

# Tentamen Gegevensbanken (19.211074.1) — 28 januari 2011

CONTROLEER EERST OF ALLE BLADZIJDEN T/M BLZ. 16 AANWEZIG ZIJN!

NAAM, VOORLETTERS: \_\_\_\_\_

STUDENTNUMMER: \_\_\_\_\_

OPLEIDING: \_\_\_\_\_

De uitwerkingen moeten op deze opgavenformulieren worden genoteerd in de daarvoor bestemde vakken. Alle overige ruimte kun je zo nodig als **kladpapier** gebruiken en wordt niet bekeken en niet beoordeeld.

[Reeds op Blackboard aangekondigd:] Bij het tentamen mogen geen boeken en dergelijke gebruikt worden behoudens **één dubbelzijdig gebruikt** vel van A4-formaat met daarop eigen aantekeningen of kopieën van delen van het boek; **kopieën van tentamenuitwerkingen en ander materiaal zijn niet toegestaan** (dat moet dan maar in eigen aantekeningen verwerkt worden).

Normering: per opgave staan de te behalen punten in de kantlijn en u krijgt 5 punten gratis; samen zijn dat 100 punten. Het eindcijfer is het aantal behaalde punten gedeeld door 10. Onleesbare tekst wordt steeds fout gerekend.

Na afloop moet de *volledige* set opgavenformulieren worden ingeleverd; het kladpapier niet. De tentamenopgaven zijn niet geheim en worden voorzien van modeluitwerkingen op Blackboard gepubliceerd. (Die modeluitwerkingen moet je op papier of elektronisch bij je hebben wanneer je je tentamen komt inzien.)

5 gratis	1	2	3	4	5	6	7	8	9	bonus	10
-------------	---	---	---	---	---	---	---	---	---	-------	----

10p.

**Opgave 1.** De casus van deze opgave betreft een ingenieursbureau. Vanuit het gezichtspunt van het bedrijf, wordt “de werkelijkheid” als volgt gekarakteriseerd:

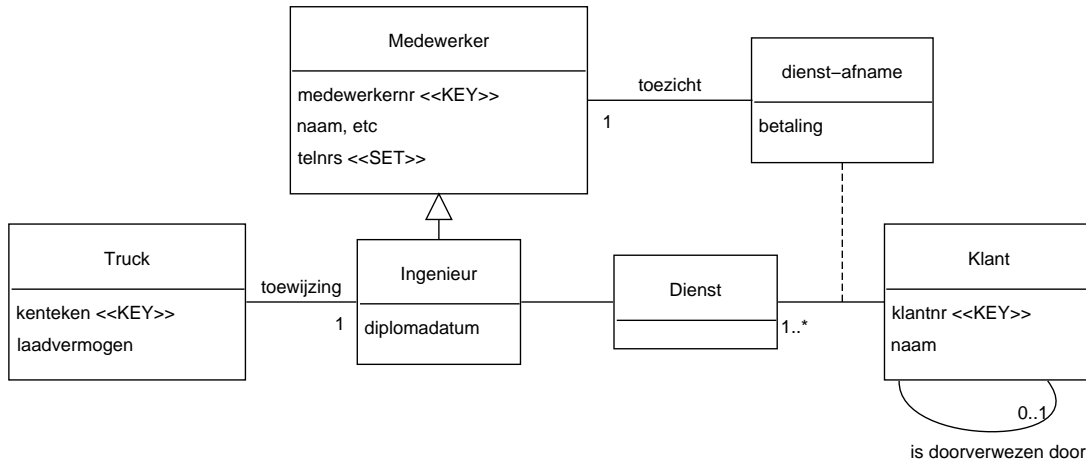
- (1) Het bureau heeft medewerkers waarvan er sommige ingenieur zijn. (2) Van een medewerker is het medewerkernummer bekend (uniek voor de medewerker), alsmede zijn naam en geboortjaar, en een stel telefoonnummers. (3) Van een ingenieur is bovendien de datum bekend waarop hij zijn ingenieursdiploma heeft behaald.
- (4) Om hun werk te verrichten gaan ingenieurs op pad met een truck; iedere truck is daarvoor toegewezen aan precies één ingenieur, maar aan sommige ingenieurs is misschien geen truck toegewezen. (5) Van een truck is het (unieke!) kenteken bekend, alsmede het laadvermogen.
- (6) Het werk dat de ingenieurs doen wordt ingedeeld in diensten: (7) iedere ingenieur levert nul of meer diensten, en iedere dienst wordt door nul of meer ingenieurs geleverd. (8) Er is tenminste één ingenieur die alle diensten levert.
- (9) Klanten nemen diensten af tegen een betaling. (10) De betaling hangt af van de afgenomen dienst én de afnemende klant. (11) Iedere klant neemt tenminste één dienst af, mogelijk meer. (12) Iedere klant heeft een klantnummer (uniek voor de klant), en naam.
- (13) Bij iedere afname van een dienst door een klant is er precies één medewerker die toezicht houdt (op die afname).
- (14) Soms krijgt het bureau een klant doordat een bestaande klant de andere op het bureau attent heeft gemaakt. (15) We spreken dan van doorverwijzing. (16) Een klant is door hooguit één andere klant doorverwezen.

Geef in het antwoordblok een Entity-Relationship diagram dat zoveel als mogelijk de volgende eigenschappen heeft:

- iedere instantie van het ERD beschrijft een mogelijke “werkelijkheid *op één tijdstip*”,
- iedere mogelijke “werkelijkheid *op één tijdstip*” kan gerepresenteerd worden als een instantie van het ERD,
- het ERD is opgebouwd met *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.* Doe het eerst op kladpapier en dan pas in het net.

Niet-geschreven multipliciteiten staan voor 0..\*.



(Zie Toelichting.)

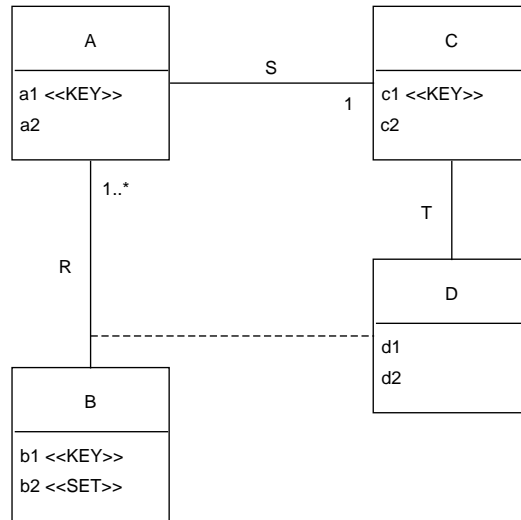
Hier is ruimte voor eventuele toelichting:

Wel in de casus maar niet in het ERD: (8) Er is tenminste één ingenieur die alle diensten levert.

