

Naam: [REDACTED]	Studentnummer: [REDACTED]	Bladnr.: 1
Adres: [REDACTED]	Studierichting: [REDACTED]	Tentamen: Databases
Postcode en Woonplaats: [REDACTED]	Jaar van eerste inschrijving: [REDACTED]	Datum: 05-04-2004
		Naam docent: Balsters / Songeja

1 ~~van~~ Attributen die niet uit fd's te bepalen zijn, zijn A, B en E; deze moeten dus in een sleutel zitten en vormen samen ook de enige minimale sleutel.  $\int$

~~WVCDT heeft geen fd's niet in BCNF, want  $\{A, B, C\}$  heeft de fd  $AB \rightarrow C$  terwijl AB geen superkey is.~~

We maken een BCNF-decompositie:

$$D_1 = \{\{A, B, C, D, E\}\}$$

Dit is geen BCNF vanwege  $AB \rightarrow C$  terwijl AB geen superkey is; er moet dus worden gesplitst:

$$D_2 = \{\{A, B, D, E\}, \{A, B, C\}\}$$

$\{A, B, D, E\}$  moet nog gesplitst vanwege  $B \rightarrow D$ :

$$D_3 = \{\{A, B, E\}, \{B, D\}, \{A, B, C\}\}$$

Hierin zitten geen fd's meer die met BCNF conflicteren, dus is het een geldige BCNF-decompositie. Het juiste antwoord is dus c.  $\int$

2 a. is onjuist, dat is precies andersom. b is onjuist, want 4NF is strikter dan BCNF. c. is juist omdat 4NF ~~van~~ BCNF impliceert.  $\int$

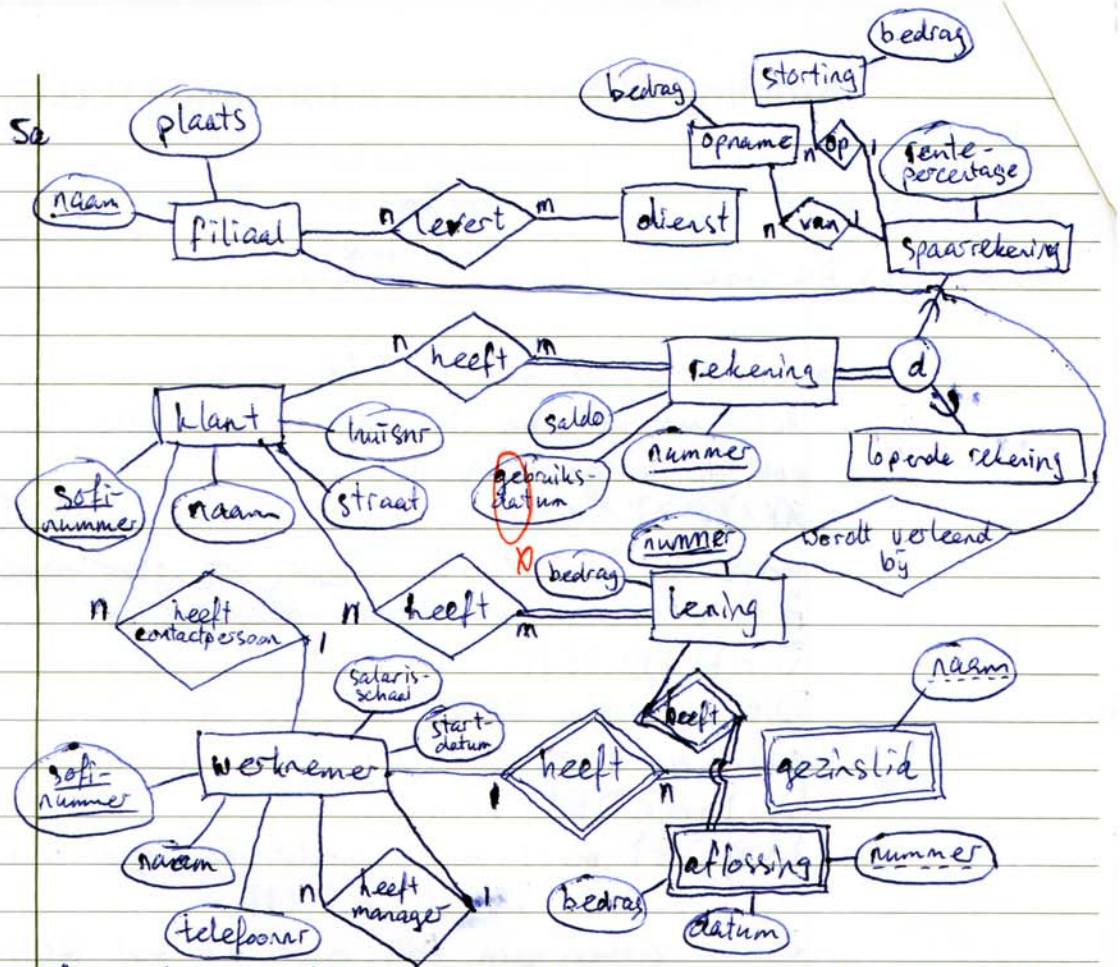
3 a. is juist. b is onjuist want een lossless join "decompositie" kan ook de relatie zelf zijn, die niet in 3NF hoeft te zijn. c is onjuist, want er mag ook een secundaire sleutel zijn. d is onjuist, want BCNF-decomposities zijn niet altijd dependency preserving.

