

$8 \frac{1}{2} + 7 + 9 + 5 + 10 \rightarrow 8$

## Afdeling Wiskunde en Informatica R.U.G.

Naam: [REDACTED]	Studentnummer: [REDACTED]	Bladnr.: 1/5
Adres: [REDACTED]	Studierichting: [REDACTED]	Tentamen: DB
Postcode en Woonplaats: [REDACTED]	Jaar van eerste inschrijving: [REDACTED]	Datum: 10-3-97
		Naam docent: J. Gejan

1 a Als meerdere gebruikers gebruik maken van een DBMS, dan worden ze "geïnterleaveed". Oftewel: ~~opdrachten~~ opdrachten van meerdere gebruikers worden om en om afgehandeld. Als deze gebruikers nu allemaal van dezelfde variabele gebruik maken, dan kan er iets fout gaan. Omdat:

- iemand gebruik maakt van een waarde van de variabele die al weer veranderd is.
- een update van een variabele verloren gaat omdat een andere gebruiker een update heeft gedaan.

een concreet voorbeeld.

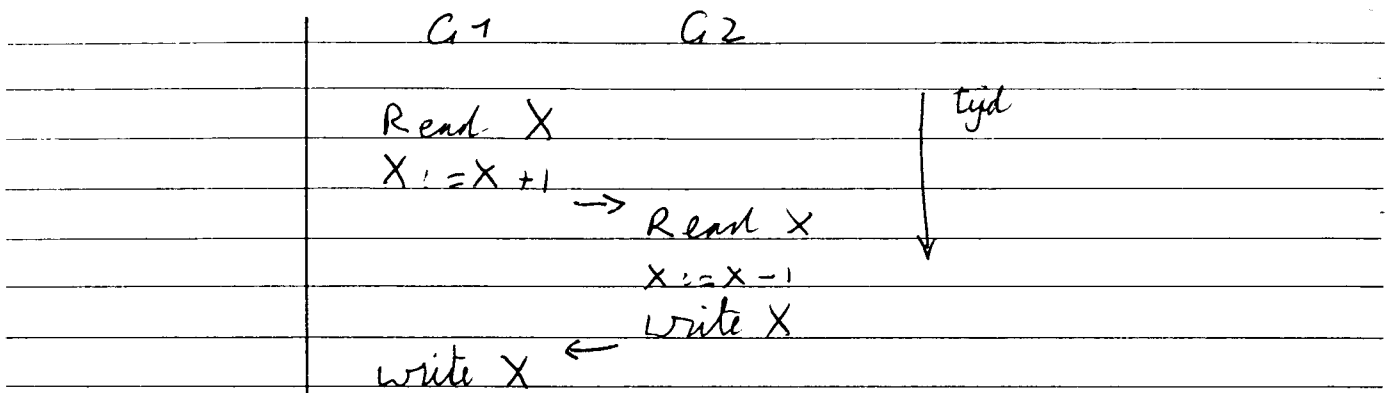
Stel gebruiker 1 wil het volgende:

```
read X
X := X + 1
write X
```

gebruiker 2 wil:

```
read X
X := X - 1
write X
```

Stel  $X$  was 3 en eerste wordt gebruiker 1 bediend en dan <sup>m</sup>de gebruiker 2. Dan wordt  $X$  dus eerst verhoogd tot 4 en daarna weer verlaagd naar 3. Echte als de interleaving plaats vindt kan het het volgende gebeuren: 200.



Na afloop van deze commands is X gelijk aan 4!

Er is dus iets fout gegaan.

Het oplossen van deze problemen leed concurrent control. Een voorbeeld hiervan is Time Stamping. Dit werkt als volgt:

Elke proces (aanvraag van een gebruiker om in te lezen of te schrijven) krijgt een nummer. Deze nummers worden in oplopende volgorde uitgedeeld en heten de 'Time stamp' van het proces ( $TS(X)$ )

Elke database variabele  $X$  krijgt twee getallen toegekend, nl de read- $TS(X)$  en de write- $TS(X)$ . Deze bevatten resp. de ~~laagste~~ laagste timestamp van de laatste succesvolle read en write van deze variabele. Voor het lezen en schrijven van een variabele gelden nu de volgende regels:

write X:

als  $TS(X) < \text{read-}TS(X)$  of  $TS(X) < \text{write-}TS(X)$  dan aanvraag afbreken en <sup>met</sup> nieuwe timestamp proberen andern aanvraag toelassen en  $\text{write-}TS(X) := TS(X)$ .

read X:

als  $TS(X) < \text{write-}TS(X)$  dan afbreken en met nieuwe TS proberen andern aanvraag toelassen en  $\text{read-}TS(X) := \max(TS(X), \text{read-}TS(X))$ .

