

Tentamen Gegevensbanken (19.211074.1) — 3 februari 2012

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 volgende casus gaat over trainingscursussen die door een bedrijf georganiseerd worden. De cursussen worden gevolgd door studenten en gegeven door docenten.

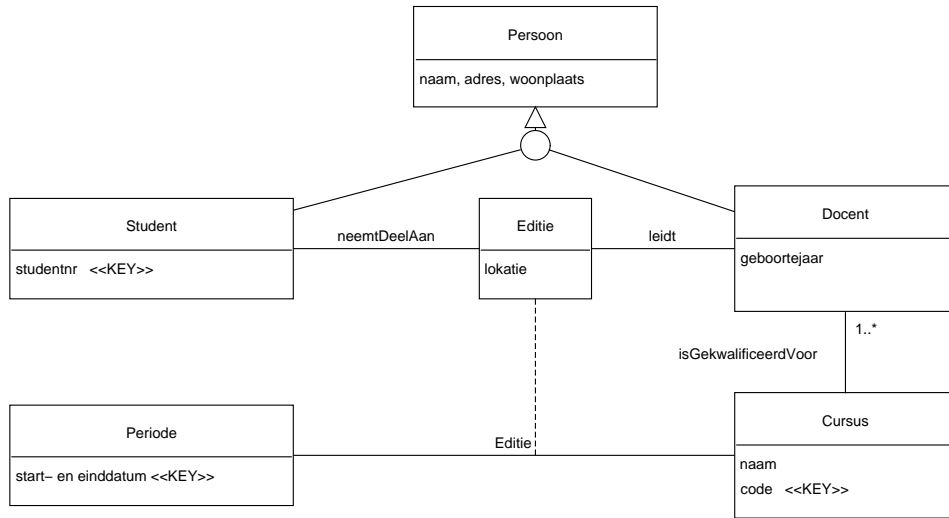
- Iedere cursus heeft een naam en een code. Iedere cursus kan verscheidene malen gegeven worden; iedere keer dat een cursus gegeven wordt spreken we van een 'editie' van de cursus. Een editie van een cursus is volkomen bepaald door de cursus en een start- en einddatum.
- Een student kan *deelnemen aan* een editie, en *volgt* dan bijhorende cursus.
- Een docent kan *gekwalificeerd* zijn voor een cursus, en kan edities *leiden* van cursussen waarvoor hij gekwalificeerd is. Voor iedere cursus is er minstens één docent die daarvoor gekwalificeerd is.
- Van studenten en docenten is de naam, het adres en de woonplaats bekend. Van studenten is bovendien een (identificerend) studentnummer bekend, en van docenten hun geboortjaar.
- Van elke cursuseditie is de lokatie bekend, alsmede het aantal studenten dat eraan deelneemt.

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



s volgt c == (exists e. neemtDeelAan e & e=Editie(..., c))

d leidt e & e=Editie(..., c) ==> d isGekwalificeerdVoor c

Periode is een zelfstandig entiteitstype om Editie goed te kunnen modelleren, maar in de casus is Periode zelf niet van belang als entiteitstype.

(Zie Toelichting.)

Hier is ruimte voor eventuele toelichting (en verklaring van verschillen tussen casus en ERD):

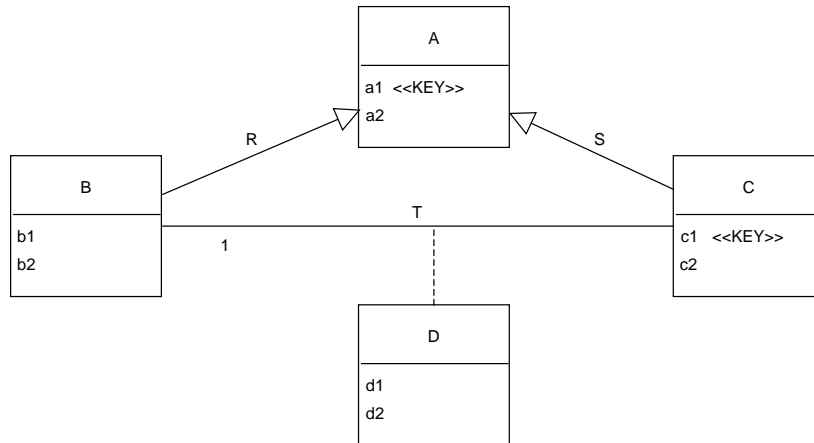
Het aantal studenten dat aan een editie deelneemt is te bepalen uit de relatie *neemtDeelAan*, en is dus niet nodig als apart attribuut in *Editie*. Het is fout om een attribuut *aantalDeelnemers* in *Editie* te hebben als niet tegelijk het verband met de relatie *neemtDeelAan* wordt vermeld. Relatie *volgt* tussen *Student* en *Cursus* mag ook expliciet aangegeven worden, maar ook dan moet vermeld worden dat *volgt* louter een samenstelling is van *neemtDeelAan* en *Editie*.

