

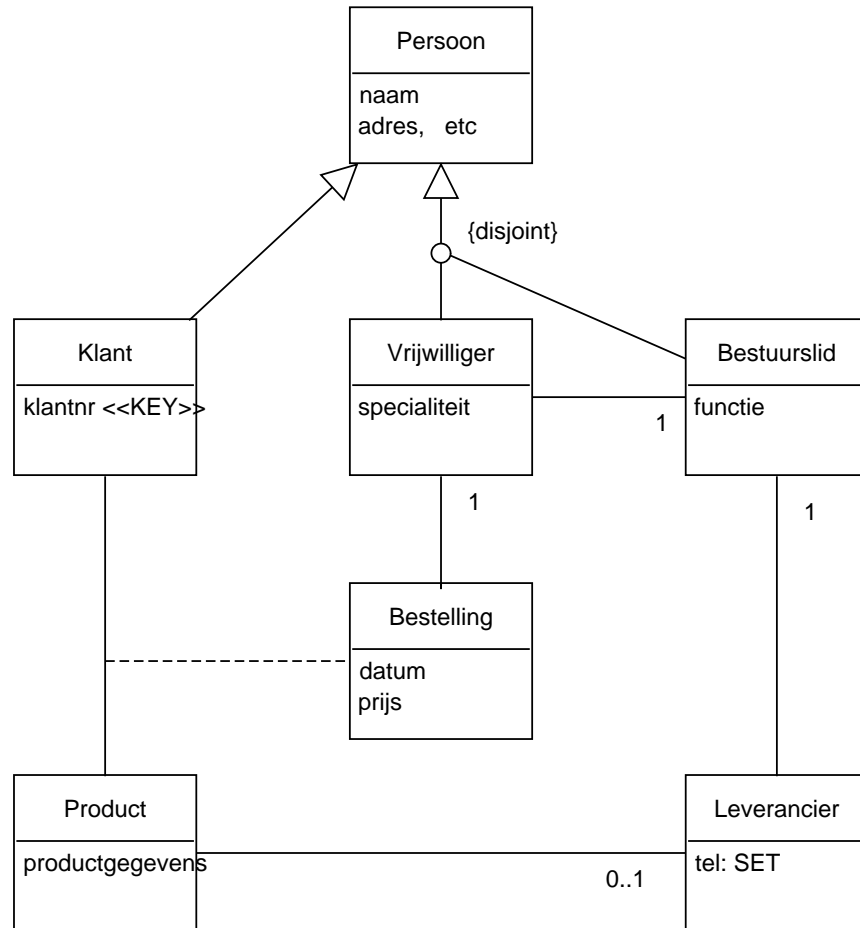
10p.

Opgave 2. Stichting *IAPC* is een ICT winkel en adviescentrum “voor studenten door studenten”. Je kunt bij *IAPC* terecht voor producten uiteenlopend van kabels tot complete PC’s en voor al je technische vragen. (*IAPC* zoekt nog één of twee bestuursleden voor 2006-2007.) De beschrijving hieronder komt misschien niet exact overeen met de werkelijkheid — en daarom gaat het over *IAPC'* in plaats van *IAPC*.

- *IAPC'* maakt onderscheid tussen klanten, vrijwilligers en bestuursleden, maar een vrijwilliger en een bestuurslid kan ook klant zijn. Vrijwilligers staan in de winkel en geven advies; zij zijn geen bestuurslid. Klanten hebben een (identificerend) klantnummer.
- Een klant kan een product bestellen. Per bestelling (van een product door een klant) is bekend op welke datum dat is gebeurd en tegen welke prijs en welke vrijwilliger daarbij betrokken was.
- Ieder product wordt geleverd door hooguit één leverancier. Leveranciers leveren in het algemeen meerdere producten. Stichting *IAPC'* houdt van ieder product enige productgegevens bij.
- Bestuursleden functioneren als kontaktpersonen naar leveranciers; iedere leverancier heeft precies één kontaktpersoon. Iedere leverancier heeft een stel telefoonnummers.
- Iedere vrijwilliger heeft één bestuurslid als kontaktpersoon. Van iedere persoon houdt *IAPC'* gegevens bij zoals naam, adres, woonplaats en bovendien ook van ieder bestuurslid zijn bestuursfunctie en van iedere vrijwilliger zijn specialiteit.

Geef in het antwoordblok een Entity-Relationship diagram dat bovenstaande situatie *zo precies mogelijk* modelleert en gebruik daarbij *zo geschikt mogelijke* constructies. Zowel de ERD-notatie uit het boek als ook de UML notatie (class diagram) is toegestaan, maar een mengeling van beide niet.

