

# Försättsblad till skriftlig tentamen vid Linköpings universitet



Datum för tentamen	2019-08-16
Sal (1)	<u>TER1(5)</u>
Tid	8-12
Utb. kod	732G16
Modul	TEN1
Utb. kodnamn/benämning Modulnamn/benämning	Databaser: design och programmering Tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	10
Jour/Kursansvarig Ange vem som besöker salen	Eva Ragnemalm
Telefon under skrivtiden	070-1907391
Besöker salen ca klockan	ca kl 10.00
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Erika Larsson erika.larsson@liu.se 013-281868
Tillåtna hjälpmedel	ordbok för översättning från svenska till modersmål, vid behov.
Övrigt	Följ instruktionerna på tentaomslaget. Svara på EN uppgift per papper.
Antal exemplar i påsen	

Linköpings Universitet  
Institutionen för datavetenskap  
Eva Ragnemalm

# TENTAMEN

## 732G16 Databaser Design och programmering

Datum: 2019-08-16 kl 8-12

Jourhavande lärare: Eva Ragnemalm

Tel: 070-1907391

Besöker tentamenslokalen: c:a kl 10

Hjälpmedel: ordbok för översättning från svenska till modersmål, vid behov.

Följ instruktionerna på tentaomslaget. Svara på EN uppgift per papper.

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

Betyg: G: 32p VG: 47p Max: 62p

1. **Begrepp:** Förklara kort likheter och skillnader mellan dessa par av begrepp. Använd gärna exempel. (6p)
  - a. Konceptuell datamodell och implementationsmodell.
  - b. DDL och DML
  - c. Databasintegritet och referensintegritet

**Kombinationsuppgift (Normalisering, Fysiska databasen, SQL:** Här finns information relevant för uppgift 2-4. En flygplats lagrar information om flighter och biljetter i en databas. En flight är flygresan från denna flygplats till en destinationsflygplats som görs en viss veckodag, har en avgångstid, ankomsttid (i lokal tid) och en flygtid. En flight återkommer alltså varje vecka. En biljett gäller en viss flight en viss vecka ett visst år och är antingen ekonomi eller business-klass. Biljetterna har ett unikt biljettnummer. Varje biljett kan gälla för flera resenärer, och varje resenär representeras med personnummer, namn och en tilldelad sittplats. En viss resenär finns bara i databasen med sin aktuella biljett.

En klandig programmerare gjorde nedanstående tabeller för att hantera denna information.

```
Flight(ID, destination, veckodag, avgångstid, ankomsttid, flygtid)
```

```
Biljett(resenärPnr, namn, biljettNr, klass, flight, år, vecka, sittplats)
```

2. Vilken normalform uppfyller Biljett? Varför? Motivera utifrån normaliseringsvillkoren och visa vilka funktionella beroenden du ser utifrån beskrivningen. De är inte så många. ResenärPnr är enda kandidatnyckel. (6p)
3. Programmeraren behöver hjälp med hanteringen av databasen. Skriv SQL-frågor för följande åtgärder: (6p)
  - a. Vilken destination, avgångstid, ankomsttid och flygtid gäller för biljett nummer BC123? Se till att din utskrift bara genererar en enda rad oavsett hur många resenärer som ingår i biljetten.
  - b. Hur många passagerare finns i Business-klass på flight nr SK721 onsdag vecka 43 (i år). Antag att klass="Business" för dessa.
  - c. Resenären 891231-1234 med biljett nummer BC123 avbokar sin resa. Ta bort personen utan att ta bort ev. andra passagerare på den biljetten.
4. Med tiden blev tabellen Biljett ganska stor. Den sorteras på resenärPnr (resenärens personnummer) men vid incheckning söker man efter alla resenärer på en viss biljett, alltså med ett visst biljettnummer. Databasen lagras på en hårddisk med accesstid på 10ms ( $10^{-3}$ s) och 2048 byte

stora block. Varje rad i Biljett tar 500 bytes och tabellen har 10 200 rader. Biljettnumret tar upp 10 byte och en pekare till ett block 10 byte. (10p)

- Hur stor plats (räknat i block) tar tabellen Biljett?
- Hur lång tid tar det att söka ut alla poster som har ett visst biljettnummer? (Observera att man vet inte hur många personer (poster) det finns för varje biljett). Du kan svara i antal accesser eller sekunder.
- Hur lång tid tar samma sökning om man skapar ett index på biljettnummer med filen fortfarande sorterad på personnummer?
- Hur skulle man kunna snabba upp denna sökning ännu mer? Förklara kort varför det går snabbare.

