

Tentamen Gegevensbanken (211074) — 29 januari 2010

CONTROLEER EERST OF ALLE BLADZIJDEN T/M BLZ. 15 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 kun je zo nodig als **kladpapier** gebruiken en wordt niet bekeken en niet beoordeeld.

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

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

Na afloop moet de *volledige* set opgavenformulieren worden ingeleverd; het kladpapier niet. De tentamenopgaven zijn niet geheim en worden voorzien van modeluitwerkingen op 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
-------------	---	---	---	---	---	---	---	---	---	----	-------

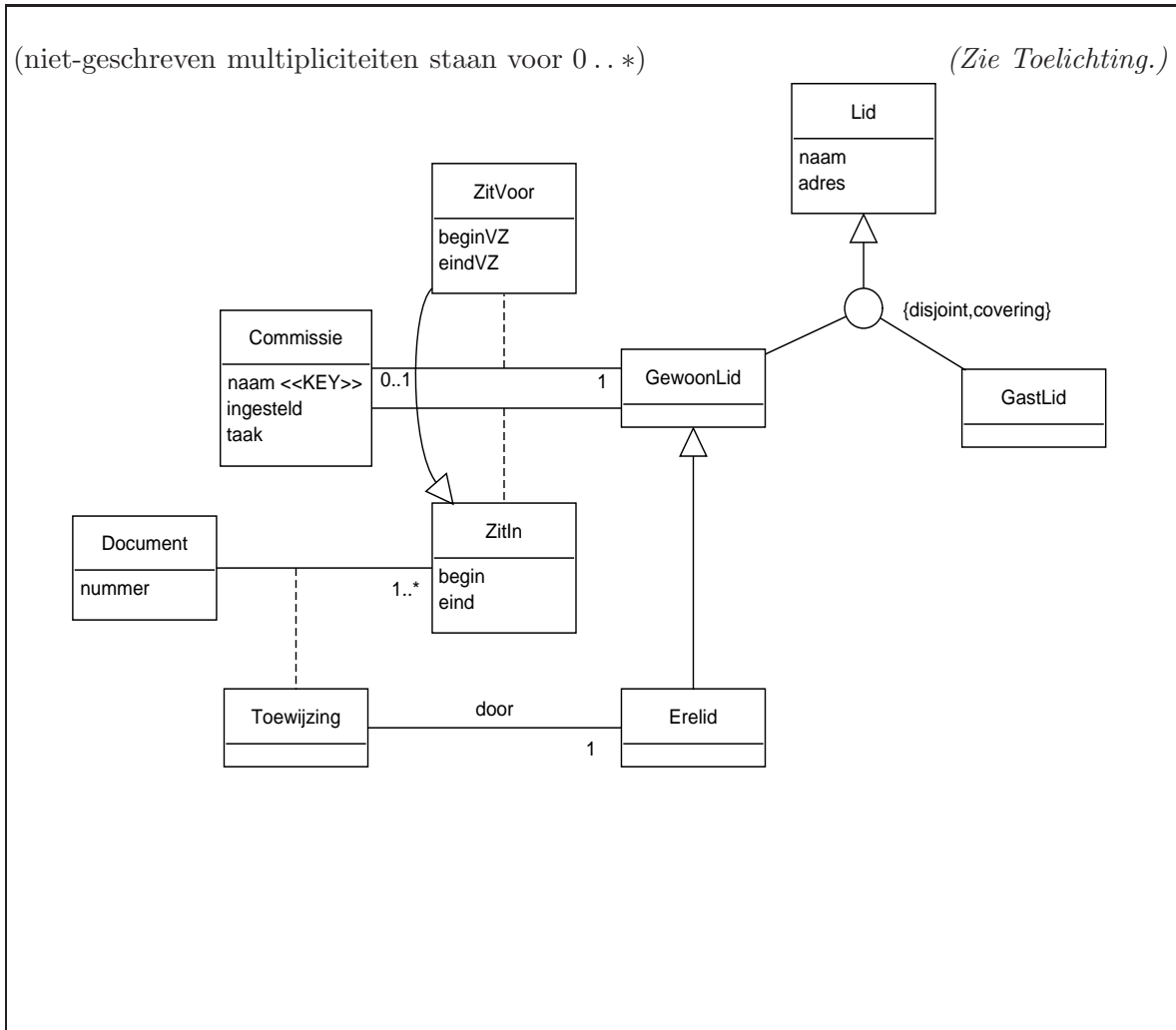
10p. **Opgave 1.** Een gezelligheidsvereniging kent commissies, leden, documenten, etc.



