

Linköpings Universitet
Institutionen för datavetenskap
Eva L. Ragnemalm

TENTAMEN

729G28 Webprogrammering och databaser

Datum: **2017-10-24**

Tid: **14-18**

Jourhavande lärare: **Eva Ragnemalm**

Tel: **070-1907391**

Besöker tentamenslokalen c:a kl 15

Hjälpmedel: **Inga**

Följ instruktionerna på tentaomslaget.
Gör bara en uppgift per papper.

Poängfördelning

Uppgift	Poäng
1	5
2	6
3	6
4	12
5	10
6	8

G: 25 p VG: 37 p Max: 47p

1. **Begrepp: Tips:** För varje fråga svara 1, X eller 2: (5p)

a. *Metadata* är:

1. Meteorologisk data om väder.
- x. Data som söks fram vid avancerade sökningar i flera databaser.
2. Data om data i databasen.

b. En *implementationsmodell* beskriver:

1. Vilka tabeller databasen ska innehålla.
- x. Vilka begrepp i den verkliga världen som ska finnas representerade i databasen.
2. I vilket format data ska lagras i databasen, t.ex. om det ska vara heltal eller strängar.

c. En *sambandstyp* i ett ER-diagram sammanbinder:

1. En entitetstyp med dess attribut.
- x. Två eller fler entitetstyper med varandra.
2. Två entitetstyper med varandra.

d. Ett *fullt funktionellt beroende* är:

1. ett specialfall av funktionellt beroende sådant att determinanten är minimal.
- x. ett funktionellt beroende där determinantens värde beror av ett annat attribut i relationen.
2. ett funktionellt beroende där man kan ta bort minst ett attribut ur determinanten och fortfarande ha ett funktionellt beroende.

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

Vara

ArtNr	Namn	Antal	Leverantor
1	Jackor med fler fickor än du kan hitta	20	1
2	Trebenta byxor	150	3
3	Bapelsiner	35	4
4	Prins Korv	50	4
5	Drottning Sylt	10	4
6	Mollijoxer	250	3
7	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 integer, Namn varchar (50), Antal integer,
Leverantor integer, Primary key (ArtNr));
```

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

