

Tentamen Gegevensbanken (19.211074.1) — 28 oktober 2010

CONTROLEER EERST OF ALLE BLADZIJDEN T/M BLZ. 14 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	10	bonus	11
-------------	---	---	---	---	---	---	---	---	---	----	-------	----

10p.

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

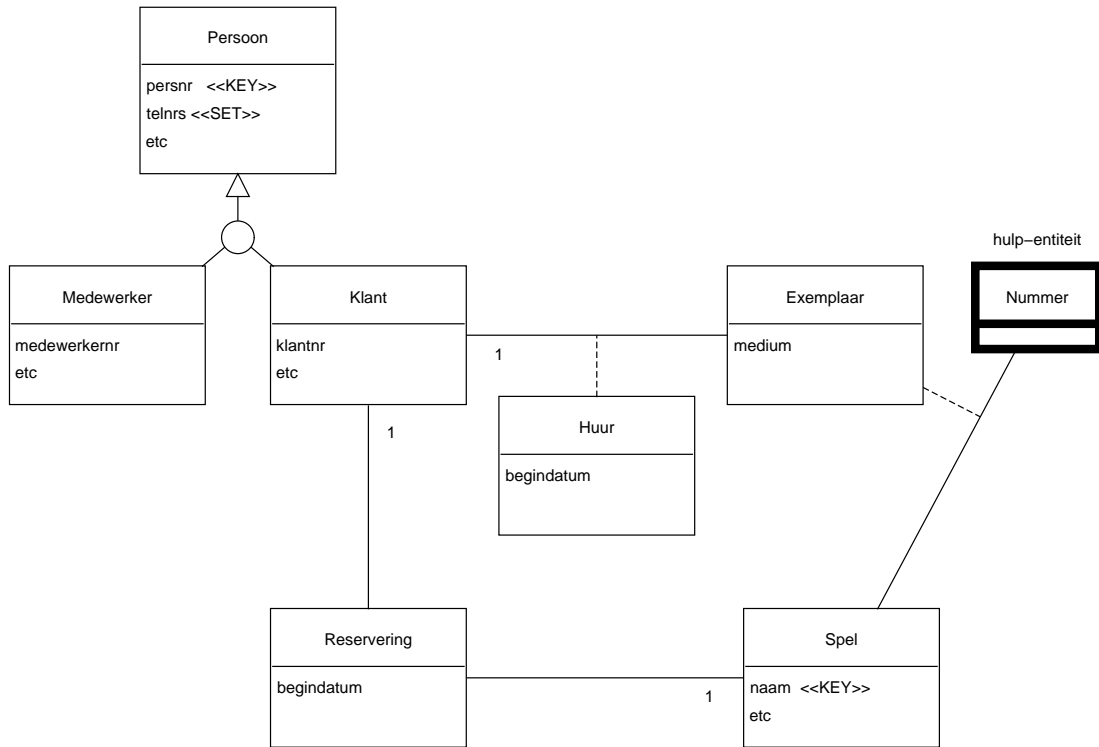
- De relevante personen zijn klanten en medewerkers. Van iedere persoon is de naam bekend, alsmede het adres, en verscheidene telefoonnummers. Van iedere klant is, naast de persoonsgegevens, een klantnummer bekend en een rekeningnummer. Van iedere medewerker is, naast de persoonsgegevens, een medewerkersnummer bekend en het jaar van aanstelling. Het is toegestaan dat medewerkers ook klant zijn. Om personen te identificeren, wordt er voor iedere persoon (klant, medewerker) een uniek persoonsnummer aangemaakt en opgeslagen.
- Van ieder spel is de naam bekend, de uitgever, de huurprijs en een beschrijving. Er zijn geen twee verschillende spellen met eenzelfde naam. Van een spel zijn er verscheidene (nul of meer) exemplaren; per spel worden de exemplaren van elkaar onderscheiden door een exemplaarnummer (denk aan: 001, 002, 003, ...). Van ieder exemplaar is bekend of het verhuurd is en wat het soort medium is (weergegeven als een tekst).
- Reserveringen (door klanten) en verhuur (aan klanten) gaan per periode; een periode wordt gekarakteriseerd door één datum (de begindatum), de duur is vast (en doet niet ter zake in deze opgave). Een *spel* kan op één moment méermalen gereserveerd zijn, zelfs voor eenzelfde periode en zelfs door eenzelfde klant. Een *exemplaar* kan op één moment hooguit éénmaal verhuurd zijn.

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

- iedere instantie van het ERD beschrijft een mogelijke “werkelijkheid *op één tijdstip*”,
- iedere mogelijke “werkelijkheid *op één tijdstip*” kan gerepresenteerd worden als een instantie van het ERD,
- het ERD is opgebouwd met *zo geschikt mogelijke* constructies.

Zowel de ERD-notatie uit het boek als ook de UML notatie (class diagram) is toegestaan, maar een mengeling van beide niet. Doe het eerst op kladpapier en dan pas in het net.

Niet-geschreven multipliteiten staan voor 0..*.



