Tentamen Gegevensbanken (211074) — 30 oktober 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 kopieen van *delen van het boek*; kopieen 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.)

Niet invullen

5 gratis	1	2	3	4	5	6	7	8	9	9bonus	10	11

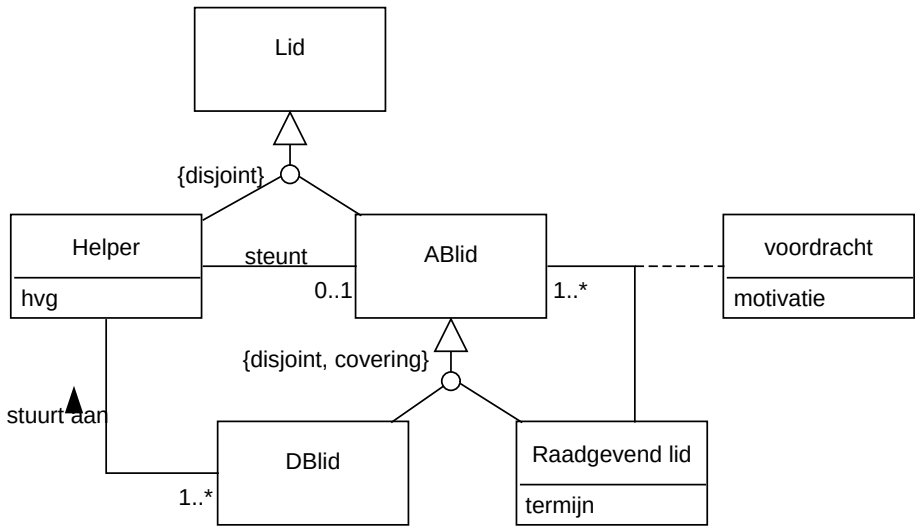
10p. **Opgave 1.** Studentenvereniging *AICP* (niet te verwarren met *IAPC*) heeft een interne organisatie die soms verkeerd begrepen wordt. Daarom wil zij de verschillende soorten *leden* en *hun onderlinge relaties* met een plaatje (Entity Relationship Diagram) beschrijven. Hier volgt een beschrijving in woorden.

- De dagelijkse bestuursklussen worden uitgevoerd door leden van het Dagelijks Bestuur. Dergelijke leden worden benoemd voor de duur van één jaar.
- Het Algemeen Bestuur bestaat uit enerzijds de leden van het Dagelijks Bestuur en anderzijds uit "wijze mannen" van *AICP* (vaak maar niet altijd: leden van een Dagelijks Bestuur uit een vorig jaar). Iemand kan meerdere jaren (de *termijn*) lid zijn van het Algemeen Bestuur.
- De taak van de leden uit het Algemeen Bestuur is het Dagelijks Bestuur met raad (en soms daad) bij te staan; daarom worden de leden van het Algemeen Bestuur die niet lid zijn van het Dagelijks Bestuur ook wel *raadgevers* genoemd.
- Leden van het Dagelijks Bestuur zijn sowieso lid van het Algemeen Bestuur; elk ander lid van het AB is benoemd op voordracht van een of meer leden van het Algemeen Bestuur. Per voordracht is een *motivatie* gegeven, waarin het voordragende lid redenen geeft waarom het voorgedragen lid geschikt is om in het Algemeen Bestuur te zitten.
- Sommige leden van de vereniging maken zich verdienstelijk door hand- en spandiensten te verlenen; zij worden helpers genoemd en zijn geen bestuurslid. Volgens de regels van *AICP*, biedt een helper steun aan hooguit één lid van het Algemeen Bestuur. Van ieder helper is bekend hoe vaak hij in het verleden helper is geweest: de *hulpvaardigheidgraad*, afgekort tot *hvg*.
- Om te zorgen dat alle hulp doelmatig en doelgericht is, wordt iedere helper door tenminste één lid van het Dagelijks Bestuur aangestuurd.

(De vraag staat op de volgende pagina!)

Onleesbare tekst wordt fout gerekend.

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



(Zie Toelichting.)

Geef in het antwoordblok (vorige pagina) 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 in het net.

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 hóógstens twee) die *wel* uit de casustekst volgen maar *niet* in uw ER-diagram opgenomen zijn:

Onleesbare tekst wordt fout gerekend.