Uit (8) volgt dat iedere dienst door tenminste één ingenieur geleverd wordt; dus mag de multipliciteit van de relatie tussen *Ingenieur* en *Dienst* aan de kant van *Ingenieur* beperkt worden tot 1..\*.

De casus impliceert niet dat multipliciteit van *toewijzing* aan de kant van *Truck* beperkt is tot 0..\*.

10p. **Opgave 2.** Beschouw het volgende ERD in de notatie van de UML:



Let op: attribuut  $b_2$  in  $B$  is heeft *SET*'s als waarde.

Vertaal het ER-diagram naar een databaseschema dat *precies* de informatie op kan slaan die past in het ER-diagram, en voldoet aan de volgende eigenschappen:

- er zijn geen NULLs nodig vanwege de vertaling,
- er wordt geen redundantie geïntroduceerd door de vertaling,
- alle attributen hebben atomaire waarden,

en verder, zoveel als mogelijk na vervulling van voorgaande eisen,

- er zijn zo *weinig mogelijk* tabelschema's.

Geef de tabelschema's in SQL syntaxis waarbij iedere domein-indicatie weggelaten mag worden; een voorbeeld van de vorm van zo'n schema is:

$X(x_1, \dots, \text{primary key } (x_i, x_j \dots), \text{foreign key } (x_m, x_n \dots) \text{ references } Y(y_1, y_2, \dots), \dots)$

Gebruik afkortingen, zoals "PK" voor "primary key", etc., en gebruik zonnodig ook CHECKs.

Onleesbare tekst wordt fout gerekend.

$A (a_1 \text{ PK}, a_2, c_1, \text{FK } (c_1) \text{ REF } C (c_1))$	-- $c_1$ representeert $S$ en de 1-multipliciteit
$B (b_1 \text{ PK}, \text{check } (b_1 \text{ in } (\text{select } b_1 \text{ from } D)))$	-- de check representeert de $1..*$ van $R$
$B' (b_1, b_2, \text{PK } (b_1, b_2), \text{FK } (b_1) \text{ REF } B (b_1))$	-- $B'$ representeert de SET van waarden per $b_1$
$C (c_1 \text{ PK}, c_2)$	
$D (a_1, b_1, d_1, d_2, \text{PK } (a_1, b_1), \text{FK } (a_1) \text{ REF } A (a_1), \text{FK } (b_1) \text{ REF } B (b_1))$	
$T (c_1, a_1, b_1, \text{PK } (c_1, a_1, b_1), \text{FK } (c_1) \text{ REF } C (c_1), \text{FK } (a_1, b_1) \text{ REF } D (a_1, b_1))$	
	-- let op: de FKs van $T$ zijn $(c_1)$ en $(a_1, b_1)$
(Relatie $R$ en $D$ zijn dezelfde relatie; ' $R$ ' en ' $D$ ' zijn synoniemen.) <span style="float: right;">(Zie Toelichting.)</span>	

10p.

**Opgave 3.** 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:

$$\bar{R} = ABCDE$$

$$\mathcal{F} = \{A \rightarrow D, B \rightarrow C, C \rightarrow B, CD \rightarrow A, D \rightarrow E\}$$

- (1) 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  $Y$ .
- (2) Omcirkel bij iedere  $X \rightarrow Y$  het deel  $X$  wanneer dat *sleutel* is van  $\mathbf{R}$ .
- (3) Onderstreep bij iedere  $X \rightarrow Y$  het deel  $X$  wanneer dat *supersleutel* is van  $\mathbf{R}$ .
- (4) Omcirkel de *nummers* van de functionele afhankelijkheden die een schending vormen van de BCNF-eis.

Onleesbare tekst wordt fout gerekend.

	<u><math>X</math></u>	$\rightarrow$	<u><math>Y</math></u>
1		$\rightarrow$	
2	<u><math>A</math></u>	$\rightarrow$	$DE$
3	<u><math>B</math></u>	$\rightarrow$	$C$
4	<u><math>C</math></u>	$\rightarrow$	$B$
5	<u><math>D</math></u>	$\rightarrow$	$E$
6	<u><math>E</math></u>	$\rightarrow$	
7	<u><math>AB</math></u>	$\rightarrow$	$CDE$
8	<u><math>AC</math></u>	$\rightarrow$	$BDE$
9	<u><math>AD</math></u>	$\rightarrow$	$E$
10	<u><math>AE</math></u>	$\rightarrow$	$D$
11	<u><math>BC</math></u>	$\rightarrow$	
12	<u><math>BD</math></u>	$\rightarrow$	$ACE$
13	<u><math>BE</math></u>	$\rightarrow$	$C$
14	<u><math>CD</math></u>	$\rightarrow$	$ABE$
15	<u><math>CE</math></u>	$\rightarrow$	$B$
16	<u><math>DE</math></u>	$\rightarrow$	
17	<u><math>ABC</math></u>	$\rightarrow$	$DE$
18	<u><math>ABD</math></u>	$\rightarrow$	$CE$
19	<u><math>ABE</math></u>	$\rightarrow$	$CD$
20	<u><math>ACD</math></u>	$\rightarrow$	$CD$
21	<u><math>ACE</math></u>	$\rightarrow$	$BD$
22	<u><math>ADE</math></u>	$\rightarrow$	
23	<u><math>BCD</math></u>	$\rightarrow$	$AE$

(Zie Toelichting.)

10p. **Opgave 4.** Beschouw het relatieschema  $\mathbf{R} = (ABCDE, \mathcal{F})$ , waarbij:

$$\mathcal{F} = \{AB \rightarrow D, CD \rightarrow A, B \rightarrow C, C \rightarrow B\}$$

**Let op:**  $E$  zit wel in  $\mathbf{R}$  maar komt niet voor in  $\mathcal{F}$ .

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:

	$X$	$\rightarrow$	$Y$		$X$	$\rightarrow$	$Y$
3	$B$	$\rightarrow$	$C$	19	$ABE$	$\rightarrow$	$CD$
4	$C$	$\rightarrow$	$B$	20	$ACD$	$\rightarrow$	$B$
7	$AB$	$\rightarrow$	$CD$	21	$ACE$	$\rightarrow$	$BD$
8	$AC$	$\rightarrow$	$BD$	23	$BCD$	$\rightarrow$	$A$
12	$BD$	$\rightarrow$	$AC$	25	$BDE$	$\rightarrow$	$AC$
13	$BE$	$\rightarrow$	$C$	26	$CDE$	$\rightarrow$	$AB$
14	$CD$	$\rightarrow$	$AB$	28	$ABCE$	$\rightarrow$	$D$
15	$CE$	$\rightarrow$	$B$	29	$ABDE$	$\rightarrow$	$C$
17	$ABC$	$\rightarrow$	$D$	30	$ACDE$	$\rightarrow$	$B$
18	$ABD$	$\rightarrow$	$C$	31	$BCDE$	$\rightarrow$	$A$

Geef alle functionele afhankelijkheden in  $\mathcal{F}$  die een schending vormen van de BCNF-conditie voor  $\mathbf{R}$ . Beargumenteer uw antwoord.