Zie ook de uitgebreide toelichting elders.

(Zie Toelichting.)

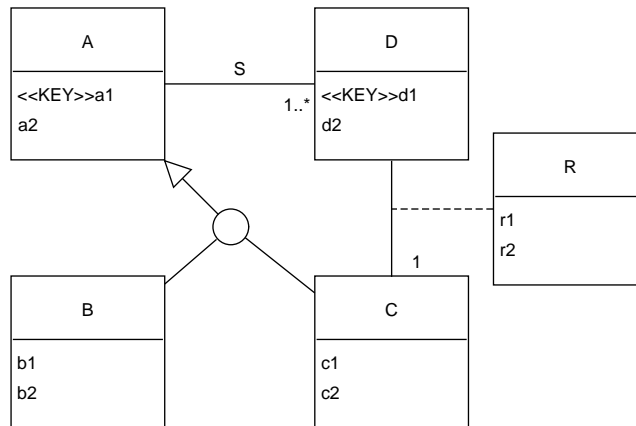
Hier is ruimte voor eventuele toelichting:

In *Medewerker* kan het attribuut *medewerkernr* desgewenst ook key zijn; dit is niet echt duidelijk in de casustekst. Net zo voor *klantnr* in *Klant*.

De specialisatie/generalisatie mag desgewenst *covering* verklaard worden omdat er in de casus geen andere personen een rol spelen dan medewerkers en klanten.

Een exemplaar is verhuurd precies wanneer het in de relatie *Huur* zit. Wanneer er een apart attribuut *verhuurd* wordt opgenomen, dan moet er ergens (in woorden of formules) vermeld worden: $e.verhuurd \Leftrightarrow \exists k \bullet (k, e) \in Huur$ (want anders zijn er instantie mogelijk die niet een werkelijkheid representeren).

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



(Bij de generalisatie/specialisatie is ‘disjointness’ en ‘covering’ *niet* gegeven. Dus *B* en *C* kunnen dezelfde instanties hebben, en *A* kan instanties hebben die niet tot *B* of *C* behoren.)

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:

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

U mag afkortingen gebruiken, zoals “PK” voor “primary key”, etc., en er mogen CHECKs voorkomen.

Onleesbare tekst wordt fout gerekend.

<code>A(a1 PK, a2, CHECK (a1 in (select a1 from S)))</code>
<code>B(a1 PK, b1, b2, FK (a1) REF A(a1))</code>
<code>C(a1 PK, c1, c2, FK (a1) REF A(a1))</code>
<code>D(d1 PK, d2, r1, r2, a1 NOT NULL, FK (a1) REF C(a1))</code>
<code>S(a1, d1, PK (a1, d1), FK (a1) REF A(a1))</code>
<i>(Zie Toelichting.)</i>

10p.

Opgave 3. Beschouw het relatieschema $\mathbf{R} = (\bar{R}, \mathcal{F})$, waarbij de attribootverzameling \bar{R} en de verzameling \mathcal{F} van functionele afhankelijkheden als volgt luiden:

$$\bar{R} = ABCDE$$

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

- (1) Geef in het antwoordblok in iedere regel een zo groot mogelijk rechterlid Y zó dat de functionele afhankelijkheid $X \rightarrow Y$ volgt uit de hierboven gegeven verzameling \mathcal{F} (met andere woorden: Y is de closure $X_{\mathcal{F}}^+$). U hoeft de leden van X niet op te nemen in Y .
- (2) Omcirkel bij iedere $X \rightarrow Y$ het deel X en/of Y wanneer dat *sleutel* is van \mathbf{R} .
- (3) Onderstreep bij iedere $X \rightarrow Y$ het deel X en/of Y wanneer dat *supersleutel* is van \mathbf{R} .
- (4) Omcirkel de *nummers* van de functionele afhankelijkheden die een schending vormen van de BCNF-eis.

Onleesbare tekst wordt fout gerekend.

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

(Zie Toelichting.)

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

$$\mathcal{F} = \{AB \rightarrow D, \quad C \rightarrow B, \quad DE \rightarrow C\}$$

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 drie sleutels, namelijk: ABE, ACE, ADE . Voor iedere FD in \mathcal{F} geldt: die FD is een schending van de BCNF-conditie, omdat die niet-triviaal is en het linkerlid ervan geen sleutel omvat.

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

Onleesbare tekst wordt fout gerekend.

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

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

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

($ABE \rightarrow C$ zit niet in \mathcal{F} , maar wel in \mathcal{F}^+ en bevat alleen maar attributen uit \bar{R}_2 .)

\mathbf{R}_1 staat in BCNF en \mathbf{R}_2 bevat maar één schending van de BCNF-conditie (zie het antwoord op devierde vraag van deze opgave; het volgt ook uit het feit dat hier één iteratiestap van het BCNF-algoritme is toegepast.)

Zijn er in bovenstaande decompositiestap van \mathbf{R} naar $\mathbf{R}_1, \mathbf{R}_2$ functionele afhankelijkheden verloren gegaan? Zo ja, geef dan zo'n functionele afhankelijkheid.