(Niet-vermelde multipliciteiten leggen geen beperkingen op: 0..*.)



(Zie Toelichting.)

Het zou kunnen zijn (maar het is niet de bedoeling) dat uw diagram ten onrechte of met opzet méér eigenschappen modelleert dan de casus aangeeft. Om zo'n "fout" enigszins te herstellen is er deze vraag: Noem en motiveer hoogstens twee eigenschappen die *wel* in het diagram staan maar *niet* uit de letterlijke tekst van de casus volgen (indien zulke eigenschappen bestaan):

Onleesbare tekst wordt fout gerekend.

Geen.

Het zou kunnen zijn (maar het is niet de bedoeling) dat uw diagram onbedoeld of met opzet minder eigenschappen modelleert dan de casus aangeeft. Om dat gebrek enigszins te herstellen is er deze vraag: Geef, in precieze bewoordingen of formules, hoogstens twee eigenschappen die *wel* uit de casustekst volgen maar *niet* in uw ER-diagram opgenomen zijn (indien zulke eigenschappen bestaan):

Onleesbare tekst wordt fout gerekend.

Wanneer $(c1, d1)$ een sleutel van S wordt verklaard te zijn, dan moet op z'n minst hier
vermeld worden dat de $0..1$ multipliciteit bij S nog niet in het databaseschema verwerkt is.

10p.

Opgave 3. Bij de sportvereniging *AltijdDoorgaan* spelen verscheidene entiteiten een rol, zo als: speler, groep, trainer, klasse, etc. We definiëren een relatie \mathcal{R} die die entiteiten aan elkaar relateert, op de volgende manier.

Een tuple (S, G, T, B, K, C, D, V) zit in relatie \mathcal{R} precies wanneer het volgende geldt:

1. S is een *Speler* van *AltijdDoorgaan*.
2. G geeft de *Groep* aan waarin speler S speelt.
3. T is de *Trainer* van groep G .
4. B geeft aan of speler S de contributie *Betaald* heeft.
5. K is de *Klasse* van de sportbond waarin groep G speelt.
6. C is de *Chairman* van de sportvereniging *AltijdDoorgaan*.
7. D is een van de *Dagen* (maandag, . . . , zaterdag) waarop groep G speelt.
8. V is het *Veld* waarop groep G op dag D speelt.

Neem het volgende aan:

- a. Relatie \mathcal{R} beschrijft de situatie van één tijdstip (namelijk: 10 november 2006).
- b. Een lid speelt maar in één groep.
- c. Een groep speelt maar in één klasse.
- d. Een groep speelt per dag hooguit één keer.
- e. Vereniging *AltijdDoorgaan* heeft drie velden waarop gespeeld kan worden.

Geef voor ieder van de functionele afhankelijkheden hieronder, met een letter W of O aan of die *Waar* of *Onwaar* is in relatie \mathcal{R} , en motiveer kort uw keuze. (De motivatie telt mee in de beoordeling.)

Onleesbare tekst wordt fout gerekend.

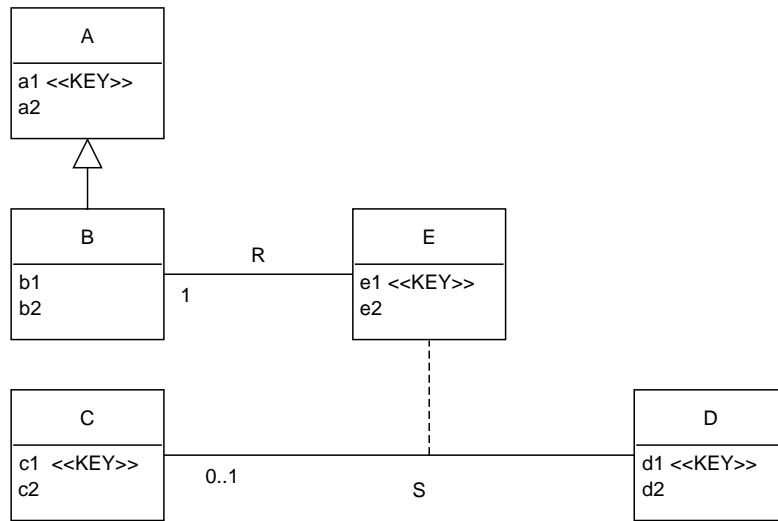
FD	W/O	Motivatie voor uw keuze
$S \rightarrow T$	W	Wegens $S \rightarrow G$ (2 of b) en $G \rightarrow T$ (3)
$T \rightarrow S$	O	Een trainer kan meerdere groepen, dus spelers, trainen.
$GT \rightarrow S$	O	Een groep (dus ook de trainer ervan) kan meerdere spelers hebben.
$G \rightarrow C$	W	De chairman ligt vast (6), er geldt zelfs $\emptyset \rightarrow C$.
$CG \rightarrow K$	W	Er geldt zelfs $G \rightarrow K$ wegens “de” in (5 of c).
$DV \rightarrow G$	O	Ondanks (7,8) kan het best zo zijn dat G_0 en G_1 beide op maandag op veld 1 spelen.

