

Tentamen i EDAF75

13 mars 2018

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 + 3 + 4 + 4.

Uppgift 1

Vi vill implementera en databas för en Amazon-liknande plattform som på en web-sida låter olika företag sälja produkter.

Varje produkt har en unik kod och en beskrivning, och den kan säljas av flera företag.

Företagen har namn och adresser, vi antar att namnen är unika. Ett företag (en 'säljare') anger för varje produkt de erbjuder till försäljning vilket pris de vill ha – priset på en produkt kan ändras över tiden, men får inte ändras under en dag (dvs prisändringar sker alltid vid midnatt). Erbjudanden lagras i databasen, med information om pris och giltighetstid (start- och slutdatum) – vi vill i databasen hålla koll på alla historiska priser, så gamla erbjudanden får inte tas bort. Ett företag kan sluta sälja en produkt genom att inte längre ha ett aktuellt erbjudande för produkten (enligt giltighetstiden).

En kund har ett unikt personnummer, ett namn och en adress. Kunder kan göra beställningar på web-sidan – varje beställning får en unik kod, och kan innehålla flera produkter, från flera företag. Exempelvis kan man i en order beställa två paket tennisbollar från "Dennis Tennis" och en rökdykardräkt från "Acme Explosives".

Vi vill spara samtliga beställningar från samtliga kunder i vår databas, så att vi i efterhand kan skriva ut när en beställning gjordes, vilka varor som köptes, hur många av varje vara som köptes, och hur mycket de kostade.

Varje kund kan dessutom ha ett antal önskelistor, varje önskelista har ett namn (exempelvis "julklappsönsknningar" och "saker att köpa när jag har onödigt mycket pengar"), önskelistorna innehåller en eller flera produkter, och kan delas med andra användare (dvs flera andra användare kan ges tillstånd att se dem).

Vi antar i uppgiften att samtliga produkter som säljs finns i tillräckligt antal.

- 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 primärnycklar och främmande nycklar.

- (c) Skriv en SQL-sats som ger pris och beskrivning för samtliga produkter som "Acme Explosives" erbjuder idag ('2018-03-13').

Uppgift 2

En databas över tågavgångar, passagerare och tågbiljetter ska utvecklas. Till varje tågavgång hör ett tågnummer, datum, avreseort, ankomstort, avgångstid och ankomsttid. En tågavgång identifieras unikt av tågnumret tillsammans med datumet. För enkelhetens skull stödjer databasen inte mellanstopp, utan varje tåg går enbart mellan två platser. Passagerare bor på en adress i en stad och identifieras unikt av deras namn. Passagerare köper biljetter och varje passagerare kan enbart köpa en biljett per tågavgång. Varje biljett ger medlemspoäng som kan användas till medlemsförmåner.

Databasen består av följande relationer (där primärnycklar är understrukna, och främmande nycklar är *kursiva*).

```
train_trips(trip_nbr, trip_date, origin, destination, departure_time, arrival_time)
tickets(trip_nbr, trip_date, passenger_name, points)
passengers(passenger_name, city, address)
```

Skriv SQL-satser i frågorna (b)-(g)

- Rita ett ER-diagram som beskriver databasen.
- Skapa tabellen tickets.
- Hämta numret på alla tågavgångar idag (2018-03-13), sortera efter avgångstid.
- Hämta avgångstid, avreseort och namn för alla passagerare som åker den 13 mars 2018.
- Ta fram namnet på alla passagerare som bor i Lund och som aldrig har köpt någon biljett.
- Hämta antalet tågavgångar per dag, sortera efter datum.
- Hämta namn och totala medlemspoäng för samtliga passagerare (de som aldrig har köpt någon biljett skall få 0 poäng) – sortera i första hand efter poäng (flest poäng först), i andra hand efter namn.

Uppgift 3

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

$$FD_1: AB \rightarrow C$$

$$FD_2: A \rightarrow D$$

$$FD_3: D \rightarrow AE$$

$$FD_4: E \rightarrow F$$

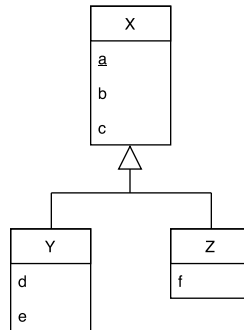
- Bestäm relationens nycklar, motivera ditt svar.
- Visa att relationen inte är i BCNF, och avgör om den är i 3NF.
- Dela upp relationen i mindre relationer som är i BCNF.

Uppgift 4

Vad är en trigger i en databas? På vilka händelser kan en trigger reagera? Ge exempel då det kan vara bra att använda en trigger.

Uppgift 5

Översätt nedanstående ER-diagram till relationer. Entitetstyperna Y och Z ärver från entitetstypen X (entitetstypen X är inte abstrakt, vilket medför att en entitet antingen är ett X, Y eller Z). I kursen har vi sett tre olika tekniker för att översätta arv till relationer. Gör en översättning för *varje* sådan teknik.



Uppgift 6

Vi har ett antal mängder av heltal, till exempel $\{1, 3, 5\}$, $\{3, 4, 5\}$ och $\{5, 6\}$. Vi kan representera detta med en relation

`sets(set_id, element)`

där `set_id` är numret på en mängd, och `element` är ett element. Mängderna i exemplet ovan representeras av följande instans av relationen:

set_id	element
1	1
1	3
1	5
2	3
2	4
2	5
3	5
3	6

Det är enkelt att skriva en SQL-sats som ger de element som förekommer i någon av mängderna:

```
SELECT DISTINCT element
FROM sets
```

Det är inte lika enkelt att skriva en SQL-sats som ger de element som förekommer i *alla* mängderna (i exemplet ovan är det bara värdet 5 som förekommer i alla mängderna). Skriv SQL-kod för att lösa uppgiften, du får använda vyer, om du vill. Uppgiften kan lösas på många sätt; enklast är nog att utnyttja GROUP BY.