

Linköpings Universitet
Institutionen för datavetenskap
Eva Ragnemalm

TENTAMEN

732G16 Databaser

Design och programmering

Datum: **2017-08-16 kl 8-12**

Jourhavande lärare: **Eva Ragnemalm**

Tel: **070-1907391**

Besöker tentamenslokalen c:a kl 10

Hjälpmedel: Svensk-kinesisk ord-
bok vid behov.

Följ instruktionerna på tentaomslaget.
Om du skriver flera uppgifter på
samma papper, lämna gott om plats
för kommentarer.

Poängfördelning

Uppgift	Poäng
1	5
2	4
3	4
4	7
5	8
6	2
7	5
8	2
9	6
10	10

Betyg: G: 27p VG: 40p Max: 53p

1. **Begrepp:** Redogör kortfattat, gärna med exempel, för följande begrepp: (5 p)

- a) Databas
- b) Datamodell
- c) Metadata
- d) Databashanterare
- e) Databasinstans

Kombinationsuppgift: Bakgrund för uppgift 2-3.

Du har en databas som skapats med nedanstående kommandon:

```
CREATE TABLE course
    (coursecode CHAR (6) CONSTRAINT pk_course PRIMARY KEY,
    name VARCHAR2(30), points INTEGER, examiner INTEGER);
CREATE TABLE employee
    (empnr INTEGER CONSTRAINT pk_employee PRIMARY KEY,
    name VARCHAR2(30), institution VARCHAR(15));
ALTER TABLE course
    ADD CONSTRAINT fk_course_employee
    FOREIGN KEY (examiner) REFERENCES employee (empnr);
```

Den har innehåll enligt nedan:

Course:

732G16	Databaser, design och programmering	7	123
TDDB38	Tekniska Databaser	6	456
729G28	Webprogrammering och databaser	7	123

Employee:

123	Eva Ragnemalm	IDA
456	Lena Strömbäck	IDA

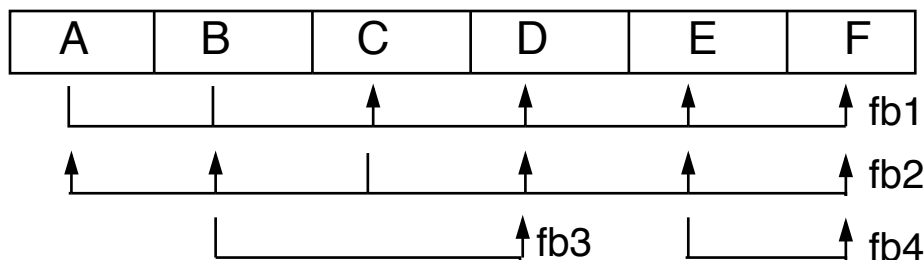
2. **Begrepp, SQL:** För varje nedanstående SQL-kommando berätta om det går att genomföra eller inte (dvs resulterar i felmeddelande). För kommandon som resulterar i felmeddelanden, beskriv orsaken och ge ett lämpligt kommando som kan användas först för att kommandot ska gå att genomföra så som det är skrivet. Det föreslagna kommandot får inte ändra databasens schema. (4p)

- a) `DELETE FROM employee WHERE empnr=123;`
- b) `INSERT INTO course VALUES('TDD38', 'Databasteknik', 8, 999);`
- c) `DELETE FROM course WHERE kursansvarig=123;`
- d) `INSERT INTO employee VALUES(789, 'Johan Falkenjack', 'IDA');`

3. **SQL:** Givet relationerna Employee och Course ovan, skriv ett SQL-uttryck för respektive uppgift nedan: (4p)

- a) lista alla kurser som ges av Eva Ragnemalm
- b) Hur många kurser ger respektive lärare? Lista namn och det antal kurser den läraren ger för alla lärare som ger någon kurs.