Onleesbare tekst wordt fout gerekend.

Schema  $\mathbf{R}$  heeft precies vier sleutels, namelijk:  $ABE, ACE, BDE, CDE$ . Voor iedere FD in  $\mathcal{F}$  geldt: die FD is een schending van de BCNF-conditie, omdat die niet-triviaal is en het linkerlid ervan geen sleutel omvat.

Geef een lossless decompositie van  $\mathbf{R}$  in precies twee schema's, zeg  $\mathbf{R}_1$  en  $\mathbf{R}_2$ , zodanig dat  $\mathbf{R}_1$  en  $\mathbf{R}_2$  samen minstens één schending minder hebben dan  $\mathbf{R}$ . Verklaar uw werkwijze en geef heel precies aan wat de attributen en functionele afhankelijkheden van  $\mathbf{R}_1$  en  $\mathbf{R}_2$  zijn.

Onleesbare tekst wordt fout gerekend.

Neem een FD uit  $\mathcal{F}$  die de BCNF-conditie schendt; bijvoorbeeld  $AB \rightarrow D$ . Splits  $\bar{R}$  in  $\bar{R}_1 = ABD$  (alle attributen van de FD) en  $\bar{R}_2 = ABCE$  (alles zonder de attributen uit het rechterlid van de FD). Neem  $\mathbf{R}_1 = (\bar{R}_1, \mathcal{F}_1)$  en  $\mathbf{R}_2 = (\bar{R}_2, \mathcal{F}_2)$ , waarbij  $\mathcal{F}_i$  een basis is voor de verzameling van FDs uit  $\mathcal{F}^+$  waarin uitsluitend attributen van  $\bar{R}_i$  voorkomen:

$$\mathbf{R}_1 = (ABD, \{AB \rightarrow D, BD \rightarrow A\})$$

$$\mathbf{R}_2 = (ABCE, \{B \rightarrow C, C \rightarrow B\})$$

NB1.  $BD \rightarrow A$  zit niet in  $\mathcal{F}$  maar wel in  $\mathcal{F}^+$  (zie regel 12 in de tabel) en komt dus in  $\mathcal{F}_1$ .

NB2.  $CD \rightarrow A$  zit weliswaar niet in  $\mathcal{F}_1 \cup \mathcal{F}_2$  maar wel in  $(\mathcal{F}_1 \cup \mathcal{F}_2)^+$  en gaat dus niet verloren.

NB3.  $\mathbf{R}_1$  staat in BCNF (de keys zijn  $AB$  en  $BD$ ) en  $\mathbf{R}_2$  bevat precies twee schendingen van de BCNF-conditie; samen dus minder dan de drie schendingen in  $\mathbf{R}$ . (Zie Toelichting.)

Construeer een lossless decompositie van  $\mathbf{R}$  tot schema's die ieder in BCNF staan. (U mag gebruik maken van en verwijzen naar de vorige antwoorden.) Verklaar iedere stap zodat het voor de corrector duidelijk is hoe u te werk gaat.

Onleesbare tekst wordt fout gerekend.

Let op:  $BD \rightarrow A$  volgt uit  $\mathcal{F}$  en zal dus (bij een correcte redenering) een FD worden van elke component waarin die attributen zitten. Net zo voor  $AC \rightarrow D$ .

We passen het BCNF-algoritme toe.

- De eerste stap is in de vorige vragen gedaan en levert bovengenoemde decompositie  $\{\mathbf{R}_1, \mathbf{R}_2\}$  van  $\mathbf{R}$  op. Schema  $\mathbf{R}_1$  staat in BCNF (de enige FD is een sleutel-beperking).

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

In  $\mathbf{R}_2$  zijn  $ABE$  en  $ACE$  de sleutels, dus  $B \rightarrow C$  en  $C \rightarrow B$  uit  $\mathcal{F}_2$  zijn beide een schending van de BCNF-conditie. We kiezen  $B \rightarrow C$  ter eliminatie. Dus splitsen we  $\bar{R}_2$  in  $\bar{R}_{2a} = BC$  en  $\bar{R}_{2b} = ABCE = ABE$ . Dit levert schema's  $\mathbf{R}_{2j} = (\bar{R}_{2j}, \mathcal{F}_{2j})$ ; hierin is  $\mathcal{F}_{2j}$  (een basis voor) de inperking van  $\mathcal{F}^+$  (of  $\mathcal{F}_2^+$ ) tot  $\bar{R}_{2j}$ . Dus

$$\mathbf{R}_{2a} = (BC, \{B \rightarrow C, C \rightarrow B\}) \text{ en}$$

$$\mathbf{R}_{2b} = (ABE, \{ \}).$$

Schema  $\mathbf{R}_{2a}$  en  $\mathbf{R}_{2b}$  staan beide in BCNF.

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

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

$$"ABCDE = ABD \bowtie BC \bowtie ABE \text{ geldt in } \mathbf{R}."$$

- 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.

(Zie Toelichting.)

Zijn alle functionele afhankelijkheden uit  $\mathcal{F}$ , van het oorspronkelijke schema  $\mathbf{R}$ , behouden onder bovenstaande decompositie van de oorspronkelijke  $\mathbf{R}$ ?

Onleesbare tekst wordt fout gerekend.

Ja; alle functionele afhankelijkheden zijn behouden (zie NB2 in het tweede antwoordblok).

10p.



**Opgave 5.** Bij een wielervedstrijd spelen verscheidene entiteiten een rol: ploeg, renner, sponsor, volgauto, etcetera. We definiëren een relatie  $\mathcal{R}$  die deze entiteiten aan elkaar relateert, op de volgende manier.

