

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

Tentamen i Databasteknik

för D1, SDU1 m fl

tisdag 12 januari 2010

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 20 poäng. För den som följt kursen hösten 2009 ger varje i tid inlämnad inlämningsuppgift en extra poäng. Den som <i>inte</i> gått kursen hösten 2009 får dessa (fem) extrapoäng ändå.
Resultat och lösningar:	Meddelas på kursens hemsida eller via e-post senast tisdag 2 februari 2010.
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.

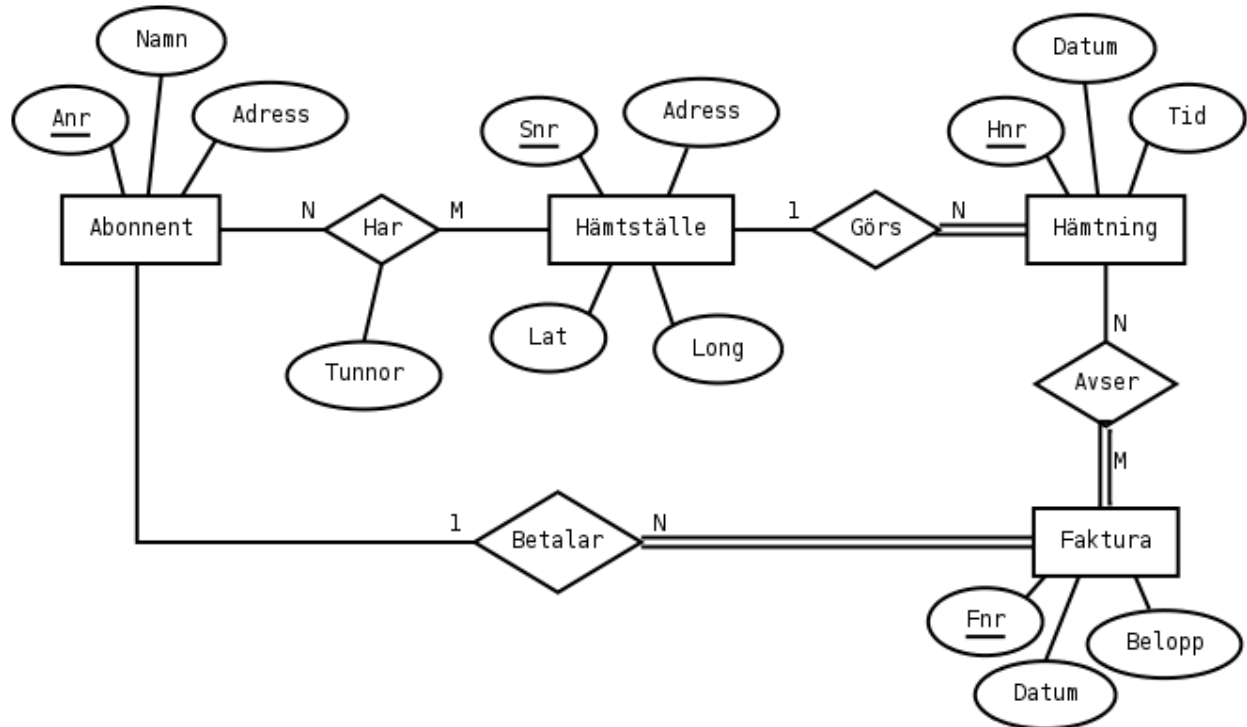
- Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng. Oklara formuleringar kommer att misstolkas.
 - Skriv den personliga tentamenskoden på varje inlämnat blad. Skriv *inte* namn eller personnummer på bladen.
 - Lös bara en uppgift per blad. Skriv bara på en 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

Tekniska verken i Örebro kommun ansvarar bland annat för sophämtningen, och man behöver förstås en databas för att hålla reda på verksamheten.

