

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

Tentamen i

Databasteknik

för D1 m fl

onsdag 22 augusti 2018

Gäller som tentamen för:
DT105G Databasteknik, provkod 0100

| | |
|------------------------------------|--|
| Hjälpmedel: | Ordbok för översättning. |
| Poängkrav: | Maximal poäng är 34. För godkänt betyg krävs 18 poäng, varav minst fyra poäng på uppgift 1 och minst fem poäng på uppgift 2. |
| Resultat: | Meddelas på kursens hemsida eller via e-post senast onsdag 12 september 2018. |
| Återlämning av tentor: | Elektroniskt via Studentforum. |
| 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 och tvetydiga 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

Här är en grantopp med några kottar:



Vi ska lagra data om skogar, träd och kottar. Kottarna har längd och vikt. Varje träd har en höjd och noll eller flera kottar. En skog innehåller noll eller flera träd.

Det man behöver hålla reda på är, mer detaljerat, följande:

- **Skogar.** Varje skog har ett unikt **namn** och den innehåller ett eller flera träd.
- **Träd.** Varje träd har en höjd, som mäts i meter, och noll eller flera kottar.
- **Kottar.** Varje kotte har en längd och en vikt, och den sitter på ett visst träd. Lösa kottar som ramlat ner från träden har vi inte med i databasen.
- Vi tar också **fotografier**. Varje foto har tagits en viss dag, och det visar noll eller flera av träden, och noll eller flera av kottarna. Samma kottar och träd kan förstås finnas med på flera fotografier.

Om man ska lagra sina data i en relationsdatabas behöver man ge träden, kottarna, fotografierna och gärna också skogarna unika identifierare, till exempel unika nummer.

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 (6 p)

Implementera den beskrivna databasen i relationsmodellen, dvs översätt ER-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. Definiera gärna vyer om det underlättar, men skapa inte nya tabeller.

- a) (1p) Vilka skogar har namn som börjar på **M**, som till exempel **Markaskogen** och **Mörkmården**?
- b) (2p) Vad heter de skogar där det finns något träd som finns med på något foto taget idag (**2018-08-22**)?
- c) (2p) Hur många träd finns det i Markaskogen?
- d) (2p) Hur högt är det högsta trädet i Markaskogen?
- e) (3p) Hur många kottar har det högsta trädet i Markaskogen?

Uppgift 4 (4 p)

Fråga a och b i uppgiften ovan körs väldigt ofta, men kanske med andra konstanter, till exempel att man söker efter skogar som börjar på **K**. De tar för lång tid att köra, och vi vill skapa index för att snabba upp dem. Från början finns det inga index alls i databasen, inte ens på primärnycklar. I databasen finns det tusentals skogar och fotografier, många miljoner träd och minst en miljard kottar.

a) Ange vilka index man bör skapa för att frågorna a och b ska gå snabbare att köra.

b) Ange ett index som *inte* skulle göra någon av frågorna snabbare, och förklara, för båda frågorna, varför det inte förbättrar tiden.

Uppgift 5 (3 p)

Många användare behöver jobba med den beskrivna databasen.

a) Föreslå en databashanterare som kan vara lämplig att använda. Motivera varför den är lämplig!

b) Föreslå en databashanterare som inte är lämplig. Motivera varför den inte är lämplig!

Uppgift 6 (3 p)

Som vi sett i kursen kan både triggers och lagrade procedurer innehålla en sorts utökad SQL, med konstruktioner från vanliga programspråk som if-satser och loopar.

Vad är det för skillnad på en trigger och en lagrad procedur? Hur används de?

Uppgift 7 (3 p)

Man kan bygga en databasbaserad webbplats på olika sätt. En metod är CGI-script, men de har ganska dåliga prestanda och är inte så vanliga längre.

Hur fungerar CGI-script, och varför har de så dåliga prestanda, särskilt när man använder dem för en databasbaserad webbplats?
