

Linköpings Universitet
Institutionen för datavetenskap
Eva Ragnemalm

T E N T A M E N

732G16 Databaser Design och programmering

Datum: 2018-08-14 kl 8-12

Jourhavande lärare: Jalal Maleki/Eva Ragnemalm

Tel: 070-1907391

Besöker tentamenslokalen: c:a kl 10

Hjälpmedel: Svensk-kinesisk, svensk-engelsk, svensk-norsk ordbok (vid behov).

Följ instruktionerna på tentaomslaget. Svara inte på flera uppgifter på samma papper.

Uppgift	Poäng
1	4
2	8
3	6
4	3
5	1
6	6
7	9
8	20
9	16
10	6

Betyg: G: 40p VG: 60p Max: 79p

1. **Begrepp:** Begreppstips: För varje delfråga svara 1, X eller 2: (4p)

- a) En databas' *schema* är:
1. Databasens faktiska innehåll vid ett visst ögonblick i tiden.
 - x. Beskrivningen av hur data i databasen ska vara strukturerat.
 2. Förändringen av databasens innehåll mellan två olika tider.
- b) En *datamodell på implementationsnivå* beskriver:
1. I vilket format data ska lagras i databasen, t.ex. om det ska vara heltal eller strängar.
 - x. Vilka begrepp i den verkliga världen som ska finnas representerade i databasen.
 2. Vilka tabeller databasen ska innehålla.
- c) Ett *fullt funktionellt beroende* är:
1. ett specialfall av funktionellt beroende sådant att determinanten är minimal.
 - x. ett funktionellt beroende där determinantens värde beror av ett annat attribut i relationen.
 2. ett funktionellt beroende där man kan ta bort minst ett attribut ur determinanten och fortfarande ha ett funktionellt beroende.
- d) *Referensintegritet* betyder att:
1. när en tabell har ett referensattribut som refererar till en annan tabell, och en rad i tabellen har ett visst värde på referensattributet så måste det värdet finnas i den tabell som attributet refererar till.
 - x. vi får inte lagra persondata utan personens medgivande.
 2. när vi konverterar ett ER-diagram till relationsschema måste tabellerna referera tillbaka till entitetstyperna genom att ha samma namn.

2. **Normalisering:** Du har en relation som har attributen A, B, C, D, E, F, där {A, B} är en kandidatnyckel och C en annan kandidatnyckel men inga fler kandidatnycklar finns. Du vet att relationen uppfyller 2:a normalform men INTE 3:e normalform (och därmed inte heller BCNF). Vilka av följande fulla funktionella beroenden KAN då finnas i relationen? Gör bedömningen på en deluppgift i taget. (8 p)

- a) $A \rightarrow D$
- b) $\{A, B\} \rightarrow D$
- c) $\{A, D\} \rightarrow E$
- d) $B \rightarrow C$
- e) $B \rightarrow F$
- f) $C \rightarrow A$
- g) $D \rightarrow C$
- h) $\{D, E\} \rightarrow F$

3. **Relationsalgebra:** Förklara med hjälp av exempel följande relationsalgebraoperationer. Med andra ord: hitta på innehållet i två relationer och visa vad resultatet av operationen blir. Använd

samma relationer i alla deluppgifter. För b och c får du hitta på villkor och mängd också, så att operationen fungerar. (6 p)

- a) Cartesisk produkt
- b) Sammansättning (join)
- c) Projektion

Kombinationsuppgift Design, Relationsmodell, Fysiska Databasen, SQL: följande innehåller information till uppgift 4-8.

Företaget Strömljus tillverkar, reparerar och säljer spisar. Man har ett antal modeller som säljs i butiken. Man utför både garantireparationer och andra reparationer på företagets spisar (men man befattar sig inte med annat), och reparationskostnaderna beror på om garantin fortfarande gäller (inom 6 månader från köpet är det alltid gratis), vilka delar som bytts ut och den arbetstid som nedlagts. Nu vill man lägga upp en databas över spisar, kunder, reparationer och reservdelar. Vid förfrågan anser chefen att följande tabeller måste finnas i databasen:

Spis: serienummer (unikt för varje spis), modellbeteckning, inspektionsdatum, inspektörID.

Kund: Kundnr, namn, gatuadress, postnummer, postort, epostadress, telefonnummer.

Reparation: reparationsnummer (ett unikt id-nummer), kund, spis, reparatör, åtgärd (en kort textuell beskrivning), reservdel, pris (för reservdelen), antal (av reservdelen), arbetstid, reparationsdatum.