"Vaak maar niet altijd:..." (tweede punt uit de opsomming) is een eigenschap die niet in een ER-diagram aangegeven kan worden.

De taak van een lid van het Algemeen Bestuur staat niet in het ER-diagram vermeld.

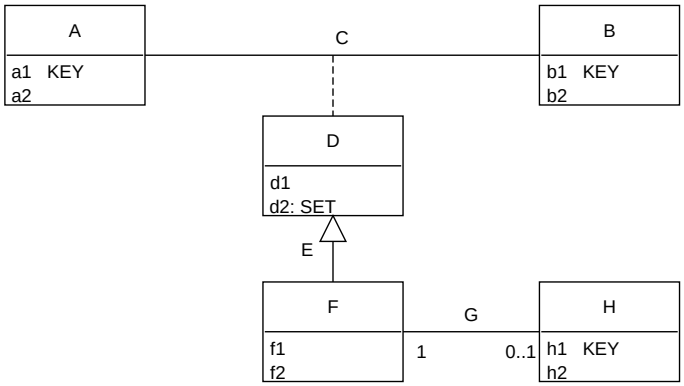
Er zijn realistische maar niet expliciet genoemde cykeleigenschappen in de casus; deze zijn geen van alle opgenomen in, of vermeld bij, het ERD.

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 hóógstens twee) die *wel* in het diagram staan maar *niet* uit de letterlijke tekst van de casus volgen:

Onleesbare tekst wordt fout gerekend.

Geen

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, er 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 domeinindicatie weggelaten mag worden; een voorbeeld van de vorm van zo'n schema is:

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

Onleesbare tekst wordt fout gerekend.

$A(a1, a2, \text{primary key}(a1))$
$B(b1, b2, \text{primary key}\,(b1))$
$D1(a1, b1, d1, \text{primary key}\,(a1, b1),$
$\quad \text{foreign key}\,(a1)\,\text{references}\,A(a1),$
$\quad \text{foreign key}\,(b1)\,\text{references}\,B(b1))$
$D2(a1, b1, d2lid, \text{primary key}\,(a1, b1, d2lid), \text{foreign key}\,(a1, b1)\,\text{references}\,D1(a1, b1))$
$F(a1, b1, f1, f2, \text{primary key}\,(a1, b1), \text{foreign key}\,(a1, b1)\,\text{references}\,D1(a1, b1))$
$H(h1, h2, a1, b1, \text{primary key}\,(h1), \text{foreign key}\,(a1, b1)\,\text{references}\,F(a1, b1), \text{unique}(a1, b1))$
Eigenlijk moet de eigenschap `not null` toegevoegd worden aan iedere foreign key (en in
instanties van het ERD mogen de key-attributen niet de waarde 'onbekend/null' hebben).
De verzameling $d2$ van een D-instantie van het ERD, met primary key waarde (x, y), wordt
uit de tabelinhouden berekend als de verzameling $\{d : D2 \mid d(a1, b2) = (x, y) \bullet d(d2lid)\}$.
(Zie Toelichting.)

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

10p. **Opgave 3.** Beschouw soap series op de TV, met gegevens zoals naam, omroep, acteur, rol, et cetera. Deze gegevens samen vormen als volgt een relatie $\mathcal{R}$:

Een tuple (S, O, A, G, R, K, T) zit in relatie $\mathcal{R}$ precies wanneer het volgende geldt:

1. S is een Serie.
2. O is de Omroep die serie S uitzendt.
3. A is een Acteur van serie S.
4. G is het Geboortejaar van acteur A.
5. R is een Rol gespeeld door acteur A in serie S.
6. K is een Kostuum dat hoort bij rol R.
7. T is een Telefoonnummer dat hoort bij serie S (denk aan 09-nummer!)

Voorts gelden er de volgende eigenschappen op ieder tijdstip:

a. Een acteur van een serie kan ook in andere series spelen.
b. Er is geen rol die in één serie door verschillende acteurs gespeeld wordt.
c. Een acteur kan verschillende rollen spelen.
d. Verschillende series hebben verschillende telefoonnummers.