Onleesbare tekst wordt fout gerekend.

Afhankelijkheid $DE \rightarrow C$ zit wel in \mathcal{F} maar niet in $(\mathcal{F}_1 \cup \mathcal{F}_2)^+$, en is dus verloren gegaan.

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

Onleesbare tekst wordt fout gerekend.

We passen het BCNF-algoritme toe.

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

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

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

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

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

Schema \mathbf{R}_{2a} en \mathbf{R}_{2b} staan beide in BCNF. De FDs van \mathbf{R}_2 zijn niet allemaal behouden: $ABE \rightarrow C$ is verloren gegaan.

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

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

“ $ABCDE = ABD \bowtie BC \bowtie ACE$ geldt in \mathbf{R} .”

Wanneer je met een andere schending begint kun je een andere decompositie bereiken:

Beginnen we met eliminatie van schending $DE \rightarrow C$ uit \mathbf{R} , dan krijgen we:

- $\mathbf{R}_1 = (CDE, \{DE \rightarrow C\})$ — staat in BCNF

- $\mathbf{R}_2 = (ABDE, \{AB \rightarrow D, DE \rightarrow B\})$

$DE \rightarrow B$ zit niet in \mathcal{F} maar wel in \mathcal{F}^+ en bevat alleen maar attributen uit \bar{R}_2 .

$C \rightarrow B$ is niet behouden gebleven; die zit wel in \mathcal{F} maar niet in $(\mathcal{F}_1 \cup \mathcal{F}_2)^+$.

$AB \rightarrow D$ en $DE \rightarrow B$ vormen beide een schending van de BCNF-conditie voor \mathbf{R}_2 .

\mathbf{R}_2 moet nog verder gedeconponeerd worden.

⋮

Beginnen we met eliminatie van schending $C \rightarrow B$ uit \mathbf{R} , dan krijgen we:

- $\mathbf{R}_1 = (BC, \{C \rightarrow B\})$ — staat in BCNF.

- $\mathbf{R}_2 = (ACDE, \{DE \rightarrow C, AC \rightarrow D\})$

$AC \rightarrow D$ zit niet in \mathcal{F} maar wel in \mathcal{F}^+ en bevat alleen maar attributen uit \bar{R}_2 .

$AB \rightarrow D$ is niet behouden gebleven; die zit wel in \mathcal{F} maar niet in $(\mathcal{F}_1 \cup \mathcal{F}_2)^+$.

$DE \rightarrow C$ en $AC \rightarrow D$ vormen beide een schending van de BCNF-conditie voor \mathbf{R}_2 .

\mathbf{R}_2 moet nog verder gedeconponeerd worden.

⋮

(Zie Toelichting.)

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

Onleesbare tekst wordt fout gerekend.

Nee; $DE \rightarrow C$ is verloren gegaan (in de 1ste stap).

10p.

Opgave 5. Bij een webshop voor boeken spelen verscheidene entiteiten een rol: boek, winkelwagen, klant, magazijn, etc. We definiëren een relatie \mathcal{R} die die entiteiten aan elkaar relateert, op de volgende manier.

Een tuple (B, A, W, K, N, V, M) zit op tijdstip t in relatie \mathcal{R} precies wanneer op tijdstip t al het volgende geldt:

1. B is een *Boek*-identificatie (bijvoorbeeld: een isbn-nummer).
4. A is het *Aantal* exemplaren van boek B dat in winkelwagen W zit.
5. W is een *Winkelwagen*.
6. K is een *Klant* die winkelwagen W gebruikt.
7. N is een naam van klant K .
8. V is de *Voorraad* (aantal exemplaren) van boek B in magazijn M .
9. M is een *Magazijn*.

Neem het volgende aan:

- a. Een klant kan verscheidene winkelwagens tegelijk gebruiken.
- b. Verschillende klanten gebruiken verschillende winkelwagens.
- c. Een klant heeft precies één naam.

Geef voor ieder van de functionele afhankelijkheden hieronder, met een letter W of O aan of die *Waar* of *Onwaar* is in relatie \mathcal{R} , en motiveer kort uw keuze met verwijzing naar 1..9 en a..c. (De motivatie telt even zwaar mee in de beoordeling als het antwoord W/O.)

Onleesbare tekst wordt fout gerekend.

FD	W/O	Motivatie voor uw keuze
$BW \rightarrow A$	W	Volgt uit gegeven 4.
$WA \rightarrow B$	O	Verschillende boeken kunnen in gelijke hoeveelheid in één W liggen.
$K \rightarrow N$	W	Volgt uit aanname c. (Volgt niet uit 7: " N is een naam van K ".)
$K \rightarrow W$	O	Dit is gegeven a: " $K \not\rightarrow W$ "
$BM \rightarrow V$	W	Volgt uit gegeven 8.
$B \rightarrow V$	O	In verschillende magazijnen kan de V voor B verschillend zijn.
$N \rightarrow K$	O	Verschillende klanten kunnen gelijke namen hebben.
$W \rightarrow K$	W	Dit is aanname b.