Geef voor ieder van de volgende MultiValued Dependencies aan of die *Waar* of *Onwaar* is in relatie \mathcal{R} :

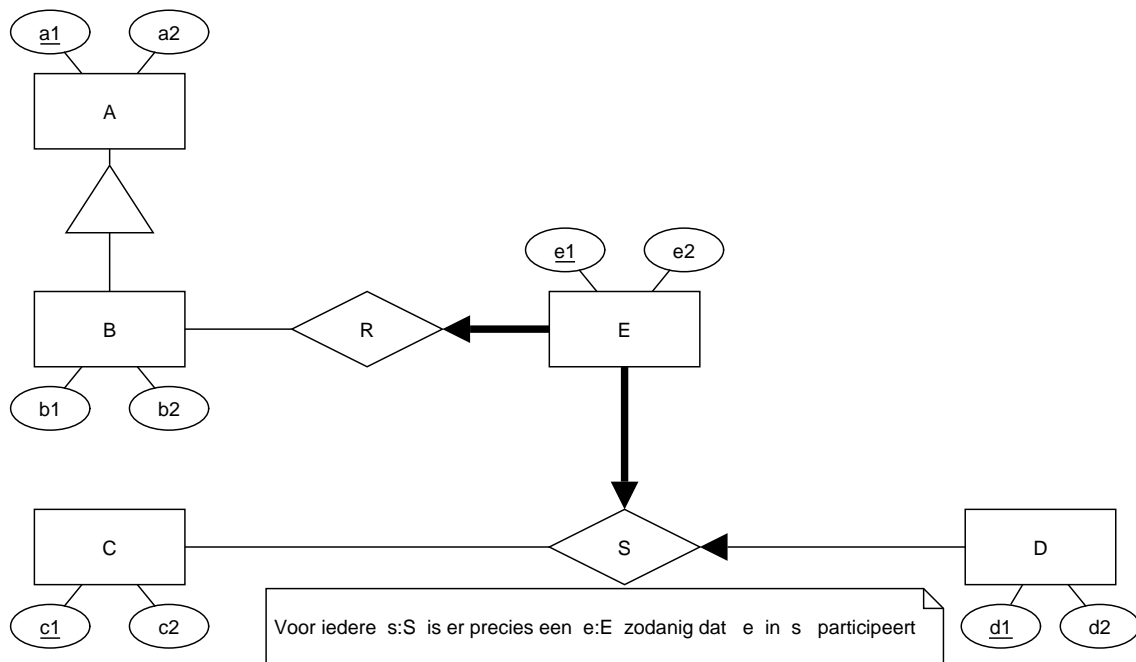
Onleesbare tekst wordt fout gerekend.

MVD	W/O	
$SGTBKCDV = SGTBD \bowtie GKCDV$	W	GD is key van rechtercomponent.
$SGTBKCDV = SGTBD \bowtie SKCDV$	W	SD is key van rechtercomponent.
$SGTBKCDV = SGTBKDV \bowtie C$	W	\emptyset is key van rechtercomponent. (Zie Toelichting.)

10p. **Opgave 4.** Beschouw het volgende ERD in de notatie van de UML (boven) of Kifer (onder):



(niet-geschreven multipliciteiten staan voor 0..*)



Het ER-diagram kan op verschillende manieren vertaald worden naar een databaseschema dat geschikt is om informatie die past in het ER-diagram, op te slaan. U dient een manier te kiezen waarbij er zo weinig mogelijk relatieschema's in het databaseschema zijn, maar met de beperkingen dat er geen NULLs nodig zijn vanwege de vertaling, er geen redundantie geïntroduceerd wordt door de vertaling, en alle attributen atomaire waarden hebben.

Geef de relatieschema's en geef bij ieder relatieschema *alle* sleutels en en alle foreign keys:

Onleesbare tekst wordt fout gerekend.