Geef voor ieder van de functionele afhankelijkheden hieronder, met een letter W aan of die volgens bovenstaande definitie altijd Waar is in relatie $\mathcal{R}$, en met een letter O of die afhankelijkheid volgens bovenstaande definitie Onwaar kan zijn 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
$A \rightarrow G$	W	volgens regel 4 (*het*).
$S, R \rightarrow O$	W	S alleen al bepaalt O volgens 2.
$T \rightarrow S$	W	eigenschap d (alléén eigenschap 7 is wordt fout gerekend)
$S \rightarrow T$	O	een serie kan best meerdere tel.nrs hebben
$S, A, R \rightarrow K$	O	een rol van een acteur in een serie kan meerdere kostuums hebben
$A, R, K \rightarrow S$	O	een acteur met een rol en kostuum kan in meerdere series zijn
$R, K, S \rightarrow A$	W	volgens c geldt al $R, S \rightarrow A$

10p. **Opgave 4.** Beschouw het relatieschema $\mathbf{R} = (\bar{R}, \mathcal{F})$, waarbij de attribuutverzameling $\bar{R}$ en de verzameling $\mathcal{F}$ van functionele afhankelijkheden als volgt luiden:

$$\begin{aligned} \bar{R} &= ABCDEF \\ \mathcal{F} &= \{A \to F, \quad BC \to A, \quad BCD \to E, \quad EF \to A\} \end{aligned}$$

(1) Geef in het antwoordblok in iedere regel een zo groot mogelijk rechterlid Y zó dat de functionele afhankelijkheid $X \to Y$ 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 **R**.
(3) *Onderstreep* in het antwoordblok iedere *supersleutel* van **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	$\to$	Y
43	$\underline{ABCD}$	$\to$	EF
44	$ABCE$	$\to$	F
45	$ABCF$	$\to$	
46	$ABDE$	$\to$	F
47	$ABDF$	$\to$	
48	$ABEF$	$\to$	
49	$ACDE$	$\to$	F
50	$ACDF$	$\to$	
51	$ACEF$	$\to$	
52	$ADEF$	$\to$	
53	$\underline{BCDE}$	$\to$	AF
54	$\underline{BCDF}$	$\to$	AE
55	$BCEF$	$\to$	A
56	$BDEF$	$\to$	A
57	$CDEF$	$\to$	A
58	$\underline{ABCDE}$	$\to$	F
59	$\underline{ABCDF}$	$\to$	E
60	$ABCEF$	$\to$	
61	$ABDEF$	$\to$	
62	$ACDEF$	$\to$	
63	$\underline{BCDEF}$	$\to$	A
64	$\underline{ABCDEF}$	$\to$	

(Zie Toelichting.)

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

$$\mathcal{F} \;=\; \{ABC{\rightarrow}D, \quad D{\rightarrow}FG, \quad E{\rightarrow}D\}$$

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.

Onleesbare tekst wordt fout gerekend.

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

We passen het BCNF-algoritme toe.

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

Van de gegeven $\mathcal{F}$ zijn alle leden een schending van de BCNF-eis voor $\mathbf{R}$; bijvoorbeeld voor $ABC{\rightarrow}D$: de FD is niet-triviaal en het linkerlid is geen sleutel: $ABC^+_{\mathcal{F}} = ABCDFG \neq ABCDEFG$. We kiezen (zomaar) de eerste, $ABC{\rightarrow}D$, ter eliminatie. Dus splitsen we $\bar{R}$ in $\bar{R}_1 = ABCD$ en $\bar{R}_2 = ABC\not\!\!D EFG = ABCEFG$. 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, $\mathbf{R}_1 = (ABCD, \{ABC{\rightarrow}D\})$ en $\mathbf{R}_2 = (ABCEFG, \{ABC{\rightarrow}FG,\ E{\rightarrow}FG\})$. Let op: de $ABC{\rightarrow}FG$ zit weliswaar niet in $\mathcal{F}$, maar wel in $\mathcal{F}^+$ en dus in $\mathcal{F}_2$. De afhankelijkheid $D{\rightarrow}FG$ is verloren gegaan.

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

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

- We bekijken nu $\mathbf{R}_2 = (ABCEFG, \{ABC{\rightarrow}FG,\ E{\rightarrow}FG\})$.