Time Op deze manier worden problemen met de variabelen voorkomen. Bovendien heeft Time Stamping geen last van deadlock.

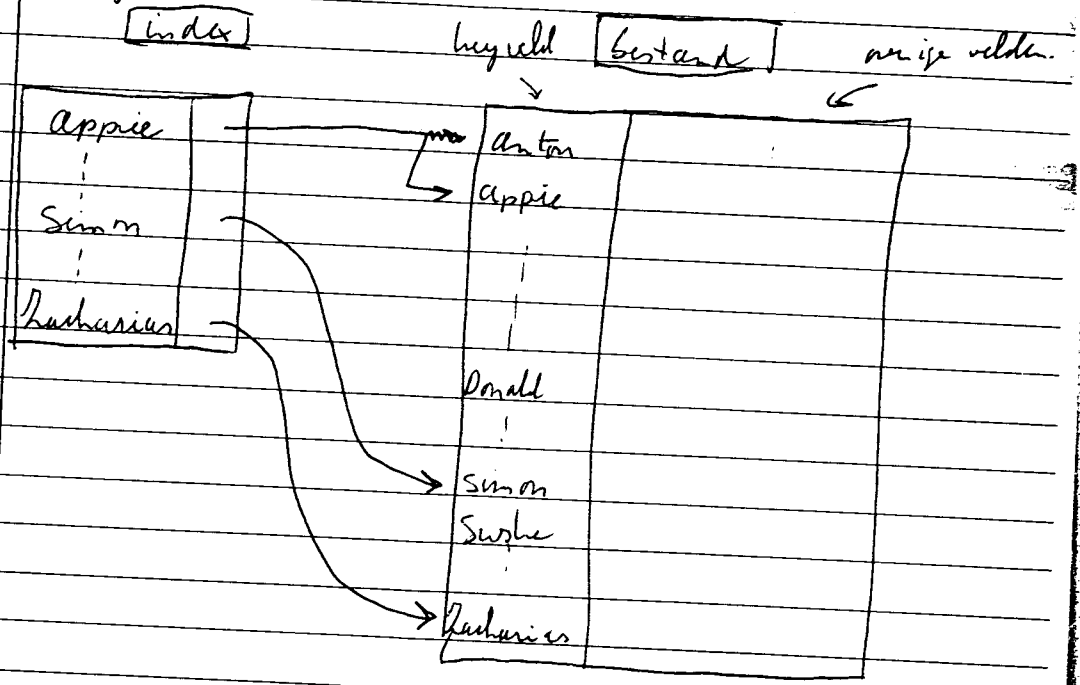
# Afdeling Wiskunde en Informatica R.U.G.

Naam: <span style="background-color: black; color: black;">[REDACTED]</span>	Studentnummer:	Bladnr.: 2/5
Adres:	Studierichting:	Tentamen: OR
Postcode en Woonplaats:	Jaar van eerste inschrijving:	Datum: Naam docent:

b. Een primary index in the (ordering key) is:

een index op een bestand dat geordend is opgedragen volgens een sleutel van dat bestand. Dit wil zeggen dat elke waarde van het sleutelveld precies één keer voorkomt in het bestand. De index bevat een aantal sleutelwoorden van dit bestand en wijst dan regelrecht naar het blok of record van het element te met het bewuste veld zich bevindt op schijf. ~~De index is een~~

skema:



✓ Z'n index is vordense, d.w.z. per disktrek wordt de key van elke record in de index opgehaald (anchor).

$\leftarrow$  Definitie  
 DW DDL: Data Description Language.  
 DML: Data Manipulation Language  
 DBA: Database Administrator.

De DDL wordt gebruikt om de structuur van de data te specificeren. Met de DML kan de data verwijzen worden en gemanipuleerd. Iemand die de DDL en DML gebruikt is de DBA, welke zorgt voor het onderhoud aan de database.