Toegestaan met toelichting: *covering* en *disjoint* aanduidingen bij de specialisatieknoop. Deze eigenschappen zijn wel realistisch, maar staan niet letterlijk gegeven in de casustekst.

10p.



Opgave 2. Beschouw het volgende ER-diagram in de notatie van de UML (niet-geschreven multipliciteiten staan voor 0..*):



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 zonodig ook CHECKs.

Onleesbare tekst wordt fout gerekend.

$A(a_1, a_2,$	
primary key (a1))	
$B(a_1, b_1, b_2,$	
primary key (a1),	
foreign key (a1) references A(a1))	
$C(a_1, c_1, c_2, a_1', d_1, d_2,$	
primary key (a1), unique (c1), check (c1 is not null)	
-- primary key (c1), unique (a1), check (a1 is not null)	alternatief voor vorige regel
foreign key (a1) references A(a1),	
foreign key (a1') references B(a1), check (a1' is not null))	(Zie Toelichting.)

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:

$$\bar{R} = ABCDE$$

$$\mathcal{F} = \{AB \rightarrow CD, D \rightarrow CE, E \rightarrow A\}$$

- (1) Geef in het antwoordblok in iedere genummerde 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 de *sleutels* van \mathbf{R} .
- (3) Onderstreep in het antwoordblok de *supersleutels* 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.

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

(Zie Toelichting.)

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

$$\mathcal{F} = \{AB \rightarrow E, C \rightarrow D, D \rightarrow A, E \rightarrow BC\}$$

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: AB, BC, BD, E . Alleen functionele afhankelijkheden $C \rightarrow D$ en $D \rightarrow A$ zijn een schending van de BCNF-conditie voor \mathbf{R} want voor hen (en voor geen ander lid van \mathcal{F}) geldt: die FD is niet-triviaal en het linkerlid ervan omvat geen sleutel.

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.

We passen één stap van het BCNF-algoritme toe (dat garandeert losslessness en vermindering van het aantal FD's die de BCNF-conditie schenden).

Neem een FD uit \mathcal{F} die de BCNF-conditie schendt; bijvoorbeeld $C \rightarrow D$.

Splits \bar{R} in $\bar{R}_1 = CD$ (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:

$$\mathbf{R}_1 = (CD, \{C \rightarrow D\})$$

$$\mathbf{R}_2 = (ABCE, \{AB \rightarrow E, E \rightarrow BC, C \rightarrow A\})$$

NB1. $C \rightarrow A$ zit niet in \mathcal{F} maar wel in \mathcal{F}^+ en komt dus in \mathcal{F}_2 .

NB2. $D \rightarrow A$ zit niet in $\mathcal{F}_1 \cup \mathcal{F}_2$ en zelfs niet in $(\mathcal{F}_1 \cup \mathcal{F}_2)^+$ en blijft dus niet behouden.

NB3. \mathbf{R}_1 staat in BCNF (de key is C) en \mathbf{R}_2 bevat precies één schending van zijn BCNF-conditie, namelijk $C \rightarrow A$; samen hebben \mathbf{R}_1 en \mathbf{R}_2 dus minder schendingen dan \mathbf{R} .

NB4. De decompositie is verliesvrij want $\bar{R}_1 \cap \bar{R}_2 = C$, is een key van één van $\mathbf{R}_1, \mathbf{R}_2$ (namelijk van \mathbf{R}_1). (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 kort maar zódanig dat het voor de corrector duidelijk is hoe u te werk gaat.

