

Tentamen Gegevensbanken (211074) — 1 februari 2008

CONTROLEER EERST OF ALLE BLADZIJDEN T/M BLZ. 14 AANWEZIG ZIJN!

Vul het tentamenbriefje volledig in, zódanig dat BEIDE DOORSLAGEN goed leesbaar zijn.

NAAM, VOORLETTERS: _____

STUDENTNUMMER: _____

OPLEIDING: _____

De uitwerkingen moeten op deze opgavenformulieren worden genoteerd in de daarvoor bestemde vakken. Alle overige ruimte en de achterkanten van de bladen kun je zo nodig als kladpapier gebruiken en wordt niet bekeken en niet beoordeeld. Gebruik van ander kladpapier is toegestaan.

Het gebruik van boeken, dictaten en dergelijke is *niet* toegestaan, behoudens één vel van A4 formaat met *eigen* aantekeningen (dubbelzijdig, getypt of geschreven) of kopieën van *delen van het boek*; kopieën van ander materiaal (zoals tentamenuitwerkingen) zijn niet toegestaan.

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 Teletop 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	8bonus	9	10
-------------	---	---	---	---	---	---	---	---	--------	---	----

10p.

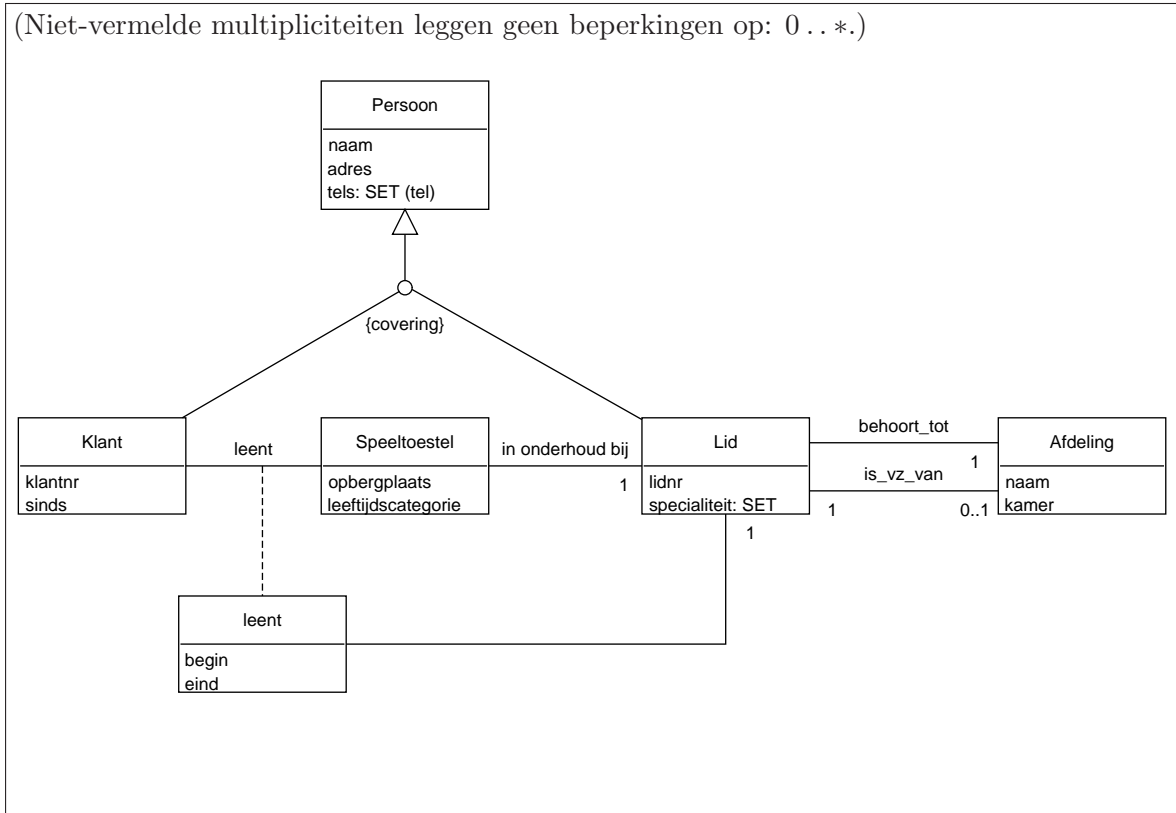
Opgave 1. De speeltuinvereniging *Altijd Kind* bezit een grote verzameling speeltoestellen die uitgeleend worden aan buurtverenigingen voor straatfeesten en dergelijke. Hieronder staat een specificatie van de gegevens die relevant zijn voor de werkzaamheden van *Altijd Kind*:

- Leden van de speeltuinvereniging worden geïdentificeerd met een lidnummer; ieder lid heeft een stel specialiteiten (zoals 'houtbewerking', 'rekenaar', etc.). Ieder lid behoort tot precies één afdeling van *Altijd Kind*.
- Van een afdeling is de naam bekend alsmede de kamer waar de afdeling zetelt.
- Elke afdeling heeft een lid van *Altijd Kind* als voorzitter; een voorzitter behoort tot de afdeling die hij voorziet.
- Van ieder speeltoestel is de opbergplaats bekend en de leeftijdscategorie.
- Klanten kunnen speeltoestellen lenen. Klanten worden geïdentificeerd met een klantnummer; van hun wordt ook bijgehouden sinds wanneer ze klant zijn bij *Altijd Kind*.
- Ieder speeltoestel is in onderhoud bij één lid van *Altijd Kind*.
- Per uitlening van een speeltoestel aan een klant is de begin- en einddatum van de uitleningstermijn bekend, alsmede het lid dat betrokken is in die uitlening.
- Per klant en lid wordt ook de naam, het adres en een stel telefoonnummers bijgehouden. Een lid kan ook klant zijn, maar krijgt dan geen speciale privileges en wordt in die hoedanigheid net zo behandeld als iedere andere klant.

Geef in het antwoordblok een Entity-Relationship diagram dat de gegevens voor één tijdstip 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. Doe het eerst op kladpapier en dan pas hieronder in het net.

Onleesbare tekst wordt fout gerekend.

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



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 de eigenschappen (zoveel mogelijk maar hógstens twee) die *wel* uit de casustekst volgen maar *niet* in uw ER-diagram opgenomen zijn:

Onleesbare tekst wordt fout gerekend.

In het ERD staat niet: ‘een voorzitter behoort tot de afdeling die hij voorzit’.

(NB. Deze cycle-eigenschap (over de twee relaties ‘is_vz_van’ en ‘behoort_tot’) is niet in een ERD weer te geven.)

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 de eigenschappen (zoveel mogelijk maar hógstens twee) die *wel* in het diagram staan maar *niet* uit de letterlijke tekst van de casus volgen:

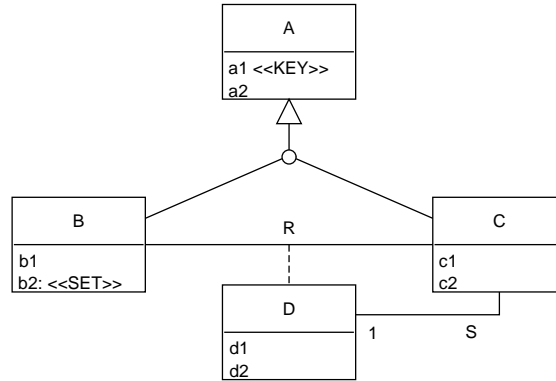
Onleesbare tekst wordt fout gerekend.

(NB. De eigenschap ‘een lid is voorzitter van hoogstens één afdeling’, aangegeven met de multipliciteit 0..1 bij de associatie *is_vz_van* aan de kant van *Afdeling*, volgt uit de volgende twee eigenschappen uit de casus: ‘een voorzitter behoort tot de afdeling die hij voorzit’ en ‘ieder lid behoort tot precies één afdeling’.)

(NB. De eigenschap ‘covering’ bij de generalisatie/specialisatieknoop volgt uit het feit dat in de casus alleen maar van klanten en leden de attribuutwaarden *naam*, *adres* en *tels* worden bijgehouden.)

10p.

Opgave 2. Beschouw het volgende ERD in de notatie van de UML (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, en geen redundantie geïntroduceerd wordt door de vertaling, en alle attributen atomaire waarden hebben.

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

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

Onleesbare tekst wordt fout gerekend.

$A(a1, a2, \text{primary key } (a1))$
$B(a1, b1, \text{primary key } (a1), \text{foreign key } (a1) \text{ references } A(a1))$
$B2s(a1, b2, \text{primary key } (a1, b2), \text{foreign key } (a1) \text{ references } B(a1))$
$C(a1, c1, c2, a1B, a1C,$
primary key (a1),
foreign key (a1) references A(a1),
foreign key (a1B, a1C) references D(a1B, a1C))
$D(a1B, a1C, d1, d2,$
primary key (a1B, a1C),
foreign key (a1B) references B(a1),
foreign key (a1C) references C(a1))
<i>(Zie Toelichting.)</i>

Zijn er instanties van het databaseschema die niet gerepresenteerd kunnen worden als instanties van het ERD? Zo ja, maak dit duidelijk (geef een voorbeeld) en leg kort uit waarom dit mogelijk is.

Onleesbare tekst wordt fout gerekend.

