

Databases (INFODB)

28 juni 2005

Vooraf

- Bij elke vraag wordt verwacht dat je laat zien hoe je aan het antwoord komt (tenzij anders wordt vermeld).
- Je mag ten hoogste één A4 met aantekeningen raadplegen.
- Gebruikte afko's:
 - 2PL: two-phase locking
 - 3NF: derde normaalvorm
 - BCNF: Boyce-Codd normaalvorm
 - DP: dependancy preserving
 - FD: functional dependancy
 - RA: relationele algebra
 - SQL = SQL (Structural Query Language)

Opgave 1. Algemeen

(20 punten)

Bij de onderstaande vragen zijn er meerdere antwoorden correct. Je dient alle correcte antwoorden aan te geven. Een toelichting is niet noodzakelijk.

Vraag 1

Welke van de onderstaande uitspraken zijn waar?

- A: Elke verliesvrije decompositie is DP.
- B: Elk relatieschema in 4NF is automatisch ook in BCNF.
- C: Een relatieschema met drie attributen is altijd in 3NF maar niet altijd in BCNF.
- D: Een attribuut kan niet tegelijkertijd foreign key en primary key zijn.
- E: Een primary key is altijd een candidate key en een candidate key is altijd een superkey.

Vraag 2

Welke van de onderstaande expressies is equivalent aan $S - (S - R)$? (We gaan ervan uit dat alle tabellen vrij zijn van duplicaten.)

- A: $R - S$
- B: $R \cap S$
- C: $S - R$
- D: $R \cup S$
- E: $R \div S$

Vraag 3

Welke van de onderstaande afleidingsregels voor FD's zijn geldig?

A: $X \rightarrow Y, XY \rightarrow ZU \Rightarrow X \rightarrow Z$

B: $XY \rightarrow Z, U \rightarrow Y \Rightarrow XU \rightarrow Z$

C: $XY \rightarrow Z, Y \rightarrow U \Rightarrow XU \rightarrow Z$

Vraag 4

Stel we hebben een I/O-mechanisme volgens de principes *deferred update* en *no flush forced*. Welke van de onderstaande uitspraken zijn waar?

A: BEFORE-images zijn van belang bij een recovery.

B: Het schrijven van een update naar het externe geheugen vindt mogelijkterwijs plaats na de commit.

C: Het regelmatig draaien van een checkpoint is zinvol.

D: De disk I/O-manager heeft maximale vrijheid bij het bepalen wanneer een update naar het externe geheugen wordt geschreven

Opgave 2

(25 punten)

Een zweefvliegclub heeft een database met daarin een tabel waarin gegevens van de vliegers worden bijgehouden, een tabel waarin gegevens van vliegtuigen worden bijgehouden en een tabel waarin gegevens van vluchten worden bijgehouden. Elke vlieger heeft een uniek brevetnummer. Daarnaast wordt van elke vlieger de naam, het geslacht en de geboortedatum bijgehouden. Vak elk vliegtuig wordt het merk, het type en het bouwjaar geregistreerd. Eveneens heeft elk vliegtuig een uniek callsign (van de gedaante PH-123). De registratie van een vlucht gaat vergezeld van de datum, de starttijd, de duur (in minuten) en het starttype (sleepstart of lierstart). Het databaseschema is als volgt:

Vlieger (*brevetnr*, naam, geslacht, geboortedatum)

Vlucht (*brevetnr*, *callsign*, *datum*, *starttijd*, *duur*, *starttype*)

Vliegtuig (*callsign*, merk, type, bouwjaar)

We hebben de volgende queries:

Q1: Geef de namen van de vliegers die tenminste één vlucht gemaakt hebben.

Q2: Geef de namen en de totale vluchtduur voor de vliegers die het grootste aantal vluchten gemaakt hebben.

Q3: Geef de namen van de vliegers die in alle vliegtuigen gevlogen hebben.

Q4: Geef de namen van de vliegers die geen enkele vlucht gemaakt hebben.

Hieronder volgen expressies in de RA of in SQL. Geef aan welke queries corresponderen met welke expressie. De relatie tussen queries en expressies is many-to-many en optioneel.

E1: $\pi_{callsign}((\pi_{brevetnr, callsign}(Vlucht) \div \pi_{brevetnr}(Vlieger)) \bowtie Vliegtuig)$

E2: $\pi_{naam}((\pi_{brevetnr}(Vlieger) - \pi_{brevetnr}(Vlucht)) \bowtie Vlieger)$

E3: $\pi_{naam}((Vlieger \bowtie Vlucht) \div \pi_{callsign}(Vliegtuig))$

E4: $\pi_{naam}((\pi_{brevetnr}(Vlieger) \cap \pi_{brevetnr}(Vlucht)) \bowtie Vlieger)$

E5: $\pi_{naam}((\pi_{brevetnr, callsign}(Vlucht) \div \pi_{callsign}(Vliegtuig)) \bowtie Vlieger)$