In $\mathbf{R}_2$ zijn $ABC{\rightarrow}FG$ en $E{\rightarrow}FG$ beide een schending van de BCNF-conditie; voor $ABC{\rightarrow}FG$ is de reden dat die niet-triviaal is en ABC geen sleutel is in $\mathbf{R}_2$ (want $ABC^+_{\mathcal{F}_2} = ABCFG \neq ABCEFG$). We kiezen (zomaar) $ABC{\rightarrow}FG$ ter eliminatie. Dus splitsen we $\bar{R}_2$ in $\bar{R}_{2a} = ABCFG$ en $\bar{R}_{2b} = ABCE\not\!\!F\not\!\!G = ABCE$. 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 $\mathbf{R}_{2a} = (ABCFG, \{ABC{\rightarrow}FG\})$ en $\mathbf{R}_{2b} = (ABCE, \{\ \})$. Let op: de $ABC{\rightarrow}FG$ zit in $\mathcal{F}_2^+$ en dus in $\mathcal{F}_{2a}$.

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

- We bekijken nu $\mathbf{R}_{2b} = (ABCE, \{\ \})$. Deze staat in BCNF.

- Dus $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b}\}$ is een decompositie van $\mathbf{R}$ waarvan alle componenten in BCNF staan. Er zijn functionele afhankelijkheden verloren gegaan; met name is dat $D{\rightarrow}FG$.

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

"$ABCDEFG = ABCD \bowtie ABCFG \bowtie ABCE$ geldt in $\mathbf{R}$."

Wanneer je met een andere schending begint kun je mogelijk een andere decompositie bereiken. Met name geldt ook in $\mathbf{R}$: $ABCDEFG = ABCE \bowtie xD \bowtie yFG$ voor iedere keuze van x uit $\{ABC, E\}$ en iedere keuze van y uit $\{ABC, D, E\}$.

10p. **Opgave 6.** Leg uit wat *Atomicity* betekent in de context van transacties.

Onleesbare tekst wordt fout gerekend.

Atomicity betekent dat van iedere transactie de uitvoering ervan *geheel* of *helemaal niet* plaats vindt, voor zover andere normaal-eindigende transacties (of observators van stabiele toestanden van de database) dat kunnen waarnemen.

Dus wijzigingen aangebracht door een transactie T worden ongedaan gemaakt (indien T niet voltooid kan worden) zó dat andere transacties die wel tot voltooing komen, die wijzigingen niet kunnen waarnemen. Dit geldt zelfs na een abort van T of een crash van het systeem.

Geef een voorbeeld waarin de eigenschap *Atomicity* geschonden wordt.

Onleesbare tekst wordt fout gerekend.

Een transactie *Overboeking*, bestaande uit een actie *afschrijving* van de ene rekening en een actie *bijschrijving* op een andere rekening. Wanneer *Overboeking* zó zou worden uitgevoerd dat de ene actie wel wordt gedaan en de andere niet, dan is *Atomicity* geschonden.

Ander voorbeeld: Als $T_1 = w_1(x); \ldots$ en $T_2 = r_2(x)$ en de uitvoering is: $w_1(x)$; $r_2(x)$; $commit_2$; $abort_1$, dan is *Atomicity* geschonden.

Benoem de techniek die gebruikt wordt om *Atomicity* te waarborgen; leg kort uit *hoe* die techniek werkt.

Onleesbare tekst wordt fout gerekend.

Logging (of preciezer: *write-ahead logging*).

Wanneer een transactie T voortijdig geeindigd is (door een abort van T of door een crash van het systeem), dan worden (onmiddellijk, of na opstarten na een crash) alle alreeds uitgevoerde acties van T teruggedraaid (*roll-back*); dat kan met behulp van de *before images* die bij iedere database-wijziging in de log geschreven worden vóórdat de actie op de database wordt uitgevoerd.

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 7.** Beschouw de volgende zoekopdracht:

Geef iedere klasse waarvan er tenminste één schip in het zelfde jaar te water is gelaten als het jaar waarin het in een zeeslag gezonken is.

Geef voor deze vraag een *afleiding in kleine stappen* in Verzamelingsnotatie naar een vorm die dicht aansluit bij SQL, *zonder Class in de from clausule*, met zo weinig mogelijk subqueries en geen group by clausule. Gebruik de uitdrukking $sameYear(x, y)$ als test of datums x en y in het zelfde jaar vallen. Kort tabel- en attribuutnamen af tot hun eerste letter. Het begin is al gegeven.