Nee

Aan het ERD wordt nu de volgende eigenschap (“integrity constraint”) toegevoegd:

Iedere B -instantie is ook een C -instantie.

Geef aan hoe u deze eigenschap in het database schema verwerkt, *uitsluitend door toevoegingen te doen aan de bestaande relatieschemas*, op zodanige manier dat in elke databasevulling die eigenschap zal gelden:

Onleesbare tekst wordt fout gerekend.

Voeg toe aan schema B : foreign key $(a1)$ references $C(a1)$.

10p.

Opgave 3. Beschouw een voortdurend veranderende organisatie met entiteiten zoals afdeling, eenheid, manager, werknemer, ondersteuner, salarisschaal, uitstapje. Deze entiteiten tezamen vormen als volgt een relatie \mathcal{R} .

Een tuple (A, E, M, W, O, S, D) zit op zeker tijdstip in \mathcal{R} precies wanneer op dat tijdstip het volgende geldt:

1. A is een Afdeling.
2. E is een Eenheid binnen afdeling A .
3. M is de Manager van eenheid E .
4. W is een Werknemer van de organisatie (niet noodzakelijk van A of E).
5. O is een Ondersteuner van werknemer W voor werkzaamheden binnen eenheid E .
6. S is de Salarisschaal van werknemer W .
7. D is de Datum van een door manager M te organiseren uitstapje voor afdeling A .

Voorts gelden er de volgende eigenschappen op ieder tijdstip:

- a. Iedere werknemer heeft per eenheid hoogstens één ondersteuner.
- b. Een manager van eenheid E behoort tot afdeling waarvan E een eenheid is.
- c. Een manager kan uitstapjes organiseren voor andere afdelingen dan zijn eigen afdeling.
- d. Afdelingen hebben geen eenheden gemeenschappelijk.

Geef voor ieder van de (functionele / multivalued) afhankelijkheden hieronder, met een letter W of O aan of die altijd 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
$O, E \rightarrow W$	O	O kan binnen E twee werknemers W_1 en W_2 ondersteunen.
$O, W \rightarrow E$	O	Zelfs met (5) en a: O kan W ondersteunen bij zowel E_1 als ook E_2 .
$O, W, E \rightarrow M$	W	Volgt al uit $E \rightarrow M$ (regel 3: de manager).
$W \rightarrow S$	W	Regel (6): de salarisschaal.
$D, M \rightarrow A$	O	Zie c: M kan uitstapjes organiseren voor A_1 en A_2 op één datum D .
$M, E \rightarrow A$	W	Uit d volgt al $E \rightarrow A$. (Motivatie "regel b" is fout.)

Onleesbare tekst wordt fout gerekend.

MVD	W/O	Motivatie voor uw keuze
$AEMWOSD = AMD \bowtie AMD EWOS$	W	triviaal; ook: AMD (= lhs \cap rhs) is superkey van lhs
$AEMWOSD = EWO D \bowtie EWO AMS$	W	EWO (= lhs \cap rhs) is superkey van rhs
$AEMWOSD = D \bowtie AEMWOS$	O	verband tussen D en M, A gaat verloren

10p.

Opgave 4. 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:

$$\bar{R} = ABCDEF$$

$$\mathcal{F} = \{ABC \rightarrow D, \quad DE \rightarrow C, \quad CD \rightarrow E, \quad EF \rightarrow A\}$$

- (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 in het antwoordblok iedere *sleutel* van \mathbf{R} .
- (3) Onderstreep in het antwoordblok iedere *supersleutel* van \mathbf{R} .
- (4) Omcirkel in het antwoordblok de *nummers* van de functionele afhankelijkheden die een schending vormen van de BCNF-eis.

Onleesbare tekst wordt fout gerekend.

	$X \rightarrow Y$
43	$ABCD \rightarrow E$
44	$ABCE \rightarrow D$
45	<u>ABCF</u> $\rightarrow DE$
46	$ABDE \rightarrow C$
47	$ABDF \rightarrow$
48	$ABEF \rightarrow$
49	$ACDE \rightarrow$
50	$ACDF \rightarrow E$
51	$ACEF \rightarrow$
52	$ADEF \rightarrow C$
53	$BCDE \rightarrow$
54	<u>BCDF</u> $\rightarrow AE$
55	<u>BCEF</u> $\rightarrow AD$
56	<u>BDEF</u> $\rightarrow AC$
57	$CDEF \rightarrow A$
58	$ABCDE \rightarrow$
59	<u>ABCDF</u> $\rightarrow E$
60	<u>ABCEF</u> $\rightarrow D$
61	<u>ABDEF</u> $\rightarrow C$
62	$ACDEF \rightarrow$
63	<u>BCDEF</u> $\rightarrow A$
64	<u>ABCDEF</u> \rightarrow

(Zie Toelichting.)