Een tuple  $(P, R, W, S, K, B)$  zit op tijdstip  $t$  in relatie  $\mathcal{R}$  precies wanneer op tijdstip  $t$  al het volgende geldt:

1.  $P$  is de Ploegnaam van een ploeg die aan de wedstrijd deelneemt.
2.  $R$  is het Rugnummer van een wielrenner uit de ploeg met naam  $P$ .
3.  $W$  is de naam van de Wielrenner met rugnummer  $R$ .
4.  $S$  is een Sponsor van de ploeg met naam  $P$ .
5.  $K$  is het Kenteken van een volgauto van de ploeg met naam  $P$ .
6.  $B$  is het Bedrag dat sponsor  $S$  aan wielrenner  $R$  geeft in deze wedstrijd.

Neem het volgende aan:

- a. Verschillende ploegen hebben verschillende ploegnamen.
- b. Een rugnummer identificeert een wielrenner uniek.
- c. Een ploeg kan verschillende sponsors hebben.

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 met verwijzing naar 1..6 en a..c. (De motivatie telt even zwaar mee in de beoordeling als het antwoord W/O.)

Onleesbare tekst wordt fout gerekend.

FD	W/O	Motivatie voor uw keuze
$R \rightarrow P$	W	Volgt uit 2&b
$W \rightarrow P$	O	Verschillende renners (uit verschillende ploegen) kunnen gelijknamig zijn.
$P \rightarrow R$	O	Een ploeg kan verscheidene wielrenners (dus rugnummers) hebben.
$K \rightarrow P$	W	Volgt uit 5.
$P \rightarrow K$	O	Een ploeg kan verscheidene volgauto's hebben.
$BS \rightarrow R$	O	Een sponsor kan aan verschillende renners eenzelfde bedrag geven.
$SR \rightarrow B$	W	Volgt uit 6&b.
$B \rightarrow SR$	O	Er geldt al $B \not\rightarrow R$ : Verschillende renners kunnen eenzelfde bedrag krijgen.
$RB \rightarrow S$	O	Verschillende sponsors kunnen een renner eenzelfde bedrag geven.

Geef voor ieder van de volgende MultiValued Dependencies aan of die Waar of Onwaar is in relatie  $\mathcal{R}$ , en geef zo mogelijk een korte motivatie:

Onleesbare tekst wordt fout gerekend.

MVD	W/O	
$PRWSKB = PRWSK \bowtie SB$	O	$SR \rightarrow B$ wordt geschonden
$PRWSKB = PRWSK \bowtie SRB$	W	doorsnee der operanden ( $=SR$ ) is key in rechter operand
$PRWSKB = PRWSK \bowtie PSRB$	W	doorsnee der operanden ( $=PSR$ ) is superkey in rechter operand (Zie Toelichting.)



10p.

**Opgave 6.** Gegeven zijn twee transacties waarvan de executievolgorde van de individuele operaties er schematisch als volgt uit zien:

$T_1$ :  $read_1(x)$ ;  $write_1(x)$ ;  $read_1(y)$ ;  $write_1(y)$ .

$T_2$ :  $read_2(x)$ ;  $write_2(x)$ .

Hierbij staan  $x$  en  $y$  voor verschillende, al bestaande, rijen van een tabel uit de database. De *read*-operaties lezen de rij, en de *write*-operaties schrijven een nieuwe waarde in de gegeven rij.

Geef, indien mogelijk, een serialiseerbare schedule voor de executie van beide transacties die verschilt van “*eerst* alles van de een en *dan* alles van de ander”. (Als het niet mogelijk is, geef dan de reden daarvan.)

Onleesbare tekst wordt fout gerekend.

$read_1(x)$ ;  $write_1(x)$ ;  $read_2(x)$ ;  $write_2(x)$ ;  $read_1(y)$ ;  $write_1(y)$ .

Algemener:

een ‘verweving’ van  $T_1$  en  $T_2$  met volgorde ...  $write_1(x)$  ...  $read_2(x)$  ...  $write_1(y)$  ...

Geef, indien mogelijk, een niet-serialiseerbare schedule voor de executie van beide transacties. (Als het niet mogelijk is, geef dan de reden daarvan.)

Onleesbare tekst wordt fout gerekend.

Algemeen geformuleerde oplossing: een ‘verweving’ van  $T_1$  en  $T_2$  met één van deze volgordes:

...  $read_2(x)$  ...  $write_1(x)$  ...  $write_2(x)$  ... of ...  $read_1(x)$  ...  $write_2(x)$  ...  $write_1(x)$  ...

Geef, indien mogelijk, een schedule voor een executie van beide transacties zó dat  $T_1$  een *phantom* ervaart. (Als het niet mogelijk is, geef dan de reden daarvan.)

Onleesbare tekst wordt fout gerekend.

Niet mogelijk; daarvoor moeten er (door  $T_2$ ) nieuwe rijen *toegevoegd* kunnen worden, én moet

$T_1$  tweemaal “collecties” van rijen kunnen lezen (i.p.v. alleen maar “uniek bepaalde” rijen).

Geef, indien mogelijk, een schedule voor een executie van beide transacties zó dat  $T_1$  een *dirty read* doet. (Als het niet mogelijk is, geef dan de reden daarvan.)

Onleesbare tekst wordt fout gerekend.

$read_2(x)$ ;  $write_2(x)$ ;  $read_1(x)$ ; ... abort van  $T_2$ ; ... commit van  $T_1$

We willen de gelijktijdige executie van transacties  $T_1$  en  $T_2$  zo min mogelijk beperken, maar wel zó dat  $T_1$  geen *dirty reads* zal doen. Welke transactie(s) moeten we dan op welk isolatieniveau zetten? (Herinner je dat de isolatieniveaus, in opklimmende sterkte van beperkingen, de volgende zijn: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE.)

Onleesbare tekst wordt fout gerekend.

Zet het isolatieniveau van  $T_1$  op READ COMMITTED.

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 klasse zijn: het land (*country*), het aantal kanonnen (*guns*), de diameter in centimeters van de kanonloop (*bore*), en de waterverplaatsing (*displacement*, gemeten in tonnen). Van een schip 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).

Wanneer we spreken van het *type* van een schip, dan bedoelen we het *type* van de klasse van dat schip; net zo voor de attributen *country*, *guns*, *bore*, *displacement*. Dus alle schepen van een klasse komen uit één land: het land dat in de klasse genoemd staat.

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*)

10p. **Opgave 7.** Beschouw de volgende zoekopdracht:

Geef ieder schip dat in een zeeslag is gezonken.

Geef voor deze vraag een **afleiding in kleine stappen** in Verzamelingsnotatie naar een vorm die dicht aansluit bij een SQL query die **geen subqueries** heeft en **zo weinig mogelijk tabellen** in de from clause. Het begin is al gegeven. Kort identifiërs tot hun eerste letter af.