Let op: $BC \rightarrow E$ volgt uit \mathcal{F} , en zal dus (bij een correcte redenering) een FD worden van elke component die tenminste BCE bevat. Net zo: $C \rightarrow A$ volgt uit \mathcal{F} en zal dus een FD worden van elke component die tenminste AC bevat. Dit geldt ook net zo voor $BD \rightarrow E$, $E \rightarrow A$, $E \rightarrow B$, $E \rightarrow D$, en andere.

We passen het BCNF-algoritme toe.

- Een eerste stap is al gegeven in de vorige vraag, met als resultaat $\mathbf{R}_1 = (\bar{R}_1, \mathcal{F}_1) = (CD, \{C \rightarrow D\})$ en $\mathbf{R}_2 = (\bar{R}_2, \mathcal{F}_2) = (ABCE, \{AB \rightarrow E, E \rightarrow BC, C \rightarrow A\})$.

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

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

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

In \mathbf{R}_2 is $C \rightarrow A$ een schending van de BCNF-conditie (en geen andere FD van \mathcal{F}_2): die FD is niet-triviaal en het linkerlid is geen sleutel in \mathbf{R}_2 (want $C_{\mathcal{F}_2}^+ = A \neq ABCE$). We kiezen dus $C \rightarrow A$ ter eliminatie. Dus splitsen we \bar{R}_2 in $\bar{R}_{2a} = AC$ en $\bar{R}_{2b} = \cancel{A}BCE = BCE$. 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 $\mathbf{R}_{2a} = (AC, \{C \rightarrow A\})$ en $\mathbf{R}_{2b} = (BCE, \{BC \rightarrow E, E \rightarrow BC\})$. Let op: de $BC \rightarrow E$ zit weliswaar niet in \mathcal{F}_2 , maar toch in \mathcal{F}_{2b} zoals aan het begin is uitgelegd.

- We bekijken nu $\mathbf{R}_{2a} = (AC, \{C \rightarrow A\})$. Deze staat in BCNF.

- We bekijken nu $\mathbf{R}_{2b} = (BCE, \{BC \rightarrow E, E \rightarrow BC\})$. Zowel BC als ook E is een key van \mathbf{R}_{2a} , dus staat \mathbf{R}_{2a} in BCNF.

- Dus $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b}\}$ is een decompositie van \mathbf{R} waarvan alle componenten in BCNF staan.

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

$$"ABCDE = CD \bowtie AC \bowtie BCE \text{ geldt in } \mathbf{R}."$$

Wanneer je met een andere schending begint kun je mogelijk een andere decompositie bereiken.
(Zie Toelichting.)

Is iedere functionele afhankelijkheid behouden gebleven onder bovenstaande decompositie van het oorspronkelijke schema \mathbf{R} ? Zo nee, geef een FD die niet behouden is:

$D \rightarrow A$ is niet behouden gebleven, al in de eerste stap.

10p.

Opgave 5. Bij een tennisvereniging spelen de volgende begrippen een rol: lidcode (van een lid van de vereniging), functie (die een lid in de vereniging vervult), rechten (horend bij een functie), speelsterkte (van een lid), telefoonnummer (van een lid), provider (van een telefoonabonnement van een lid — met deze kennis probeert de vereniging telefoneerkosten te besparen), e-mailadres (van een lid), betalingsachterstand (van een lid betreffende de contributie), etc.

We definiëren een relatie \mathcal{R} die deze entiteiten aan elkaar relateert.

Een tuple (L, F, R, S, T, P, E, B) zit op tijdstip t in relatie \mathcal{R} precies wanneer op tijdstip t het volgende geldt:

1. L : is de Lidcode van een lid (L identificeert een lid)
2. F : is een van de functies die lid L vervult
3. R : is een van de rechten die hoort bij functie F
4. S : is de speelsterkte van lid L
5. T : is een van de telefoonnummers van lid L
6. P : is de provider van telefoonnummer T
7. E : is een van de e-mail adressen van lid L
8. B : is de betalingsachterstand (in euro's) van lid L

Toelichting:

9. ieder lid vervult tenminste één functie.
10. voor sommige functies zijn er géén rechten
11. ieder lid heeft tenminste één e-mailadres
12. mogelijk delen sommige leden samen een telefoonnummer
13. mogelijk delen sommige leden samen een e-mailadres

Geef in het antwoordblok voor ieder van de functionele afhankelijkheden (FD) en multi-valued dependencies (MVD), met een letter W of O aan of die *Waar* of *Onwaar* is in relatie \mathcal{R} .