Onleesbare tekst wordt fout gerekend.

	"iedere klasse met minstens één schip dat in het jaar van te water lating gezonken is"
=	$\{c : C \mid (\exists\, s : S \mid$ "s is van klasse c" $\bullet$ "s gezonken in jaar van te water lating"$) \bullet c.name\}$
=	$\{c \mid (\exists\, s \mid s.c{=}c.n \bullet (\exists\, b \mid sameYear(s.l, b.d) \bullet$ "s gezonken in b"$)) \bullet c.name\}$
=	$\{c \mid (\exists\, s \mid s.c{=}c.n \bullet (\exists\, b \mid sameYear(s.l, b.d) \bullet (\exists\, o \mid s.n{=}o.s \wedge o.b{=}b.n \bullet o.r{=}sunk))) \bullet c.n\}$
=	[shunting]
	$\{c;\, s \mid s.c{=}c.n \wedge (\exists\, b \mid sameYear(s.l, b.d) \bullet (\exists\, o \mid s.n{=}o.s \wedge o.b{=}b.n \bullet o.r{=}sunk)) \bullet c.n\}$
=	[shunting]
	$\{c;\, s;\, b \mid s.c = c.n \wedge sameYear(s.l, b.d) \wedge (\exists\, o \mid s.n = o.s \wedge o.b = b.n \bullet o.r = sunk) \bullet c.n\}$
=	[shunting]
	$\{c;\, s;\, b;\, o \mid s.c = c.n \wedge sameYear(s.l, b.d) \wedge s.n = o.s \wedge o.b = b.n \wedge o.r = sunk \bullet c.n\}$
=	[gelijkheid] $\downarrow\downarrow$
	$\{c;\, s;\, b;\, o \mid s.c = c.n \wedge sameYear(s.l, b.d) \wedge s.n = o.s \wedge o.b = b.n \wedge o.r = sunk \bullet s.n\}$
=	[shunting]
	$\{s;\, b;\, o \mid (\exists\, c \bullet s.c{=}c.n) \wedge sameYear(s.l, b.d) \wedge s.n{=}o.s \wedge o.b{=}b.n \wedge o.r{=}sunk \bullet s.n\}$
=	[In *Ship* is *classname* foreign key verwijzend naar $Class(name)$]
	[dus, voor elke s geldt: $\exists\, c \bullet s.c = c.n$]
	$\{s;\, b;\, o \mid true \wedge sameYear(s.l, b.d) \wedge s.n = o.s \wedge o.b = b.n \wedge o.r = sunk \bullet s.n\}$
=	[propositie logica]
	$\{s;\, b;\, o \mid sameYear(s.l, b.d) \wedge s.n = o.s \wedge o.b = b.n \wedge o.r = sunk \bullet s.n\}$
	We hebben kortheidshalve vaak $c : C$ afgekort tot c, en net zo bij $s{:}S$, $b{:}B$, $o{:}O$.

Geef een SQL formulering van de beschouwde vraag; de SQL formulering moet dicht aansluiten bij de zojuist gegeven uitdrukking. Gebruik *DISTINCT* alleen wanneer het nodig is.

Onleesbare tekst wordt fout gerekend.

select distinct $s.classname$

from $S\ s,\ O\ o,\ B\ b$

where $sameYear(s.l, b.d)$ and $s.n = o.s$ and $o.b = b.n$ and $o.r = sunk$

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.

$\pi_{cn}\ (\sigma_{sameYear(sl,bd) \wedge r = sunk}\ (S' \bowtie O' \bowtie B'))$

Hierbij zijn met renaming nieuwe relaties gedefinieerd:

$S' = S[sn, cn, sl]$ en $O' = O[sn, bn, r]$ en $B' = B[bn, bd]$

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, battlename</u>, result)	*O (<u>s, b</u>, r)*

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

Geef voor ieder schip s dat aan tenminste twintig zeeslagen heeft deelgenomen, het aantal, a, van andere schepen die gezonken zijn in een zeeslag waaraan s deelnam. Sla de schepen s over waarvoor dat aantal a nul is.