Mattetips:  $2048/10=204.8$ ,  $2048/20=102.4$ ,  $2048/500=4.096$ ,  $10200/2=5100$ ,  $10200/4=2550$ ,  $10200/100=102$ ,  $10200/102=100$ ,  $10200/500=20.4$ ,  $10200/2048=4.98$ ,  $\log_2(100)=6.64$ ,  $\log_2(102)=6.67$ ,  $\log_2(500)=8.97$ ,  $\log_2(2048)=11$ ,  $\log_2(2550)=11.3$ ,  $\log_2(5100)=12.3$ ,  $\log_2(10200)=13.3$

5. **Transaktioner:** En transaktion ska vara ACID. Förklara vad varje bokstav står för. Beskriv också kort hur man i en databashanterare uppnår detta. (8p)

6. **Design:** Försäkringsbolaget **Trygga Handen** behöver en databas för sina försäkringar och försäkringstagare. Man vill också kunna hålla reda på de skador som anmälts. Försäkringstagarna registreras med namn, personnummer, adress (bestående av gatans namn, nummer, lägenhetsnummer/trappor, postnummer och postort). Varje försäkringstagare kan ha flera försäkringar hos Trygga Handen, men varje försäkring har en försäkringstagare. Varje försäkring gäller ett visst objekt (en bil, hus, båt osv).

Man lagrar information om skador. En skada är alltid relaterad till ett visst försäkringsobjekt och klassificeras i olika typer (trafikolycka, vattenskada, stöld osv) samt typ av ersättning (reparationskostnader, skadestånd, ersättningsobjekt osv) samt storleken på ersättning (i kronor). Man lagrar också datum för skadan, en fritext-beskrivning av skadan och en utredningsansvarig person (en av de anställda på försäkringsbolaget). Typ av ersättning och storleken på ersättningen sätts till null respektive noll medan utredning pågår. En skada identifieras genom vilket objekt som skadats och vilket datum skadan skedde (om ett objekt utsatts för flera typer av skador samma dag sätts skadetyper till "flera" och beskrivs i fritexten).

För varje försäkring anges vilka försäkringsprodukter som ingår (för hus tex hemförsäkring, all-risk, reseförsäkring mm och för bil t.ex trafik, delkasko osv). Dessa kan vara flera per försäkring. Den aktuella årspremien lagras också, samt till vilket datum försäkringen gäller (den s.k. huvudförfallodagen. Notera att ett datum består av år, månad och dag). Man använder ett speciellt försäkringsnummer som identifikation på försäkringen.

Försäkringsobjektet, det som är försäkrat, beskrivs med identifiering (för bilar registreringsnummer, för hus fastighetsbeteckning osv), tillverkningsår (byggår eller dylikt) samt en fritext-beskrivning som tar upp all information som premien beräknas på. Ett objekt kan naturligtvis vara inblandat i flera skaderapporter men hör till en viss försäkring.

Rita upp ett ER-diagram för denna databas. Glöm inte markera nycklar med understrykning, deltagande med enkel/dubbelstreck och kardinalitet med 1/N/M. (16p)

7. **Relationsmodell:** Ett nytt rapporteringssystem för folkhälsa planeras i Sverige. Socialstyrelsen (SOS) har redan designat ett ER-diagram över hur de vill att databasen ska vara strukturerad, men har inte lyckats översätta ER-diagrammet till en relationsdatabas. Hjälps SOS genom att översätta diagrammet till en lämplig relationsdatabas i enlighet med översättningsstegen beskrivna i kursboken och/eller föreläsningarna. För denna fråga ska du **inte** normalisera relationerna. Markera relationernas primärnycklar med understrykning och främmande nycklar med pil från referensattributet till den utpekade (refererade) relationen. Tänk på hur du sätter namn på relationer och attribut så att den som rättar förstår var de kommer ifrån. (10p)