Motiveer kort uw keuze zo *mogelijk* met verwijzing naar gegevens 1 . . 13.

De motivatie telt even zwaar mee in de beoordeling als het antwoord *Waar/Onwaar*.

FD/MVD	W/O	motivatie
$L \rightarrow FRSTPEB$	O	een lid L kan meerdere combinaties van ⟨functie, rechten, telnr, provider, e-mailadres en betalingsachterstand⟩ hebben (een lid kan zelfs al meerdere functies hebben, etc)
$FRSTPEB \rightarrow L$	O	een $FRSTPEB$ combinatie kan bij verschillende leden horen (zie ook 12 en 13)
$L \rightarrow S$	W	een lid L heeft één speelsterkte (zie 4 — dit geldt op ieder tijdstip, zie de definitie van \mathcal{R})
$ET \rightarrow L$	O	een ET kan, volgens 12 en 13, gedeeld worden door meerdere leden
$B \rightarrow L$	O	een betalingsachterstand B kan bij verschillende leden horen
$LTE = LT \bowtie LE$	W	telefoonnummers en e-mailadressen van een lid komen in alle combinaties voor in \mathcal{R}
$LSBE = LSB \bowtie LE$	W	$LSB \cap LE (= L)$ is een key in LSB

(Zie Toelichting.)

10p.

Opgave 6. Geef een definitie (of nauwkeurige beschrijving) van de volgende begrippen.

System catalog:

Onleesbare tekst wordt fout gerekend.

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

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

Referential integrity:

Onleesbare tekst wordt fout gerekend.

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

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

OLAP, of alleen het karakteristieke verschil tussen OLAP en OLTP

(OLAP = Online Analytical Processing, OLTP = Online Transaction Processing):

Onleesbare tekst wordt fout gerekend.

Zie hoofdstuk 1. OLAP is het gebruik en de analyse van gegevens in een database voor ondersteuning van management beslissingen. De queries duren soms heel lang, vergeleken met transacties die een database up-to-date houden (OLTP), en behoeven in het algemeen geen accurate data te hebben.

DBA (= Database Administrator):

Onleesbare tekst wordt fout gerekend.

Zie hoofdstuk 1.

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

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*)

Samengevat:

<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>)

Ter herinnering:

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 7.** Beschouw de volgende zoekopdracht:

Vind iedere klasse waarvan er minstens een schip bij een zeeslag betrokken is.

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; dat moet je gebruiken. Kort identifiers tot hun eerste letter af, en schrijf desgewenst c voor $c : C$, etc.

Onleesbare tekst wordt fout gerekend.

"iedere klasse waarvan er minstens een schip bij een zeeslag betrokken is"
= $\{c : C \mid \text{"een schip van } c \text{ is bij een zeeslag betrokken"} \bullet c.n\}$
= $\{c : C \mid (\exists s : S \mid "s \text{ is van klasse } c" \bullet (\exists b : B \bullet "s \text{ is betrokken in } b")) \bullet c.n\}$
= $\{c : C \mid (\exists s : S \mid s.c = c.n \bullet (\exists b : B \bullet (\exists o : O \mid o.b = b.n \bullet o.s = s.n))) \bullet c.n\}$
= $\{c : C \mid (\exists s : S \mid s.c = c.n \bullet (\exists b : B; o : O \mid o.b = b.n \bullet o.s = s.n)) \bullet c.n\}$
= [shunting]
$\{c : C \mid (\exists s : S \mid s.c = c.n \bullet (\exists o : O \mid (\exists b : B \bullet o.b = b.n) \bullet o.s = s.n)) \bullet c.n\}$
= [In O is b een foreign key naar $B(n)$, dus voor iedere o geldt: $\exists b : B \bullet o.b = b.n$]
$\{c : C \mid (\exists s : S \mid s.c = c.n \bullet (\exists o : O \mid true \bullet o.s = s.n)) \bullet c.n\}$
= $\{c : C \mid (\exists s : S \mid s.c = c.n \bullet (\exists o : O \bullet o.s = s.n)) \bullet c.n\}$
= $\{c : C \mid (\exists s : S; o : O \mid s.c = c.n \bullet o.s = s.n) \bullet c.n\}$
= [shunting]
$\{c : C; s : S; o : O \mid s.c = c.n \wedge o.s = s.n \bullet c.n\}$
= $\{c : C; s : S; o : O \mid s.c = c.n \wedge o.s = s.n \bullet s.c\}$
= [shunting]
$\{s : S; o : O \mid (\exists c : C \bullet s.c = c.n) \wedge o.s = s.n \bullet s.c\}$
= [In S is c een foreign key naar $C(n)$, dus voor iedere s geldt: $\exists c : C \bullet s.c = c.n$]
$\{s : S; o : O \mid true \wedge o.s = s.n \bullet s.c\}$
= $\{s : S; o : O \mid o.s = s.n \bullet s.c\}$

