

Tentamen i EDAF75

19 mars 2019

Skrivtid: 8-13

- SKRIV BARA PÅ ENA SIDAN AV PAPPRET – tentorna kommer att scannas in, och endast framsidorna rättas.
- SKRIV INTE MED FÄRGPENNA – enda tillåtna färg är svart/blyerts.
- SKRIV TYDLIGT – om texten inte går att läsa kan du inte få några poäng.
- SÄTT IDENTITET OCH SIDNUMMER PÅ VARJE INLÄMNAT BLAD, kontrollera att sidnumret på din sista sida är samma som det antal blad du markerar på omslagspappret.
- Preliminär maxpoäng på uppgifterna: 13 + 13 + 13 + 4 + 3 + 4.

Uppgift 1

En institution vill ha en databas för att administrera övningar i olika kurser. En kurs har en kurskod och ett namn, och ger ett givet antal poäng. Till varje kurs hör ett antal övningar, de ser väsentligen likadana ut varje gång kursen ges – de har ett namn ("Övning 1"), en beskrivning (uppgiftstexten), och en längd (dvs för hur lång tid vi behöver boka övningssalarna).

En kurs kan ges flera gånger varje år, men kan påbörjas högst en gång per läsperiod. Varje kursomgång¹ hanteras av en kursansvarig lärare, och under kursomgången ges alla de övningar som hör till kursen – övningarna ges vid olika övningstillfällen. Varje övningstillfälle ges i ett antal salar, leds av en eller flera övningsledare (som kan gå mellan salarna), och behandlar någon av kursens övningar (en övning kan behandlas vid flera övningstillfällen, men varje övningstillfälle behandlar bara en övning). En övningsledare har en 'stil'-identitet och ett namn, och har en timlön som kan varierar mellan olika övningsledare, beroende på olika faktorer. En kursansvarig lärare har en 'lucast'-identitet, och ett namn. Övningssalarna har unika namn, rymmer ett givet antal studenter, och har ett givet timpris.

- Gör en E/R-modell som beskriver databasen, redovisa med ett UML-diagram, markera nycklar och svaga entitetstyper ('weak entity sets'), och skriv ut multipliciteter på samtliga associationer.
- Skapa relationer för din E/R-modell – markera tydligt primärnycklar och främmande nycklar.
- Skriv en SQL-sats som skriver ut tid och rumsnamn för alla rumsbokningar för kursen med namnet "Databasteknik" under "2019-vt1" (alltså första läsperioden på våren 2019). Sortera på tid, och därefter rumsnamn.

¹Du kan i uppgiften använda det engelska ordet *class* för att beskriva en kursomgång.

Uppgift 2

Ett telefonbolag har abonnenter (*users*) som har abonnemang (*subscriptions*). Alla utgående telefonsamtal (*calls*) registreras med uppgift om bland annat vilket nummer som ringer (*caller_no*) och vilket nummer som blir uppringt (*callee_no*) – vi förutsätter att alla samtal rings mellan registrerade telefonnummer.

Detta beskrivs av följande databasschema:

```
users(user_id, username, address)
subscriptions(phone_no, user_id)
calls(call_id, caller_no, callee_no, start_time, finish_time, cost)
```

Skriv SQL-satser i frågorna (b)-(g)

- Rita ett ER-diagram som beskriver databasen, rita diagrammet enligt anvisningarna i problem 1a.
- Skriv SQL-satser för att skapa tabellen *calls*.
- För de 10 senaste samtalen: skriv ut vilket nummer som ringde, och vilken kostnaden var för samtalet.
- Skriv ut namn och adress till kunden som äger abonnemanget med nummer 0707-123456.
- Skriv ut namn och adress (sorterat efter namn) till de kunder som har mer än ett abonnemang.
- Skriv ut namn, adress och total samtalskostnad för de kunder som ringt för sammanlagt mer än 1000 kr.
- Skriv ut telefonnummer för de abonnemang som aldrig använts (aldrig ringts från eller till).

Uppgift 3

I relationen $R(A, B, C, D, E, F)$ gäller följande funktionella beroenden:

$$FD_1: AB \rightarrow C$$

$$FD_2: A \rightarrow D$$

$$FD_3: D \rightarrow AE$$

$$FD_4: E \rightarrow F$$

- Bestäm relationens nycklar, motivera ditt svar.
- Visa att relationen inte är i BCNF.
- Avgör om den är i 3NF eller ej – motivera ditt svar.
- Dela upp relationen i mindre relationer som är i BCNF, och som ger samma relation när vi joinar ihop dem igen (dvs vi tappar inga rader, och får inga nya rader). Motivera ditt svar genom att i varje steg visa vilka nycklar och funktionella beroenden som finns (även för två-attributsrelationer).

Uppgift 4

- (a) Vilka är de två huvudsakliga problem man löser med transaktioner. Relatera problemen till akronymen ACID.
- (b) Förklara kortfattat vad följande isolationsnivåer innebär, och rangordna dem (säkrast först):
- READ COMMITTED
 - READ UNCOMMITTED
 - REPEATABLE READ
 - SERIALIZABLE

Uppgift 5

Visa att en relation med bara två attribut alltid är i BCNF.

Uppgift 6

I denna uppgift kommer vi att använda tabeller som alla har attributet a, som är definierat som:

```
CREATE TABLE ... (  
  a      INT,  
  ...  
);
```

- (a) Det finns visserligen en standard för SQL, men nästan alla databas-system avviker på olika sätt – exempelvis finns det några system som saknar funktionen `intersect`, dvs funktionen som ger snittet mellan två mängder. I SQLite3, PostgreSQL och MySQL kan vi skriva:

```
SELECT a  
FROM r1  
INTERSECT  
SELECT a  
FROM r2
```

för att få de a-värden som förekommer i både r1 och r2. Skriv SQL-kod som ger samma resultat utan att använda `INTERSECT` eller andra mängdfunktioner.

- (b) Skriv en SQL-query som för tabeller r1 och r2 motsvarande dem i uppgift 6 (a) ger alla a-värden som förekommer i en av tabellerna, men inte båda (*exklusivt eller*).

I denna uppgift får du gärna använda mängdfunktionerna `UNION` (unionen, dvs raderna från båda tabellerna, utan dubletter), `INTERSECT` (snittet, dvs de rader som finns i båda tabellerna, se ovan), och `EXCEPT` (differensen, dvs de rader som finns i den första tabellen men inte i den andra), men du får även lösa det på annat sätt om du vill.