

# Tentamen

## Databasteknik

**2014-08-22, 14.00-19.00**

Inga hjälpmedel.

Bedömning (preliminär): uppgifterna ger maximalt  $13 + 13 + 13 + 4 + 4 + 3 = 50$  poäng. För godkänt betyg krävs 27 poäng (3/27, 4/35, 5/43).

- 
1. Ett studieförbund ger kurser på olika orter. Det finns kurser i många ämnen: engelska på olika nivåer, tyska, knyppling, krokiteckning, datakörkort, osv. Kurserna som ges på en viss ort beskrivs i en katalog som skickas ut till hushållen inför varje nytt läsår. I katalogen finns för varje kurs anmälningsskod, namn och en beskrivande text. Kurser kan ges vid flera tillfällen under året. För varje tillfälle anges ansvarig lärare och när kursen ges (startdatum, antal veckor, veckodag och tid). Varje kurstillfälle har en avgift som kursdeltagarna ska betala.

Uppgifterna om kurserna och kurstillfällena hämtas från studieförbundets centrala databas. Denna innehåller alla uppgifter som studieförbundet behöver för att administrera kursverksamheten. Till exempel anges inte bara ansvarig lärare för en kurs utan också vilken lärare som ska undervisa vid varje undervisningstillfälle. Detta därför att lärarna får betalt per timme, och man vill enkelt kunna beräkna ersättningen till lärarna och skicka ut löneaviser och uppgifter till skatteverket.

Kursdeltagarna får fakturor på anmälningssavgiften innan kursen startar. Påminnelse skickas ut om avgiften inte betalas.

- Utveckla en ER-modell som beskriver databasen. Hitta på lämpliga attribut till entitetsmängderna och sambanden.
  - Översätt ER-modellen till en relationsmodell. Samtliga relationer bör vara i BCNF. Motivera att så är fallet eller varför du valt en annan normaliseringsgrad. Ange primärnycklar och främmande nycklar för alla relationer.
  - Utgående från dina relationer i uppgift b, skriv en SQL-sats som ger de totala intäkterna för de kurser som har getts under 2014 i Lund. Resultatet ska innehålla kurskod, kursnamn och totala intäkter, sorterat efter kurskod.
-

2. Infektionskliniken på ett sjukhus ska undersöka effekten av antibiotikabehandling av olika sjukdomar och har utvecklat en databas med följande relationer:

Patients(id, name, address)  
 Bacteria(name, description)  
 Antibiotics(name, description, source)  
 Treatments(patientId, bacteriaName, antibioticName, effect)

Attributet *effect* i Treatments anger på en skala 0–10 hur effektiv behandlingen mot en viss bakterie med ett visst antibiotikum var för en viss patient. Attributet *source* i Antibiotics är tillverkarens namn.

- Rita ett ER-diagram som beskriver databasen.
- Skapa tabellen Treatments. Ange rimliga integritetsvillkor.

I uppgift c till f ska du skriva SQL-frågor som hämtar information från databasen (du får använda vyer).

- Skriv ut namnen på de antibiotika som inte har använts i någon behandling.
- Skriv ut antalet patienter som har behandlats för en infektion förorsakad av bakterien *S. aureus*.
- Skriv ut id-nummer, namn och adress för de patienter som fått mer än en behandling.
- Skriv ut namn och leverantör för de antibiotika som visat sig vara effektiva i behandling av infektioner förorsakade av bakterien *S. aureus*. Ett antibiotikum anses vara effektivt mot en bakterie om varje enskild behandling har en effekt av minst 7.

3. I relationen R(A, B, C, D, E) finns följande funktionella beroenden:

FD1.  $A \rightarrow B$   
 FD2.  $BC \rightarrow E$   
 FD3.  $DE \rightarrow A$

- Vilka nycklar finns i relationen?
- Visa att relationen inte är i BCNF, men att den är i 3NF.
- Dela upp relationen i mindre relationer som är i BCNF.

4. Betrakta följande tabell:

```
create table Persons (
  pNbr char(11) primary key,
  name varchar(40),
  address varchar(60)
);
```

I databashanteraren exekveras satsen `create index NameIndex on Persons(name)`. Vad har detta för konsekvenser för användningen av databasen? Visa på både fördelar och nackdelar.

5. Innan en SQL-fråga exekveras av en databashanterare så optimeras frågan. En vanlig optimering är "pushing selection". Vad innebär det? Varför innebär det normalt en optimering?

6. I OS X sparas information som program behöver i egenskapslistor, "property lists". En egenskapslista är en XML-fil som följer nedanstående (förenklade) DTD. ENTITY-elementet fungerar ungefär som en deklaration av en textkonstant; med %namn; får man värdet av konstanten.

```
<!ENTITY % plistObject "(array | dict | integer | string)" >
<!ELEMENT plist %plistObject;>
<!ATTLIST plist version CDATA "1.0" >
<!-- Collections -->
<!ELEMENT array (%plistObject;)*>
<!ELEMENT dict (key, %plistObject;)*>
<!ELEMENT key (#PCDATA)>
<!-- Primitive types -->
<!ELEMENT string (#PCDATA)>
<!ELEMENT integer (#PCDATA)> <!-- An integer number in base 10 -->
```

Vilka är felen i nedanstående XML-fil? Markera och förklara varje fel.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist SYSTEM "file:///System/Library/DTDs/PropertyList.dtd">
<plist version="1.0">
<dict>
  <key>Author</key>
  <string>William Shakespeare</string>
  <key>Lines</key>
  <array>
    <string>It is a tale told by an idiot,</string>
    <string>Full of sound and fury, signifying nothing.</string>
  </array>
  <key>Birthdate</key>
  <year>1564</year>
  <key>Deathdate</key>
  <year>1616</year>
</dict>
<dict>
  <key>Author</key>
  <string>Christopher Marlowe</string>
  <key>Birthdate</key>
  <year>1564</year>
  <key>Deathdate</key>
  <year>1593</year>
</plist>
```