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

Onleesbare tekst wordt fout gerekend.

```
select distinct s.c
from S s, O o
where O.s = s.n
```

Geef een formulering van de vraag in Relationale Algebra, zonder het Cartesisch product \times te gebruiken maar in plaats daarvan alleen de natural join \bowtie (die koppelt rijen precies wanneer gelijknamige attributen gelijke waarden hebben):

Onleesbare tekst wordt fout gerekend.

```
 $\pi_c(S' \bowtie O')$ 
hierbij is met renaming gedefinieerd:  $S' = S[sn, c, l]$  en  $O' = O[sn, b, r]$ .
```

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

Geef voor ieder land waarvoor er tenminste drie klassen zijn, deze twee gegevens: (i) het aantal schepen van dat land dat in een zeeslag is gezonken, en (ii) de datum van de vroegste zeeslag waaraan een schip van dat land heeft deelgenomen.

Het vergelijken van data d, d' mag je uitdrukken met $d < d'$ en $d \leq d''$ en $\min(d, d')$, etc.

Onleesbare tekst wordt fout gerekend.

```
select c.country, count (distinct s.n), min (b.date)
from C c, S s, O o, B b
where c.name = s.class and s.name = o.shipname and o.battlename = b.name
group by c.country
having count (distinct c.name)  $\geq$  3
```

Indien een schip maar éénmaal kan zinken (een realistische veronderstelling, die echter niet in de casus vermeld is), dan is in de query het deel “count (distinct s.name)” gelijkwaardig met “count (s.name)” en zelfs met “count (*)”.

Ter herinnering:

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> , b, r)

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.) Beschouw de volgende vraag:

Geef ieder schip s waarvoor geldt dat ieder ouder schip s' heeft deelgenomen aan een zeeslag waaraan ook s heeft deelgenomen.

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.

(We korten $s : S$ af tot s , et cetera. Alleen de laatste regel wordt gevraagd.)

$\{s \mid \text{"ieder ouder schip } s' \text{ is betrokken in een zeeslag waarin ook } s \text{ is betrokken"} \bullet s.n\}$

$= \{s \mid (\forall s' \mid \text{"}s' \text{ ouder dan } s" \bullet (\exists b \mid \text{"}s \text{ betrokken in } b" \bullet \text{"}s' \text{ betrokken in } b")) \bullet s.n\}$

$= \{s \mid (\forall s' \mid s'.l < s.l \bullet (\exists b \mid (\exists o \bullet b.n = o.b \wedge o.s = s.n) \bullet (\exists o' \bullet b.n = o'.b \wedge o'.s = s'.n))) \bullet s.n\}$

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.

	(Alleen de laatste regel wordt gevraagd)
=	[shunting, predicate logic]
=	$\{s \mid (\forall s' \mid s'.l < s.l \bullet (\exists b; o; o' \mid b.n = o.b \wedge o.s = s.n \bullet b.n = o'.b \wedge o'.s = s'.n)) \bullet s.n\}$
=	$\{s \mid (\forall s' \mid s'.l < s.l \bullet (\exists b; o; o' \mid b.n = o.b \wedge o.s = s.n \bullet o.b = o'.b \wedge o'.s = s'.n)) \bullet s.n\}$
=	[shunting]
=	$\{s \mid (\forall s' \mid s'.l < s.l \bullet (\exists o; o' \mid (\exists b \bullet b.n = o.b) \wedge o.s = s.n \bullet o.b = o'.b \wedge o'.s = s'.n)) \bullet s.n\}$
=	[In O is b een foreign key naar $B(n)$, dus voor iedere o geldt: $\exists b \bullet b.n = o.b$]
=	$\{s \mid (\forall s' \mid s'.l < s.l \bullet (\exists o; o' \mid true \wedge o.s = s.n \bullet o.b = o'.b \wedge o'.s = s'.n)) \bullet s.n\}$
=	$\{s \mid (\forall s' \mid s'.l < s.l \bullet (\exists o; o' \mid o.s = s.n \bullet o.b = o'.b \wedge o'.s = s'.n)) \bullet s.n\}$
=	$\{s \mid \neg (\exists s' \mid s'.l < s.l \bullet \neg (\exists o; o' \mid o.s = s.n \bullet o.b = o'.b \wedge o'.s = s'.n)) \bullet s.n\}$
=	
=	
=	
=	
=	
=	
=	

