

Databases (INFODB)

24 januari 2007

- Schrijf en formuleer duidelijk.
- Bij elke vraag wordt verwacht dat je laat zien hoe je aan het antwoord komt (tenzij anders wordt vermeld).
- Je mag uitsluitend originele indexcards met aantekeningen raadplegen.
- Gebruikte afko's:
 - 2PL = two-phase locking
 - BCNF = Boyce-Codd normaalvorm
 - DP = dependency preserving
 - FD = functional dependency
 - RA = relationele algebra
 - SQL = SQL (Structured Query Language)

Opgave 1. Algemeen

(20 punten)

Bij de onderstaande vragen zijn er 0 of 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 : Binnen een decompositie komt elk attribuut in maximaal één relatie voor.
- B : Bij een verliesvolle (“lossy”) decompositie verlies je tuples.
- C : Een functional dependency is een speciaal geval van een key.
- D : Een decompositie is alleen lossless als de decompositie de DP eigenschap heeft.
- E : De attributen van een candidate key vormen altijd een strikte deelverzameling van een superkey.

Vraag 2: Welke van de onderstaande uitspraken zijn waar?

- A : BCNF is de ultieme normaalvorm in termen van het uitbannen van redundantie.
- B : Een relatieschema met twee attributen is altijd in 3NF.
- C : Een relatieschema met twee attributen is altijd in BCNF.
- D : Een relatieschema met twee attributen is altijd in 4NF.
- E : Bij het 3NF algoritme maak je eerst de closure van je FD verzameling.

Vraag 3: Welke van de onderstaande afleidingsregels voor FD's zijn geldig?

- A : $X \rightarrow Z, X \rightarrow Y \Rightarrow Y \rightarrow Z$.
- B : $X \rightarrow Z \Rightarrow X \rightarrow Y, Y \rightarrow Z$.
- C : $XY \rightarrow Z, Y \rightarrow U \Rightarrow XU \rightarrow Z$.

D : $X \rightarrow Y, XY \rightarrow ZU \Rightarrow X \rightarrow Z$.

E : $XY \rightarrow Z, U \rightarrow Y \Rightarrow XU \rightarrow Z$.

Vraag 4: Stel we hebben een I/O-mechanisme volgens de principes *immediate update* en *flush forced*. Welke van de onderstaande uitspraken zijn waar?

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

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

C : Het regelmatig draaien van een checkpoint is niet zinvol.

D : Bij een recovery kan een UNDO van sommige transacties noodzakelijk zijn.

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

Opgave 2: Queries

(20 punten)

Een bedrijf dat gegevens bijhoudt over de woningmarkt heeft een database met daarin een tabel waarin gegevens over woningen worden bijgehouden en een tabel waarin de verkoop- transacties worden bijgehouden. Elke woning heeft een uniek objectnummer waaronder het bij het bedrijf bekend staat, een adres (inclusief huisnummer, toevoeging huisnummer en plaats), een type (vrijstaande woning, hoekwoning, tussenwoning, appartement), een woonoppervlakte, een grondoppervlakte, een bouwjaar en een beschrijving. Verkooptrans- acties hebben een verkoopprijs, een verkoopdatum en een referentie naar een woning op basis van het objectnummer. We gaan er in het onderstaande vanuit dat geen enkele tabel leeg is. Het databaseschema van de database is als volgt (primary keys zijn in cursief):

- Woning (*objectnr*, adres, type, woonopp, grondopp, bouwjaar, beschrijving)
- Verkoop (*objectnr*, datum, verkoopprijs)

Gegeven zijn de volgende queries:

Q1 : Geef de adressen van de woningen die ten minste één keer verkocht zijn.

Q2 : Geef het bouwjaar van de oudste woning die op 23-01-2007 verkocht is.

Q3 : Geef de adressen van de woningen die nooit verkocht zijn.

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

E1 : $\pi_{\text{adres}}((\pi_{\text{objectnr}}(\text{Woning}) \cap \pi_{\text{objectnr}}(\text{Verkoop})) \bowtie \text{Woning})$

E2 : $\pi_{\text{adres}}(\text{Woning} \bowtie \text{Verkoop})$

E3 : $\pi_{\text{adres}}((\pi_{\text{objectnr}}(\text{Verkoop}) - \pi_{\text{objectnr}}(\text{Woning})) \bowtie \text{Woning})$

E4 : $\pi_{\text{adres}}((\pi_{\text{objectnr}}(\text{Woning}) \cup \pi_{\text{objectnr}}(\text{Verkoop})) \bowtie \text{Woning})$

E5 : $\pi_{\text{adres}}((\pi_{\text{objectnr}}(\text{Woning}) - \pi_{\text{objectnr}}(\text{Verkoop})) \bowtie \text{Woning})$