4a De 3-schema architectuur bestaat uit 3 lagen: ~~op~~ opslag-niveau, dataniveau en viewniveau. Het opslagniveau zorgt voor de opslag van de data. Het dataniveau representeert de data zelf. Het viewniveau levert een bepaalde kijk op deze data.

b Afbeeldingen tussen schemaniveaus zijn nodig zodat deze op een eenduidige ~~manier~~ en consistente manier kunnen communiceren.

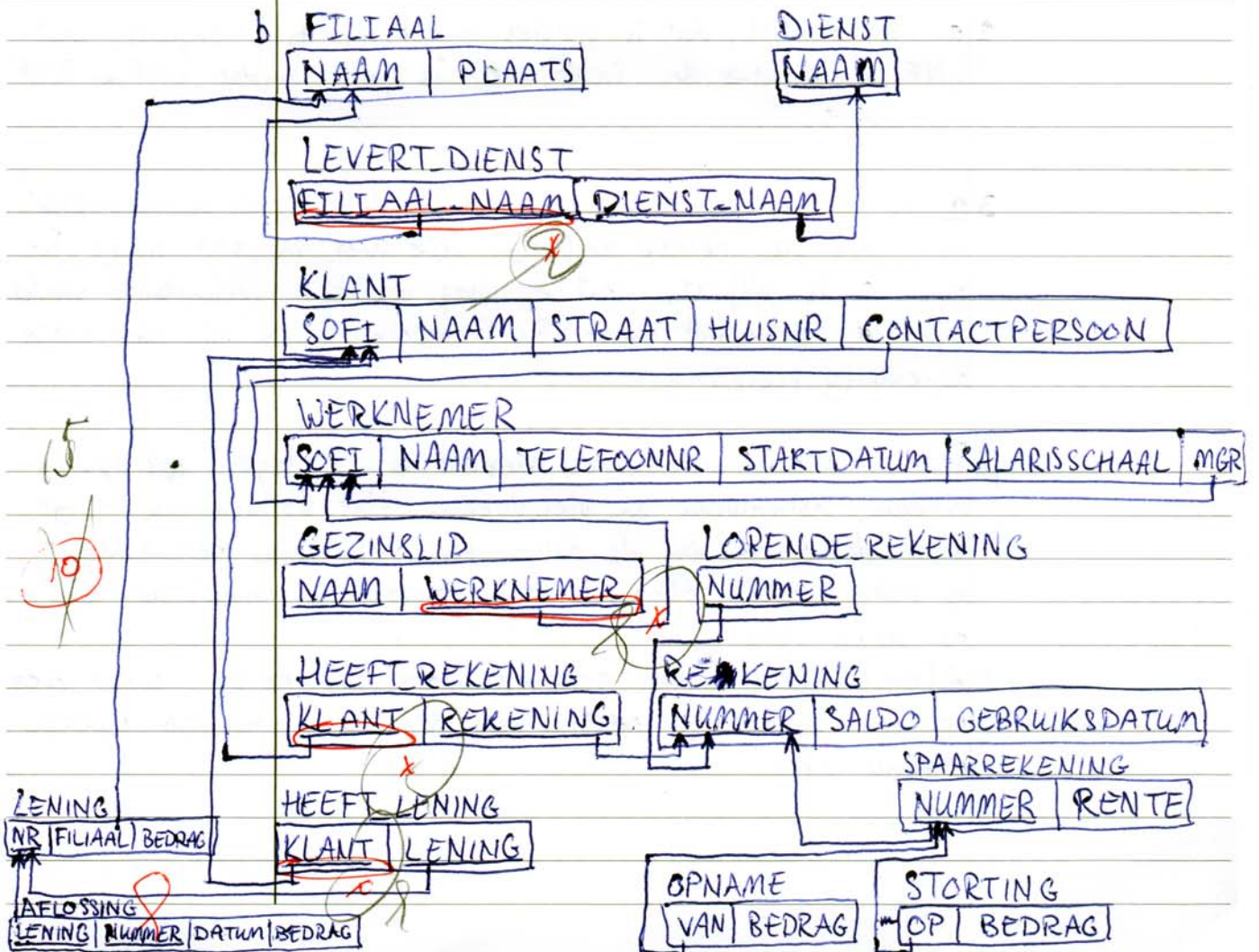
c





8-14

NB. Wat klopt er aan de relatie bij de klant naar werknemer? Excuses voor de rommel.



5-10

5 Wederom excuses voor de rommel.  
 (versvolg) Een aantal punten ontbreken in de beschrijving die waarschijnlijk wel gewenst zijn:

- woonplaats van een klant
- datum van stortingen en opnames
- naam en beschrijving van diensten
- bedrag van stortingen en opnames.

Hiervan heb ik ~~alleen~~ een aantal wel in de diagrammen opgenomen.

✓ passagier?

```

6 FIND FIRST Vlucht;
  while not eof eof do
    if v v := GET Vlucht;
    if v.Van = "Schiphol" then
      FIND FIRST v IN Boeking OWNED BY v;
      while not eos do
        p := GET Passagier FROM Boeking;
        if p.Naam = "Jansen" then
          FIND OWNER IN Vliegt WITH v;
          q := GET Piloot FROM Vliegt;
          print q.Naam;
          break;
        end;
      end;
      passagier FIND NEXT IN Boeking;
    end;
  end;
  FIND NEXT IN Vlucht;
end;
  
```

N.B.: waarschijnlijk is de syntax hier volledig fout; ik hoop echter dat de naderker het met mij eens is dat begrip van de achterliggende principes belangrijker is dan de precieze syntax. 😊

10 c is de juiste query. We willen alle waarden die èn in R zitten, èn in S of T. (Er van uitgaande dat de laatste OR in query C er niet hoort te staan.)



12a Product (maker, model)

PC (model, prijs)

Laptop (model, schermgrootte, prijs)

Antwoord d is juist. Wat er fout is aan alledrie de queries, is dat ze alleen prijzen vergelijken van pc's en laptops met hetzelfde model, en niet met dezelfde maker.

15 Statement a is juist. b doet niets met pc's, en c is syntactisch fout (BY).

19 Constraint a is juist. Deze telt de <sup>pc's</sup> makers ~~van~~ ~~per~~ maker en levert alleen die makers waarvoor dit meer dan 1 is. Als ~~dit~~ deze bestaat is de constraint geschonden.