Formuleer de vraag in SQL, *zonder overbodige subqueries en tabellen*:

Onleesbare tekst wordt fout gerekend.

=	select $s.n$ from S s where not exists (
	select * from S s' where $s'.l < s.l$ and not exists (
	select * from O o , O o' where $o.s = s.n$ and $o.b = o'.b$ and $o'.s = s'.n$))

10p.



Opgave 10. Een fietsenmaker bestelt regelmatig fietsen bij de fabrikant. Zodra een bestelling geleverd wordt, neemt het aantal in bestelling zijnde fietsen af, terwijl de voorraad fietsen juist toeneemt. Schematisch ziet de reeks van achtereenvolgens uitgevoerde operaties (tezamen: transactie T) voor een levering van 10 fietsen er als volgt uit:

Transactie T : $lees(B, b)$
 $schrijf(B, b - 10)$
 $lees(V, v)$
 $schrijf(V, v + 10)$

Hierbij zijn B en V attributen van de bestellingentabel, en b en v zijn programmavariabelen in de code van de transactie.

Leg uit wat er verstaan wordt onder *durability*:

Onleesbare tekst wordt fout gerekend.

De effecten van een transactie blijven behouden, ook al crasht het systeem of is er stroomuitval, of zoiets (lang nadat de transactie voltooid is). (Herstel gaat door <i>replay</i> : het logboek wordt 'voorwaarts' nagedaan, dus er staan steeds de nieuwe waarden, <i>after images</i> .)

Om *durability* te realiseren gebruikt het databasesysteem onder andere een logboek, de *log*. Geef in de kolom 'waarde B ' en 'waarde V ' de waarden die B en V na afloop van de betreffende operatie hebben in de database (de beginwaarden zijn gegeven) en in de kolom 'logboek' de gegevens die in het logboek geschreven worden om *durability* te kunnen realiseren:

Onleesbare tekst wordt fout gerekend.

operatie	waarde B	waarde V	logboek
	20	50	
$lees(B, b)$	20	50	$\langle T, \text{begin} \rangle$
$schrijf(B, b-10)$	10	50	$\langle T, B, 10 \rangle$ ("Nieuwe waarde van B is 10")
$lees(V, v)$	10	50	
$schrijf(V, v+10)$	10	60	$\langle T, V, 60 \rangle$ ("Nieuwe waarde van V is 60")
	10	60	$\langle T, \text{commit} \rangle$