4. **Normalisering, begrepp:** Relationen som illustreras i figuren nedan har de atomära attributen A-F (det är namnen på attributen) och de funktionella beroenden fb1-4 som symboliseras av pilarna i figuren. (7p)



- a) Är {A, D, E} en *supernyckel* till relationen? Varför/inte?
- b) Är {C, D} en *kandidatnyckel* till relationen? Varför/inte?
- c) Vilken normalform uppfyller relationen? Varför? (Motivera genom att markera kandidatnycklar och primattribut och sedan utifrån dem förklara normaliseringsvillkoren och vilka som är uppfyllda).
- d) Beskriv kort vilka problem man kan få om databasen man arbetar med inte är normaliserad.

Kombinationsuppgift: Bakgrund för uppgift 5 -9.

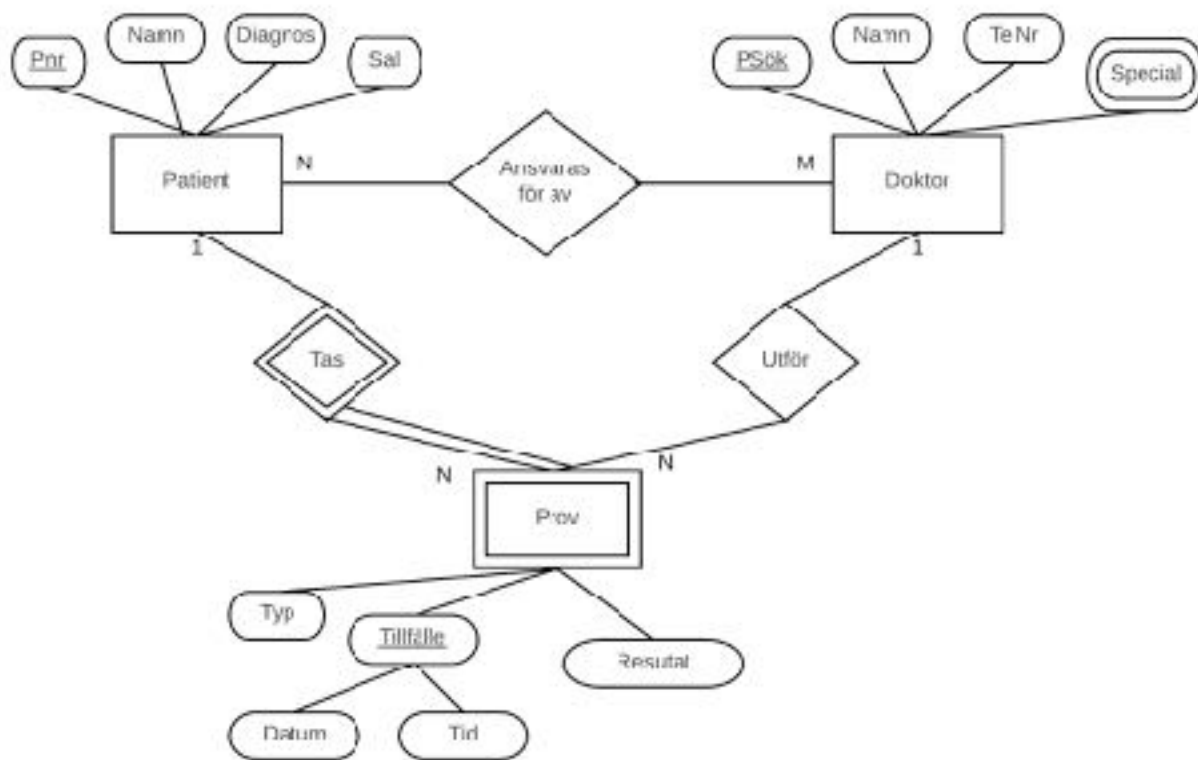
Sjukhemmet Solåsen behöver hjälp med designen av sin databas. De vill hålla reda på patienter och doktore och de prover som tas. För patienterna behöver man hålla reda på dem själva (namn och personnummer), deras diagnos och vilken sal de ligger på. Man måste också hålla rätt på vilken läkare som är ansvarig för respektive patient (flera läkare kan vara ansvariga för en patient om det är ett komplext fall, och en läkare är ansvarig för flera patienter). För läkarna vill man också hålla reda på deras specialiteter (en läkare kan ha flera specialstområden), telefonnummer och personsökarnummer. När det gäller proverna som tas på patienter måste man kunna lagra vilken typ av prov det är (blodsocker, gastroskopi, sänka etc), när det togs, resul-

tat av analys (en text) samt för vissa typer vem som utförde det. Enklare prov, som blodtryck och blodsockermätning utförs av laboratoriepersonal och lagras då inte. Mer avancerade prov som gastroskopi utförs av läkarna och detta lagras då.