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

Onleesbare tekst wordt fout gerekend.

MVD	W/O	
$BTPAWKNVM = BTPAW \bowtie AWKN \bowtie KNVM$	O	$BM \rightarrow V$ wordt geschonden
$BTPAWKNVM = BTP \bowtie AW \bowtie KN \bowtie VM$	O	$W \rightarrow K$ wordt geschonden
$BTPAWKNVM = BTPWAKN \bowtie BMV$	W	
$BTPWAKN \cap BMV (= B)$ is key in een component, nml. "schema" BMV (Zie Toelichting.)		

5p. **Opgave 6.** Een bedrijf heeft een database met personeelsgegevens. De database bevat een tabel T met het volgende schema:

(*Naam ...*, *Adres ...*, *Afd ...*, *Salaris ...*, *.....*)

De secretaresse van de afdeling Inkoop heeft username U voor de database. Het is gewenst dat de secretaresse lees-toegang heeft (en niet meer dan dat) tot de *Naam* en *Adres*-velden van de personeelsleden van afdeling 'Inkoop'. Geef de SQL statements waarmee deze toegangsrechten voor U gerealiseerd worden:

Onleesbare tekst wordt fout gerekend.

```
create view X as (select Naam, Adres from T where Afd='Inkoop');  
grant select on X to U;
```

Zie het boek, Ch 2 (en slide 9 van GB2010-hc3a-Ch3+5.3-handouts.pdf).

5p. **Opgave 7.** Definieer of beschrijf of verklaar de volgende begrippen.

Physical Data Independence:

Onleesbare tekst wordt fout gerekend.

Zie het boek (en slide 6 van GB2010-hc3a-Ch3+5.3-handouts.pdf).

Referential Integrity / Referential Integrity constraint:

Onleesbare tekst wordt fout gerekend.

Zie het boek, sectie 2.2, 2.3, 3.2.2 (en slide 5 van GB2010-hc3a-Ch3+5.3-handouts.pdf).

In de volgende opgavenserie wordt het volgende databaseschema gebruikt:

Class (*name*, *type*, *country*, *guns*, *bore*, *displacement*)

Ship (*name*, *classname*, *launched*)

Battle (*name*, *date*)

Outcome (*shipname*, *battlename*, *result*)

De attributen die tot de sleutel behoren zijn onderstreept.

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

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

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

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

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

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

C (*n*, *t*, *c*, *g*, *b*, *d*)

S (*n*, *c*, *l*)

B (*n*, *d*)

O (*s*, *b*, *r*)

10p. **Opgave 8.** Beschouw de volgende vraag:

Welke schepen zijn, volgens de database, in twee of meer zeeslagen betrokken?

Geef voor deze vraag een afleiding in Verzamelingsnotatie in stappen naar een vorm die dicht aansluit bij en met . Het begin is al gegeven. Kort de identifiers tot hun eerste letter af.

Onleesbare tekst wordt fout gerekend.

	“de schepen die in minstens twee zeeslagen betrokken zijn”
=	$\{s : S \mid \text{“}s \text{ is in minstens twee zeeslagen betrokken”} \bullet s.n\}$
=	$\{s : S \mid (\exists b_1, b_2 : B \bullet \text{“}s \text{ is betrokken in } b_1 \text{ en } b_2\text{”}) \bullet s.n\}$
=	$\{s : S \mid (\exists b_1, b_2 : B \bullet b_1 \neq b_2 \wedge \text{“}s \text{ is betrokken in } b_1 \text{ en } b_2\text{”}) \bullet s.n\}$
=	$\{s : S \mid (\exists b_1, b_2 : B \bullet b_1 \neq b_2 \wedge (\exists o_1, o_2 : O \bullet o_1.b = b_1.n \wedge o_2.b = b_2.n \wedge o_1.s = s.n = o_2.s)) \bullet s.n\}$
=	$\{s : S \mid (\exists b_1, b_2 : B; o_1, o_2 : O \bullet b_1 \neq b_2 \wedge o_1.b = b_1.n \wedge o_2.b = b_2.n \wedge o_1.s = s.n = o_2.s)\} \bullet s.n\}$
=	[In B is n een primary key]
	$\{s : S \mid (\exists b_1, b_2 : B; o_1, o_2 : O \bullet b_1.n \neq b_2.n \wedge o_1.b = b_1.n \wedge o_2.b = b_2.n \wedge o_1.s = s.n = o_2.s)\} \bullet s.n\}$
=	$\{s : S \mid (\exists b_1, b_2 : B; o_1, o_2 : O \bullet o_1.b \neq o_2.b \wedge o_1.b = b_1.n \wedge o_2.b = b_2.n \wedge o_1.s = s.n = o_2.s)\} \bullet s.n\}$
=	$\{s : S \mid (\exists o_1, o_2 : O \bullet o_1.b \neq o_2.b \wedge (\exists b_1, b_2 : B \bullet o_1.b = b_1.n \wedge b_2.n = o_2.b) \wedge o_1.s = s.n = o_2.s)\} \bullet s.n\}$
=	[in O is b een FK naar $B(n)$; dus bestaan de b_1 en b_2]
	$\{s : S \mid (\exists o_1, o_2 : O \bullet o_1.b \neq o_2.b \wedge o_1.s = s.n = o_2.s)\} \bullet s.n\}$
=	$\{s : S; o_1, o_2 : O \mid o_1.b \neq o_2.b \wedge o_1.s = s.n = o_2.s \bullet s.n\}$
=	$\{s : S; o_1, o_2 : O \mid o_1.b \neq o_2.b \wedge o_1.s = s.n = o_2.s \bullet o_1.s\}$
=	$\{o_1, o_2 : O \mid o_1.b \neq o_2.b \wedge (\exists s : S \bullet o_1.s = s.n) \wedge o_1.s = o_2.s \bullet o_1.s\}$
=	[$O.o$ is foreign key naar $S.n$; dus bestaat de s]
	$\{o_1, o_2 : O \mid o_1.b \neq o_2.b \wedge o_1.s = o_2.s \bullet o_1.s\}$
	Desgewenst kan $s : S$ afgekort worden tot s , en net zo voor o_1, o_2, b_1, b_2 .