- Iedere commissie heeft een naam (uniek binnen de vereniging) en een datum van instelling en een taak.
- Er zijn drie soorten leden: gewone leden, ereleden en gastleden. Een erelid is óók een gewoon lid, maar een gastlid is géén gewoon lid. De vereniging houdt van ieder lid de naam en het adres bij.
- Een lid (maar gastleden niet) kan zitting hebben in een commissie. Per commissie is er precies één voorzitter; de voorzitter is een gewoon lid dat zitting heeft in de commissie, en deze kan niet tevens voorzitter van een andere commissie zijn.
- Het is op ieder tijdstip voor de vereniging van belang om te weten, van iemand die op dat moment zitting heeft in een commissie, wanneer de huidige zittingstermijn begon en zal ophouden. Ook van een huidig voorzitterschap is het relevant te weten wanneer dat begonnen is en zal ophouden.
- Ieder document heeft een uniek nummer.
- Ieder document is toegewezen aan één of méér van de leden die zitting hebben in een commissie; zo'n toewijzing is door een erelid gedaan.

Geef in het antwoordblok een Entity-Relationship diagram dat de gegevens *voor één tijdstip zo precies mogelijk* modelleert. Gebruik daarbij *zo geschikt mogelijke* constructies.

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



Het zou kunnen zijn (maar het is niet de bedoeling) dat uw diagram *minder* of *meer* eigenschappen modelleert dan de casus aangeeft. Om die fout enigszins te herstellen kunt u die hier noemen (en motiveren):

Nota Bene: de multipliciteit 1 van *ZitVoor* bij *GewoonLid* tesamen met het feit dat *ZitVoor* een subset is van *ZitIn* impliceert dat *ZitIn* de multipliciteit 1..* bij *GewoonLid* heeft! Dat had er voor de duidelijkheid bij vermeld mogen worden.

10p.



Opgave 2. Beschouw de wereld zoals gezien door de ogen van een organisatiebureau. Er zijn evenementen, opdrachtgevers, betalingen, medewerkers, en zo voorts. Deze vormen samen als volgt een relatie \mathcal{R} .

Een tuple $(E, M, T, O, A, B, N, W, F)$ zit op tijdstip t in \mathcal{R} precies wanneer al het volgende geldt op tijdstip t :

1. E is een *Evenement*
2. M is een (interne) *Medewerker* die ingezet wordt bij E
3. T is de *Taak* die M bij E heeft
4. O is de (externe) *Opdrachtgever* van E
5. A is het *Adres* van O
6. B is een van de *Betalingen* voor E
7. N is het *betalingNummer* van B
8. W is een *Willekeurige medewerker* van het organisatiebureau
9. F is de *Functie* van W .

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 bij keuze
$E \rightarrow MT$	O	Bij één evenement kunnen meerdere M of T horen
$MT \rightarrow E$	O	Een medewerker kan met eenzelfde taak bij meerdere E werken
$T \rightarrow EM$	O	Een taak kan bij meerdere E of door meerdere M gedaan worden
$E \rightarrow WF$	O	Bij een evenement zijn er meerdere willekeurige medew's of functies
$WF \rightarrow E$	O	Bij een willekeurige medewerker+functie kunnen er meerdere E zijn
$E \rightarrow BN$	O	Bij een evenement kunnen er meerdere betalingen zijn (zie 5)
$O \rightarrow A$	W	Bij een opdrachtgever hoort hooguit één adres (zie 6)
$ETOABNWF \rightarrow M$	O	Bij een stel $ETOABNWF$ kunnen er meerdere M zijn

Geef een MVD (multivalued dependency) die waar is voor \mathcal{R} en die niet te schrijven is als een functionale afhankelijkheid — *indien er zo'n MVD bestaat*:

Onleesbare tekst wordt fout gerekend.