Samma prov kan tas på samma patient flera gånger, dock max en gång per dag. En läkare behöver inte vara ansvarig för en patient för att utföra ett prov på en patient. Man vill kunna ställa frågor av följande typ till systemet:

- Finns namn och personsökarnummer för den läkare som är ansvarig för patienten med personnummer 281012-1234.
- Vilka prover har utförts på patienten med personnummer 120212-4321, när, och med vilka resultat? Sortera i datumordning så att man kan se förändring i tillståndet.
- Hur många gastroskopier (Prov = gastroskopi) har de olika läkarna utfört (ange i listan läkarens namn samt antal gånger han/hon utfört gastroskopi).

Ett ER-diagram för denna skulle kunna se ut så här:



5. **Databasesign:** Konvertera detta ER-diagram till ett relationsschema (tabeller). Markera primärnycklar med understrykning och främmande nycklar (referensattribut) som pilar som vi gjort på föreläsningarna. (8p)

Fortsättning: En klantig programmerare, som inte begrep sig på ER-diagram, föreslog den design som redovisas nedan:

Patient (Personnummer, Efternamn, Förnamn, Diagnos, Sal, Prov, Provdatum, Resultat, ProvtagandeLäkare)

Läkare (Efternamn, Förnamn, Personsökarnr, Telnr, Patient)

Fältet Läkare i relationen Patient innehåller läkarens personsökarnummer.

6. **Relationsalgebra:** Skriv ett relationalalgebra-uttryck för fråga a ovan givet relationerna Läkare och Patient. (2p)

Fortsättning: Den inkompetente programmeraren skrev sedan några program som hanterade de relationer han hade (relationerna Läkare och Patient ovan), en relation per fysisk fil. Notera att filen Patient innehåller en post per prov som gjorts på en viss patient. Filerna lagrades på en server centralt så att alla skulle kunna komma åt dem från sina kontorsdatorer när de ville. Servern har en hårddisk som har blockstorleken 4096 byte och accesstid 10 ms.

Filen med patienter och prover (enligt ovanstående) har 20000 poster och fylls på genom att nya poster läggs till i slutet. Varje post tar 1000byte.

7. **Fysiska databasen:** Hur lång tid tar det (i genomsnitt) att söka ut ett visst (godtyckligt) prov för en godtycklig patient? Hur lång tid tar det att söka ut alla en viss patients prover? (man vet inte hur många det finns). Ignorera den tid det tar att skicka informationen från servern till kontorsdatorn. (5p)

8. **Fysiska databasen:** Hur skulle man kunna snabba upp sökningen? Förklara med exempel. (2p)

Fortsättning: Programmet för att lägga in ett prov fungerade så att det först öppnade Patient-filen som låg på servern, läste in den till en intern fil, stängde den externa (den på servern), uppdaterade patientposten, skrev tillbaks hela den ändrade interna filen till servern (skriver över den gamla filen).

Programmet som lade in en ny patient fungerade så att det först öppnade Patientfilen, läste in den till en intern fil, stängde den externa (den på servern), lade in patientposten, öppnade Läkarefilen (på samma sätt som patientfilen), lade in en ny patient på den aktuella läkaren, skrev tillbaks båda de interna filerna till servern (skriver över de båda gamla).