Geef een SQL formulering van de beschouwde vraag; de SQL formulering moet dicht aansluiten bij de zojuist gegeven uitdrukking (en dus geen subqueries hebben):

Onleesbare tekst wordt fout gerekend.

	select distinct $o.s$ from O $o1$, O $o2$ where $o1.b \neq o2.b$ and $o1.s = o2.s$
=	(zie vierde regel van onder in bovenstaande afleiding)
	select distinct $s.n$ from S s , O $o1$, O $o2$ where $o1.b \neq o2.b$ and $o1.s = s.n$ and $s.n = o2.s$
	<i>(Zie Toelichting.)</i>

Geef ook een formulering van de vraag in Relationele Algebra; u mag géén Cartesisch product \times gebruiken (maar wél de natural join \bowtie).

Onleesbare tekst wordt fout gerekend.

	$\pi_s(\sigma_{o1b \neq o2b}(O' \bowtie O''))$	$= \pi_s(\sigma_{o1b \neq o2b}(S \bowtie O' \bowtie O''))$
	Hierbij is met renaming gedefinieerd: $O' = O[s, o1b, o1r]$ en $O'' = O[s, o2b, o2r]$.	

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 9.** Formuleer in SQL met een group-by query:



Geef, voor iedere zeeslag waarbij er minstens 10 landen betrokken zijn, de eerste te waterlating (*launched*) van de schepen die bij die zeeslag betrokken zijn.

We zeggen dat een land *L* *betrokken* is bij een zeeslag *B* als er een schip betrokken is bij *B* en de klasse van dat schip heeft *country = L*.

Onleesbare tekst wordt fout gerekend.

```
select o.battlename, min (s.launched)
```

```
from O o, S s, C c
```

```
where o.shipname = s.name and s.classname = c.name
```

```
group by o.battlename
```

```
having count (distinct c.country) ≥ 10
```

Afkortingen zijn toegestaan

Een extra tabel *B b* in de from-clause geeft geen puntenaftrek

5p.

Opgave 10. (Goede beantwoording levert 5 bonuspunten boven op de punten die voor deze opgave gegeven worden. Daardoor kan het totaal aantal behaalde punten boven de 100 uitkomen.) Beschouw de volgende vraag:

Geef ieder land met de eigenschap dat er in *iedere* zeeslag waaraan dat land deelneemt, een schip gezonken is.

(We zeggen “een land L neemt deel aan zeeslag B ” als: er een schip deelneemt aan zeeslag B en het land van de klasse van dat schip is L .) 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.

$\{c_0 \mid (\forall b \mid (\exists s \bullet (\exists o \bullet b.n = o.b \wedge o.s = s.n) \wedge (\exists c \bullet s.c = c.n \wedge c.c = c_0.c))$
$\bullet (\exists s' \bullet (\exists o' \bullet b.n = o'.b \wedge o'.s = s'.n \wedge o'.r = \text{sunk})) \bullet c_0.c\}$
<i>(Zie Toelichting.)</i>

Formuleer de vraag in verzamelingsnotatie op een manier die zo *direct mogelijk* aansluit bij de SQL query die u zo dadelijk gaat geven. (Een afleiding op kladpapier zou u hierbij kunnen helpen.)

Onleesbare tekst wordt fout gerekend.

$\{c_0 \mid \neg (\exists o, s, c \mid o.s=s.n \wedge s.c=c.n \wedge c.c=c_0.c \bullet \neg (\exists o' \bullet o.b=o'.b \wedge o'.r=\text{sunk})) \bullet c_0.c\}$

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

Onleesbare tekst wordt fout gerekend.

<pre>select distinct c0.c from C c0 where not exists (select * from O o, S s, C c where o.s = s.n ∧ s.c = c.n ∧ c.c = c0.c and not exists (select * from O o' where o'.r = sunk))</pre>

10p.