$EMTOABNWF = EMTOABN \bowtie WF$
$EMTOABNWF = EBN \bowtie E \langle \text{alles behalve } EBN \rangle$
$EMTOABNWF = EOA \bowtie E \langle \text{alles behalve } EOA \rangle$
$EMTOABNWF = EMT \bowtie E \langle \text{alles behalve } EMT \rangle$ (Zie Toelichting.)

10p. **Opgave 3.** Beschrijf wat een phantom is:

Onleesbare tekst wordt fout gerekend.

Als transactie T alle rijen van een tabel leest die aan conditie P voldoen en daarna, maar nog tijdens de executie van T , een andere transactie een rij aan de tabel toevoegt die ook aan P voldoet, dan is die nieuwe rij een <i>phantom</i> voor T .
(Dit kan gebeuren als T alleen op de gelezen rijen een lock zet, en géén schrijfflock op de hele tabel. Bij tweede lezing van die rijen ziet T dan een nieuwe rij verschijnen die hij zelf niet heeft toegevoegd.)

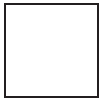
Geef (informeel) twee transacties T_1 en T_2 en een schedule voor gelijktijdige executie van hun zodanig dat T_1 een phantom ervaart:

Onleesbare tekst wordt fout gerekend.

tijdstip	acties van transactie T_1	acties van transactie T_2
1	select * from R where P	
2		insert into R “een rij die aan P voldoet”
3	select * from R where P (levert extra rij op: de <i>phantom</i>)	

(Zie Toelichting.)

10p.



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

$$\bar{R} = ABCDE$$

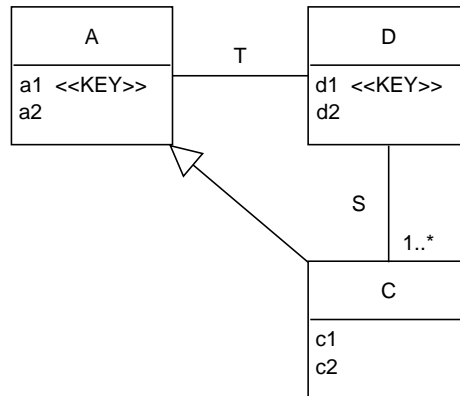
$$\mathcal{F} = \{AB \rightarrow C, \quad C \rightarrow B, \quad C \rightarrow D, \quad BE \rightarrow A\}$$

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

Onleesbare tekst wordt fout gerekend.

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

10p. **Opgave 5.** Beschouw het volgende ERD in de notatie van de UML:



Het ER-diagram kan op verschillende manieren naar een databaseschema vertaald worden dat geschikt is om *precies* de informatie die past in het ER-diagram, op te slaan. U dient een manier te kiezen waarbij: (1) er *één* relatieschema is, genaamd *A*, voor entiteiten *A* en *C* samen, en verder –zoveel als mogelijk na vervulling van voorgaande eis– (2) er *zo weinig mogelijk* relatieschema’s in het databaseschema zijn, maar met de beperkingen dat (3) er geen NULLs nodig zijn vanwege de vertaling, (4) er geen redundantie geïntroduceerd wordt door de vertaling, en (5) alle attributen atomaire waarden hebben.

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

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

(U mag “primary key” afkorten tot “PK”, en zo voorts, en er mogen CHECKs voorkomen.)

Onleesbare tekst wordt fout gerekend.