```
Alter table Vara add constraint varuLeverantor Foreign key (Leveran-
tor) references Levarantor (IDnr);
```

- a. Skriv ett SQL-kommando som listar namnet på alla varor och för varje vara även namnet på leverantören. (3p)

- b. Skriv ett SQL-kommando som listar leverantörernas IDnr och för varje leverantör hur många olika varor de levererar (t.ex. nr 4 levererar 3 olika varor). (3p)

3. **Begrepp:** Nedan listas ett antal kommandon mot ovanstående (se uppgift 2) databas.

1. `Insert into Vara values (8, "Mackapärer", 23, 3);`
2. `Insert into Leverantor values (4, "Feletonbolaget", "Ringvägen 3", "Sverige");`
3. `Insert into Vara values (10, "Varm glass", 50, 5);`
4. `Drop table Vara;`
5. `Delete from Leverantor where Land="USA";`
6. `Delete from Vara where Leverantor=1;`
7. `Delete from Levarantor where Land="Danmark";`
8. `Create table Bestallning (OrderNr integer, vara integer, antal integer, kund integer, Primary key (OrderNr));`
9. `Alter table Bestallning add constraint BestallnVara Foreign Key (vara) references Vara(ArtNr);`

- a. Vilka av de ovanstående kommandona påverkar databasens *schema* och vilka påverkar dess *instans* (bortse från ev. problem med kommandot)? (2p)
- b. Vissa av kommandona ovan kommer inte att gå att genomföra på grund av integritetsproblem. Vilka? Förklara kort, för respektive kommando, varför det inte kan genomföras. (4p)

4. **Design, 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 passagerare 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, t.ex. ARN - Arlanda) en viss tid en viss dag i veckan, t.ex. Må 07:15. Flighter identifieras med en kod, t.ex. SK721. Varje flight flygs sedan samma dag och tid vecka efter vecka året om.

- Ett visst flygplan flyger en specifik flight en viss vecka ett visst år, t.ex. 44-17 för vecka 44 2017 (men ett annat flygplan kan flyga flighten en annan vecka). Med det flygplanet reser passagerare som får sig tilldelade en viss sittplats, t.ex. 29A, som normalt inte är samma passagerare en annan vecka. Vid en viss flygtur får planet som flyger flighten tilldelat en gate, t.ex. nr 31, där det plockar upp sina passagerare. Gaten kan också variera från vecka till vecka. 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 ett visst år. För varje passagerare på en flygtur vill man spara namn, biljettnummer, vilken flight det är, vilken vecka vilket år. Denna information lagras i databasen 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 (dvs en flygtur)

bestäms en tid innan avgång och avgörs av behovet av olika platser. Alla passagerare på den flighten den veckan reser naturligtvis med just det planet, som körs fram till en enda gate.

- En sittplats har ett nummer, en klassificering (1:a klass eller ekonomiklass) och hör till ett visst plan. Sittplatser kan naturligtvis vara tomma.

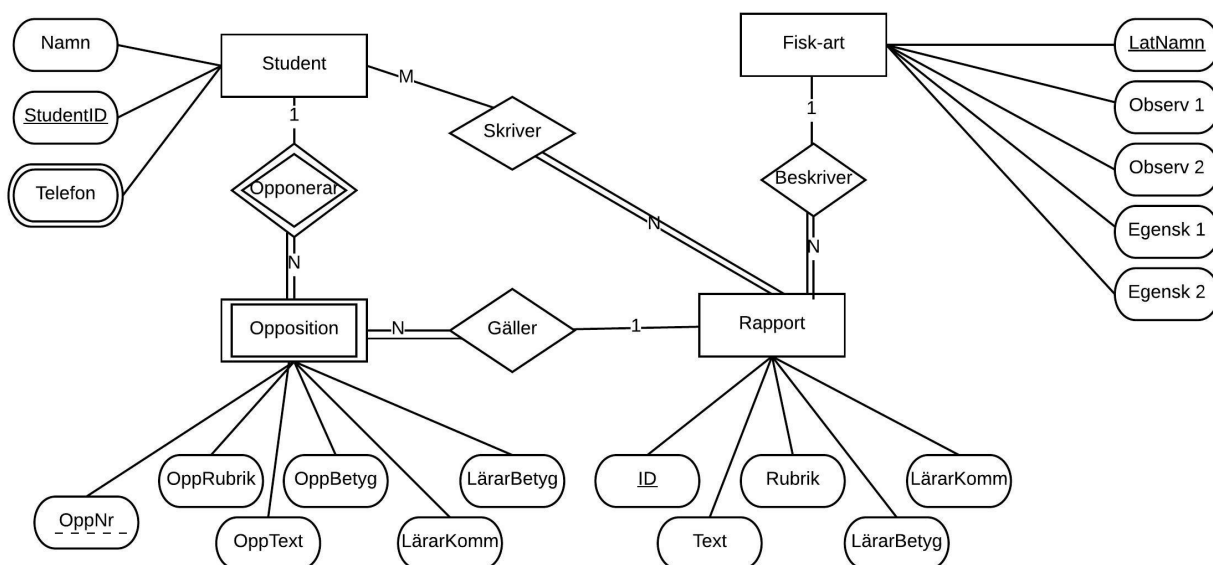
Designa en ER-modell för denna databas. Ange nycklar för entitetstyperna samt kardinalitet och deltagande för sambandstyperna. Om du behöver semantiska integritetsvillkor, anteckna dem vid sidan om diagrammet. För alla attribut, visa ett exempel på hur innehållet kan tänkas se ut. (12p)

5. **Normalisering, Design:** En klantig programmerare lade upp följande tabell för Linköpings flygplats, se uppgift 4 ovan (funkar inte riktigt enligt specifikation, programmeraren fick inte med allt - den var ju klantig):

Flyg (FlightNr, Destination, veckodag, tid, vecka-år, planNamn, planTyp, antalFörsta, antalEkonomi, 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 under relationen som på föreläsningen eller skriva dem på formen {determinant}->{de attribut som bestäms}). Var försiktig så att du inte hittar på beroenden som inte finns. (3p)
 - Vilka problem relativt specifikationen har denna tabell (vilka krav/behov är inte uppfyllda)? (2p)
 - Antag att alla attribut är atomära. Vilken normalform uppfyller tabellen Flyg? 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)
6. **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 betygsätter både rapporter och oppositioner och detta lagras i anslutning till respektive dokument. Både rapporter



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

och opposition kan få betygen U/G/VG.

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