Onleesbare tekst wordt fout gerekend.

select $o1.s$, count (distinct $o2.s$) as a -- eventueel: count($o2.s$), of: count($*$)

from $O\ o1,\ O\ o2$

where $o1.b = o2.b$ and $o1.s <> o2.s$ and $o2.r = sunk$

group by $o1.s$

having count ($o1.b$) >= 20

De variant 'eventueel...' geldt onder aanname dat een schip hooguit éénmaal zinkt.

(Zie Toelichting.)

5p. **Opgave 9.** (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 in iedere zeeslag waarin het deel neemt, er een schip s' is dat ook aan die zeeslag deel neemt maar met een ander resultaat dan s.

(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).

$\{s \mid (\forall\, b \mid \text{“}s \text{ in } b\text{”} \bullet (\exists\, s' \mid \text{“}s' \text{ in } b \bullet \text{“resultaten van } s \text{ en } s' \text{ in } b \text{ verschillen”})) \bullet s.n\}$

$= \{s \mid (\forall\, b \mid (\exists\, o \bullet s.n = o.s \wedge o.b = b.n) \bullet (\exists\, s' \mid (\exists\, o' \bullet s'.n = o'.s \wedge o'.b = b.n) \bullet$
“resultaten van s en s' in b verschillen”)) $\bullet s.n\}$

$=$ [shunting, tweemaal]

$\{s \mid (\forall\, b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists\, s', o' \mid s'.n = o'.s \wedge o'.b = b.n \bullet$
“resultaten van s en s' in b verschillen”)) $\bullet s.n\}$

$=$ [bij gegeven $o, o' \mid s.n = o.s \wedge s'.n = o'.s \wedge o.b = b.n = o'.b$ is “...”$=(o.r \neq o'.r)$]

$\{s \mid (\forall\, b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists\, s', o' \mid s'.n = o'.s \wedge o'.b = b.n \bullet o.r \neq o'.r)) \bullet s.n\}$

$=$ [shunting]

$\{s \mid (\forall\, b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists\, o' \mid (\exists\, s' \bullet s'.n = o'.s) \wedge o'.b = b.n \bullet o.r \neq o'.r)) \bullet s.n\}$

$=$ [in O is s foreign key verwijzend naar $Ship(n)$]

$\{s \mid (\forall\, b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists\, o' \mid o'.b = b.n \bullet o.r \neq o'.r)) \bullet s.n\}$

$= \{s \mid (\forall\, b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists\, o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$

$=$ [shunting]

$\{s \mid (\forall\, o \mid s.n = o.s \wedge (\exists\, b \bullet o.b = b.n) \bullet (\exists\, o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$

$=$ [in O is b foreign key verwijzend naar $B(n)$]

$\{s \mid (\forall\, o \mid s.n = o.s \bullet (\exists\, o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$

$= \{s \mid \neg\, (\exists\, o \mid s.n = o.s \bullet \neg\, (\exists\, o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$

$=$ select $s.n$ from S s where not exists (

select $*$ from O o where $s.n = o.s$ and not exists (

select $*$ from O o' where $o'.b = o.b$ and $o.r \neq o'.r$)) *(Zie Toelichting.)*

5p. **Opgave 10.** Leg uit wat *Physical Data Independence* betekent.

Onleesbare tekst wordt fout gerekend.

De eigenschap dat een applicatieprogrammeur zich niet bezig hoeft te houden met de manier/structuur waarop data fysiek op schijf wordt opgeslagen (het *physical schema*), maar in plaats daarvan de data en bewerkingen in een *conceptual schema* (dat wil zeggen, het relationele model, geformaliseerd in SQL) kan uitdrukken. Het voordeel hiervan is dat de manier van fysieke dataopslag kan wijzigen zónder dat er iets in de applicatie-programma's gewijzigd hoeft worden. Zie Hoofdstuk 3 van Kifer's boek.

Hoe of waardoor wordt *Physical Data Independence* bereikt?

Onleesbare tekst wordt fout gerekend.

Het DBMS verzorgt de vertaling van het *conceptual schema* (SQL) naar het *physical schema*.