Onleesbare tekst wordt fout gerekend.

	“ieder schip dat in een zeeslag zinkt”
=	$\{s : S \mid \text{“}s \text{ zinkt in een zeeslag”} \bullet s.n\}$
=	$\{s : S \mid (\exists b : B \bullet \text{“}s \text{ zinkt in } b\text{”}) \bullet s.n\}$
=	$\{s : S \mid (\exists b : B \bullet (\exists o : O \mid o.s=s.n \wedge o.b=b.n \bullet o.r=sunk)) \bullet s.n\}$
=	[logica]
	$\{s : S \mid (\exists b : B; o : O \mid o.s=s.n \wedge o.b=b.n \bullet o.r=sunk) \bullet s.n\}$
=	[shunting]
	$\{s : S \mid (\exists o : O \mid o.s=s.n \wedge (\exists b : B \bullet o.b=b.n) \bullet o.r=sunk) \bullet s.n\}$
=	[in $O$ is $b$ een foreign key naar $B(n)$ ; dus voor iedere $o$ geldt: $\exists b : B \bullet o.b=b.n$ ]
	$\{s : S \mid (\exists o : O \mid o.s=s.n \bullet o.r=sunk) \bullet s.n\}$
=	[shunting]
	$\{s : S; o : O \mid o.s=s.n \wedge o.r=sunk \bullet s.n\}$
=	[substitutie van gelijken voor gelijken (als voorbereiding voor de volgende stap)]
	$\{s : S; o : O \mid o.s=s.n \wedge o.r=sunk \bullet o.s\}$
=	[shunting]
	$\{o : O \mid (\exists s : S \bullet o.s=s.n) \wedge o.r=sunk \bullet o.s\}$
=	[in $O$ is $s$ een foreign key naar $S(n)$ ; dus voor iedere $o$ geldt: $\exists s : S \bullet o.s=s.n$ ]
	$\{o : O \mid o.r=sunk \bullet o.s\}$

Geef een SQL formulering van de beschouwde vraag; de SQL formulering moet dicht aansluiten bij de zojuist gegeven uitdrukking (en dus geen subqueries hebben en zo weinig mogelijk tabellen in de from clause). Gebruik *DISTINCT* alleen wanneer het nodig is.

Onleesbare tekst wordt fout gerekend.

select distinct $o.s$ from $O$ $o$ where $o.r = sunk$
‘Distinct’ is niet nodig wanneer in de database de eigenschap geldt dat een schip maar één keer kan zinken. Deze eigenschap is niet gegeven, maar is wel realistisch en aannemelijk.

Geef ook een formulering van de vraag in Relationele Algebra:

Onleesbare tekst wordt fout gerekend.

$\pi_n (\sigma_{r=\text{sunk}} O)$

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:

Wat is, voor iedere zeeslag na het jaar 1600 waarbij tenminste drie schepen zijn *betrokken*, de naam van die zeeslag en het aantal in-die-zeeslag-gezonken schepen?

Onleesbare tekst wordt fout gerekend.

Definieer "de groep voor  $b$ " als: (1) de schepen die in  $b$  *gezonken* zijn (dus de 'count van de groep' komt in de select expressie), of: (2) de schepen die in  $b$  *betrokken* zijn (dus de 'count van de groep' komt in de having conditie). Hieronder staat (1) uitgewerkt.

select  $b.name$ , count (\*)

from  $B$   $b$ ,  $O$   $o$

where  $b.date \geq 1600$  and  $3 \leq$  (select count (\*) from  $O$   $o'$  where  $o'.battlename = b.name$ )

and  $b.name = o.battlename$  and  $o.result = \text{sunk}$

group by  $b.name$

(Zie Toelichting.)

5p.

**Opgave 9.** (Goede beantwoording levert 5 bonuspunten boven op de punten die voor deze opgave gegeven worden. Daardoor kan het totaal aantal behaalde punten boven de 100 uitkomen.) Beschouw de volgende vraag:

Van welke landen zijn *alle* schepen gezonken?

Formuleer de vraag geheel in verzamelingsnotatie op een manier die zo direct mogelijk aansluit bij de gegeven formulering van de vraag.

Onleesbare tekst wordt fout gerekend.

$$\begin{aligned} & \{c : C \mid (\forall s : S \mid \text{"s is van land } c.c" \bullet \text{"s is gezonken"}) \bullet c.c\} \\ = & \{c : C \mid (\forall s : S \mid (\exists c' : C \bullet s.c = c'.n \wedge c'.c = c.c) \bullet \text{"s is gezonken"}) \bullet c.c\} \\ = & \{c : C \mid (\forall s : S \mid (\exists c' : C \bullet s.c = c'.n \wedge c'.c = c.c) \bullet (\exists o : O \mid o.r = sunk \bullet o.s = s.n)) \bullet c.c\} \\ & \text{(Alleen de laatste regel werd gevraagd.)} \qquad \qquad \qquad \text{(Zie Toelichting.)} \end{aligned}$$

Formuleer de vraag in verzamelingsnotatie op een manier die zo direct mogelijk aansluit bij de SQL query die u zo dadelijk gaat geven.

Onleesbare tekst wordt fout gerekend.

$$\begin{aligned} = & \text{[shunting]} \\ & \{c : C \mid (\forall s : S; c' : C \mid s.c = c'.n \wedge c'.c = c.c \bullet (\exists o : O \mid o.r = sunk \bullet o.s = s.n)) \bullet c.c\} \\ = & \{c : C \mid \neg (\exists s : S; c' : C \mid s.c = c'.n \wedge c'.c = c.c \bullet \neg (\exists o : O \mid o.r = sunk \bullet o.s = s.n)) \bullet c.c\} \\ = & \{c : C \mid \neg (\exists s : S; c' : C \bullet s.c = c'.n \wedge c'.c = c.c \wedge \neg (\exists o : O \mid o.r = sunk \bullet o.s = s.n)) \bullet c.c\} \\ & \text{(Alleen de laatste of een-na-laatste regel werd gevraagd.)} \end{aligned}$$

Formuleer de vraag in SQL, zonder overbodige subqueries en tabellen.