$A(a_1, a_2, c_1, c_2,$	-- Een rij in <i>A</i> behoort tot <i>C</i> precies wanneer <i>c1</i> niet null is
PK (<i>a1</i>),	
CHECK (“ <i>c1</i> is null \Leftrightarrow <i>c2</i> is null”))	
$D(d_1, d_2,$	
PK (<i>d1</i>),	
CHECK (<i>d1</i> in (select <i>d1</i> from <i>S</i>)))	-- de 1..* eigenschap
$S(d_1, a_1,$	
PK (<i>d1, a1</i>), FK (<i>d1</i>) REF <i>D</i> (<i>d1</i>), FK (<i>a1</i>) REF <i>A</i> (<i>a1</i>),	
CHECK (“de rij <i>a</i> in <i>A</i> met <i>a.a1</i> = <i>a1</i> heeft ook: <i>a.c1</i> is not null”)	
	-- <i>a1</i> identificeert een <i>C</i> -rij!
)	
$T(a_1, d_1,$	
PK (<i>a1, d1</i>), FK (<i>a1</i>) REF <i>A</i> (<i>a1</i>), FK (<i>d1</i>) REF <i>D</i> (<i>d1</i>))	(Zie Toelichting.)

10p. **Opgave 6.** Beschouw het relatieschema $\mathbf{R} = (ABCDEF, \mathcal{F})$, waarbij:

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

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

Onleesbare tekst wordt fout gerekend.

Schema \mathbf{R} heeft precies vier sleutels, namelijk: ADE, BDE, CDE, FDE . Voor iedere FD in \mathcal{F} geldt: die FD is een schending van de BCNF-conditie, omdat die niet-triviaal is en het linkerlid 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} . (Als u het BCNF-algoritme toepast, dan krijgt u na één stap zo'n decompositie.) 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 $A \rightarrow B$. Splits \bar{R} in $\bar{R}_1 = AB$ (alle attributen van de FD) en $\bar{R}_2 = ACDEF$ (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:

$$\mathbf{R}_1 = (AB, \{A \rightarrow B\})$$

$$\mathbf{R}_2 = (ACDEF, \{A \rightarrow C, CD \rightarrow F, F \rightarrow A\})$$

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

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

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

Onleesbare tekst wordt fout gerekend.

Let op: $A \rightarrow C$ volgt uit \mathcal{F} en zal dus (bij een correcte redenering) een FD worden van elke component met attributen $A\dots C\dots$, ook al zit B daar niet bij. Net zo voor $CD \rightarrow AB$ en $F \rightarrow BC$ en "elke vervanging van C door B ".

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-beperving, key constraint).

- We bekijken nu $\mathbf{R}_2 = (ACDEF, \{A \rightarrow C, CD \rightarrow F, F \rightarrow A\})$.

In \mathbf{R}_2 zijn ADE , CDE en DEF de sleutels, dus ieder van \mathcal{F}_2 is een schending van de BCNF-conditie. We kiezen $A \rightarrow C$ ter eliminatie. Dus splitsen we \bar{R}_2 in $\bar{R}_{2a} = AC$ en $\bar{R}_{2b} = A\cancel{C}DEF = ADEF$. 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

$\mathbf{R}_{2a} = (AC, \{A \rightarrow C\})$ en

$\mathbf{R}_{2b} = (ADEF, \{AD \rightarrow F, F \rightarrow A\})$.

De FD $CD \rightarrow F$ van \mathcal{F}_2 zit niet in $(\mathcal{F}_{2a} \cup \mathcal{F}_{2b})^+$ en is dus verloren gegaan. Schema \mathbf{R}_{2b} staat niet in BCNF; schema \mathbf{R}_{2a} wel.

- We bekijken nu $\mathbf{R}_{2b} = (ADEF, \{AD \rightarrow F, F \rightarrow A\})$.