5p. **Opgave 11.** Noem de vier isolatieniveaus, en noem voor ieder de anomalie die in dat niveau (en sterkere niveaus) onmogelijk is.

Onleesbare tekst wordt fout gerekend.

In volgorde van toenemende sterkte:

- read uncommitted: geen anomalieen onmogelijk
- read committed: dirty reads onmogelijk
- repeatable read: unrepeatable reads (en lost updates) onmogelijk
- serializable: phantoms onmogelijk.

Toelichtingen

Toelichting bij antwoord 1.
De volgende entiteiten en eigenschappen zijn in de casus gegeven:

entiteittypen *ABlid*, *DBlid*, *Helper*;
$DBlid \cap ABlid = \varnothing$, $ABlid \cap Helper = \varnothing$ (dus ook $DBld \cap Helper = \varnothing$)
entiteittype $Raadgevendlid = Wijzeman = ABlid \setminus DBlid$;
(dus $Raadgevendlid \cap DBlid = \varnothing$ en $Raadgevenlid \cup DBlid = ABlid$)
relatie *steunt* : $Helper$——$_{0..1}ABlid$
relatie *stuurt* : $DBlid_{1..*}$——$Helper$
relatie *draagtvoor* : $ABlid_{1..*}$——$Raadgevendlid$

Met ieder paar in *draagtvoor* is bovendien een *motivatie* geassocieerd.

Het entiteittype *Lid* is in de uitwerking verzonnen om gemakkelijk $ABlid \cap Helper = \varnothing$ aan te geven. Desgewenst kun je ook een van de volgende twee herkennen in de casus (en bijgevolg opnemen in het ER-diagram):

relatie *geeftraad* : $ABlid$———$DBlid$, of
relatie *geeftraad* : $Raadgevendlid$———$DBlid$.

De multipliciteit van *steunt* : $Helper$——$_{0..1}ABlid$ is expliciet in de casus gegeven; desgewenst kun je stellen dat een helper die geen hulp biedt geen helper is, en dus de volgende multipliciteit aangeven: *steunt* : $Helper$———$_1ABlid$.

Toelichting bij antwoord 2.
Relatie C is per definitie identiek aan relatie D; het zijn synoniemen. En er is expliciet gevraagd zo weinig mogelijk tabellen te gebruiken.

De generalisatie/specialisatie E wordt gerepresenteerd door de foreign key eigenschap in tabel F. De F-instanties worden niet opgenomen in de tabel voor de D-instanties omdat dan NULLs nodig zouden zijn (en dat mag niet volgens de opgave) om een D-instantie te representeren die geen F-instantie is.

Er is geen aparte tabel nodig om de instanties van de G-relatie te representeren; vanwege de '1'-multipliciteit van relatie G kan een G-instantie met als sleutelwaarde (x, y) (d.w.z., x is de waarde van sleutel-van-F, en y is de waarde van sleutel-van-H) gerepresenteerd worden in tabel H door de rij met precies x en y als waarden van de attributen sleutel-van-F en sleutel-van-H (en in de gegeven uitwerking is $(a1, b1)$ de sleutel-van-F, en is $(h1)$ de sleutel-van-H). De multipliciteit 0..1 van de relatie G wordt weergegeven door de eigenschap *uniq*(sleutel-van-F) in schema H: een waarde van sleutel-van-F kan in hoogstens één rij van tabel H voorkomen.

Een andere manier om de multipliciteit 0..1 van de relatie G weer te geven is door $h1$ op te nemen in F met de eigenschap 'foreign key $(h1)$ references $H(h1)$'. Maar dan is een NULL waarde nodig indien er een F-instantie gerepresenteerd moet worden die *niet* met een H-instantie is geassocieerd. Er mogen volgens de opgave geen NULLs nodig zijn.

Omdat alle foreign keys not null zijn, kan in schema H de eigenschap 'primary key $(h1)$, unique $(a1, b1)$' ook geformuleerd worden als 'check (h1 is not null), uniq $(h1)$, primary key $(a1, b1)$'.

Bedenk dat 'foreign key (x, y) references $Z(x, y)$' een *sterkere* eigenschap is dan 'foreign key (x) references $Z(x)$, foreign key (y) references $Z(y)$', en bovendien vereist de laatste

uitdrukking dat zowel x als ook y een key is in Z (terwijl die eerste uitdrukking alleen maar vereist dat (x, y) een sleutel is in Z)!