A (a1, a2, primary key a1)
B (a1, b1, b2, primary key a1, foreign key a1 references A(a1))
C (c1, c2, primary key c1)
D (d1, d2, primary key d1)
E (e1, e2, a1, c1, d1, primary key e1, alternatief: primary key (d1), foreign key a1 references B(a1), foreign key c1 references C(c1), foreign key d1 references D(d1))
<i>(Zie Toelichting.)</i>

Het zou kunnen zijn dat er eigenschappen in het ER-diagram staan die niet vermeld zijn in het databaseschema in het vorige antwoord en daar ook niet uit volgen. Geef die eigenschappen indien er zulke eigenschappen zijn:

Onleesbare tekst wordt fout gerekend.

Zie onderdeel (2) en (3) van de toelichting op het vorige antwoord.

10p.

Opgave 5. Beschouw het relatieschema $\mathbf{R} = (\bar{R}, \mathcal{F})$, waarbij de attribootverzameling \bar{R} en de verzameling \mathcal{F} van functionele afhankelijkheden als volgt luiden:

$$\bar{R} = ABCDE$$

$$\mathcal{F} = \{A \rightarrow D, \quad B \rightarrow A, \quad BC \rightarrow D, \quad DE \rightarrow A\}$$

Geef de sleutels van \mathbf{R} :

Onleesbare tekst wordt fout gerekend.

BCE

Geef alle supersleutels van \mathbf{R} :

Onleesbare tekst wordt fout gerekend.

BCE, ABCE, BCDE, ABCDE

Geef in het antwoordblok in iedere regel een zo groot mogelijk rechterlid Y zó dat de functionele afhankelijkheid $X \rightarrow Y$ volgt uit de hierboven gegeven verzameling \mathcal{F} (met andere woorden: Y is de closure $X_{\mathcal{F}}^+$). U hoeft de leden van X niet op te nemen in Y .

Onleesbare tekst wordt fout gerekend.

	X	\rightarrow	Y
1		\rightarrow	
2	A	\rightarrow	D
3	B	\rightarrow	AD
4	C	\rightarrow	
13	BE	\rightarrow	AD
14	CD	\rightarrow	
15	CE	\rightarrow	
16	DE	\rightarrow	A
22	ADE	\rightarrow	
23	BCD	\rightarrow	A
24	BCE	\rightarrow	AD
25	BDE	\rightarrow	A
31	$BCDE$	\rightarrow	A
32	$ABCDE$	\rightarrow	

Zet in onderstaande tabel een kruisje bij iedere schending van de BCNF-conditie voor **R**.
 (Sommige van onderstaande functionele afhankelijkheden gelden niet in **R**.)

Onleesbare tekst wordt fout gerekend.

	$X \rightarrow Y$
?	$A \rightarrow B$
x	$B \rightarrow D$
?	$C \rightarrow ABDE$
?	$AB \rightarrow E$
?	$AC \rightarrow BC$
x	$BE \rightarrow AD$
	$CD \rightarrow$
x	$DE \rightarrow A$
x	$BCD \rightarrow A$
	$BCE \rightarrow AD$
x	$CDE \rightarrow A$
	$BCDE \rightarrow A$
	$ABCDE \rightarrow ABCDE$

(Zie Toelichting.)

10p. **Opgave 6.** Beschouw het relatieschema $\mathbf{R} = (ABCDE, \mathcal{F})$, waarbij:

$$\mathcal{F} = \{AC \rightarrow B, \quad C \rightarrow D, \quad D \rightarrow E\}$$

Ter informatie: de niet-triviale leden $X \rightarrow Y$ uit \mathcal{F}^+ met $X \cap Y = \emptyset$ en Y maximaal (dus $Y = X_{\mathcal{F}}^+ \setminus X$), staan opgesomd in de volgende tabel:

	X	\rightarrow	Y
1	C	\rightarrow	DE
2	D	\rightarrow	E
3	AC	\rightarrow	BDE
4	AD	\rightarrow	E
5	BC	\rightarrow	DE
6	BD	\rightarrow	E
7	CD	\rightarrow	E
8	ABC	\rightarrow	DE
9	ABD	\rightarrow	E
10	ACD	\rightarrow	BE
11	ACE	\rightarrow	BD
12	BCD	\rightarrow	E
13	BCE	\rightarrow	D
14	$ABCD$	\rightarrow	E
15	$ABCE$	\rightarrow	D
16	$ACDE$	\rightarrow	B

Construeer in het antwoordblok een lossless decompositie van \mathbf{R} tot schema's die ieder in BCNF staan. *Geef ook aan of de functionele afhankelijkheden behouden blijven onder de decompositie.* Geef bij iedere stap een verklaring zodat het voor de corrector heel duidelijk is hoe u te werk gaat.