10p. **Opgave 5.** Beschouw het relatieschema $\mathbf{R} = (ABCDEFGH, \mathcal{F})$, waarbij:

$$\mathcal{F} = \{ABC \rightarrow DE, \quad DE \rightarrow ABC, \quad F \rightarrow G, \quad G \rightarrow H, \quad H \rightarrow F\}$$

Voer in het antwoordblok de volgende opdrachten uit:

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

$$\mathcal{F} = \{ABC \rightarrow DE, DE \rightarrow ABC, F \rightarrow G, G \rightarrow H, H \rightarrow F\}$$

Onleesbare tekst wordt fout gerekend.

Let op: $G \rightarrow F$ volgt uit \mathcal{F} en zal dus (bij een correcte redenering) een FD worden van een component met attributen $FG\dots$, ook al zit H daar niet bij. Net zo: $H \rightarrow G$ volgt uit \mathcal{F} , en zal dus een FD worden van een component met attributen $GH\dots$, ook al zit F daar niet bij.

We passen het BCNF-algoritme toe.

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

Van de gegeven \mathcal{F} zijn alle leden een schending van de BCNF-eis voor \mathbf{R} ; bijvoorbeeld voor $ABC \rightarrow DE$: de FD is niet-triviaal en het linkerlid is geen sleutel: $ABC_{\mathcal{F}}^+ = ABCDE \neq ABCDEFGH$. We kiezen (zomaar) de eerste, $ABC \rightarrow DE$, ter eliminatie. Dus splitsen we \bar{R} in $\bar{R}_1 = ABCDE$ en $\bar{R}_2 = ABC\cancel{D}\cancel{E}FGH = ABCFGH$. 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 . Dus, $\mathbf{R}_1 = (ABCDE, \{ABC \rightarrow DE, DE \rightarrow ABC\})$ en $\mathbf{R}_2 = (ABCFGH, \{F \rightarrow G, G \rightarrow H, H \rightarrow F\})$.

- We bekijken nu $\mathbf{R}_1 = (ABCDE, \{ABC \rightarrow DE, DE \rightarrow ABC\})$.

In \mathbf{R}_1 is zowel ABC als ook DE een sleutel, dus zijn $ABC \rightarrow DE$ en $DE \rightarrow ABC$ geen schending van de BCNF-conditie, en dus staat \mathbf{R}_1 in BCNF.

- We bekijken nu $\mathbf{R}_2 = (ABCFGH, \{F \rightarrow G, G \rightarrow H, H \rightarrow F\})$.

In \mathbf{R}_2 zijn $F \rightarrow G$, $G \rightarrow H$ en $H \rightarrow F$ alledrie een schending van de BCNF-conditie; voor $F \rightarrow G$ is de reden dat die niet-triviaal is en F geen sleutel is in \mathbf{R}_2 (want $F_{\mathcal{F}_2}^+ = FGH \neq ABCFGH$). We kiezen (zomaar) $F \rightarrow G$ ter eliminatie. Dus splitsen we \bar{R}_2 in $\bar{R}_{2a} = FG$ en $\bar{R}_{2b} = ABC\cancel{F}\cancel{G}H = ABCFH$. Dit levert schema's $\mathbf{R}_{2j} = (\bar{R}_{2j}, \mathcal{F}_{2j})$, waarbij \mathcal{F}_{2j} (een basis voor) de inperking is van \mathcal{F}^+ (of \mathcal{F}_2^+) tot \bar{R}_{2j} . Dus $\mathbf{R}_{2a} = (FG, \{F \rightarrow G, G \rightarrow F\})$ en $\mathbf{R}_{2b} = (ABCFH, \{F \rightarrow H, H \rightarrow F\})$. Let op: de $G \rightarrow F$ zit weliswaar niet in \mathcal{F}_2 , maar toch in \mathcal{F}_{2a} zoals aan het begin is uitgelegd. Net zo zit $F \rightarrow H$ in \mathcal{F}_{2b} .

- We bekijken nu $\mathbf{R}_{2a} = (FG, \{F \rightarrow G, G \rightarrow F\})$. Deze staat in BCNF.