Opgave 11. Beschouw het relatieschema $\mathbf{R} = (ABCDE, \emptyset)$; dus de attribuutverzameling is $ABCDE$ en er zijn geen functionele afhankelijkheden. (Het domein van ieder attribuut is: de verzameling van gehele getallen.)

Voor dit schema bekijken we de decompositie in de twee schema's (ABC, \emptyset) en (CDE, \emptyset) . Gevraagd wordt een relatie-instantie r te geven en dan daarmee aan te tonen dat de decompositie niet lossless is (dus *wel* "informatie verliest").

Geef de relatie-instantie r , zonder toelichting of motivatie (de waarden voor A en B zijn al gegeven; er zijn precies twee rijen in r):

Onleesbare tekst wordt fout gerekend.

$r =$	A	B	C	D	E
	0	0	c	2	2
	1	1	c	3	3

Toon aan met behulp van de gegeven r dat de decompositie niet lossless is:

Onleesbare tekst wordt fout gerekend.

De decompositie van r bestaat uit de twee volgende relatie-instanties r_1 en r_2 :

$$r_1 = \pi_{ABC}(r) = \begin{array}{|c|c|c|} \hline A & B & C \\ \hline 0 & 0 & c \\ \hline 1 & 1 & c \\ \hline \end{array} \quad \text{en} \quad r_2 = \pi_{CDE}(r) = \begin{array}{|c|c|c|} \hline C & D & E \\ \hline c & 2 & 2 \\ \hline c & 3 & 3 \\ \hline \end{array}$$

De natural join $r_1 \bowtie r_2$ van deze twee is niet gelijk aan de oorspronkelijke r :

$$r_1 \bowtie r_2 = \begin{array}{|c|c|c|c|c|} \hline A & B & C & D & E \\ \hline 0 & 0 & c & 2 & 2 \\ \hline 0 & 0 & c & 3 & 3 \\ \hline 1 & 1 & c & 2 & 2 \\ \hline 1 & 1 & c & 3 & 3 \\ \hline \end{array} \neq r$$

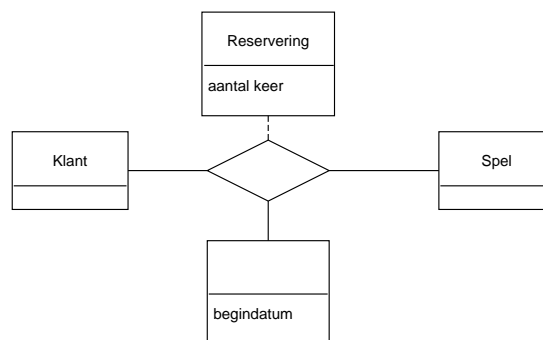
(Zie Toelichting.)

Toelichtingen

Toelichting bij antwoord 1.

- Het is fout om *Exemplaar* als een specialisatie te modelleren van *Spel*. Het zou onder andere betekenen dat er minstens zoveel spellen bestaan als exemplaren.
- Het diagram modelleert, zoals gevraagd(!), de casus-werkelijkheid van één tijdstip. Met name: een exemplaar wordt aan hoogstens één klant tegelijk verhuurd; dat wordt weergegeven met de multipliciteit 0..1 bij *Klant* in de relatie *Huur*.
- In de klassieke ERD-notatie kan een *Exemplaar* als zwakke entiteit weergegeven worden (dus met vette lijnen getekend): een exemplaar kan niet bestaan zonder het bijhorende *Spel* en het exemplaarnummer is slechts een *partial key*. In de UML-notatie is het beter (mooier, explicieter) om *Exemplaar* als een relatie-entiteit weer te geven tussen *Spel* en een hulp-entiteit *Exemplaarnummer*. In ons diagram wordt een exemplaar gekarakteriseerd door een nummer en spel (samen de key) en daarnaast nog het medium; nummer is dus een partial key van *Exemplaar* en is op zichzelf géén key van *Exemplaar*, want exemplaren van verschillende spellen kunnen eenzelfde nummer hebben. Entiteit *Nummer* is op zichzelf niet interessant (er zal geen vraag zijn naar “alle mogelijke exemplaarnummers”), wat aangegeven wordt door de dikke lijnen (liever had ik dat met stippelijnen gedaan).
- Er kunnen verschillende reserveringen r en r' bestaan, met eenzelfde begindatum, die gerelateerd zijn aan eenzelfde klant en spel; precies zoals uitdrukkelijk vermeld in de casus. Zonder de multipliciteiten ‘1’ zou een reservering door een aantal klanten *gezamenlijk* gedaan kunnen worden, en een aantal spellen *gezamenlijk* in één reservering betrokken kunnen zijn.

Een alternatief voor de modellering van reserveringen is deze:



Wanneer een (klant,spel,begindatum)-combinatie in relatie *Reservering* zit, geeft *aantal_keer* aan hoe vaak dat spel door die klant voor die datum gereserveerd is (want dat aantal moet groter dan één kunnen zijn, volgens de casus).