We passen het BCNF-algoritme toe.

- We bekijken $\mathbf{R} = (ABCDE, \mathcal{F})$.

Van de gegeven \mathcal{F} zijn $C \rightarrow D$ en $D \rightarrow E$ een schending van de BCNF-eis voor \mathbf{R} , want die FD's zijn niet-triviaal en hun linkerleden bevatten geen sleutel van \mathbf{R} — hetgeen uit de gegeven opsomming van \mathcal{F}^+ blijkt: $C_{\mathcal{F}}^+ = DE \neq ABCDE$ en $D_{\mathcal{F}}^+ = E \neq ABCDE$. We kiezen (zomaar) de eerste, $C \rightarrow D$, ter eliminatie. Dus splitsen we $\bar{R} = ABCDE$ in $\bar{R}_1 = CD$ en $\bar{R}_2 = ABC\cancel{D}E = ABCE$.

Dit levert schema's $\mathbf{R}_i = (\bar{R}_i, \mathcal{F}_i)$, waarbij \mathcal{F}_i (een basis voor) de inperking is van \mathcal{F}^+ tot \bar{R}_i .

(Let op: vergeet de closure \dots^+ bij \mathcal{F} niet!) Dus, $\mathbf{R}_1 = (CD; \{C \rightarrow D\})$ en, **LET OP**, $\mathbf{R}_2 = (ABCE; \{AC \rightarrow B, C \rightarrow E\})$. De $C \rightarrow E$ zit weliswaar niet in \mathcal{F} , maar toch in \mathcal{F}_2 omdat $C \rightarrow E$ in \mathcal{F}^+ zit en dus ook, per definitie, in \mathcal{F}_2 . (Merk ook op dat de afhankelijkheid $D \rightarrow E$ verloren is gegaan!)

- We bekijken nu $\mathbf{R}_1 = (CD, \mathcal{F}_1)$.

Omdat er in \mathcal{F} en dus in \mathcal{F}_1 geen niet-triviale FD's van de vorm $\emptyset \rightarrow Y$ zijn, en \mathbf{R}_1 slechts twee attributen heeft, staat \mathbf{R}_1 in BCNF.

- We bekijken nu $\mathbf{R}_2 = (ABCE, \{AC \rightarrow B, C \rightarrow E\})$.

Nu is $C \rightarrow E$ een schending van de BCNF-eis voor \mathbf{R}_2 , want die FD is niet-triviaal en C bevat geen sleutel van \mathbf{R}_2 : $C_{\mathcal{F}_2}^+ = E \neq ABCE$ (de enige sleutel in \mathbf{R}_2 is AC).

Dus splitsen we $\bar{R}_2 = ABCE$ in $\bar{R}_{2a} = ABC\cancel{E} = ABC$ en $\bar{R}_{2b} = CE$.

Dit levert schema's $\mathbf{R}_{2j} = (\bar{R}_{2j}, \mathcal{F}_{2j})$, waarbij \mathcal{F}_{2j} de inperking is van \mathcal{F}^+ (of \mathcal{F}_2^+) tot \bar{R}_{2j} . (Let op: vergeet de closure \dots^+ bij \mathcal{F} niet!)

- We bekijken nu $\mathbf{R}_{2a} = (ABC, \{AC \rightarrow B\})$ en $\mathbf{R}_{2b} = (CE, \{C \rightarrow E\})$. Omdat in \mathbf{R}_{2a} het stel AC een sleutel is, en $AC \rightarrow B$ dus geen schending van de BCNF-conditie voor \mathbf{R}_{2a} , is \mathbf{R}_{2a} in BCNF. Analoog voor \mathbf{R}_{2b} .

- Dus $\{\mathbf{R}_{2a}, \mathbf{R}_{2b}\}$ is een decompositie van \mathbf{R}_2 , en $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b}\}$ van \mathbf{R} .

- De functionele afhankelijkheden zijn niet behouden; met name is $D \rightarrow E$ verloren gegaan (al in de eerste decompositiestap).

- NB. Omdat dit een lossless decompositie is (een eigenschap van het toegepaste BCNF-algoritme), schrijven we ook wel:

“ $ABCD = ABC \bowtie CE \bowtie CD$ geldt in \mathbf{R} .”

Wanneer je met een andere schending begint kun je mogelijk een andere decompositie bereiken, en een andere uitslag voor het behoud van de functionele afhankelijkheden.

