

Örebro universitet
Akademin för naturvetenskap och teknik
[Thomas Padron-McCarthy \(thomas.padron-mccarthy@oru.se\)](mailto:thomas.padron-mccarthy@oru.se)

Tentamen i

Databasteknik

för D1, SDU1 m fl

lördag 5 mars 2011

Gäller som tentamen för:

DT1012 Datateknik A, Databasteknik, provkod 0100
DT1007 Datateknik A, Tillämpad datavetenskap, provkod 0310

Hjälpmedel:	Inga hjälpmedel.
Poängkrav:	Maximal poäng är 35. För betyget 3 respektive G krävs 17 poäng.
Resultat och lösningar:	Meddelas på kursens hemsida eller via e-post senast lördag 26 mars 2011.
Återlämning av tentor:	Efter att resultatet meddelats kan tentorna hämtas på universitetets centrala tentamensutlämning.
Examinator och jourhavande:	Thomas Padron-McCarthy, telefon 070 - 73 47 013.

- Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng. Oklara formuleringar kommer att misstolkas.
 - Skriv den personliga tentamenskoden på varje inlämnat blad. Skriv *inte* namn eller personnummer på bladen.
 - Skriv bara på en sida av papperet. Använd inte röd skrift.
 - Antaganden utöver de som står i uppgifterna måste anges. Gjorda antaganden får inte förändra den givna uppgiften.
 - Skriv gärna förklaringar om hur du tänkt. Även ett svar som är fel kan ge poäng, om det finns med en förklaring som visar att huvudtankarna var rätt.
-

LYCKA TILL!

Scenario till uppgifterna

Vi ska bygga en datorbaserad kalender, där personer kan boka möten och annat. Kalendern ska lagra sina data i en databas, och det man behöver hålla reda på är följande saker:

1. **Dagar**. Vi gör helt enkelt så att vi lägger in alla de dagar som man ska kunna boka saker på. Varje dag har ett **datum**, till exempel **2011-03-05**. För att underlätta programmeringen får varje datum dessutom ett unikt **nummer**, till exempel **17**.
2. **Personer** som ska kunna boka in saker. Varje person har ett unikt **nummer** och ett (inte nödvändigtvis unikt) **namn**.
3. **Lokaler** där man ska kunna boka saker. Varje lokal har ett unikt **nummer**, ett unikt **namn** och ett antal **platser**, som anger hur många personer som maximalt får plats i lokalen.
4. **Bokningar**. Varje bokning hör till en viss **dag** och den har en **starttid** och en **sluttid**. Varje bokning hör också ihop med en eller flera **personer**. (Om flera personer ska ha ett möte, gör de alltså en enda bokning för det mötet.) Varje bokning kan också, men måste inte, höra ihop med en **lokal**. Dessutom har bokningen en **rubrik**, och den kan ha en **kommentar**.

Uppgift 1 (5 p)

Rita ett ER- eller EER-diagram för den beskrivna databasen. Använd informationen i scenariot ovan, men tänk också på att det ska gå att svara på frågorna i uppgift 3 nedan.

ER- och EER-diagram kan ritas på flera olika sätt. Om du använder en annan notation än kursboken, måste du förklara den notation som du använder.

Uppgift 2 (5 p)

Implementera den beskrivna databasen i relationsmodellen, dvs översätt ER- eller EER-diagrammet till tabeller.

Du behöver inte skriva **create table**-kommandon i SQL, men du ska ange vilka relationer som finns och vilka attribut varje relation innehåller. Ange också alla kandidatnycklar, vilken av dessa som är primärnyckel, samt vilka referensattribut som finns och vad de refererar till.

Implementationen ska vara bra.

Uppgift 3 (10 p)

Formulera följande frågor i SQL:

- a) Vilket nummer har lokalen **T120**?
- b) Vilka bokningar finns det i lokalen **T122** idag (**2011-03-05**)? Det räcker med rubrikerna i svaret.
- c) Jag heter **Thomas**. Det finns bara en Thomas i databasen. Vilka bokningar har jag inbokade idag (**2011-03-05**)? Det räcker med rubrikerna i svaret.
- d) Hur många bokningar har jag (**Thomas**) totalt?
- e) När börjar den tidigaste bokningen i **T124** idag (**2011-03-05**)?

Definiera gärna vyer om det underlättar frågorna.

Uppgift 4 (3 p)

Databasen innehåller många tusen rader i varje tabell. De tre SQL-frågorna a, b och c i uppgiften ovan körs väldigt ofta (men med andra konstanter, till exempel andra personnamn än **Thomas**) och behöver snabbas upp. Vi märker att det inte finns några index alls i databasen, inte ens på nycklar. Vilka index bör man skapa för att just de frågorna ska gå snabbt att köra? Motivera svaret!

Uppgift 5 (3 p)

Vilken databashanterare bör vi välja för vår databas? Nämn minst tre olika databashanterare, och tala om varför, eller varför inte, de passar för den här tillämpningen.

Uppgift 6 (5 p)

Det kan uppstå krockar mellan bokningar, dels genom att en person är uppbokad för två eller flera aktiviteter samtidigt, och dels genom att en lokal är uppbokad för två eller flera aktiviteter samtidigt.

Vi ska komplettera det ursprungliga ER-diagrammet från uppgift 1 med:

1. **Krockar**. En krock har ett unikt **nummer**, en **starttid** och en **sluttid**, och den hör ihop med två eller flera **bokningar**.
2. En krock kan vara en **personkrock**, som hör ihop med en viss **person**.
3. Alternativt kan en krock kan vara en **lokalkrock**, som hör ihop med en viss **lokal**.

a)

Komplettera det ursprungliga ER-diagrammet från uppgift 1 med de nya entitetstyperna.

Du behöver inte rita om hela diagrammet från uppgift 1, utan det räcker att visa vilka entitetstyper från uppgift 1 som de nya entitetstyperna ska kopplas ihop med.

b)

Översätt till tabeller. Redovisa svaret på ett tydligt och lättfattligt sätt.

Uppgift 7 (4 p)

En förening har sex olika medlemmar, och lagrar dem i en tabell som ser ut så här:

Nummer	Namn	Adress	Postnummer	Ort
1	Bo Ek	Granvägen 16	702 21	Örebro
2	Anna Berg	Granvägen 23	702 21	Örebro
3	Lars Skog	Tallvägen 4	702 21	Örebro
4	Bo Lund	Storgatan 3	703 61	Örebro
5	Bo Lund	Storgatan 3	642 37	Flen
6	Bo Lund	Storgatan 3	642 37	Flen

a) Vilken är den högsta normalform som tabellen uppfyller? Motivera svaret.

b) Baserat på ditt svar på fråga a, vad bör man göra? Varför?