Hierin zijn beide FDs een schending van de BCNF-conditie; bijvoorbeeld, voor $AD \rightarrow F$ is de reden dat die niet-triviaal is en AD geen sleutel is in \mathbf{R}_{2b} (want $AD_{\mathcal{F}_{2b}}^+ = ADF \neq ADEF$). We kiezen (zomaar) $AD \rightarrow F$ ter eliminatie (met de keuze $F \rightarrow A$ ben je sneller klaar...). Dus splitsen we \bar{R}_{2b} in $\bar{R}_{2b1} = ADF$ en $\bar{R}_{2b2} = ADE\cancel{F} = ADE$. Dit levert schema's $\mathbf{R}_{2bi} = (\bar{R}_{2bi}, \mathcal{F}_{2bi})$; hierin is \mathcal{F}_{2bi} (een basis voor) de inperking van \mathcal{F}^+ (of \mathcal{F}_{2b}^+) tot \bar{R}_{2bi} . Dus

$\mathbf{R}_{2b1} = (ADF, \{AD \rightarrow F, F \rightarrow A\})$ en

$\mathbf{R}_{2b2} = (ADE, \{ \})$.

De FDs van \mathbf{R}_{2b} zijn behouden. Schema \mathbf{R}_{2b2} staat in BCNF (het heeft geen niet-triviale FDs). In schema \mathbf{R}_{2b1} is $F \rightarrow A$ een schending van de BCNF-conditie. Decompositie van \mathbf{R}_{2b1} levert: $\mathbf{R}_{2b1x} = (AF, \{F \rightarrow A\})$ en $\mathbf{R}_{2b1y} = (DF, \{ \})$, die beide in BCNF staan en waarbij $AD \rightarrow F$ verloren is gegaan.

- Dus $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b1x}, \mathbf{R}_{2b1y}, \mathbf{R}_{2b2}\}$ is een decompositie van \mathbf{R} waarvan alle componenten in BCNF staan. Er zijn functionele afhankelijkheden verloren gegaan.

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

Zijn alle functionele afhankelijkheden behouden onder bovenstaande decompositie?

Onleesbare tekst wordt fout gerekend.

Nee; $A \rightarrow B$ is verloren gegaan (in de 1ste stap) en daarna ook nog, bijvoorbeeld, $AD \rightarrow F$.

10p. **Opgave 7.** Een bibliotheek wil een database opzetten volgens de volgende richtlijnen:



- Er is een tabel *Boek* met attributen *isbn*, *titel*, *auteur*. Attribuut *isbn* is de primary key van *Boek*.
- Er is een tabel *Exemplaar* met attributen *isbn*, *volgnummer*, *gewicht*, *kast*. De primary key van *Exemplaar* is het paar (*isbn*, *volgnummer*). Een exemplaar *E* hoort bij het boek dat hetzelfde *isbn* heeft als *E*.

Bij iedere wijziging van de database moet het volgende gelden:

- Een exemplaar *E* kan niet aan de database toegevoegd worden wanneer er geen boek in de database bestaat met hetzelfde *isbn* als *E*.
- Wanneer van een boek in de database de *isbn*-waarde wijzigt, moet dat bij alle exemplaren die bij dat boek horen, ook eensgelijks gebeuren (zodat die exemplaren bij hetzelfde boek blijven horen).
- Een verwijdering van een boek moet tot gevolg hebben dat alle exemplaren die bij dat boek horen, ook verwijderd worden.

Geef de SQL create statements voor *Boek* en *Exemplaar* zodanig dat bovenstaande eisen vervuld zijn. U mag géén triggers gebruiken, en u hoeft de domeinen van attribuutwaarden (zoals CHAR(20) en INTEGER) niet te vermelden.

Ter herinnering, foreign key constraints zien er als volgt uit:

foreign key (*x*, *x'*, ...) references *T*(*y*, *y'*, ...) on *event action* on *event' action'* ...