In de volgende opgavenserie wordt het volgende databaseschema gebruikt:

```
Class (name, type, country, guns, bore, displacement)
Ship (name, classname, launched)
Battle (name, date)
Outcome (shipname, battlename, result)
```

De attributen die tot de sleutel behoren zijn onderstreept.

In *Ship* is *classname* een foreign key verwijzend naar *Class* (*name*).

In *Outcome* is *shipname* een foreign key verwijzend naar *Ship* (*name*).

In *Outcome* is *battlename* een foreign key verwijzend naar *Battle* (*name*).

Schepen die volgens eenzelfde ontwerp worden gebouwd vormen samen een klasse (*class*). Klassen komen in twee typen (*type*): *bb* voor *battleship* en *bc* voor *battlecruiser*. De overige attributen van een *class* zijn: het land (*country*), het aantal kanonnen (*guns*), de diameter in centimeters van de kanonsloop (*bore*), en de waterverplaatsing (*displacement*, gemeten in tonnen). Alle schepen van een klasse komen uit één land: het land dat in de klasse genoemd staat. Van een *ship* is, naast de naam (*name*) en de klassenaam (*classname*), ook nog bekend wanneer het te water is gelaten (*launched*). Van een zeeslag (*battle*) is de naam (*name*) en datum (*date*) bekend. Relatie *Outcome* geeft aan hoe schepen de zeeslagen hebben doorstaan: gezonken, beschadigd of okay (*result* = *sunk*, *damaged*, en *ok*, respectievelijk).

U mag identifiers tot hun eerste letter afkorten. Het schema luidt dan:

```
C (n, t, c, g, b, d)
S (n, c, l)
B (n, d)
O (s, b, r)
```


Geef ook een formulering van de vraag in Relationale Algebra:

Onleesbare tekst wordt fout gerekend.

$\pi_{sn} (\sigma_{os=sn \wedge ob=bn \wedge or=sunk} (S' \times B' \times O'))$	=	$\pi_{sn} (\sigma_{or=sunk} (S'' \bowtie B'' \bowtie O''))$
Hierbij zijn met renaming nieuwe relaties gedefinieerd:		
$O' = O[os, ob, or]$ en $S' = S[sn, sc, sl]$ en $B' = B[bn, bd]$ en		
$O'' = O[s, b, or]$ en $S'' = S[s, sc, sl]$ en $B'' = B[b, bd]$.		

Class (<u>name</u> , type, country, guns, bore, displacement)	C (<u>n</u> , t, c, g, b, d)
Ship (<u>name</u> , classname, launched)	S (<u>n</u> , c, l)
Battle (<u>name</u> , date)	B (<u>n</u> , d)
Outcome (<u>shipname</u> , <u>battlename</u> , result)	O (<u>s</u> , <u>b</u> , r)

10p. **Opgave 8.** Formuleer in SQL met een group-by query:



Geef voor ieder land waarvan de schepen hooguit 100 ton van elkaar verschillen in waterverplaatsing (displacement), het aantal kanonnen dat een schip van dat land *gemiddeld* heeft.

(Dus, per land dat aan de voorwaarde voldoet, wordt het gemiddelde genomen over al de schepen van dat land.) Sla de landen zonder schepen over; voor zo'n land is het gemiddelde aantal kanonnen niet bepaald.

Onleesbare tekst wordt fout gerekend.

select <i>c.country</i> , avg(<i>c.guns</i>) from <i>Class c</i> , <i>Ship s</i> where <i>c.n = s.c</i>
group by <i>c.country</i>
having <i>max(displacement) - min(displacement) ≤ 100</i>
Weglaten van <i>Ship s</i> heeft tot gevolg dat het gemiddelde verkeerd wordt berekend: niet over alle schepen van dat land, maar over alle klassen van dat land.

10p. **Opgave 10.** Geef een definitie (of nauwkeurige beschrijving) van de volgende begrippen.
System catalog:

Onleesbare tekst wordt fout gerekend.

Zie het boek (gebruik de index of kijk direct naar Section 3.3.2).

Een *system catalog* is een tabel waarin de beschrijving (door middel van kolommen *Attribuutnaam*, *Relatiennaam*, *Positie*, *Waardedomein*, etc) staat van de tabellen van een database.

Referential integrity:

Onleesbare tekst wordt fout gerekend.

Zie het boek (gebruik de index of kijk direct naar pagina 43).

Referential integrity is de eigenschap dat rijen/objecten waarnaar verwezen wordt ook daadwerkelijk bestaan. De Foreign Key constraint is een belangrijk voorbeeld van een Referential Integrity constraint.

