

Tentamen i
Databasteknik
för D1, SDU1 m fl

onsdag 19 augusti 2009 kl 08:15 - 12:15

Gäller som tentamen för:
DT1012 Datateknik A, Databasteknik, provkod 0100
DT1007 Datateknik A, Tillämpad datavetenskap, provkod 0310

Hjälpmedel:	Inga hjälpmedel.
Poängkrav:	Maximal poäng är 40. För betyget 3 respektive G krävs 20 poäng.
Resultat och lösningar:	Meddelas på kursens hemsida eller via e-post senast onsdag 9 september 2009.
Återlämning av tentor:	Efter att resultatet meddelats kan tentorna hämtas på institutionen. Man kan också få sin rättade tenta hemskickad.
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.
 - Lös bara en uppgift per blad. 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

Musiktjänsten **Snuttify** gör det möjligt för användare att lyssna på musik via nätet.

Personen som sitter dygnet runt på Snuttifys kontor och spelar skivor börjar emellertid tröttna på det, så nu vill man ha en datoriserad lösning som bygger på en databas.

Man behöver lagra följande saker i databasen:

- **Användare.** Varje användare har ett unikt nummer, till exempel **17**, ett riktigt namn, en adress, ett logginnamn och ett lösenord.
- **Genrer**, dvs musiktyper, till exempel **Death Metal** och **Christian Gangsta Rap**.
- **Artister** som gjort musiken. Varje artist har ett unikt nummer, ett namn och en huvudsaklig genre, som den artisten gjort mest musik inom.
- **Skivor.** Varje skiva i databasen har ett unikt nummer, ett namn, och har gjorts av en eller flera artister. Även skivan har en huvudsaklig genre.
- **Låtar.** Musiken lagras inte direkt i databasen, utan varje låt lagras som en separat fil. Varje låt i databasen har ett namn, ett unikt nummer, ett filnamn där den ligger lagrad, och en längd, mätt i sekunder. Varje låt hör till en viss skiva, och en viss genre.
- Vi vill kunna låta varje användare sätta betyg (**0** till **10**) på varje låt som hon eller han lyssnat på, och vi vill lagra alla dessa betyg i databasen.



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 4 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 (6 p)

Implementera den beskrivna databasen i relationsmodellen, dvs översätt 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 (5 p)

Att "implementationen ska vara bra" i frågan ovan innebär bland annat att tabellerna ska uppfylla vissa normalformer.

a) (2p) Vilken normalform vill man normalt att tabellerna i en databas ska uppfylla? Varför?

b) (3p) Välj *en* av tabellerna i din lösning på uppgift 2 ovan, och ange vilken som är den högsta normalform som den uppfyller. Visa också varför det är så!

Uppgift 4 (10 p)

Formulera följande frågor i SQL.

- (1p) Vilket lösenord har användaren med logginnamnet **I33th4xx0r**?
- (2p) Vilka låtar finns med på skivan **Burt's nr. 2** med bandet **Burt's**?



- (2p) Vilka av artisterna som finns med i databasen har inga skivor inlagda?
- (2p) Hur många låtar hör till genren **Techno-Industrial**?
- (3p) Vilken låt har fått högst genomsnittsbetyg?

Definiera gärna vyer om det underlättar, men skapa inte nya tabeller.

Uppgift 5 (3 p)

Databasen innehåller realistiska mängder data, dvs miljoner användare, tiotusentals skivor, och så vidare. De två SQL-frågorna a och b i uppgiften ovan körs väldigt ofta (men med andra konstanter) 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? Förklara varför!

Uppgift 6 (3 p)

Snuttifys databas ska kunna administreras av Snuttifys personal via webben. Vilka tekniker kan man använda för att de ska kunna komma åt databasen via webben? Förklara kort hur de olika teknikerna fungerar.

Uppgift 7 (3 p)

Vilken databashanterare bör Snuttify välja för sin databas? Nämn några förslag på databashanterare, och tala om varför, eller varför inte, de passar för Snuttifys tillämpning.

Uppgift 8 (5 p)

a) (3p) Visa med ett par enkla exempel vad man har SQL-kommandona **grant** och **revoke** till. Glöm inte att förklara vad som händer!

b) (2p) Visa hur man kan kombinera **grant** och **revoke** med vyer för att ge en mer finkornig kontroll av vad användarna får göra med databasen.