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

Onleesbare tekst wordt fout gerekend.

create table <i>Boek</i> (<hr/> <i>isbn</i> , <i>titel</i> , <i>auteur</i> , primary key (<i>isbn</i>)) <hr/>
create table <i>Exemplaar</i> (<hr/> <i>isbn</i> , <i>volgnummer</i> , <i>gewicht</i> , <i>kast</i> , <hr/> primary key (<i>isbn</i> , <i>volgnummer</i>), <hr/> foreign key (<i>isbn</i>) references <i>Boek</i> (<i>isbn</i>) <hr/> on delete cascade on update cascade) <hr/>
NB. In <i>Exemplaar</i> is <i>isbn</i> not null (nodig vor de eerste eis) want het is een onderdeel van de primary key.

De eisen voor databasewijzigingen kunnen ook met triggers vervuld worden in plaats van constraints in de create table statements. Geef de SQL create statements voor *Boek* en *Exemplaar* en voor een geschikte trigger zodanig dat *de laatste van* bovenstaande eisen vervuld wordt door de trigger.

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

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

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

Onleesbare tekst wordt fout gerekend.

```
create table Boek -- als voorheen
```

```
create table Exemplaar -- als voorheen maar zonder "on delete cascade"
```

```
create trigger OnDeleteCascade
```

```
  after delete on Boek
```

```
  referencing old as O
```

```
  for each row
```

```
  delete from Exemplaar E where E.isbn = O.isbn
```

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 8.** Beschouw de volgende zoekopdracht:



Geef iedere zeeslag waarbij er een schip uit NL betrokken is.

(Een schip komt uit land NL als voor de klasse van dat schip geldt: $country = NL$.) Geef voor deze vraag een afleiding in Verzamelingsnotatie in kleine stappen naar een vorm die dicht aansluit bij SQL-*zonder*-subqueries. Het begin is al gegeven. Kort tabel- en attribuutnamen af tot hun eerste letter.

Onleesbare tekst wordt fout gerekend.

“(geef) iedere zeeslag waarbij er een schip uit NL betrokken is”
= $\{b : B \mid \text{“er is een schip uit NL betrokken bij } b\text{”} \bullet b.n\}$
= $\{b : B \mid (\exists s : S \mid \text{“}s \text{ is uit NL”} \bullet \text{“}s \text{ betrokken bij } b\text{”}) \bullet b.n\}$
= $\{b : B \mid (\exists s : S \mid (\exists c : C \mid c.n = s.c \bullet c.c = NL) \bullet (\exists o : O \mid s.n = o.s \bullet o.b = b.n)) \bullet b.n\}$
= $\{b : B \mid (\exists s : S; c : C \mid c.n = s.c \wedge c.c = NL \bullet (\exists o : O \mid s.n = o.s \bullet o.b = b.n)) \bullet b.n\}$
= $\{b : B \mid (\exists s : S; c : C; o : O \mid c.n = s.c \wedge c.c = NL \bullet s.n = o.s \wedge o.b = b.n) \bullet b.n\}$
= $\{b : B; s : S; c : C; o : O \mid c.n = s.c \wedge c.c = NL \wedge s.n = o.s \wedge o.b = b.n \bullet b.n\}$
= [ter voorbereiding van de volgende stappen: vervang $b.n$ door $o.b$]
$\{b : B; s : S; c : C; o : O \mid c.n = s.c \wedge c.c = NL \wedge s.n = o.s \wedge o.b = b.n \bullet o.b\}$
= [shunting: “ $b : B$ naar binnen brengen”]
$\{s : S; c : C; o : O \mid (\exists b : B \bullet o.b = b.n) \wedge c.n = s.c \wedge c.c = NL \wedge s.n = o.s \bullet o.b\}$
= [$O(b)$ is FK naar $B(n)$, dus voor elke o geldt $(\exists b : B \mid o.b = b.n)$]
$\{s : S; c : C; o : O \mid c.n = s.c \wedge c.c = NL \wedge s.n = o.s \bullet o.b\}$
Desgewenst had $b : B$ afgekort mogen worden tot b . Net zo voor s, c, o . (Zie Toelichting.)