- We bekijken nu $\mathbf{R}_{2b} = (ABCFH, \{F \rightarrow H, H \rightarrow F\})$. Zowel $F \rightarrow H$ als ook $H \rightarrow F$ vormen een schending van de BCNF-conditie in \mathbf{R}_{2b} ; bijvoorbeeld voor $F \rightarrow H$: die FD is niet triviaal en het linkerlid is geen sleutel (want $F_{\mathcal{F}_{2b}}^+ = FH \neq ABCFH$). We kiezen (zomaar) $F \rightarrow H$ ter eliminatie. Dus splitsen we \bar{R}_{2b} in $\bar{R}_{2b1} = FH$ en $\bar{R}_{2b2} = ABC\cancel{F}\cancel{H} = ABCF$. Dit levert schema's $\mathbf{R}_{2bx} = (\bar{R}_{2bx}, \mathcal{F}_{2bx})$, waarbij \mathcal{F}_{2bx} (een basis voor) de inperking is van \mathcal{F}^+ (of \mathcal{F}_{2b}^+) tot \bar{R}_{2bx} . Dus $\mathbf{R}_{2b1} = (FH, \{F \rightarrow H, H \rightarrow F\})$ en $\mathbf{R}_{2b2} = (ABCF, \{\})$. Beide staan in BCNF.

- Dus $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b1}, \mathbf{R}_{2b2}\}$ is een decompositie van \mathbf{R} waarvan alle componenten in BCNF staan. De functionele afhankelijkheden zijn allemaal behouden; met name is $G \rightarrow H$ afleidbaar uit $G \rightarrow F$ in \mathbf{R}_{2a} en $F \rightarrow H$ in \mathbf{R}_{2b1} .

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

$$“ABCDEFGH = ABCDE \bowtie FG \bowtie FH \bowtie ABCF \text{ geldt in } \mathbf{R}.”$$

Wanneer je met een andere schending begint kun je mogelijk een andere decompositie bereiken. Met name geldt ook in \mathbf{R} : $ABCDEFGH = ABCDE \bowtie FGH \bowtie ABCF$.

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 kanonsloop (*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 6.** Beschouw de volgende zoekopdracht:

Geef alle paren van verschillende landen die in eenzelfde zeeslag betrokken zijn.

(Een country $cntry$ is betrokken in battle b als er een class c , een ship s en een outcome o bestaat met $cntry = c.c$ en $c.n = s.c$ en $s.n = o.s$ en $o.b = b.n$.)

Geef voor deze vraag een *afleiding* in Verzamelingsnotatie naar een vorm die dicht aansluit bij SQL met zo weinig mogelijk subqueries en geen group by clause. Het begin is al gegeven. Kort tabel- en attribuutnamen af tot hun eerste letter.

Onleesbare tekst wordt fout gerekend.

	“alle $cntry_1, cntry_2$ met $cntry_1 \neq cntry_2$, betrokken in eenzelfde zeeslag”
=	$\{cntry_1, cntry_2 : txt \mid cntry_1 \neq cntry_2 \wedge$
	$(\exists b : B \bullet \text{“}cntry_1 \text{ betrokken in } b\text{”} \wedge \text{“}cntry_2 \text{ betrokken in } b\text{”}) \bullet (cntry_1, cntry_2)\}$
=	$\{cntry_1, cntry_2 : txt \mid cntry_1 \neq cntry_2 \wedge$
	$(\exists b : B \bullet (\exists c_1 : C; s_1 : S; o_1 : O \bullet cntry_1 = c_1.c \wedge c_1.n = s_1.c \wedge s_1.n = o_1.s \wedge o_1.b = b.n) \wedge$
	$(\exists c_2 : C; s_2 : S; o_2 : O \bullet cntry_2 = c_2.c \wedge c_2.n = s_2.c \wedge s_2.n = o_2.s \wedge o_2.b = b.n))$
	$\bullet (cntry_1, cntry_2)\}$
=	$\{cntry_1, cntry_2 : txt; b : B; c_1, c_2 : C; s_1, s_2 : S; o_1, o_2 : O \mid cntry_1 \neq cntry_2 \wedge$
	$cntry_1 = c_1.c \wedge c_1.n = s_1.c \wedge s_1.n = o_1.s \wedge o_1.b = b.n \wedge$
	$cntry_2 = c_2.c \wedge c_2.n = s_2.c \wedge s_2.n = o_2.s \wedge o_2.b = b.n \bullet (cntry_1, cntry_2)\}$
=	$\{b : B; c_1, c_2 : C; s_1, s_2 : S; o_1, o_2 : O \mid$
	$(\exists cntry_1, cntry_2 : txt \bullet cntry_1 = c_1.c \wedge cntry_2 = c_2.c) \wedge c_1.c \neq c_2.c \wedge$
	$c_1.n = s_1.c \wedge s_1.n = o_1.s \wedge o_1.b = b.n \wedge c_2.n = s_2.c \wedge s_2.n = o_2.s \wedge o_2.b = b.n \bullet (c_1.c, c_2.c)\}$
=	[aanname: het domein van <i>country</i> in <i>Class</i> is <i>txt</i>]
	$\{b : B; c_1, c_2 : C; s_1, s_2 : S; o_1, o_2 : O \mid c_1.c \neq c_2.c \wedge$
	$c_1.n = s_1.c \wedge s_1.n = o_1.s \wedge o_1.b = b.n \wedge c_2.n = s_2.c \wedge s_2.n = o_2.s \wedge o_2.b = b.n \bullet (c_1.c, c_2.c)\}$
=	$\{c_1, c_2 : C; s_1, s_2 : S; o_1, o_2 : O \mid (\exists b : B \bullet o_1.b = b.n) \wedge c_1.c \neq c_2.c \wedge$
	$c_1.n = s_1.c \wedge s_1.n = o_1.s \wedge c_2.n = s_2.c \wedge s_2.n = o_2.s \wedge o_2.b = o_1.b \bullet (c_1.c, c_2.c)\}$
=	[in O is b foreign key verwijzend naar $B(n)$, dus voor o_i geldt: $\exists b \bullet o_i.b = b.n$]
	$\{c_1, c_2 : C; s_1, s_2 : S; o_1, o_2 : O \mid c_1.c \neq c_2.c \wedge$
	$c_1.n = s_1.c \wedge s_1.n = o_1.s \wedge c_2.n = s_2.c \wedge s_2.n = o_2.s \wedge o_2.b = o_1.b \bullet (c_1.c, c_2.c)\}$