E6 :

```
SELECT Woning.bouwjaar
FROM Verkoop, Woning
WHERE Woning.bouwjaar <= MIN(Woning.bouwjaar)
AND Verkoop.objectnr = Woning.objectnr
AND Verkoop.datum = '23-01-2007'
GROUP BY Woning.bouwjaar
```

E7 :

```
SELECT Woning.bouwjaar
FROM Verkoop, Woning
WHERE Verkoop.datum = '23-01-2007'
AND Verkoop.objectnr = Woning.objectnr
GROUP BY Woning.bouwjaar
HAVING Woning.bouwjaar < ALL (
    SELECT Woning.bouwjaar
    FROM Verkoop, Woning
    WHERE Verkoop.datum = '23-01-2007'
    AND Verkoop.objectnr = Woning.objectnr
    GROUP BY Woning.bouwjaar)
```

E8 :

```
SELECT Woning.bouwjaar
FROM Verkoop, Woning
WHERE Woning.bouwjaar <= MIN(Woning.bouwjaar)
AND Verkoop.objectnr = Woning.objectnr
AND Verkoop.datum = '23-01-2007'
```

E9 :

```
SELECT Woning.adres
FROM Woning
WHERE Woning.objectnr IN (
    SELECT Verkoop.objectnr
    FROM Verkoop)
```

E10 :

```
SELECT Woning.bouwjaar
FROM Verkoop, Woning
WHERE Verkoop.datum = '23-01-2007'
AND Verkoop.objectnr = Woning.objectnr
GROUP BY Woning.bouwjaar
HAVING Woning.bouwjaar =< ALL (
    SELECT Woning.bouwjaar
    FROM Verkoop, Woning
    AND Verkoop.objectnr = Woning.objectnr
    GROUP BY Woning.bouwjaar)
```

```
E11: SELECT MIN(Woning.bouwjaar)
FROM Verkoop, Woning
WHERE Verkoop.datum = '23-01-2007'
AND Verkoop.objectnr = Woning.objectnr
GROUP BY Woning.bouwjaar
```

```
E12: SELECT MIN(Woning.bouwjaar)
FROM Verkoop, Woning
WHERE Verkoop.datum = '23-01-2007'
AND Verkoop.objectnr = Woning.objectnr
```

```
E13: SELECT Woning.adres
FROM Woning
WHERE NOT EXISTS
    (SELECT Verkoop.objectnr
    FROM Verkoop)
```

E14 :

```
SELECT Woning.bouwjaar
FROM Verkoop, Woning
WHERE Verkoop.datum = '23-01-2007'
AND Verkoop.objectnr = Woning.objectnr
GROUP BY Woning.bouwjaar
HAVING Woning.bouwjaar <= ALL (
    SELECT Woning.bouwjaar
    FROM Verkoop, Woning
    WHERE Verkoop.datum = '23-01-2007'
    AND Verkoop.objectnr = Woning.objectnr
    GROUP BY Woning.bouwjaar)
```

E15 :

```
SELECT Woning.adres
FROM Woning
WHERE NOT EXISTS (
    SELECT Verkoop.objectnr
    FROM Verkoop
    WHERE Verkoop.objectnr = Woning.objectnr)
```

E16 :

```
SELECT Woning.adres
FROM Woning
WHERE EXISTS (
    SELECT Verkoop.objectnr
    FROM Verkoop
    WHERE Verkoop.objectnr = Woning.objectnr)
```

Opgave 3: Concurrency

(25 punten)

We beschouwen de onderstaande twee schedules.

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

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

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

Opgave 4: Normalisatie

(30 punten)

Ter herinnering wordt hier de decompositiestelling vermeld:

Stel we hebben een relatieschema $R(XYZ)$.
Als $X \rightarrow Y$,
dan is de decompositie $R_1(XY), R_2(XY)$ verliesvrij.

Gegeven een relatieschema $R(ABCDE)$ en een set FDs

$$F = \{A \rightarrow BCD, BC \rightarrow D, AD \rightarrow DE\}.$$

Hieronder zijn vijf mogelijke decomposities gegeven:

1. $(ABCD), (ADE)$
2. $(ABCD), (AE)$
3. $(ABC), (BCD), (ADE)$
4. $(ABC), (BCD), (AE)$
5. $(AD), (AE), (ABC)$

Beschouw de volgende drie eigenschappen:

DC : de decompositie kan vanuit R verkregen worden door het (herhaald) toepassen van de decompositiestelling en is derhalve verliesvrij.

BCNF : de decompositie is in **BCNF** .

DP : de decompositie is dependency preserving.

Ga voor elk van de vijf decomposities na of eigenschap **DC** geldt. Zo ja, geef dan de decompositievolgorde en geef dan ook aan of eigenschap **BCNF** geldt. Als **BCNF** geldt, geef dan ook aan of **DP** geldt. Licht de antwoorden met betrekking tot **DC** kort toe. De antwoorden met betrekking tot **BCNF** en **DP** hoeven niet toegelicht te worden.