Så här ser kan ett ER-diagram för databasen se ut:



Man behöver lagra följande saker i databasen:

- **Abonnenter**. Det kan till exempel vara en villaägare, ett företag eller en hyresvärd. Det är abonnenterna som betalar för sophämtningen.
- **Hämtställen**, där man ställer ut sina soptunnor och sopbilarna sen kommer och hämtar soporna. De flesta hämtställen hör till en enda abonnent, men ett hämtställe kan också delas mellan flera abonnenter, till exempel om flera hushåll ska ställa fram sina sopor på en viss plats om sopbilen inte kan köra ända fram till varje hus. En abonnent kan också ha flera hämtställen, till exempel om en hyresvärd har flera hyreshus. För varje hämtställe där en abonnent ställer ut soptunnor, ska man också ange hur många soptunnor som ska hämtas för just den abonnenten på just det hämtstället.
- **Hämtningar**, som är när en sopbil åker och hämtar soptunnor på ett hämtställe.
- **Fakturer**. Abonnenterna ska ju betala för sophämtningen, så man vill lagra fakturer ("räkningar") i databasen också. En faktura gäller för ett antal hämtningar, och för en viss abonnent. (Det kommer att finnas hämtningar i databasen som inte hör ihop med någon faktura, helt enkelt för att man inte hunnit skicka ut fakturer på dem än.)

Uppgift 1 (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 2 (10 p)

Formulera följande frågor i SQL:

- Vilken position (som anges med koordinaterna **Lat** och **Long**) har hämtstället på **Grangrensvägen 2**?
- På vilka hämtställen gjordes det hämtningar **2010-01-12** klockan **09:03**? Vi vill veta adresserna för dessa hämtställen.
- Ange namn och adress för de abonnenter som har hämtställe på **Talltoppstrappan 9**.
- Jag heter **Thomas Padron-McCarthy**. När har mina sopor hämtats? Jag vill veta datum och tid.
- Vilka abonnenter har aldrig fått sina sopor hämtade? Vi vill veta de abonnenternas namn och adress.
- Vilken abonnent har flest soptunnor? Vi vill veta namn och adress för den abonnenten.

Uppgift 3 (3 p)

Databasen innehåller minst flera tusen rader i varje tabell. De tre SQL-frågorna a, b och c 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? Motivera svaret!

Uppgift 4 (3 p)

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

Uppgift 5 (3 p)

Visa med ett par enkla exempel från databasen i scenariot vad man har SQL-kommandona **grant** och **revoke** till. Glöm inte att förklara vad som händer!

Uppgift 6 (6 p)

En databas innehåller båtar. Alla båtar har ett unikt nummer, en vikt och ett namn. En del av båtarna är fiskebåtar, och varje fiskebåt har en fångst, som mäts i ton. En del av båtarna är ubåtar, och varje ubåt har ett maxdjup. En del av ubåtarna är militär-ubåtar. En del av militär-ubåtarna är attack-ubåtar och en del är missil-ubåtar. Varje attack-ubåt har en tuffhet, och varje missil-ubåt har ett missilantal.

Dessutom finns det personer i databasen. Varje person har ett namn och ett unikt nummer. En del av personerna är sjömän, och andra är kaptener. Varje båt har en kapten. Ubåtarna har dessutom en besättning på en eller flera sjömän.

a) Rita ett EER-diagram över den beskrivna databasen.

b) Översätt EER-diagrammet till tabeller.

I b-uppgiften behöver du 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.

Uppgift 7 (4 p)

Ibland talar man om **ACID-transaktioner**, där bokstäverna i förkortningen **ACID** står för

- **atomicitet**,
- **consistency preserving**, dvs "konsistensbevarande" eller "konsekvensbevarande",
- **isolering** och
- **durability**, dvs "hållbarhet".

Förklara för var och en av de fyra egenskaperna vad den egentligen betyder, och ge ett exempel på vad som kan hända om den saknas.