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

Onleesbare tekst wordt fout gerekend.

```
select distinct c1.c, c2.c
from C c1, C c2, S s1, S s2, O o1, O o2
where c1.c <> c2.c and
      c1.n=s1.c and s1.n=o1.s and c2.n=s2.c and s2.n=o2.s and o2.b=o1.b
NB Elk landenpaar  $l_1, l_2$  wordt tweemaal opgeleverd, nml. als  $(l_1, l_2)$  en als  $(l_2, l_1)$ ; dit kan voorkomen worden door  $c1.c <> c2.c$  te vervangen door  $c1.c < c2.c$ .
```

Geef ook een formulering van de vraag in Relationele Algebra waarin zoveel mogelijk gebruik gemaakt wordt van de natural join (\bowtie) (in plaats van andere vormen van joins of het Cartesische product):

Onleesbare tekst wordt fout gerekend.

```
 $\pi_{cc1, cc2} (\sigma_{cc1 \neq cc2} (C1 \bowtie S1 \bowtie O1 \bowtie C2 \bowtie S2 \bowtie O2))$ 
Hierbij zijn met renaming nieuwe relaties gedefinieerd:
C1 = C[ $cn1, cc1, \dots$ ] en S1 = S[ $sn1, cn1, sl1$ ] en O1 = O[ $sn1, bn, r1$ ]
C2 = C[ $cn2, cc2, \dots$ ] en S2 = S[ $sn2, cn2, sl2$ ] en O2 = O[ $sn2, bn, r2$ ]
```

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

10p. **Opgave 7.** Formuleer in SQL met een group-by query:

Geef voor ieder schip dat met minstens acht andere schepen aan zeeslagen heeft deelgenomen, het aantal zeeslagen waaraan het heeft deelgenomen.

Onleesbare tekst wordt fout gerekend.

```
select o.s, count (distinct o.b)
from O o, O o1
where o.b = o1.b and o.s <> o1.s
group by o.s
having count (distinct o1.s) >= 8 (Zie Toelichting.)
```

5p. **Opgave 8.** (Goede beantwoording levert 5 bonuspunten boven op de 5 punten die voor deze opgave gegeven worden. Daardoor kan het totaal aantal behaalde punten op 105 uitkomen.)
 Formuleer in *Verzamelingsnotatie én in SQL*:

Geef ieder schip s waarvoor geldt dat er *voor iedere* zeeslag waaraan s heeft deelgenomen, er een schip is dat éérder dan s te water is gelaten en ook aan die zeeslag heeft deelgenomen.

Het vergelijken van datums, zoals $Ship(launched)$, mag met $<$ gedaan worden. (U hoeft geen afleiding te geven, maar een afleiding, hieronder of op kladpapier, zou u wel kunnen helpen.)

Onleesbare tekst wordt fout gerekend.

