skapat vyn OrderSummor med kolumnerna OrderNr och OrderSumma

**Lösning:**

```
SELECT "Kund"."Namn",
AVG( "OrderSumma" ), SUM( "OrderSumma" )
FROM "Kund", "Order", "OrderSummor"
WHERE "Kund"."KundNr" = "Order"."KundNr"
AND "Order"."OrderNr" = "OrderSummor"."OrderNr"
GROUP BY "Kund"."Namn"
```

- k) (3p) Lista namn på alla kunder som lagt mindre än 3 ordrar. Använd en nästlad SQL-fråga. **Lösning:**

```
SELECT "Namn" FROM "Kund"
WHERE "KundNr" IN
( SELECT "KundNr" FROM "Order"
GROUP BY "KundNr"
HAVING COUNT( * ) < 3 )
```

- l) (3p) Lista hur mycket pengar Kalle har lagt på respektive produkt.

**Lösning:**

```
SELECT "Produkt"."ProdId", SUM( "Pris" * "Antal" )
FROM "Innehåller", "Kund", "Order", "Produkt"
WHERE "Kund"."Namn" = 'Kalle'
AND "Kund"."KundNr" = "Order"."KundNr"
AND "Order"."OrderNr" = "Innehåller"."OrderNr"
AND "Innehåller"."ProdId" = "Produkt"."ProdId"
GROUP BY "Produkt"."ProdId"
```