Efter ett tag upptäckte man att slumpvisa (såvitt man kunde se) ändringar försvann fastän läkaren som ändrat svor på att han/hon kört programmet korrekt. Man anställde då en extra sekreterare som utbildades speciellt på programmet och som sedan var den enda som införde ändringar. Alla uppdateringar i systemet meddelades till denna person, som lade in dem, istället för att varje läkare lade in dem själv. Då försvann problemet.

9. **Transaktioner:** Vad orsakade problemet (förklara i detalj vad som hände när det blev fel) och vad innehåller ett DBMS som den klantige programmeraren glömt lägga in i sitt program?

Hur skulle en programmerare som använder ett DBMS göra för att undvika problemet (beskriv lösningen detalj)? (6p)

10. **Design:** Du har fått i uppgift att utforma en databas för att hantera de olika plantorna som säljs i en trädgårdsbutik. För att kunna ge kunderna rätt råd om plantering och skötsel vill man lagra information om växtzon, lämplig planteringstid, läge och lämplig jordmån samt ordinarie pris.

Utöver växtens vanligaste svenska namn (som kan betraktas som unikt om man vill) lagrar man det latinska namnet, i två delar, familjenamn och artnamn (och för varianter skriver man även variant-namnet i artnamns-fältet). Familjenamn och artnamn ger en unik, men lång, identifikation av blom-arter.

Företaget ger ut ett reklamblad veckovis, med veckans kampanjerbjudanden. Man lagrar vilka blommor som ingår i veckans kampanj och vilket kampanjpris som gäller respektive blomma, samt start och slut-datum för kampanjen. All denna information sparas i efterhand för att kunna utvärdera vilka kampanjer som är effektiva. Man vill också lagra information om vilken sida en blomma visats på i reklambladet, eftersom kunderna ofta missförstår texten till bilderna och blandar ihop de avbildade blommornas namn. Man kan vara säker på att en viss blomma bara visas en gång i ett visst reklamblad.

Man vill också lagra information om sina leverantörer (kontaktinformation i första hand, för beställningar och betalningar, alltså företagets namn, namn på kontaktperson, telefon och adress). Varje leverantör kan oftast leverera flera olika växter. De flesta växter kan levereras av flera leverantörer, och alla sätter sitt eget leveranspris (som måste kunna lagras även om vi inte har några blommor av den sorten inne från den leverantören just nu, men det räcker att kunna spara det senaste).

Varje faktisk leverans av en växt förses också med ett batchnummer (unikt löpnummer) för att kunna identifiera vem som levererat just de plantorna (i händelse av att plantorna har någon sjukdom eller är av ovanligt dålig kvalitet eller så). Det händer ofta att man har plantor av samma sort från olika batcher ute i butiken samtidigt, men eftersom prislapparna som sätts på krukorna identifierar batchen är det ingen risk för sammanblandning. Varje batch innehåller alltså ett visst antal blommor av samma art och familj levererade från en viss leverantör. Man vill naturligtvis också hålla rätt på hur många plantor man faktiskt har kvar av varje sort och vilken batch de kommer från. Informationen om batchen ska finnas kvar när plantorna är slutsålda.

Man vill till exempel kunna göra följande utsökningar ur databasen:

- a) En kund vill ha tips om vilka växter som trivs i samma jordmån och läge som röd Isbegonia (*Begonia semperflorens* 'Inferno Red'). Lista sådana växter.
- b) Hur många vita isbegonior (*Begonia semperflorens* 'Inferno White') finns på lager?
- c) Vilka olika leverantörer kan leverera röda isbegonior, och till vilket pris? Lista namn och telefonnumret till leverantören, samt leveranspris. Sortera på leveranspris.

Designa en databas utifrån denna information. Rita upp ett ER-diagram för databasen. Ta också hänsyn till de utsökningar du ska kunna göra när du utformar databasen. Glöm inte markera kardinalitet och deltagande samt nyckelattribut. Om du lägger till attribut som inte krävs av specifikationen ovan måste du motivera varför de behövs. Tips: tänk efter på vilken detaljnivå du behöver hålla reda på växterna, tex utifrån vilka samband behöver du kunna representera? EER ska inte användas.

(10p)

Lycka till!