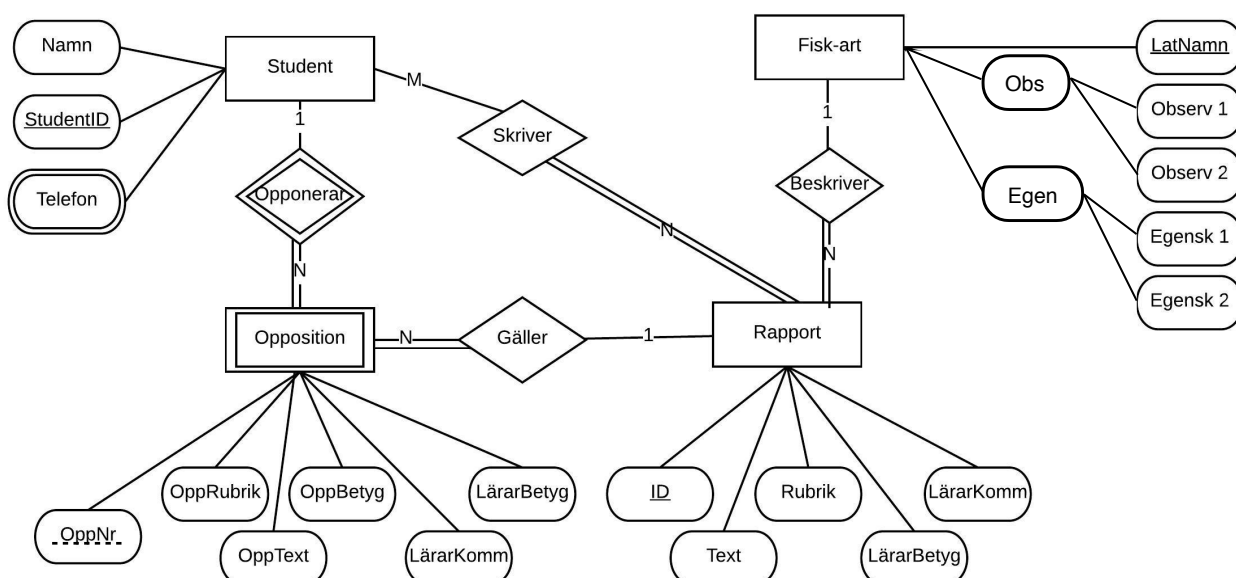
Vid en första design skapades tabeller enligt ovan i en relationsdatabashanterare och man implementerade ett system för att automatiskt fakturera kunder för reparationer baserat på de tabellerna. Reparatorerna klagade på att de bara kunde debitera en sorts reservdel per reparation (trist när de bytte t.ex både ugnslucka och barnspärr). Kunderna klagade på att de fick faktura fastän det var mindre än 6 månader sedan de köpt spisen (och alltså hade giltig garanti). Pelle på lagret insåg också att samma reservdel ibland fick olika pris (en reservdel ska alltid ha samma pris oavsett i vilken reparation den använts).

4. Vad var orsaken till de olika priserna på reservdelar? Hur kan man använda något vi lärt i kursen för förhindra att sådant här händer? (3 p)
5. Vilken information behöver lagras i databasen för att man ska kunna hantera garantireparationer (dvs att reparationer är gratis 6 månader från att kunden köpt spisen)? (1 p)
6. Man skapar en hashtabell för att lagra filen som innehåller tabellen Spis och som hashfunktion används de två sista siffrorna i serienummret (dvs serienummer mod 100). Filen innehåller efter ett tag 10 000 poster. Antag att vi byter från en hårddisk vars block rymmer 10 poster till en vars block rymmer 100 poster. Hur påverkas antalet spillblock? Antag att hashfunktionen ger mycket bra spridning. (6 p)
7. Skriv följande frågor i SQL med utgångspunkt från tabellerna ovan. (9 p)

- Beräkna totalkostnaden för en viss reparation (den med reparationsnr 1111) enligt följande: totalkostnad för reservdelar plus arbetstid * 500kr.
- Skapa en vy som underlag för fakturering, som för varje reparation anger reparationsnummer, vilken spis det gäller, kundnr och sammanlagda kostnaden för reparationen (samma beräkning som i a).
- Antag att du har en sådan vy (med attributen reparationsnummer, spisens serienummer, kundnr, kostnad), beräkna vad de olika spismodellerna kostat att reparera sammanlagt. Modellbeteckningen finns i tabellen Spis. Resultatlistan ska innehålla modellbeteckning och summan av reparationskostnaderna för alla reparationer av spisar av den modellen. Listan ska inkludera alla modeller, även de som inte behövt repareras.

8. **Design ER:** Efter en tid vill man revidera systemet och göra om designen så att man eliminerar problemen med olika priser, antal reservdelar och garantireparationer. All information som tabellerna ovan innehåller behöver kunna representeras på något sätt UTOM att Reparationsnummer inte är unikt längre utan det är ett löpnummer för reparationerna på en viss spis. Man kan alltså inte identifiera en viss reparation längre utan att veta vilken spis det var som reparerades. Man vill dessutom kunna hantera att det används flera olika reservdelar vid en reparation, identifiera garanti-reparationer, samt ha samma pris på reservdelar oavsett vilken reparation de används i. Dessutom vill man kunna lagra flera olika telefonnummer till kund.
- Gör om designen genom att göra ett ER-diagram för det förbättrade systemet. Glöm inte markera lämpliga nycklar, kardinalitet och deltagande i diagrammet som vi gjort i kursen. Om du tycker att du behöver göra ytterligare antaganden om systemet, skriv ner dem tydligt. (20 p)

9. **Relationsmodell:** Du har nedanstående ER-diagram. Konvertera det till relationsmodell. Markera primärnycklar och främmande nycklar som vi gjort i kursen. Namnge nya relationer logiskt. (16 p)



10. **Transaktioner:** Antag att du har transaktionen nedan där A, B och C är data i en databas. Placera in kommandona ReadLock (X), WriteLock(X) och Unlock(X) (där du ersätter X med någon av bokstäverna som används i koden) på lämpliga platser i koden nedan så att du låser data så kort tid som möjligt samtidigt som du uppfyller protokollet för *Rigorös tvåfaslåsnig*. (6 p)

```
Read(A)
Read(B)
B = B - A
Write (B)
Read C
C = C - B
Write(C)
Commit
```

Lycka till!