E6: $\pi_{naam}((\pi_{brevetnr}(Vlieger) \cup \pi_{brevetnr}(Vlucht)) \bowtie Vlieger)$

E7: SELECT Vlieger.naam
 FROM Vlieger
 WHERE Vlieger.brevetnr IN
 (SELECT Vlucht.brevetnr
 FROM Vlucht)

E8: SELECT Vlieger.naam, SUM(Vlucht.duur)
 FROM Vlieger, Vlucht
 WHERE Vlieger.brevetnr = Vlucht.brevetnr
 GROUP BY Vlucht.brevetnr
 HAVING COUNT(*) > ALL
 (SELECT COUNT(*)
 FROM Vlucht
 GROUP BY Vlucht.brevetnr)

E9: SELECT Vlieger.naam
 FROM Vlieger
 WHERE NOT EXISTS
 (SELECT Vlucht.brevetnr
 FROM Vlucht)

E10: SELECT Vlieger.naam, SUM(Vlucht.duur)
 FROM Vlieger, Vlucht
 WHERE Vlieger.brevetnr = Vlucht.brevetnr
 GROUP BY Vlucht.brevetnr
 HAVING MAX(COUNT(*))

E11: SELECT Vlieger.naam
 FROM Vlieger
 WHERE Vlieger.brevetnr NOT IN
 (SELECT Vlucht.brevetnr
 FROM Vlucht
 WHERE Vlucht.brevetnr <> Vlieger.brevetnr)

E12: SELECT Vlieger.naam, SUM(Vlucht.duur)
 FROM Vlieger, Vlucht
 WHERE Vlieger.brevetnr = Vlucht.brevetnr
 GROUP BY Vlucht.brevetnr
 HAVING COUNT(*) > ALL
 (SELECT COUNT(*)
 FROM Vlucht
 GROUP BY Vlucht.brevetnr)

E13: SELECT Vlieger.naam
 FROM Vlieger
 WHERE NOT EXISTS
 (SELECT Vlucht.brevetnr
 FROM Vlucht
 WHERE Vlucht.brevetnr = Vlieger.brevetnr)

```
E14: SELECT Vlieger.naam
      FROM Vlieger
      WHERE EXISTS
        (SELECT Vlucht.brevetnr
         FROM Vlucht
         WHERE Vlucht.brevetnr = Vlieger.brevetnr)
```

```
E15: SELECT Vlieger.naam
      FROM Vlieger
      WHERE NOT EXISTS
        (SELECT Vliegtuig.callsign
         FROM Vliegtuig
         WHERE NOT EXISTS
           (SELECT Vlucht.brevetnr
            FROM Vlucht
            WHERE Vlucht.brevetnr = Vlieger.brevetnr))
```

```
E16: SELECT Vlieger.naam
      FROM Vlieger
      WHERE NOT EXISTS
        (SELECT Vliegtuig.callsign
         FROM Vliegtuig
         WHERE NOT EXISTS
           (SELECT Vlucht.brevetnr
            FROM Vlucht
            WHERE Vlucht.brevetnr = Vlieger.brevetnr
              AND Vlucht.callsign = Vliegtuig.callsign))
```

Opgave 3. Normaalvormen

(30 punten)

- (i) Stel we hebben een relatieschema $R(ABCDEFGH)$ en een verzameling FD's

$$F_R = \{G \rightarrow D, C \rightarrow AD, B \rightarrow CF, BD \rightarrow G, B \rightarrow AE, CF \rightarrow E\}.$$

- a) Geef alle candidate keys voor R . Geef hierbij een korte toelichting.
 - b) Geef een verliesvrije DP 3NF-decompositie voor R . Laat zien welke methode je gebruikt.
- (ii) Stel we hebben een relatieschema $R(ABCDEFG)$ en een verzameling FD's

$$F_R = \{A \rightarrow BC, F \rightarrow EG, D \rightarrow E, B \rightarrow C\}.$$

- a) Geef een verliesvrije BCNF-decompositie van R . Laat zien welke methode je gebruikt.
- b) Beargumenteer dat met behulp van het BCNF-algoritme geen DP BCNF-decompositie van R gevonden kan worden.

Opgave 4. Concurrency

(25 punten)

We beschouwen de volgende twee schedules (z.o.z.).

- (i) Geef aan of deze schedules serialiseerbaar zijn of niet. Licht toe. Geef zo mogelijk de equivalenten seriële schedules.
- (ii) Geef eveneens aan of de schedules getolereerd worden door een 2PL-scheduler. Geef hierbij een korte toelichting.

S1				
T1	T2	T3	T4	T5
			r(z)	
w(z)				
	w(x)			
		r(z)		
r(x)				
	r(y)			
				r(x)
r(z)				
			w(y)	
				w(y)

S2				
T1	T2	T3	T4	T5
			w(z)	
r(z)				
	r(x)			
		w(z)		
w(x)				
	w(y)			
				w(x)
w(z)				
			r(y)	
				r(y)