Onleesbare tekst wordt fout gerekend.

$$\begin{aligned} = & \text{select distinct } c.c \text{ from } C \text{ } c \text{ where not exists (} \\ & \text{select * from } S \text{ } s, C \text{ } c1 \text{ where } s.c = c1.n \text{ and } c1.c = c.c \text{ and not exists (} \\ & \text{select * from } O \text{ } o \text{ where } o.r = sunk \text{ and } o.s = s.n \text{ ) )} \\ & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{(Zie Toelichting.)} \end{aligned}$$

10p. **Opgave 10.** Beschouw het volgende databaseschema:



*Person* (*id*, *name*)  
*Student*(*id*, *university*)

De bedoeling is dat iedere student ook een persoon is. Daarom willen we dat voortdurend de volgende acties worden uitgevoerd:

- (a) Wanneer een rij wordt verwijderd uit tabel *Person*, dan wordt de rij met hetzelfde *id* ook verwijderd uit tabel *Student*.
- (b) Wanneer er gepoogd wordt een rij toe te voegen aan tabel *Student*, dan wordt getest of er een rij met hetzelfde *id* bestaat in *Person*; is dat niet het geval, dan vindt de toevoeging niet plaats.

Voeg key constraints toe aan het schema waardoor de gewenste acties automatisch door het databasesysteem worden uitgevoerd. Voeg geen overbodige constraints toe.

Ter herinnering, key constraints zien er als volgt uit:

- primary key ( $x, x', \dots$ )
- foreign key ( $x, x', \dots$ ) references  $T(y, y', \dots)$  on *event* action on *event'* action' ...

Hierbij staat  $x$  voor een attribuutnaam,  $T$  voor een tabelnaam, *event* voor een gebeurtenis (*delete*, *update*) en *action* voor een actie (*set null*, *set default*, *no action*, *cascade*).

Onleesbare tekst wordt fout gerekend.

<i>Person</i> ( <i>id</i> , <i>name</i> , primary key <i>id</i> )
<i>Student</i> ( <i>id</i> , <i>name</i> , foreign key ( <i>id</i> ) references <i>Person</i> ( <i>id</i> ) on delete cascade)
Het deel 'on delete cascade' zorgt voor actie (a),
het deel 'foreign key( <i>id</i> ) references <i>Person</i> ( <i>id</i> )' zorgt voor actie (b).
Er werd niet gevraagd om in <i>Student</i> de <i>id</i> tot primary key te verklaren, en ook niet om de actie die opgeroepen wordt door 'on update cascade' toe te voegen aan de foreign key constraint.

Geef nu, voor het oorspronkelijk gegeven schema, een of meer triggers waardoor de gewenste acties (a) en (b) automatisch door het databasesysteem worden uitgevoerd.

Ter herinnering, de syntaxis van een trigger creatie luidt als volgt:

```
create trigger trigger-name
  {before | after} {insert | delete | update [ of column-name-list ] } on table-name
  [ referencing [ old as var-to-refer-to-old-tuple ]
    [ new as var-to-refer-to-new-tuple ]
    [ old table as name-to-refer-to-old-table ]
    [ new table as name-to-refer-to-new-table ] ]
  [ for each { row | statement } ]
  [ when (precondition) ]
  statement-list
```

Hierbij staat  $\{x \mid \dots\}$  voor een keuze uit  $x, \dots$ ; en  $[x]$  staat voor een keuze uit  $x$  of niets.

Onleesbare tekst wordt fout gerekend.

```
create trigger OnPersonDelete
```

```
  after delete on Person
```

```
  referencing old as DeletedPerson
```

```
  for each row
```

```
  delete from Student s where s.id = DeletedPerson.id;
```

```
create trigger OnStudentInsert
```

```
  before insert on Student
```

```
  referencing new as NewStudent
```

```
  for each row
```

```
  when (NewStudent.id not in (select id from Person))
```

```
  abort;
```

Varianten met statement level granularity ('for each statement', de default) zijn ook mogelijk.

(Zie Toelichting.)

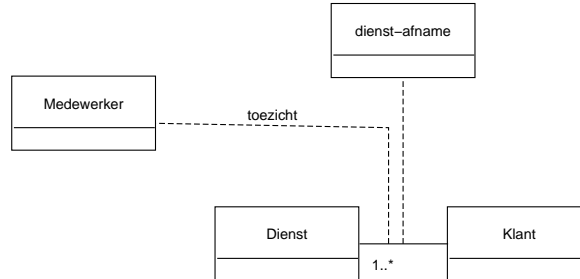




## Toelichtingen

### Toelichting bij antwoord 1.

Een veel gemaakte ernstige fout is de volgende:



Hierin zijn nu zowel *dienst-afname* als ook *Medewerker* indientiek aan de relatie tussen *Dienst* en *Klant*.

### Toelichting bij antwoord 2.

**Beoordelingsrichtlijn.** 2 punt aftrek voor foute modellering van de 1..\*-multipliciteit, en voor foute modellering van de 1-multipliciteit; 3 punt aftrek voor foute modellering van de associatie-entiteit  $R (= D)$  (waarvan 2 punten voor foute keuze van de key voor deze associatie-entiteit); 2 punten aftrek voor foute representatie van de SET waarde in  $B$ .

### Toelichting bij antwoord 3.

De functionele afhankelijkheden met een leeg rechterkant zijn triviaal en schenden de BCNF-eis dus niet. De functionele afhankelijkheden met een sleutel in het linkerlid zijn ook geen schending van de BCNF-eis.

Iedere key is ook superkey.

**Beoordelingsrichtlijn.** 4 punt voor deel (1) (1 punt aftrek voor iedere fout), 2 punt voor ieder van (2), (3), (4).

### Toelichting bij antwoord 4.

Dat het aantal schendingen in  $\mathbf{R}_1$  en  $\mathbf{R}_2$  samen minder is dan in  $\mathbf{R}$  volgt ook uit het feit dat hier één iteratiestap van het BCNF-algoritme is toegepast.

Een andere compositie wordt verkregen door, bijvoorbeeld,  $B \rightarrow C$  te kiezen ter eliminatie: Splits  $\bar{R}$  in  $\bar{R}_1 = BC$  (alle attributen van de FD) en  $\bar{R}_2 = ABDE$  (alles zonder de attributen uit het rechterlid van de FD). Neem  $\mathbf{R}_1 = (\bar{R}_1, \mathcal{F}_1)$  en  $\mathbf{R}_2 = (\bar{R}_2, \mathcal{F}_2)$ , waarbij  $\mathcal{F}_i$  een basis is voor de verzameling van FDs uit  $\mathcal{F}^+$  waarin uitsluitend attributen van  $\bar{R}_i$  voorkomen:

$$\mathbf{R}_1 = (BC, \{B \rightarrow C, C \rightarrow B\})$$

$$\mathbf{R}_2 = (ABDE, \{AB \rightarrow D, BD \rightarrow A\})$$

NB1.  $BD \rightarrow A$  zit niet in  $\mathcal{F}$  maar wel in  $\mathcal{F}^+$  (zie bijv. regel 12 in de tabel) en komt dus in  $\mathcal{F}_2$ .