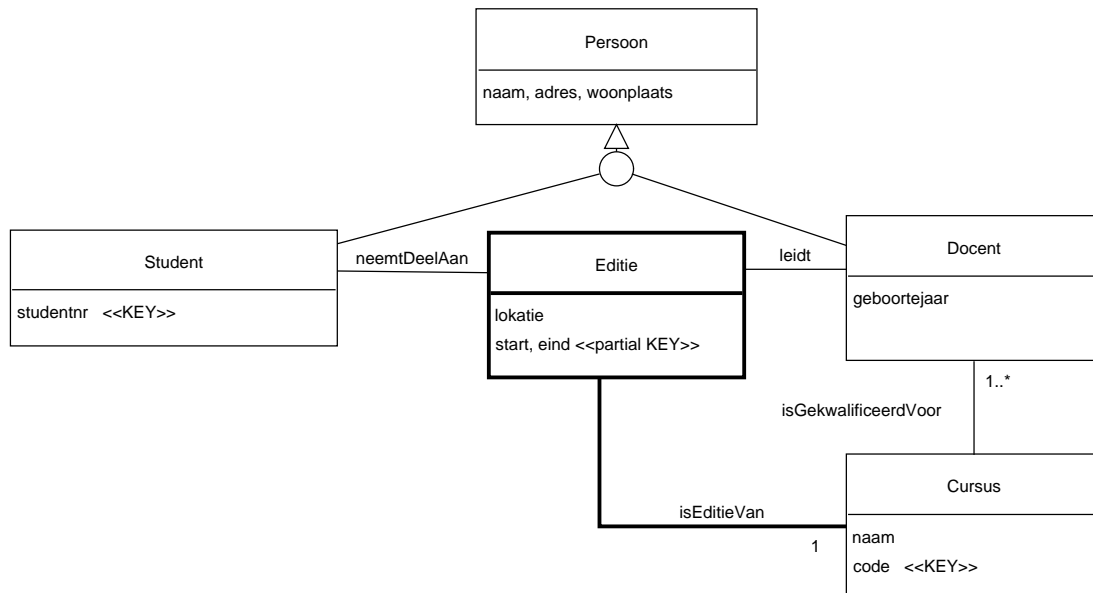
Toelichtingen

Toelichting bij antwoord 1.

Merk op dat *Student* niet alleen *studentnr* als key heeft, maar ook de key van *Persoon* (wat die key ook moge zijn); daarentegen heeft *Docent* alleen maar de key van *Persoon* als key.

Per definitie bestaat de key van *Editie* uit enerzijds de/een key van *Periode* en anderzijds de/een key van *Cursus*. Het wordt fout aangerekend als óók nog *cursuscode* als attribuut van *Editie* wordt opgegeven.

Een alternatief is om *Editie* niet een associatie-entiteit te laten zijn, maar een zelfstandige entiteit met extra attributen *start-* en *einddatum*, en een relatie (genaamd *isEditieVan*) met precies één *Cursus*. In feite is *Editie* dan een zwakke entiteit — wat in de figuur is aangegeven met dikke lijken. Het is dan fout om óók nog *cursuscode* als attribuut van *Editie* op te geven. En ook is het dan fout om alleen maar *start-* en *einddatum* als key te definiëren, want de cursus behoort ook tot de key. Attributen *start-* en *einddatum* zijn wel déél van de key; dit is automatisch in de gegeven oplossing maar niet meer in dit alternatief.



$s \text{ volgt } c \iff (\text{exists } e. \text{ neemtDeelAan } e \ \& \ e=\text{Editie}(\dots, c))$

$d \text{ leidt } e \ \& \ e=\text{Editie}(\dots, c) \implies d \text{ isGekwalificeerdVoor } c$

Toelichting bij antwoord 2.

T en *D* zijn identiek, maar staan op twee manieren getekend in het diagram. Het is in strijd met de opdracht “geen redundantie” om voor *D* of *T* een aparte tabel te hebben, want vanwege de multipliciteit kan alle *D-* en *T-*data in *C* opgeslagen worden.

De tabel voor *A* kan niet weggelaten worden door zijn attributen in *B* en *C* op te nemen, om twee redenen: (1) het is niet gegeven dat *B* en *C* tezamen *A* overdekken, dus kan sommige *A*-data niet opgeslagen worden, en (2) het is niet gegeven dat *B* en *C* elkaar niet overlappen, dus wordt sommige *A*-data gedupliceerd.

De relaties R, S, T, D zijn redundant:

Relatie R (gezien als een verzameling van paren keys) is als volgt te verkrijgen uit de database:

$R = \text{select } a.a1 \text{ as keyA, } b.a1 \text{ as keyB from } A a, B b \text{ where } a.a1 = b.a1;$

Net zo voor S, T :

$S = \text{select } a.a1 \text{ as keyA, } c.a1 \text{ as keyC from } A a, C c \text{ where } a.a1 = c.a1;$

$T = \text{select } c.a1' \text{ as keyB, } c.a1 \text{ as keyC from } C c;$

Relatie D gezien als een verzameling van paren keys, is identiek aan T .

Relatie D gezien als een verzameling van paren keys tezamen met attributen $d1, d2$, is als volgt te verkrijgen uit de database:

$D = \text{select } c.a1' \text{ as keyB, } c.a1 \text{ as keyC, } c.d1 \text{ as d1, } c.d2 \text{ as d2 from } C c;$

Toelichting bij antwoord 3.