- Het is niet nodig (en niet gewenst) om *Telefoonnummer* als een aparte entiteit te modelleren (er zal geen vraag zijn naar “alle mogelijke telefoonnummers”). Wanneer er wel een aparte entiteit *Telefoonnummer* is, zou opgemerkt moeten worden dat de entiteit

zelf niet interessant is (net als exemplaar *Nummer*) maar wel de relatie tussen *Persoon* en *Telefoonnummer*. In die relatie kan met multipliciteiten aangegeven worden dat een telefoonnummer bij hoogstens één persoon hoort — wat overigens niet gegeven is in de casus.

Beoordelingsrichtlijn. 2 punt aftrek voor: identificatie van spel en exemplaar, foute modellering van reservering of verhuur, reservering of verhuur als relatie-entiteit *Periode* gemodelleerd, “foreign keys” opgenomen in de entiteiten; 1 punt aftrek voor: multipliciteiten bij een specialisatie/generalisatie en bij een relatie-entiteit, geen vermelding van de gegeven keys, bewering dat exemplaarnummer een key is van *Exemplaar*, redundante attributen zonder expliciete vermelding van de redundantie, foute multipliciteiten.

Toelichting bij antwoord 2.

De NOT NULL bij de $a1$ in D is nodig om de relatie R juist te representeren. De eigenschap NOT NULL bij onderdelen van een primary key is overbodig, omdat die al volgt uit de primary key eigenschap. De eigenschap NOT NULL bij $a2$ in A , $b1, b2$ in B , $c1, c2$ in C , $d2, r1, r2$ in D is niet nodig: als er in de tabellen rijen met null waarden zitten, dan zouden die ook precies zo als null waarden in instanties van de entiteitstypen zitten.

Omdat er bij iedere $(d1, d2)$ in D precies één $(a1, c1, c2)$ in C is die volgens R gerelateerd is aan $(d1, d2)$, kan de sleutel van die C -instantie, namelijk $a1$, tezamen met de R -attribuutwaarden, bij die $(d1, d2)$ worden opgenomen in D ter representatie van relatie R .

Het is onmogelijk om op soortgelijke manier R op te nemen in C , of S op te nemen in A of in D .

De CHECK in tabel A garandeert de eigenschap $\forall a:A \bullet \exists s:S \bullet a.a1 = s.a1$, in het ERD weergegeven door de multipliciteit $1..*$ bij relatie S . Nota bene: in A is $a1$ géén foreign key verwijzend naar $S(a1)$, want $a1$ is op zichzelf geen key in S .

Beoordelingsrichtlijn. 3 punt aftrek voor foute modellering van de $1..*$ -multipliciteit van relatie S , en voor foute modellering van de 1-multipliciteit van relatie R (bijvoorbeeld: geen opname van R in D); 2 punt aftrek voor foute modellering van de generalisatie/specialisatie.

Toelichting bij antwoord 3.

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

Iedere key is ook superkey.

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

Toelichting bij antwoord 4.

Beoordelingsrichtlijn. 2 punt voor deel 1, 3 punt voor deel 2, 1 punt voor deel 3, $3\frac{1}{2}$ punt voor deel 4, $\frac{1}{2}$ punt voor deel 5.

Toelichting bij antwoord 5.

“FD F wordt geschonden” betekent: er zijn instanties/tabelvullingen mogelijk (die niet in

tegenspraak zijn met F) waarbij de join van de componenten een instantie/tabelvulling oplevert die de FD F tegenspreekt. Wanneer dit het geval is, kan de join dus niet gelijk zijn aan de oorspronkelijke instantie/tabelvulling.

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

Beoordelingsrichtlijn. 9 punt voor deel 1 (1 punt aftrek voor ieder fout antwoord W/O en voor iedere foute motivatie; 1 punt voor deel 2 (alleen te krijgen als alles goed is).

Toelichting bij antwoord 8.

Er is een heel andere formulering in SQL mogelijk, namelijk:

```
select o.s from O o group by o.s having count (*) > 1
```

Maar deze sluit niet dicht aan bij de formulering in verzamelingsnotatie.

Beoordelingsrichtlijn. 7 punt voor de afleiding (3 punt aftrek voor een vorm die in SQL subqueries geeft of meer dan drie tabellen in de from-clause; 2 punt aftrek voor te grote afleidingsstap), $1\frac{1}{2}$ punt voor de correcte omzetting van het resultaat van deel 1 naar SQL, $1\frac{1}{2}$ punt voor de formulering in relational algebra.

Toelichting bij antwoord 10.

