

Linköpings Universitet
Institutionen för datavetenskap
Eva L. Ragnemalm

TENTAMEN

729G28 Webprogrammering och databaser

Datum: **2018-10-23**

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 | 8 |
| 2 | 6 |
| 3 | 4 |
| 4 | 4 |
| 5 | 14 |
| 6 | 8 |

G: 23 p VG: 35 p Max: 44p

1. **Begrepp**, För varje nedanstående påstående, är det sant eller falskt? Rätt svar ger en poäng. Felaktigt svar ger minus en poäng. Uppgiften som helhet kan dock inte ge mindre än 0p. (8p)
- Alla data i en databas hänger ihop med varandra, annars är det inte en databas.
 - En databas' *schema* beskriver precis när värdena i olika tabeller får ändras.
 - Ett Relationsschema är en *konceptuell datamodell*.
 - En *sambandstyp* kan koppla ihop hur många *entitetstyper* som helst.
 - Ett *referensattribut* är ett villkor som gäller för ett attribut, t.ex. att ett lagersaldo inte får ha ett negativt värde.
 - Ett *fullt funktionellt beroende* är ett *funktionellt beroende* där man inte kan ta bort något attribut ur *determinanten* och fortfarande ha ett funktionellt beroende.
 - I en Select-sats kan en vy (skapad med Create View...) användas på samma sätt som en tabell, men vyn tar ingen plats i databasen.
 - Att ett attribut i en relation är *atomärt* betyder att det består av ett huvudvärde och ett eller flera delvärden.

Kombinationsuppgift (gemensam information för uppgift 2-4): Du har en databas som skapats med nedanstående kommandon:

```
CREATE TABLE course
  (coursecode CHAR (6) CONSTRAINT pk_course PRIMARY KEY,
   name VARCHAR(40), points INTEGER, examiner INTEGER);
CREATE TABLE employee
  (empnr INTEGER CONSTRAINT pk_employee PRIMARY KEY,
   name VARCHAR2(30), institution VARCHAR(5));
ALTER TABLE course
  ADD CONSTRAINT fk_course_employee
  FOREIGN KEY (examiner) REFERENCES employee (empnr);
```

Den har innehåll enligt nedan:

Course:

| | | | |
|---------|-------------------------------------|---|-----|
| 732G16 | Databaser, design och programmering | 7 | 123 |
| TDDDB38 | Tekniska Databaser | 6 | 456 |
| 729G28 | Webprogrammering och databaser | 7 | 123 |

Employee:

| | | |
|-----|----------------|-----|
| 123 | Eva Ragnemalm | IDA |
| 456 | Lena Strömbäck | IDA |

2. **Begrepp/SQL**: För varje nedanstående SQL-kommando berätta om det går att genomföra eller inte (dvs resulterar i felmeddelande). För kommandon som resulterar i felmeddelanden, beskriv

orsaken och ge ett lämpligt kommando som kan användas först för att kommandot ska gå att genomföra så som det är skrivet. För varje kommando utgå från innehållet ovan. (6p)

- a. `DELETE FROM course
WHERE kursansvarig=123;`
- b. `DELETE FROM employee
WHERE empnr=123;`
- c. `INSERT INTO course
VALUES ('TDDB38', 'Databasteknik', 8, 999);`
- d. `INSERT INTO employee
VALUES (789, 'Johan Falkenjack', 'IDA');`

3. **SQL:** Givet tabellerna ovan (bortse från ev. felstavningar): (4p)

- a. Vad blir svaret från detta kommando? Rita hela resultat-tabellen.
`SELECT name AS Kursansvarig, institution
FROM employee
WHERE empnr IN (SELECT examiner
FROM course
WHERE coursecode='732G16');`

- b. Vad blir svaret från detta kommando? Rita hela resultat-tabellen:
`SELECT employee.name, COUNT(*) AS Antal
FROM employee JOIN course ON employee.empnr=course.examiner
GROUP BY employee.name;`

4. **Normalisering:** Betrakta tabellerna course och employee ovan som relationer: (4p)

- a. Rita ut de fulla funktionella beroenden som kan antas finnas i course och employee så som de definierats ovan (med pilar som vi gjort i kursen).
- b. Lägg till ett attribut i course som gör att relationen bryter mot tredje normalformen. Namnge det så att det går att avgöra vad det representerar, och rita ut det fulla funktionella beroende som du introducerar.

5. **Designuppgift:** Musikfirman AllaKanSpela hyr ut musikinstrument till barn som nyss börjat lära sig spela. Kunden är den betalande föräldern, inte barnet. Musikfirman vill ha hjälp att hålla ordning på instrumenten, kontrakten och kunderna och annan data.

Varje instrument har en unik identitetskod, en typ (t.ex. blockflöjt, fiol osv.) och en viss månads-hyra som bara uppdateras när man får in instrumentet mellan uthyrningar. Varje kontrakt gäller ett visst instrument och en viss kund och från ett startdatum till ett slutdatum (som inte kan vara samma datum). Man vill ha kvar information om gamla kontrakt (vem som hyrde vad under vilken period) även efter kontrakt-tidens slut. Vilken hyra som gällde för gamla kontrakt ska **inte** sparas. Man upprättar ett kontrakt för varje instrument som hyrs ut även om kunden hyr flera instrument samtidigt. Det händer ganska ofta att samma kund hyr samma instrument igen, då upprättas ett nytt kontrakt för den nya hyresperioden. Det gamla kontraktet ska finnas kvar. En kund kan ha flera kontrakt samtidigt (man kan ju ha flera barn som spelar musik).

Om kunderna behöver man veta namn, adress (bestående av gata, nummer, postnummer och postort) och telefonnummer (vissa har flera nummer), och man använder personnummer som identifikation.

AllaKanSpela har också lärare och lektionsgrupper, som barnen som spelar instrumenten deltar i (alla elever har instrument hyrda hos AllaKanSpela). Om eleverna behöver man veta deras namn, adress (behöver inte vara samma som betalande förälders), ålder, mobilnummer (om de har), vilket eller vilka instrument de spelar på (alltså tar lektioner i), vilken/vilka lektionsgrupp/er de ingår i samt vilken kund som betalar för dem. En elev kan ta lektioner i flera instrument och deltar då i en lektionsgrupp för respektive instrument. Musikfirman vill inte lagra barnens personnummer, men du kan anta att en person inte ger flera barn samma namn.

Om lektionsgrupperna behöver man veta vilken lärare som leder dem, vilken lokal de hålls i samt vilken tid lektionen hålls (lektionerna hålls en gång i veckan, på en viss veckodag, startar ett visst klockslag och slutar ett visst annat klockslag). Om lärarna behöver man veta deras namn, adress, telefonnummer samt vilken annan lärare som kan tänkas vara vikarie ifall denna är sjuk.

Rita upp ett ER-diagram för databasen. Markera kardinalitet och deltagande samt nycklar så som vi gjort i kursen. Om du tillför attribut som inte tydligt finns med i beskrivningen måste du motivera varför de behövs. Läs beskrivningen noga så att du inte missar något. Om du tycker det saknas information eller att du behöver göra ytterligare antaganden, skriv då ner dem. Använd inte EER. (14p)

6. Nedanstående ER-diagram representerar företaget Strömljus' data om spisarna de säljer och reparerar. Konvertera det till relationsschema. Rita ut primärnycklar och främmande nycklar som vi gjort i kursen. (8p)