	Kortheidshalve laten we de typering achterwege (en korten dus $b : B$ af tot b , et cetera).
	$\{s \mid (\forall b \mid \text{“}s \text{ neemt deel aan } b\text{”} \bullet (\exists s' \mid \text{“}s' \text{ vóór } s \text{ te water”} \bullet \text{“}s' \text{ neemt deel aan } b\text{”})) \bullet s.n\}$
=	[formalisering]
	$\{s \mid (\forall b \mid (\exists o \bullet s.n=o.s \wedge o.b=b.n) \bullet (\exists s' \mid s'.l < s.l \bullet (\exists o' \bullet s'.n=o'.s \wedge o'.b=b.n))) \bullet s.n\}$
=	[shunting]
	$\{s \mid (\forall b; o \mid s.n=o.s \wedge o.b=b.n \bullet (\exists s'; o' \mid s'.l < s.l \bullet s'.n=o'.s \wedge o'.b=b.n)) \bullet s.n\}$
=	[shunting]
	$\{s \mid (\forall o \mid (\exists b \bullet o.b=b.n) \wedge s.n=o.s \bullet (\exists s'; o' \mid s'.l < s.l \bullet s'.n=o'.s \wedge o'.b=o.b)) \bullet s.n\}$
=	[in O is b foreign key naar $B(n)$; dus voor o geldt: $\exists b \bullet o.b = b.n$]
	$\{s \mid (\forall o \mid s.n=o.s \bullet (\exists s'; o' \mid s'.l < s.l \bullet s'.n=o'.s \wedge o'.b=o.b)) \bullet s.n\}$
=	[predicatencalculus]
	$\{s \mid \neg (\exists o \mid s.n=o.s \bullet \neg (\exists s'; o' \mid s'.l < s.l \bullet s'.n=o'.s \wedge o'.b=o.b)) \bullet s.n\}$
=	[vertaling naar sql]
	select $s.n$ from S s where not exists (select * from O o where $s.n=o.s$ and not exists (select * from S $s1$, O $o1$ where $s1.l < s.l$ and $s1.n = o1.s$ and $o1.b = o.b$)))
	(Zie Toelichting.)

10p.

Opgave 9. In sommige databasesystemen (DBMSs) wordt automatisch een *index* aange-
maakt voor de *primary key* van een tabel. Wat is het nut hiervan?

Onleesbare tekst wordt fout gerekend.

(1) Bij een index voor attribuut *A* zijn de rijen met gegeven waarde voor *A* efficiënt te vinden. (2) De rijen (hooguit één) met gegeven waarde voor de *primary key* moeten vaak gevonden worden: om bij toevoegingen en wijzigingen te controleren of er geen rijen met gelijke *primary keys* ontstaan, en bij joins met ander tabellen. Dus, (3) een *index* op de *primary key* verhoogt de efficiëntie van zowel deze controle, als ook de joins.

10p.

Opgave 10. In de implementatie van transacties spelen de begrippen *locking*, *read lock* en *write lock* een rol, alsmede de “spelregels” voor het verkrijgen van locks. Deze begrippen mag u bekend veronderstellen. Geef aan wat de aanduiding “*two-phase*” betekent in het begrip *two-phase* locking:

Onleesbare tekst wordt fout gerekend.

Een locking protocol is *two-phase* wanneer voor alle transacties geldt dat gedurende de transactie er éérst een fase is waarin de locks verkregen worden, en er vervolgens een fase is waarin de locks vrijgegeven worden.

Geef aan welke eigenschap wél waargemaakt wordt door *two-phase* locking en niet door een locking protocol dat niet *two-phase* is (met andere woorden, geef aan wat het nut is van *two-phase* in het locking protocol):

Onleesbare tekst wordt fout gerekend.

Serialiseerbaarheid, dat wil zeggen dat elke wijziging van de database door verscheidene *gelijktijdig* uitgevoerde transacties ook bewerkstelligd kan worden door de transacties *na* elkaar uit te voeren (in een daartoe geschikt gekozen volgorde).

Toelichtingen

Toelichting bij antwoord 2.

Entiteiten B en C zijn niet opgenomen in de tabel voor alle entiteiten A omdat dan NULLs nodig zijn om een A -instantie te representeren die geen B - of C -instantie is.

Voor iedere B -instantie met key $a1$ representeert de verzameling $\{b2 \mid (a1, b2) \in B2s\}$ de verzameling van $b2$ -waarden van die instantie.

Entiteit S is niet als aparte tabel nodig, omdat S -instanties gerepresenteerd worden in het attribootpaar $(a1B, a1C)$ van C (daarbij gebruik makend van de eigenschap dat er bij iedere C -instantie precies één D -instantie is waarmee die in relatie S zit).

Relatie R is, per definitie, identiek aan relatie-entiteit D .

Toelichting bij antwoord 4.

Let op: iedere key is ook een superkey, omgekeerd is dit niet het geval.

Toelichting bij antwoord 7.

In plaats van de having-conditie ‘count (distinct $o1.s$) ≥ 8 ’ kun je ook nemen ‘count (distinct $o1.s$) > 8 ’ indien je dan tegelijk de where-conditie ‘ $o.s \ll o1.s$ ’ schrapt.

