

Tekniska Högskolan i Linköping  
Institutionen för datavetenskap  
Eva Ragnemalm

# TENTAMEN

## 729G28 Webprogrammering och databaser

Datum: **2016-11-18**

Tid: **14-18**

Jourhavande lärare: **Eva Ragnemalm**

Tel: **070-1907391**

Besöker tentamenslokalen c:a kl 15

Hjälpmedel: **Inga**

Följ instruktionerna på tentaomslaget.

### Poängfördelning

Uppgift	Poäng
1	6
2	10
3	10
4	5
5	8

G: 21 p VG: 30 p Max: 39p

1. **Begrepp:** Redogör kortfattat för följande par av begrepp och hur de hänger ihop (6p):

- a. Databas och databashanterare
- b. ER-modell och relationsmodell
- c. Referensattribut (främmande nyckel) och referensintegritet

2. **Begrepp och SQL:** Antag att du har nedanstående tabeller:

Vara

ArtNr	Namn	Antal	Leverantor
1	Jackor med fler fickor än du kan hitta	20	1
2	Smör i solen	100	2
3	Trebenta byxor	150	3
4	Bapelsiner	35	4
5	Prins Korv	50	4
6	Drottning Sylt	10	4
7	Mollijoxer	250	3
8	Mojänger	55	3

Leverantor

IDnr	Namn	Adress	Land
1	Inteenfemma	Majvägen 5	Sverige
2	Tidig	Köbenhavnsvej 2	Danmark
3	WaitTulip	Allroad	USA
4	Kungsverket	Galavägen	Sverige

Tabellerna är skapade med följande tre kommandon:

```
create table Vara (ArtNr int, Namn varchar (50), Antal int, Leverantor
int) Primary key (ArtNr));
```

```
create table Levarantor (IDnr int, Namn varchar (25) Adress varchar
(30), Land varchar (15) Primary key (IDnr));
```

```
Alter table Vara add constraint varuLeverantor Foreign key (Leverantor)
references Leverantor (IDnr);
```

- a. Skriv ett SQL-kommando som listar namnet på alla varor och för varje vara även namnet på leverantören. (2p)
- b. Skriv ett SQL-kommando som listar alla leverantörer och för varje leverantör hur många olika varor de levererar (t.ex. Kungsverket levererar 3 olika varor). (3p)

Nedan listas ett antal kommandon mot denna databas.

```
1. Insert into Vara values (9, "Mackapärer", 23, 3);
```

```
2. Insert into Leverantor values (4, Feletonbolaget, Ringvägen 3, Sve-
rige);
```

```
3. Drop table Vara;
```

4. Alter table Vara add constraint varuLeverantor Foreign key (Leverantor) references Leverantor (IDnr);
  5. Delete from Leverantor where Land="USA";
  6. Delete from Vara where Leverantor=1;
  7. Create table Bestallning (OrderNr int, vara int, antal int, kund int, Primary key (OrderNr);
- c. Vilka av de ovanstående kommandona påverkar databasens *schema* och vilka påverkar dess *instans* (bortse från ev. problem med kommandot)? (2p)
- d. Vissa av kommandona ovan kommer inte att gå att genomföra på grund av integritetsproblem. Vilka? Förklara kort varför respektive kommando inte kan genomföras. (3p)
3. **ER-modellering:** Linköpings flygplats skall expandera och i samband med detta skall man skapa en ny relationsdatabas för en del av flygplatsens aktiviteter. Man vill hålla reda på sina flygplan, flighter och kunder för att kunna sköta bokning och skapande av tidtabeller utan för mycket krångel. Informationsbehovet i databasen beskrivs nedan. All information som lagras i databasen kommer att finnas kvar i flera år.
- En flight är en planerad resa till en viss destination (flygplats) en viss tid en viss veckodag. Flighter identifieras med en kod, t.ex. SK721.
  - En flight med ett visst nummer flygs oftast varje vecka och kan tilldelas olika flygplan och gate vid olika tillfällen (men samma skulle kunna återkomma). Vilken gate man ska gå till beror alltså på exakt vilken dag (vilken vecka) man flyger. Flighten går dock alltid till samma plats och avgår samma tid på samma veckodag.
  - En passagerare flyger en viss tur av en flight, dvs en viss vecka. För varje passagerare på en flygtur vill man spara namn, biljettnummer, vilken flight det är, vilken vecka vilket år (t.ex 4716 för vecka 47 år 2016). Denna information sparas då biljetten köps. Vid incheckning sparas incheckade bagage och tilldelad sittplats. Varje passagerare kan checka in flera bagageartiklar men kan endast tilldelas en sittplats. Varje bagageartikel tilldelas en unik kod. Biljettnummer är unikt för varje passagerare. När biljetten köps bestäms om man reser i 1:a klass eller ekonomiklass men inte förrän vid incheckning tilldelas den specifika platsen.
  - Ett flygplan har ett unikt namn (t.ex. Viking Bird 38), en ägare (t.ex. SAS) och en typ (t.ex. Boeing 747) samt ett antal sittplatser. Sittplatserna är inte beroende av flygplanstyp utan av det individuella planet, varje plan levereras med egen design och egen fördelning mellan 1:a klass och ekonomiklass). Vilket flygplan som ska flyga en viss flight en viss vecka bestäms några dagar innan avgång och avgörs av behovet av olika platser.
  - En sittplats har ett nummer, en klassificering (1:a klass eller ekonomiklass) och hör till ett visst plan. Sittplatser kan naturligtvis vara tomma. Tilldelning av sittplatser sker vid incheckning. Designa en ER-modell för denna databas. Ange nycklar för entitetstyperna samt kardinalitet och deltagande för sambandstyperna. (10p)
4. **Normalisering:** En klantig programmerare lade upp följande tabell för Linköpings flygplats, se ovan (funkar inte riktigt som ovan, programmeraren fick inte riktigt med allt):  
 Flyg (FlightNr, Destination, veckodag, tid, vecka, år, planNamn, planTyp, antalFörsta, antalAndra, PassagerarNamn, BiljettNr, BagageNr, Gate, planÄgare)  
 Primärnyckel (och enda kandidatnyckel) är {FlightNr, vecka, år, BiljettNr}. Ur ovanstående beskrivning kan du räkna ut vilka funktionella beroenden som finns. Beskriv dem (antingen genom att

