



Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	2011-06-09
Sal (1) <small>Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och ringa in vilken sal som avses</small>	TER2
Tid	8-12
Kurskod	732G16
Provkod	TEN1
Kursnamn/benämning Provnamn/benämning	Databaser: design och programmering Tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	9
Jour/Kursansvarig <small>Ange vem som besöker salen</small>	Eva Ragnemalm
Telefon under skrivtiden	073-0332469
Besöker salen ca kl.	ca kl 10
Kursadministratör/kontaktperson <small>(namn + tfnr + mailaddress)</small>	Carita Lilja
Tillåtna hjälpmedel	Inga
Övrigt	
Vilken typ av papper ska användas, rutigt eller linjerat	Rutigt
Antal exemplar i påsen	

1(5)

Tekniska Högskolan i Linköping
Institutionen för datavetenskap
Eva L. Ragnemalm

TENTAMEN

732G16 Databaser

Design och programmering

Datum: 2011-06-09 kl 08-12

Jourhavande lärare: **Eva Ragnemalm**

Tel: **073-0332469**

Besöker tentamenslokalen c:a kl 10

Hjälpmedel: **Inga**

Följ instruktionerna på tentaomslaget.

Poängfördelning

Uppgift	Poäng
1	5
2	4
3	4
4	6
5	5
6	6
7	6
8	4
9	5

G: 23p VG: 35p Max: 45p

2(5)

1. **Begrepp:** För varje påstående nedan, ange om det är sant eller falskt (Felaktigt svar ger -1p men uppgiften kan inte ge negativt resultat): (5p)

- Skillnaden mellan **data** och **metadata** är att data lagras i databasen, och metadata gör inte det.
- Skillnaden mellan **databas** och **databashanterare** är att databasen är bara datat och databashanteraren är programsystemet som används för att manipulera datat.
- Skillnaden mellan **schema** och **instans** är att schemat beskriver hur datat i databasen ska struktureras (organiseras) och instansen beskriver databasens specifika innehåll vid en viss tidpunkt.
- Skillnaden mellan **funktionellt beroende** och **fullt funktionellt beroende** är att fullt funktionellt beroende är ett specialfall av funktionellt beroende sådant att determinanten är minimal.
- Skillnaden mellan **logiskt databeroende** och **fysiskt databeroende** är att vid logiskt databeroende är data antingen sant eller falskt och vid fysiskt databeroende kan det anta vilket numeriskt värde som helst.

2. **Begrepp:** Du har en databas som skapats med nedanstående kommandon:

```
CREATE TABLE course
  (coursecode CHAR (6) CONSTRAINT pk_course PRIMARY KEY,
  name VARCHAR2(20), points INTEGER, examiner INTEGER);
CREATE TABLE employee
  (empnr INTEGER CONSTRAINT pk_employee PRIMARY KEY,
  name VARCHAR2(30), institution VARCHAR(15));
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
TDDB38	Tekniska Databaser	6	456
729G28	Webprogrammering och databaser	7	123

Employee:

123	Eva Ragnemalm	IDA
456	Lena Strömbäck	IDA

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. (4p)

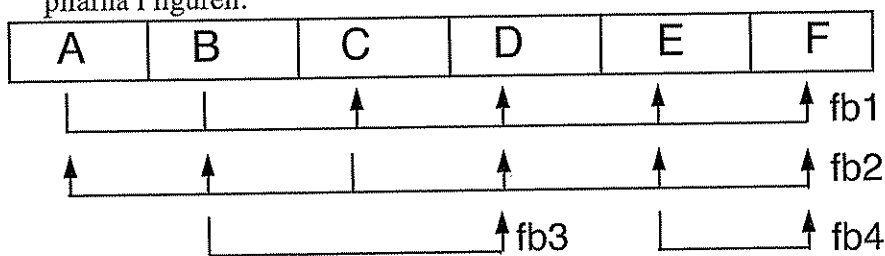
3(5)

- a) DELETE FROM employee WHERE empnr=123;
- b) INSERT INTO course VALUES ('TDDB38', 'Databasteknik', 8, 999);
- c) DELETE FROM kurs WHERE kursansvarig=123;
- d) INSERT INTO examiner VALUES (789, 'Nils Dahlbäck', 'IDA');

3. **Relationsalgebra:** Givet tabellerna Employee och course och det innehåll som anges i ovanstående uppgift, utför följande operationer: (4p)

- a) $\pi_{\text{(coursecode)}}(\text{course})$ (tips: π =pi=projektion)
- b) $\sigma_{\text{(empnr=123)}}(\text{employee})$ (tips: σ =sigma=urval)
- c) $\text{course} * \text{employee}$

4. **Normalisering:** Relationen som illustreras i figuren nedan har de atomära attributen A-F (det är namnen på attributen) och de funktionella beroenden fb1-4 som symboliseras av pilarna i figuren. 6p



- a) Är {A, D, E} en supernyckel till relationen? Varför/inte?
- b) Är {C, D} en kandidatnyckel till relationen? Varför/inte?
- c) Är E ett primattribut? Varför/inte?

5. **Fysiska databasen:** Du har en tabell över alla studenter på LiU. Den innehåller fälten: Personnummer, Efternamn, Förnamn, student-id, Adress, Postadress, telefonnr1, telefonnr2. Posterna tar 500 byte, filen har 20000 poster och är sorterad på personnummer. Hårddisken som används har blockstorlek 2048 bytes och accesstid 10 ms. Besvara nedanstående frågor (redovisa beräkningarna tydligt så finns möjlighet att få poäng även om slutsvaret inte är helt korrekt): 5p

- a) Hur lång tid tar det (maximalt) att söka ut en viss person givet personnumret?
- b) Hur lång tid tar det (i genomsnitt) att söka ut en viss person på namnet, om vi antar att det inte finns någon fler med samma namn?

Tips: $\log_2(500)=8.97$ $\log_2(2000)=10.97$ $\log_2(2048)=11$ $\log_2(5000)=12.29$ $\log_2(20000)=14.29$

6. **Transaktioner:** Din databas-server hängde sig och du blev tvungen att starta om den. Lyckligtvis behövde du inte läsa tillbaka backup-en, för inget blev skadat. Loggfilen, som byggts upp sedan senaste backup-en, ser ut som nedan. För respektive markerat komman-

4(5)

do i filen, tala om ifall det ska **repeteras** (dvs nya värdet skrivs i databasen) eller **rivas upp** (dvs gamla värdet skrivs i databasen), eller kan **ignoreras**. All information du behöver finns i loggfilen. tnnn är ett transaktions-id, konto nnn ett databasfält. 6p