NB 1: Sleutels zijn attribuutverzamelingen; zo iets als " $ABC \rightarrow \dots$ " is geen sleutel (maar misschien wel een functionele afhankelijkheid). Het omcirkelen van de hele functionele afhankelijkheid wordt fout gerekend als alleen de sleutel omcirkeld moet worden. Net zo voor het onderstrepen van de supersleutels.

NB 2: Iedere sleutel is ook een supersleutel; omgekeerd niet. Dus ook sleutels moeten onderstreept worden wanneer gevraagd wordt de supersleutels te onderstrepen.

Toelichting bij antwoord 4.

Een andere decompositie wordt verkregen door een andere schending van \mathbf{R} te kiezen ter eliminatie, bijvoorbeeld $D \rightarrow A$:

Splits \bar{R} in $\bar{R}_1 = AD$ (alle attributen van de FD) en $\bar{R}_2 = BCDE$ (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:

$$\mathbf{R}_1 = (AD, \{D \rightarrow A\})$$

$$\mathbf{R}_2 = (BCDE, \{BD \rightarrow E, E \rightarrow BC, C \rightarrow D\})$$

NB1. $BD \rightarrow E$ zit niet in \mathcal{F} maar wel in \mathcal{F}^+ en komt dus in \mathcal{F}_2 .

NB2. $AB \rightarrow E$ zit niet in $\mathcal{F}_1 \cup \mathcal{F}_2$ en zelfs niet in $(\mathcal{F}_1 \cup \mathcal{F}_2)^+$ en blijft dus niet behouden.

NB3. \mathbf{R}_1 staat in BCNF (de key is D) en \mathbf{R}_2 bevat precies één schending van de BCNF-conditie voor \mathbf{R}_2 , namelijk $C \rightarrow D$; samen dus minder dan het aantal schendingen in \mathbf{R} .

NB4. De decompositie is verliesvrij want $\bar{R}_1 \cap \bar{R}_2 = D$, is een key van één van $\mathbf{R}_1, \mathbf{R}_2$ (namelijk van \mathbf{R}_1).

Je kunt ook een schending uit $\mathcal{F}^+ \setminus \mathcal{F}$ nemen, bijvoorbeeld $C \rightarrow AD$.

Toelichting bij antwoord 4.

Een andere decompositie wordt verkregen door in de eerste stap een andere schending te elimineren, met name $D \rightarrow A$. Het resultaat hiervan staat in de toelichting op het vorige antwoord:

$$\mathbf{R}_1 = (AD, \{D \rightarrow A\}) \text{ en } \mathbf{R}_2 = (BCDE, \{BD \rightarrow E, E \rightarrow BC, C \rightarrow D\}).$$

- We bekijken nu $\mathbf{R}_1 = (AD, \{D \rightarrow A\})$. Deze staat in BCNF.
- We bekijken nu $\mathbf{R}_2 = (BCDE, \{BD \rightarrow E, E \rightarrow BC, C \rightarrow D\})$. In \mathbf{R}_2 is $C \rightarrow D$ een schending van de BCNF-conditie (en geen andere FD van \mathcal{F}_2): die FD is niet-triviaal en het linkerlid is geen sleutel want $C_{\mathcal{F}_2}^+ = D \neq \bar{R}_2$. Dus kiezen we $C \rightarrow D$ ter eliminatie. Dus splitsen we \bar{R}_2

in $\bar{R}_{2a} = CD$ en $\bar{R}_{2b} = BC \not\rightarrow E = BCE$. 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 $\mathbf{R}_{2a} = (CD, \{C \rightarrow D\})$ en $\mathbf{R}_{2b} = (BCE, \{BC \rightarrow E, E \rightarrow BC\})$. Let op: $BC \rightarrow E$ zit weliswaar niet in \mathcal{F}_2 maar toch in \mathcal{F}_{2b} zoals aan het begin van het antwoord is uitgelegd.

- Zowel \mathbf{R}_{2a} als ook \mathbf{R}_{2b} staan in BCNF.
- Dus $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b}\}$ is een decompositie van \mathbf{R} waarvan alle componenten in BCNF staan:

$$"ABCDE = AD \bowtie CD \bowtie BCE \text{ geldt in } \mathbf{R}."$$

Toelichting bij antwoord 5.

Bij de tweede FD is soms de volgende motivatie gegeven: " F is niet uniek voor L ". Dit is een foute motivatie, want waar het om gaat is dat de combinatie $FRSTPEB$ uniek is, of niet, voor L .