rita dem med pilar som på föreläsningen eller skriva dem på formen {determinant}->{de attribut som bestäms}).

- Vilken normalform uppfyller tabellen? Varför? (2p)
- Normalisera tabellen genom att undersöka tabellen med de olika normaliseringsvillkoren och dela upp den så att ingen redundans förekommer. Motivera varje uppdelning av tabellen med hjälp av normaliseringsvillkor och funktionella beroenden och se till att all information bevaras. (3p)

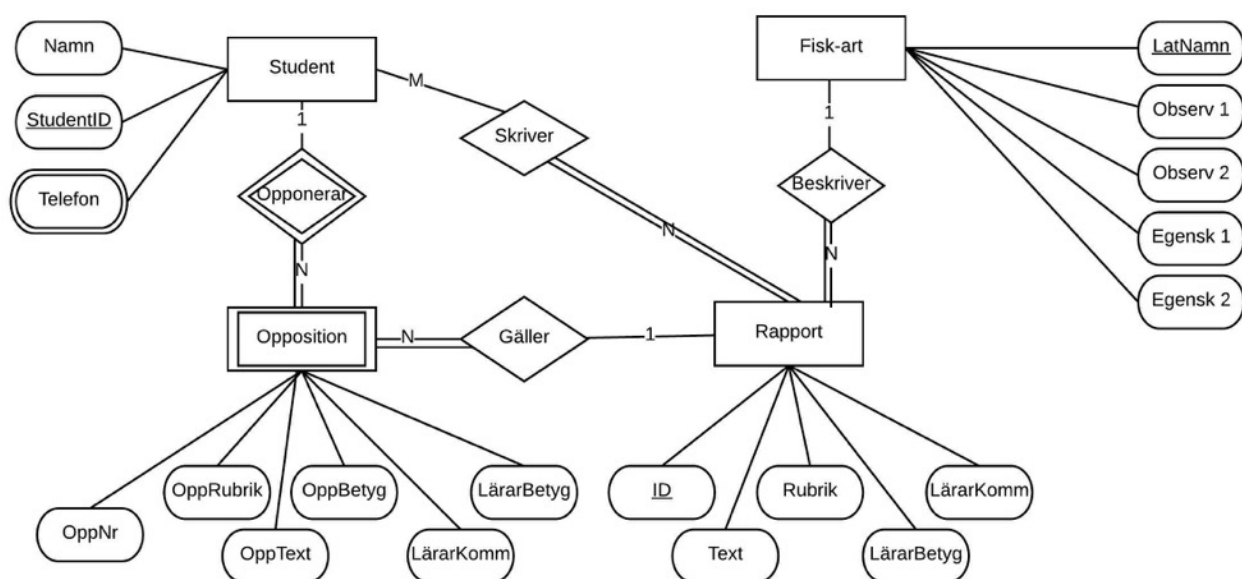
5. **Relationsmodellen:** I en kurs på Universitetet i Hawaii ska studenterna studera fiskbeståndet i korallreven. Varje student ska studera tre arter av fisk och delta i skrivandet av en rapport för respektive art. Varje rapport skrivs av en grupp studenter men avhandlar en fisk-art. Varje student ska också skriftligt opponera på och betygsätta sex andra rapporter.

Läraren vill lagra all information om detta i en databas. Om studenterna vill han lagra namn och student-id, samt telefonnummer till studenten, så att läraren kan påminna dem om inlämningar. Ofta har studenterna flera telefonnummer som de kan nå på. Studenterna lagras i databasen vid kursstart, dvs innan de gjort något i kursen. För varje rapport som sedan läggs in behöver läraren veta vilka som skrivit den, rubriken, själva texten, samt vilken fisk-art som behandlas. För enkelhets skull får varje rapport ett unikt ID.

För varje fisk-art lagras information om dess latinska namn, de två senaste observationerna i de lokala korallreven med datum och gps-koordinater, samt de två viktigaste egenskaperna som studenterna måste få med i sina rapporter.

För varje opposition finns en rapport som opponeras på och en student som opponerar. Varje opposition en student gör dokumenteras i en text med separat rubrik och opponenter sätter också ett sammanfattande betyg för rapporten. Oppositionstexterna får ett nummer som är 1 för den första oppositionen studenten gör, 2 för nästa osv. Läraren kommenterar och betygssätter både rapporter och oppositioner och detta lagras i anslutning till respektive dokument. Både rapporter och opposition kan få betygen U/G/VG.

Denna databas beskrivs i nedanstående ER-diagram. Översätt den till relationsmodellen och rita upp relationsschemat. Markera nycklar i relationsmodellen med understrykning och främmande nycklar med pilar. (8p)



Attributet OppNr är ett löpnummer för varje students oppositioner. Det är alltså partiell nyckel.