Hier is —voor de liefhebbers— een formele afleiding van de group-by query:

$$\begin{aligned}
& \{s \mid \text{“}s \text{ nam met } \geq 8 \text{ schepen deel aan zeeslagen”} \bullet (s.n, \#\text{“zeeslagen waaraan } s \text{ deel nam”})\} \\
= & \{s \mid 8 \leq \#\{s' \mid \text{“}s \text{ samen met } s' \neq s \text{ in 'n zeeslag”} \bullet s'.n\} \bullet (s.n, \#\{b \mid \text{“}s \text{ nam deel aan } b \text{”} \bullet b.n\})\} \\
= & \{s \mid 8 \leq \#A_{s.n} \bullet (s.n, \#B_{s.n})\} \\
& \text{met } A_x = \{s' \mid s'.n \neq x \wedge \text{“}x \text{ samen met } s' \text{ in één zeeslag”} \bullet s'.n\} \\
& \quad = \{s' \mid s'.n \neq x \wedge (\exists o, o'; b \mid x=o.s \wedge o.b=b.n=o'.b \wedge o'.s=s'.n) \bullet s'.n\} \\
& \quad = \{s'; o, o'; b \mid s'.n \neq x=o.s \wedge o.b=b.n=o'.b \wedge o'.s=s'.n \bullet s'.n\} \\
& \quad = \{o, o' \mid (\exists s' \bullet s'.n=o'.s) \wedge (\exists b \bullet b.n=o.b) \wedge o'.s \neq x=o.s \wedge o.b=o'.b \bullet o'.s\} \\
& \quad = \{o, o' \mid o'.s \neq x=o.s \wedge o.b=o'.b \bullet o'.s\} \\
= & \quad [\text{streef naar dezelfde vorm als die van } A_x] \\
& \{s \mid 1 \leq \#A_{s.n} \wedge 8 \leq \#A_{s.n} \bullet (s.n, \#B_{s.n})\} \\
= & \{s \mid (\exists o, o' \bullet o'.s \neq o.s = s.n \wedge o.b=o'.b) \wedge 8 \leq \#A_{s.n} \bullet (s.n, \#B_{s.n})\} \\
= & \{s; o, o' \mid o'.s \neq o.s = s.n \wedge o.b=o'.b \wedge 8 \leq \#A_{s.n} \bullet (s.n, \#B_{s.n})\} \\
= & \{o, o' \mid o'.s \neq o.s \wedge (\exists s \bullet s.n=o.s) \wedge o.b=o'.b \wedge 8 \leq \#A_{o.s} \bullet (o.s, \#B_{o.s})\} \\
= & \{o, o' \mid o'.s \neq o.s \wedge o.b=o'.b \wedge 8 \leq \#A_{o.s} \bullet (o.s, \#B_{o.s})\} \\
& \text{met } B_x = \{b \mid \text{“}x \text{ nam deel aan } b \text{”} \bullet b.n\} \\
& \quad = \{b \mid (\exists o \bullet b.n = o.b \wedge o.s = x) \bullet b.n\} \\
& \quad = \{b; o \mid b.n = o.b \wedge o.s = x \bullet b.n\} \\
& \quad = \{o \mid (\exists b \bullet b.n = o.b) \wedge o.s = x \bullet o.b\} \\
& \quad = \{o \mid o.s = x \bullet o.b\} \\
& \quad \quad [\text{streef naar gelijke vorm met } A_x] \\
& \quad = \{o \mid o.s = x \wedge (\exists o' \bullet x \neq o'.s \wedge o.b = o'.b) \bullet o.b\} \\
& \quad = \{o, o' \mid o.s = x \neq o'.s \wedge o.b = o'.b \bullet o.b\}
\end{aligned}$$

Toelichting bij antwoord 8.

Uit de een-na-laatste formulering in verzamelingsnotatie volgen onmiddellijk ook de volgende:

$$\begin{aligned}
& \{s \mid (\forall o \mid s.n=o.s \bullet o.b \in \{s'; o' \mid s'.l < s.l \wedge s'.n=o'.s \bullet o'.b\}) \bullet s.n\} \\
& \{s \mid \{o \mid s.n=o.s \bullet o.b\} \subseteq \{s'; o' \mid s'.l < s.l \wedge s'.n=o'.s \bullet o'.b\} \bullet s.n\}
\end{aligned}$$

$\{s \mid \{o \mid s.n=o.s \bullet o.b\} \setminus \{s'; o' \mid s'.l < s.l \wedge s'.n=o'.s \bullet o'.b\} = \emptyset \bullet s.n\}$

De eerste en laatste hiervan geven bij directe vertaling naar SQL de volgende queries:

```
select s.n from S s where not exists (  
  select * from O o where s.n=o.s and o.b not in (  
    select o1.b from S s1, O o1 where s1.l < s.l and s1.n = o1.s))  
= select s.n from S s where not exists (  
  select o.b from O o where s.n=o.s  
  except  
  select o1.b from S s1, O o1 where s1.l < s.l and s1.n = o1.s)
```