OLAP, of alleen het karakteristieke verschil tussen OLAP en OLTP
(OLAP = Online Analytical Processing, OLTP = Online Transaction Processing):

Onleesbare tekst wordt fout gerekend.

Zie hoofdstuk 1.

OLAP is het gebruik van gegevens in een database *voor ondersteuning van management beslissingen*. De queries *duren soms heel lang*, vergeleken met transacties die een database up-to-date houden (OLTP), en behoeven in het algemeen geen *accurate* data te hebben.

DBA (= Database Administrator):

Onleesbare tekst wordt fout gerekend.

Zie hoofdstuk 1.

De DBA is de functie/persoon die de database onderhoudt; hij verzorgt onder andere de storage allocation, optimalisatie en security betreffende de database.

De SQL-bewerkingen (select, update, delete, insert) die niet op *views* mogen plaats vinden maar wel op gewone *tables*:

Onleesbare tekst wordt fout gerekend.

Een wijziging op een view wordt gerealiseerd door wijzigingen op de onderliggende base tables. Wanneer die niet eenduidig bepaald zijn door de wijziging op de view, dan wordt die niet toegelaten. Dit doet zich onder andere voor wanneer de view uit meerdere tabellen gevormd is of via een group-by of aggregatie.

10p. **Opgave 11.** Gegeven zijn de volgende transacties:

$$T_1 = r_1(x); w_1(z); w_1(x)$$

$$T_2 = r_2(x); r_2(z); w_2(x)$$

Welke van de volgende schedules Sa en Sb zijn serialiseerbaar en waarom?

$$Sa = r_1(x); r_2(x); r_2(z); w_1(z); w_1(x); w_2(x)$$

$$Sb = r_1(x); r_2(x); w_1(z); r_2(z); w_1(x); w_2(x)$$

Onleesbare tekst wordt fout gerekend.

Geen van tweeen. Omdat $r(x)$ en $w(x)$, en ook $w(x)$ en $w(x)$, niet *commuteren*, kunnen niet alle acties van T_1 vóór die van T_2 gebracht worden door commutatie, en ook niet áchter die van T_2 . (Twee acties a en b *commuteren* wanneer a ; b het zelfde effect heeft als b ; a .)

Definieer of beschrijf in woorden wat, in de context van transacties, een *phantom* is:

Onleesbare tekst wordt fout gerekend.

Een *phantom* voor transactie T is een rij die in een tabel erbij komt (door toedoen van een andere transactie) terwijl T die tabel die tabel leest of her-leest.

Geef aan hoe phantoms voorkomen kunnen worden:

Onleesbare tekst wordt fout gerekend.

Door T te laten executeren onder een geschikt isolatieniveau; in dit geval moeten niet alleen rijen gelockt worden maar hele tabellen. Het SQL isolatieniveau **serializable** voldoet hieraan.

De write-ahead log is een techniek in die gebruikt wordt om, in de context van transacties, atomicity te realiseren. (Dit mag u bekend veronderstellen.) Geef aan wat *write-ahead* inhoudt, en wat er fout kan gaan als de log niet volgens de *write-ahead* procedure gebruikt

wordt:

Onleesbare tekst wordt fout gerekend.

Write-ahead houdt in dat bij een wijziging in de database éérs de log bijgehouden wordt en dan pas de database zelf gewijzigd wordt. (Om atomicity te realiseren wordt bij een schrijfactie de *oude* waarde in de log geschreven zo dat roll-back mogelijk is). Wanneer *write-ahead* niet gevold wordt is er bij een crash ná een schrijfactie op de database en vóór de actie op de log, de oude waarde niet meer bekend en kan er geen roll-back uitgevoerd worden.

Toelichtingen

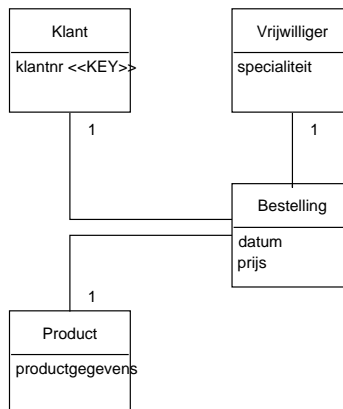
Toelichting bij antwoord 2.

(1) De twee specialisaties van *Persoon* kunnen ook samengenomen worden tot één specialisatie (waarbij de sub-entiteiten *Klant*, *Vrijwilliger* en *Bestuurslid* op gelijke voet behandeld worden). De *disjointness*-eigenschap moet in dat geval aangegeven worden met een aantekening die het diagram vergezelt: “*Vrijwilliger* en *Bestuurslid* zijn disjoint”.