NB2.  $CD \rightarrow A$  zit weliswaar niet in  $\mathcal{F}_1 \cup \mathcal{F}_2$  maar wel in  $(\mathcal{F}_1 \cup \mathcal{F}_2)^+$  en gaat dus niet verloren.

NB3.  $\mathbf{R}_1$  staat in BCNF (de keys zijn  $B$  en  $C$ ) en  $\mathbf{R}_2$  bevat precies twee schendingen van de BCNF-conditie; samen dus minder dan de drie schendingen in  $\mathbf{R}$ .

### Toelichting bij antwoord 4.

**Beoordelingsrichtlijn.** 2 punt voor deel 1, 3 punt voor deel 2, 4 punt voor deel 3, 1 punt voor deel 4.

### **Toelichting bij antwoord 5.**

“FD  $F$  wordt geschonden” betekent: er zijn instanties/tabelvullingen mogelijk (die niet in tegenpraak zijn met  $F$ ) waarbij de join van de componenten een instantie/tabelvulling oplevert die de FD  $F$  tegenspreekt. Wanneer dit het geval is, kan de join dus niet gelijk zijn aan de oorspronkelijke instantie/tabelvulling.

Merk op dat “FD  $F$  wordt geschonden” (Engels: violation) iets anders is dan “FD  $F$  blijft behouden” (Engels: preservation); het laatste betekent dat de gezamenlijke FDs van de componenten de  $F$  tot gevolg hebben. (NB: het BCNF-algoritme zal ALTIJD een decompositie opleveren waarvan de join de oorspronkelijke tabelvulling is, en dus wordt geen van de FDs geschonden.)

**Beoordelingsrichtlijn.** 8 punt voor deel 1 (1 punt aftrek voor ieder fout antwoord W/O en voor iedere foute motivatie); 2 punt voor deel 2 (alleen te krijgen als alles goed is).

### **Toelichting bij antwoord 8.**

De uitkomst wijzigt niet wanneer van de ‘count (\*)’ één of beide vervangen wordt door ‘count ( $o.shipname$ )’ of ‘count (distinct  $o.shipname$ )’. De reden is dat een rij  $o$  in  $O$  bij vaste  $o.battlename(=b.name)$  uniek bepaald wordt door de waarde van  $o.shipname$  omdat het paar ( $o.battlename, o.shipname$ ) een key is van  $O$ .

Hieronder staat (2) uitgewerkt: de groep voor  $b$  bestaat uit de schepen die in  $b$  betrokken zijn.

```
select b.name,  
       (select count (*) from O o' where o'.b = b.n and o'.r =sunk)  
from B b, O o  
where b.nane = o.battlename and b.date ≥ 1600  
group by b.name  
having 3 ≤ count (*)
```

### **Toelichting bij antwoord 9.**

We hebben  $c.c (= c.country)$  gebruikt als bron van alle landen. Dit had eventueel ook via een omweg *afgeleid* kunnen worden:

$$\begin{aligned} & \{l : Text \mid “l \text{ is een land}” \wedge (\forall s : S \mid “s \text{ is van land } l” \bullet “s \text{ is gezonken}”) \bullet l\} \\ = & \quad [c.c \text{ is de bron van alle landen}] \\ & \{l : Text \mid (\exists c : C \bullet l = c.c) \wedge (\forall s : S \mid “s \text{ is van land } l” \bullet “s \text{ is gezonken}”) \bullet l\} \\ = & \quad [\text{shunting}] \\ & \{l : Text; c : C \mid l = c.c \wedge (\forall s : S \mid “s \text{ is van land } l” \bullet “s \text{ is gezonken}”) \bullet l\} \\ = & \quad [\text{substitutie van gelijken voor gelijken; shunting}] \\ & \{l : Text; c : C \mid l = c.c \wedge (\forall s : S \mid “s \text{ is van land } c.c” \bullet “s \text{ is gezonken}”) \bullet c.c\} \\ = & \quad [\text{shunting}] \\ & \{c : C \mid (\exists l : Text \bullet l = c.c) \wedge (\forall s : S \mid “s \text{ is van land } c.c” \bullet “s \text{ is gezonken}”) \bullet c.c\} \\ = & \quad [“éénpuntsregel” voor \exists] \\ & \{c : C \mid c.c \in Text \wedge (\forall s : S \mid “s \text{ is van land } c.c” \bullet “s \text{ is gezonken}”) \bullet c.c\} \\ = & \quad [\text{aanname over domein van } Class.country] \\ & \{c : C \mid (\forall s : S \mid “s \text{ is van land } c.c” \bullet “s \text{ is gezonken}”) \bullet c.c\} \end{aligned}$$

### **Toelichting bij antwoord 9.**

Een alternatieve uitwerking, tot de vorm (“alle landen” minus “landen met een-of-meer niet-

gezonken schepen”), kan als volgt worden afgeleid:

$$\begin{aligned}
& \{c:C \mid \neg (\exists s:S; c':C \bullet s.c = c'.n \wedge c'.c = c.c \wedge (\forall o:O \mid o.r = sunk \bullet o.s \neq s.n)) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists s:S; c':C \bullet s.c = c'.n \wedge c'.c = c.c \wedge s.n \notin \{o:O \mid o.r = sunk \bullet o.s\}) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists s:S; c':C \mid s.c = c'.n \wedge s.n \notin \{o:O \mid o.r = sunk \bullet o.s\}) \bullet c'.c = c.c \bullet c.c\} \\
= & \{c:C \mid c.c \notin \{s:S; c':C \mid s.c = c'.n \wedge s.n \notin \{o:O \mid o.r = sunk \bullet o.s\}) \bullet c'.c\} \bullet c.c\} \\
= & \{c:C \mid c.c\} \setminus \{s:S; c':C \mid s.c = c'.n \wedge s.n \notin \{o:O \mid o.r = sunk \bullet o.s\}) \bullet c.c\} \\
= & (\text{select } c.c \text{ from } C \text{ c}) \\
& \text{EXCEPT} \\
& (\text{select } c.c \text{ from } S \text{ s, } C \text{ c where } s.c=c.n \text{ and } s.n \text{ not in } (\text{select } o.s \text{ from } O \text{ o where } o.r=sunk))
\end{aligned}$$

Veel gegeven fout antwoord:

$$\begin{aligned}
& \{c:C \mid (\forall s:S \mid “s \text{ is van klasse } c” \bullet “s \text{ is gezonken}”) \bullet c.c\} \\
= & \{c:C \mid (\forall s:S \mid s.c = c.n \bullet “s \text{ is gezonken}”) \bullet c.c\} \\
= & \text{landen van de klassen waarvan alle schepen gezonken zijn.}
\end{aligned}$$