Voorbeeld van een DDL is Ledgeen en het gegeven is bij opgave 3. Mijn antwoord maakt gebruik van de bijbehorende DML.  $\int$

2.  $\{A, B, C, D, E\}$

①  $AB \rightarrow CDE$

②  $BCD \rightarrow AE$

③  $BD \rightarrow E$

a. Mhv ① zien we meteen dat  $AB^+ = \{a, b, c, d, e\}$ . Alleen A of B levert niet alle attributen, dus  $\{AB\}$  is een minimale sleutel van R. de set FA's.  $\int$

b. minimal cover:

1.  $AB \rightarrow C$

2.  $AB \rightarrow D$

3.  ~~$AB \rightarrow E$~~  wv 2+6

4.  $BCD \rightarrow A$

5.  ~~$BCD \rightarrow E$~~  wv 4+2+6  $\int$

6.  $BD \rightarrow E$

schema's:  $\leftarrow$  kopie van eerste (na verwijzing)

$\times$   $\{A, B, C\}, \{A, B, D\}, \{B, C, D, A\}, \{B, D, E\}$   $\otimes$   
 $\leftarrow$  samen!

Deze schema's bevatten alle attributen.

De minimale sleutel AB komt in twee van de schema van.

Dus is aan alle eisen voldaan.  $\otimes$  is een dependency preserving, lossless join 3NF decompositie van R.

2 c {A, B, C, D, E}  
 1 AB → CDE, BCD → AE, BD → E  
 ①                      ②                      ③

We zien dat AB en BCD minimale sleutels zijn.  
 (Dit volgt direct uit de conclusie van die 2.)

BD → E goed in tegen BCNF in {A, B, C, D, E}, dan  
 splitsen we  
 ↓  
 want BD is geen minimale sleutel.

{A, B, C, D}, {B, D, E} ④

Maar zijn er geen problemen meer. Dus ④ is  
 onze lossen voor decompositie in BCNF van R. 8

X ④ is echter niet dependency preserving, want bv  
 AB → E gaat verloren.  
 jawel hoor, via via ...  
 dat is nl. dezelfde decompositie als b5, 2b).

3 'Geef van alle studenten, die een cursus volgen bij  
 jongeren (Empnr = 17), de naam en in welke taal  
 dat gebeurt.'

```

$Find Any Student;
while (db_status = 0) do begin
  $GET Student;
  $Find First Course Using Follows;
  While (db_status = 0) do begin
    $GET Course;
    $FIND OWNER Using Teachers; ✓
    $Get Teacher;
    if (Teacher.Empnr = 17) then begin
      write(student, Sname, ' ');
      $Find Owner Using Holds;
      $GET Room;
      write( Room, Roomnr);
    end;
    $Find Next Course Using Follows;
  end;
$Find Duplicate Student;
end;
  
```

for db\_status = 0

dit is wel goed, maar niet efficiënt

202

Query in woorden:

Von alle studenten

Von alle vakken die de student volgt  
hoor de docent die het vak geeft.

Als die docent nummer 17 is (Jouzejan)

schrijf de naam van de student een schema;

hoor de taal waarin het vak gegeven wordt

schrijf dat het taalnummer naar schema.

5 Select <sup>A</sup> ~~M~~. ANR, <sup>A</sup> ~~A~~. ANAAM, <sup>A</sup> ~~B~~. BUDGET,  
Sum(~~M~~. SAL) as X,

(x) Count(~~M~~. ANR) as Y.

From MW M Natural Join AFD A

(x) Where 1 < Select ~~N~~. ANR Count(~~N~~. ANR)

(x) From MW N Natural Join AFD B

Where M. ANR = N. ANR

↓ group by

De tweede select geeft het aantal medewerkers per afdeling.

6  $\{ A \in \{ ANR, ANAAM, BUDGET \} \cup M \in \{ ANAAM, SAL \} \mid$   
 $A \in v(AFD) \text{ en } M \in v(MW) \text{ en } A(MANR) = M(MNR) \text{ en}$   
 $w > 1 \mid \{ B \in v(MW) \mid B(ANR) = A(ANR) \} \mid \text{ en}$   
 $x \times A(BUDGET) < \{ \sum O \{ SAL \} \mid O \in v(MW) \text{ en } O(ANR) =$   
 $A(ANR) \}$   
}