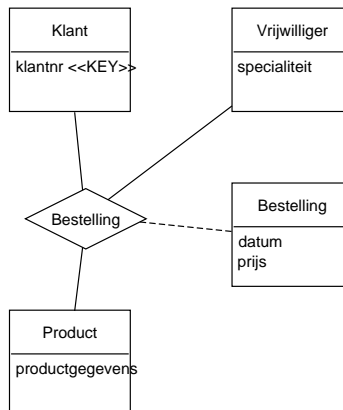
(2) De mededeling *covering* bij de specialisatie wordt niet fout gerekend; het betekent dat iedere *persoon* een *klant* of *vrijwilliger* of *bestuurslid* is.

(3) *Telefoon* als een aparte entiteit is afkeuringswaardig: over *telefoons* wordt geen aparte informatie bijgehouden, en telefoons zijn niet de entiteiten waarin de opdrachtgever geïnteresseerd is. Pas bij de omzetting naar een databaseschema zal er een aparte tabel voor telefoons komen (om de eerste normaalvorm te bereiken: alle attributen hebben atomaire waarden).

(4) Volgens bovenstaand diagram kan een klant k_0 een product p_0 hooguit éénmaal bestellen: het paar (k_0, p_0) zit in de relatie *Bestelling* of niet (“tweemaal in de relatie zitten” kan niet). Het diagramfragment hieronder laat toe dat zowel “klant k_0 bestelt product p_0 op datum d_0 ” als ook “klant k_0 bestelt product p_0 op datum d_1 ” tegelijkertijd waar zijn (dus twee *verschillende* bestellingen waarbij dezelfde klant en product betrokken zijn):



Het volgende alternatief laat, net als de modeluitwerking, *niet* toe dat een klant een product tweemaal bestelt *tenzij* dat via verschillende vrijwilligers gebeurt:



Toelichting bij antwoord 3.

Motivatie voor de antwoorden (was niet gevraagd): zie Sectie 6.6.1 van het boek.

De score voor dit onderdeel is 3 punten als alle antwoorden correct zijn, en anders 0 punten.

Toelichting bij antwoord 4.

(0) R en of S als apart relatieschema wordt fout gerekend omdat er naar zo weinig mogelijk relatieschema's gevraagd wordt.

(1) Opname van B in A vereist $NULL$ s indien een A -entiteit geen B -entiteit is (en wordt daarom fout gerekend; zie de opgave).

(2) Een alternatief zou zijn om in A alleen de entiteiten te representeren die geen B -entiteit zijn, en in B dus alle B -entiteiten met al hun A -attributen:

A (a1, a2, primary key a1)

B (a1, a2, b1, b2, primary key a1)

In dit geval mag *geen* a1-waarde van B voorkomen in $A(a1)$, en moet dat aangegeven worden in het antwoord op de volgende vraag.

(3) Zonder de multipliciteit 0..1 bij relatie S zou het paar $(c1, d1)$ is een sleutel zijn van S . Vanwege de multipliciteit ("Voor elke $d : D$ is er hoogstens één $c : C$ zodanig dat $(c, d) \in S$ ") is " $c1$ door $d1$ bepaald", en is dus $(d1)$ op z'n eentje al sleutel van S . En omdat E en S identiek zijn, is $(d1)$ dus óók een sleutel van E .

(4) Zoiets als "voor alle $a : A$ is er hooguit één $b : B$ met $a.a1 = b.a1$ " wordt fout gerekend als 'eigenschap die nog niet in het databaseschema staat' omdat die eigenschap wél volgt uit het door ons gegeven databaseschema.

(5) Opname van B in E geeft redundantie: verschillende e 's kunnen naar eenzelfde b refereren en dan zouden de attribuutwaarden van die b gedupliceerd worden.

Toelichting bij antwoord 5.

Laat \mathbf{R} een relatieschema zijn. De "BCNF-voorwaarde" is gedefinieerd voor $X \rightarrow Y$ -beweringen die functionele afhankelijkheden van \mathbf{R} zijn. Dus wanneer $X \rightarrow Y$ geen functionele afhankelijkheid van \mathbf{R} is, is het twijfelachtig of je de BCNF-voorwaarde vervuld of onvervuld moet verklaren. Vandaar de vraagtekens in het antwoord; die staan precies bij de $X \rightarrow Y$ -beweringen die géén functionele afhankelijkheden van het gegeven relatieschema zijn.