Toelichting bij antwoord 4.
Let op: iedere key is ook een superkey, omgekeerd is dit niet het geval.
BCD is de enige key van $\mathbf{R}$; geen van B, C, D kan met FDs geproduceerd worden (en samen produceren ze wel ieder van A, E, F).

Toelichting bij antwoord 8.
Hier is een informele verklaring van de constructie; links staat join van $O\ o1$ met $O\ o2$, verdeeld in groepen (groepering op het criterium "gelijke $o1.s$"), en rechts staat per groep de opgeleverde rij:

$o1.s$	$o1.b$	$o1.r$	$o2.s$ ($\neq o1.s$)	$o2.b$ ($=o1.b$)	$o2.r$ ($=sunk$)
$s1$	b_{1a}	...	s_{11}	b_{1a}	$sunk$
			s_{12}	b_{1a}	$sunk$
	b_{1b}		s_{13}	b_{1b}	$sunk$
			⋮		
	b_{1c}		s_{17}	b_{1c}	$sunk$
$s2$	b_{2a}	...	s_{21}	b_{2a}	$sunk$
			s_{22}	b_{2a}	$sunk$
			⋮		
	b_{2e}		s_{29}	b_{2e}	$sunk$
$s3$	b_{3a}	...	s_{31}	b_{3a}	$sunk$
			s_{32}	b_{3a}	$sunk$
	b_{3b}		s_{33}	b_{3b}	$sunk$
	b_{3c}		s_{34}	b_{3c}	$sunk$

ship	*aantal*
$s1$	7
$s2$	9
$s3$	4

Vanwege de groepering op $o1.s$ zijn de $s1, s2, s3, ...$ onderling verschillend. Vanwege het feit dat in O het paar (s, b) een key is, zijn *per groep* alle b_{xy} onderling verschillend. Onder aanname dat een schip hooguit éénmaal kan zinken, zijn *per groep* alle s_{xy} verschillend, en kan dus 'count($o2.s$) as a' opgeleverd worden en zelfs 'count($*$) as a'. Vanwege het feit dat groepen per definitie niet-leeg zijn, is per groep het aantal a groter dan nul (en hoeft dit niet in de having clausule getest te worden).

Wanneer niet gegeven was 'Sla de schepen s over waarvoor dat aantal a nul is', dan hadden *alle* schepen opgeleverd moeten worden (sommigen met aantal 0) en was een oplossing met een group-by nauwelijks mogelijk.

De SQL formulering kan ook iets minder compact, namelijk met een overbodige '$S\ s1$' erbij (te elimineren vanwege $s1.n = o1.s$):

```
select s1.n, count (distinct o2.s) as a          -- eventueel: count(o2.s), of: count(*)
from S s1, O o1, O o2
where o1.b = o2.b and s1.n = o1.s<>o2.s and o2.r = sunk
group by s1.n
having count (o1.b) >= 20
```

Nog minder compact, namelijk met overbodige $s1, s2, b$:

select $s1.n$, count (distinct $o2.s$) as a -- eventueel: count($o2.s$), of: count($*$)
from S $s1$, S $s2$, B b, O $o1$, O $o2$
where $o1.b = b.n = o2.b$ and $s1.n = o1.s <> o2.s = s2.n$ and $o2.r = sunk$
group by $s1.n$
having count ($o1.b$) >= 20

Toelichting bij antwoord 9.
Uit de een-na-laatste formulering in verzamelingsnotatie volgen onmiddellijk ook de volgende:
$\{s \mid (\forall\, o \mid s.n = o.s \bullet o.b \in \{o' \mid o.r \neq o'.r \bullet o'.b\}) \bullet s.n\}$
$\{s \mid \{o \mid s.n = o.s \bullet o.b\} \subseteq \{o' \mid o.r \neq o'.r \bullet o'.b\} \bullet s.n\}$
$\{s \mid \{o \mid s.n = o.s \bullet o.b\} \setminus \{o' \mid o.r \neq o'.r \bullet o'.b\} = \varnothing \bullet s.n\}$
De eerste en laatste hiervan kunnen ook direct in SQL omgezet worden (met behulp van *in* en *except*).