De formulering kan als volgt afgeleid worden (tweede regel), en verder bewerkt worden tot het antwoord op de tweede deelvraag:

$$\begin{aligned}
& \{c_0 \mid (\forall b \mid (\exists s \bullet \text{“}s \text{ neemt deel aan } b\text{”} \wedge \text{“het land van de klasse van } s \text{ is } c_0.c\text{”} \\
& \quad \bullet (\exists s' \bullet \text{“}s' \text{ is gezonken in } b\text{”})) \bullet c_0.c\} \\
= & \{c_0 \mid (\forall b \mid (\exists s \bullet (\exists o \bullet b.n = o.b \wedge o.s = s.n) \wedge (\exists c \bullet s.c = c.n \wedge c.c = c_0.c)) \\
& \quad \bullet (\exists s' \bullet (\exists o' \bullet b.n = o'.b \wedge o'.s = s'.n \wedge o'.r = \text{sunk}))) \bullet c_0.c\} \\
= & \quad [\text{herhaaldelijk shunting}] \\
& \{c_0 \mid (\forall b, o, s, c \mid b.n = o.b \wedge o.s = s.n \wedge s.c = c.n \wedge c.c = c_0.c \\
& \quad \bullet (\exists s', o' \bullet b.n = o'.b \wedge o'.s = s'.n \wedge o'.r = \text{sunk})) \bullet c_0.c\} \\
= & \quad [\text{substitutie van gelijken voor gelijken en tweemaal shunting}] \\
= & \{c_0 \mid (\forall o, s, c \mid (\exists b \bullet b.n = o.b) \wedge o.s = s.n \wedge s.c = c.n \wedge c.c = c_0.c \\
& \quad \bullet (\exists o' \bullet o.b = o'.b \wedge (\exists s' \bullet s'.n = o'.s) \wedge o'.r = \text{sunk})) \bullet c_0.c\} \\
= & \quad [\text{in } O \text{ is } b \text{ FK naar } B(n), \text{ dus bestaat de vereiste } b \text{ voor } o] \\
& \quad [\text{in } O \text{ is } s \text{ FK naar } S(n), \text{ dus bestaat de vereiste } s' \text{ voor } o'] \\
& \{c_0 \mid (\forall o, s, c \mid o.s = s.n \wedge s.c = c.n \wedge c.c = c_0.c \bullet (\exists o' \bullet o.b = o'.b \wedge o'.r = \text{sunk})) \bullet c_0.c\} \\
= & \{c_0 \mid \neg (\exists o, s, c \mid o.s = s.n \wedge s.c = c.n \wedge c.c = c_0.c \bullet \neg (\exists o' \bullet o.b = o'.b \wedge o'.r = \text{sunk})) \bullet c_0.c\}
\end{aligned}$$

Een veelvoorkomende fout is het ontbreken van het onderscheid tussen de rol die c_0 en c spelen in bovenstaande formuleringen: $c_0.c$ is louter “een willekeurig land” en c is de klasse van s (zodat de eis luidt: $c.c = c_0.c$).

We hebben $c_0.c$ gekozen als bron voor alle landen. Een manier om dit “af te leiden” is de volgende:

$$\begin{aligned}
& \{L : TXT \mid \text{“}L \text{ is een land”} \wedge \dots \bullet L\} \\
= & \quad [\text{nu wordt aangenomen: in } country \text{ attribuut van } Class \text{ staan alle landen}] \\
& \{L : TXT \mid (\exists c_0 \bullet c_0.c = L) \wedge \dots \bullet L\} \\
= & \quad [\text{shunting}]
\end{aligned}$$

$$\begin{aligned}
& \{L : TXT; c_0 \mid c_0.c = L \wedge \dots \bullet L\} \\
= & \quad \text{[voorbereiding voor volgende shunting]} \\
& \{L : TXT; c_0 \mid c_0.c = L \wedge \dots \bullet c_0.c\} \\
= & \quad \text{[shunting]} \\
& \{c_0 \mid (\exists L : TXT \bullet c_0.c = L) \wedge \dots \bullet c_0.c\} \\
= & \{c_0 \mid c_0.c \in TXT \wedge \dots \bullet c_0.c\} \\
= & \quad \text{[aannname: domein van } country \text{ attribuut in } Class \text{ is } TXT\text{]} \\
& \{c_0 \mid \dots \bullet c_0.c\}
\end{aligned}$$

Toelichting bij antwoord 11.

Hier is een alternatieve correcte redenering dat de decompositie niet lossless is:

Voor een schema dat alleen maar FD's als interrelationele constraints heeft, zegt de stelling in Sectie 6.6.1 dat een decompositie van dat schema lossless is als, en slechts als, de gezamenlijke attributen van de componenten een (super)key zijn van (minstens) een der componenten.

De opgave geeft aan dat schema **R** helemaal geen constraints heeft, dus de voorwaarde van de stelling (dat er alleen maar functionle afhankelijkheden zijn) is vervuld. De gezamenlijke attributen van de componenten is: C . Opdat C een (super)key is in minstens een der componenten, moet er gelden: $C \rightarrow AB$ of $C \rightarrow DE$. Geen van deze twee afhankelijkheden is waar, want geen ervan zit in \mathcal{F}^+ (met $\mathcal{F} = \emptyset$). Dus is de decompositie wel lossless.

Een bezwaar tegen dit antwoord is dat de redenering niet “met behulp van r ” is.