

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

tisdag 15 mars 2016

Gäller som tentamen för:  
DT105G Databasteknik, provkod 0100  
DT1026 Datateknik A, Databasteknik, provkod 0100  
DT1030 Datateknik A, Tillämpad datavetenskap, provkod 0310

---

<b>Hjälpmedel:</b>	Ordbok för översättning.
<b>Poängkrav:</b>	Maximal poäng är 35. För betyget 3 respektive G krävs 18 poäng, samt minst fyra poäng på uppgift 1 och minst fem poäng på uppgift 2.
<b>Resultat:</b>	Meddelas på kursens hemsida eller via e-post senast tisdag 5 april 2016.
<b>Återlämning av tentor:</b>	Elektroniskt via Studentforum.
<b>Examinator och jourhavande:</b>	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

Vår lokala pizzeria ska datoriseras. Vår uppgift är att hjälpa dem att skapa en databas.

Det vi ska lagra i databasen är följande:

- **Pizzor.** Varje pizza har ett unikt **nummer**, till exempel **17**, och ett unikt **namn**, till exempel **Bearnaise-symfoni**.
- **Ingredienser.** Varje ingrediens har ett unikt **nummer**, till exempel **17**, och ett unikt **namn**, till exempel **rostade champinjoner**. Varje pizza innehåller en eller flera av dessa ingredienser. Samma ingrediens kan förstås finnas på flera olika pizzor.
- **Prisklasser.** I stället för att prissätta varje pizza för sig, delar vi in dem i prisklasser. Varje pizza tillhör en prisklass. Varje prisklass har ett unikt **nummer** och ett (inte nödvändigtvis unikt) **pris**. Till exempel kan pizzan **Lyx-kebaborama** tillhöra prisklass **5**, som kostar **80** kronor.
- Vi ska också hålla reda på **pizzabagarna**. Varje pizzabagare har ett unikt **nummer**, ett (inte nödvändigtvis unikt) **namn** och ett (inte nödvändigtvis unikt) **telefonnummer**.

## 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 (7 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. Använd dina tabeller från uppgift 2. Definiera gärna vyer om det underlättar.

- Vad heter den eller de pizzabagare som har telefonnummer **070-73 47 013**?
- Här har vi en pizza som heter **Entrecote-båt**. Vad kostar den?
- Vad heter de ingredienser som pizzan **Entrecote-båt** innehåller?
- Hur många pizzor finns i prisklass 1? Vi vill alltså veta antalet, inte vilka det är.
- Vad heter den dyraste pizza som innehåller kantareller?

## Uppgift 4 (3 p)

Det trettio olika pizzor i fem olika prisklasser, tjugo ingredienser och fem pizzabagare. Databasen används en gång varje år när pizzerians ägare ska skriva ut en ny prislista.

- Vilka index är det nödvändigt att skapa? Motivera svaret!
- Föreslå en databashanterare som kan vara lämplig att använda. Motivera varför den är lämplig!

## Uppgift 5 (5 p)

En viss databashanterare har så kallad "auto-commit", vilket betyder att varje SQL-kommando räknas som en egen transaktion, så länge man inte uttryckligen startar en transaktion med kommandot "start transaction".

Vi startar två olika klientprogram som kopplar upp sig mot databasservern, och ger följande SQL-kommandon, i den angivna ordningen, i de två klienterna. Vad blir resultatet av var och en av de tio select-frågorna?

Klient 1	Klient 2
create table Djur (Nummer integer not null primary key, Namn char(10) not null); insert into Djur values (1, 'Struts'); select * from Djur; -- Fråga 1	
	select * from Djur; -- Fråga 2 insert into Djur values (2, 'Varg');
select * from Djur; -- Fråga 3 start transaction; insert into Djur values (3, 'Ko'); select * from Djur; -- Fråga 4	
	select * from Djur; -- Fråga 5
rollback; select * from Djur; -- Fråga 6	
	select * from Djur; -- Fråga 7
start transaction; insert into Djur values (4, 'Äpple');	
	select * from Djur; -- Fråga 8
commit; select * from Djur; -- Fråga 9	
	select * from Djur; -- Fråga 10

## Uppgift 6 (5 p)

Förklara följande begrepp från databasområdet:

- relation
  - referensintegritet
  - null
  - XML
  - yttre join
-