```
--filen börjar här--
  start (t123)
  read (t123, konto 567)
(a) write (t123, konto 567, 150, 350)
  read (t123, konto 568)
  write (t123, konto 568, 12 300, 12 100)
  commit (t123)
  checkpoint
  start (t124)
  read (t124, konto 345)
  start (t125)
(b) read (t125, konto 789)
(c) write (t124, konto 345, 3450, 2450)
  read (t124, konto 346)
  write (t124, konto 346, 8 330, 9 330)
  commit (t124)
(d) write (t125, konto 789, 2000, 1900)
--filen slutar här--
```

Designuppgift: USA kräver numera att varje resenär som kommer dit har ett inresetillstånd. För att ansöka om ett sådant krävs att man gör en ansökan på nätet och betalar 10 dollar, via kreditkort. Den som betalar behöver inte vara samma person som den som åker. För alla personer som söker inresetillstånd lagras passnummer, namn, adress, ansökningsdatum, tidpunkt för resans start, anledning till resan (fritext), adress i USA, svaren (ja/nej) på två frågor: om man bär på smittsamma sjukdomar eller tänker ägna sig åt brottslig verksamhet i USA, samt det slutliga beslutet (ja, nej eller ej avgjort). Man lagrar också information om vilket kreditkort som betalt ansökan och vem som äger kreditkortet (namn). Varje ansökan tilldelas ett unikt id-nummer. Nu vill man även lagra information om personer som betalar åt någon annan. För sådana vill man lagra kreditkortsnummer (är unikt), innehavarens namn och datum för gjorda ansökningar, samt passnummer för personer man betalt inresetillstånd för. Man vill kunna göra nedanstående utsökningar:

- a) Hitta alla personer (lista namn och adress och passnummer) som har betalat med kreditkortsnummer 123456778901234. Varje person ska bara finnas med en gång i listan även om det finns flera ansökningar för den personen (strunta i att man byter pass ibland).
- b) Ange namn och kreditkortsnummer, samt antal ansökningar de betalt för, för alla som betalat för mer än 10 personer.

7. Rita upp ett ER-diagram för databasen. Glöm inte markera kardinalitet och deltagande samt nycklar. Om du tycker det saknas information eller att du behöver göra antaganden, skriv då ner dem. Om du inte kan lagra information som uttryckligen efterfrågas får du poängavdrag, men du får heller inte ha redundans i systemet. (6p)

5(5)

8. Konvertera ER-diagrammet till relationsmodellen (tabeller). Glöm inte att markera nycklar och främmande nycklar. (4p)
9. Formulera sedan fråga a och b ovan i SQL. (5p)

Lycka till