Voor de liefhebber — Andere foute antwoorden:

$$\begin{aligned}
& \{c:C; s:S; o:O \mid c.n = s.c \wedge s.n = o.s \wedge \dots \bullet c.c\} \\
= & \text{landen met een schip dat aan een zeeslag heeft deelgenomen en } \dots
\end{aligned}$$

$$\begin{aligned}
& \{c:C; s:S \mid c.n = s.c \wedge (\forall o:O \mid o.s = s.n \bullet o.r = sunk) \bullet c.c\} \\
= & \{c:C; s:S \mid c.n = s.c \wedge \neg (\exists o:O \mid o.s = s.n \bullet o.r \neq sunk) \bullet c.c\} \\
= & \text{landen met een schip dat iedere zeeslag waaraan het heeft deelgenomen, heeft overleefd.}
\end{aligned}$$

$$\begin{aligned}
& \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \forall o:O \bullet s.n \neq o.s \vee o.r \neq sunk) \bullet c.c\} \\
= & \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \forall o:O \mid s.n = o.s \bullet o.r \neq sunk) \bullet c.c\} \\
= & \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \neg \exists o:O \mid s.n = o.s \bullet o.r = sunk) \bullet c.c\} \\
= & \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \neg \exists o:O \bullet s.n = o.s \wedge o.r = sunk) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists s:S \mid c.n = s.c \bullet \exists o:O \bullet s.n = o.s \wedge o.r = sunk) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists s:S; o:O \bullet c.n = s.c \wedge s.n = o.s \wedge o.r = sunk) \bullet c.c\} \\
= & \text{landen van klassen waarvan alle schepen niet gezonken zijn.}
\end{aligned}$$

$$\begin{aligned}
& \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \forall o:O \bullet s.n \neq o.s \vee o.r = sunk) \bullet c.c\} \\
= & \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \forall o:O \mid s.n = o.s \bullet o.r = sunk) \bullet c.c\} \\
= & \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \neg \exists o:O \mid s.n = o.s \bullet o.r \neq sunk) \bullet c.c\} \\
= & \{c:C \mid (\forall s:S \mid c.n = s.c \bullet \neg \exists o:O \bullet s.n = o.s \wedge o.r \neq sunk) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists s:S \mid c.n = s.c \bullet \exists o:O \bullet s.n = o.s \wedge o.r \neq sunk) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists s:S; o:O \bullet c.n = s.c \wedge s.n = o.s \wedge o.r \neq sunk) \bullet c.c\} \\
= & \text{landen van klassen die geen zeeslag-overlevende schepen hebben} \\
= & \text{landen van klassen waarvan ieder schip gezonken is in iedere zeeslag waaraan het deelgenomen heeft.}
\end{aligned}$$

landen waarvan ieder schip gezonken is

$$\begin{aligned}
& \subset \text{landen waarvan ieder schip gezonken is in iedere zeeslag waaraan het deelgenomen heeft} \\
= & \{c:C \mid (\forall s:S \mid (\exists c':C \bullet c.c = c'.c \wedge c'.n = s.c) \bullet \forall o:O \mid s.n = o.s \bullet o.r = sunk) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists s:S \mid (\exists c':C \bullet c.c = c'.c \wedge c'.n = s.c) \bullet \neg \forall o:O \mid s.n = o.s \bullet o.r = sunk) \bullet c.c\} \\
= & \{c:C \mid \neg (\exists c':C; s:S \mid c.c = c'.c \wedge c'.n = s.c \bullet \neg \forall o:O \mid s.n = o.s \bullet o.r = sunk) \bullet c.c\}
\end{aligned}$$

$= \{c : C \mid \neg (\exists c' : C; s : S \mid c.c = c'.c \wedge c'.n = s.c \bullet \exists o : O \mid s.n = o.s \bullet o.r \neq \text{sunken}) \bullet c.c\}$   
 $= \{c : C \mid \neg (\exists c' : C; s : S \bullet c.c = c'.c \wedge c'.n = s.c \wedge \exists o : O \mid s.n = o.s \bullet o.r \neq \text{sunken}) \bullet c.c\}$   
 $= \{c : C \mid \neg (\exists c' : C; s : S; o : O \bullet c.c = c'.c \wedge c'.n = s.c \wedge s.n = o.s \wedge o.r \neq \text{sunken}) \bullet c.c\}$   
 $= \{c : C \mid c.c \notin \{c' : C; s : S; o : O \mid c'.n = s.c \wedge s.n = o.s \wedge o.r \neq \text{sunken} \bullet c'.n\} \bullet c.c\}$   
 $= \{c : C \bullet c.c\} \setminus \{c' : C; s : S; o : O \mid c'.n = s.c \wedge s.n = o.s \wedge o.r \neq \text{sunken} \bullet c'.n \bullet c.c\}$   
 = alle landen met uitzondering van de landen met een schip dat een zeeslag overleefd heeft  
 = landen waarvan geen schip een zeeslag overleefd heeft  
 = landen waarvan ieder schip gezonken is in iedere zeeslag waaraan het deelgenomen heeft  
 $\subset$  landen waarvan ieder schip gezonken is of niet deelgenomen heeft aan een zeeslag.

$\{c : C \mid \neg (\exists c' : C; s : S; o : O \bullet c.c = c'.c \wedge c'.n = s.c \wedge s.n = o.s \wedge o.r = \text{sunken}) \bullet c.c\}$   
 $= \{c : C \mid c.c \notin \{c' : C; s : S; o : O \mid c'.n = s.c \wedge s.n = o.s \wedge o.r = \text{sunken} \bullet c'.n\} \bullet c.c\}$   
 $= \{c : C \bullet c.c\} \setminus \{c' : C; s : S; o : O \mid c'.n = s.c \wedge s.n = o.s \wedge o.r = \text{sunken} \bullet c'.n \bullet c.c\}$   
 = alle landen met uitzondering van de landen met (minstens) een gezonken schip  
 = landen waarvan geen schip gezonken is.

### ***Toelichting bij antwoord 10.***

Varianten met statement level granularity (de default):

```

create trigger OnPersonDelete'
  after delete on Person
  referencing old table as DeletedPersons
  for each statement
  delete from Student s where s.id in (select id from DeletedPersons);
create trigger OnStudentInsert'
  before insert on Student
  referencing new table as NewStudents
  for each statement
  when (exists ((select id from NewStudents) except (select id from Person)))
  abort;
  
```