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 $o.b$ from $S s, C c, O o$ where $c.n = s.c$ and $c.c = NL$ and $s.n = o.s$

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.

$$\pi_b (\sigma_{cc=NL} (S'' \bowtie C'' \bowtie O''))$$

Hierbij zijn met renaming nieuwe relaties gedefinieerd:

$O'' = O[s, b, or]$ en $S'' = S[s, c, sl]$ en $C'' = C[c, cc, \dots]$.

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 ieder land dat in minstens 10 zeeslagen betrokken was, de datums van de eerste en de laatste zeeslag waarin dat land betrokken was.

Hierbij staat “land x is betrokken in zeeslag y ” voor “er is een schip, van een klasse van land x , dat betrokken is in zeeslag y ”.

Veronderstel dat de waarden van het *date*-attribuut van zeeslagen met *min* en *max* geaggregeerd kunnen worden (tot een “eerste” en “laatste” datum).

Onleesbare tekst wordt fout gerekend.

```
select c.country, min (b.date), max (b.date)
```

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

```
where c.n = s.c and s.n = o.s and o.b = b.n
```

```
group by c.country
```

```
having count (distinct b.n) ≥ 10
```


Toelichtingen

Toelichting bij antwoord 1.

Merk op dat relatie *ZitVoor* óók een begin- en einddatum heeft; die verschillen van de begin- en einddatum van de relatie *ZitIn*.

Bij een relatie *Toewijzing* tussen de drie entiteiten *Document*, *ZitIn* én *Erelid*, kunnen de multipliciteiten niet meer correct in het ERD weergegeven worden (maar zouden die apart als een notitie erbij gegeven moeten worden).

Toelichting bij antwoord 2.

De FD $B \rightarrow N$ is gelijkwaardig met de MVD $\bar{R} = BN \bowtie B$ (alles van \bar{R} behalve BN).
Net zo voor $M \rightarrow T$ en $O \rightarrow A$ en $W \rightarrow F$.

Toelichting bij antwoord 3.

Individuele SQL statements worden *nooit* tegelijkertijd uitgevoerd. (Gelijktijdige uitvoering van transacties betekent dat hun SQL statements in willekeurige volgorde worden uitgevoerd, maar dus niet tegelijkertijd.) Het is dus fout om in de schedule twee SQL statements op eenzelfde tijdstip te laten uitvoeren.

Toelichting bij antwoord 5.

Een redundant attribuut *isC* in *A* wordt goedgekeurd indien het volgende afleidbaar is:

$$isC \Leftrightarrow c1 \neq NULL \wedge c2 \neq NULL$$

Het is fout om in *T* te zetten: “FK(*a1*, *d1*) REF (*A*(*a1*), *D*(*d1*))”, want (*a1*, *d1*) zijn niet samen één externe sleutel, maar *a1* en *d1* zijn beide afzonderlijk een externe sleutel. (Net zo in *S*.)

De *c1*, *c2* attributen zijn geen sleutel; dus een verwijzing naar een van deze kan geen foreign key zijn.

Toelichting bij antwoord 6.

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

$$“ABCDEF = AB \bowtie AC \bowtie AF \bowtie DF \bowtie ADE \text{ geldt in } \mathbf{R}.”$$

Toelichting bij antwoord 8.

We hebben *B* weggewerkt dankzij het feit dat *O*(*b*) een foreign key is naar *B*(*n*). Er geldt ook: *O*(*s*) is een foreign key is naar *S*(*n*). Desondanks kan *S* niet weggewerkt worden, omdat *s.c* niet in attributen van *o* uitgedrukt kan worden.

Toelichting bij antwoord 10.

Wanneer we aan de vraag toevoegen: “laat de zeeslagen waarin geen schepen betrokken zijn weg”, dan is het volgende ook een correcte SQL formulering:

```
select o.b from O o, S s where o.s = s.n group by o.b having count (distinct s.c) = 1
```

Een alternatieve formulering in verzamelingsnotatie is de volgende:

$\{b : B \mid (\forall s, s' : S \mid \text{“}s, s' \text{ betrokken in } b\text{”} \bullet s, s' \text{ hebben eenzelfde klasse}) \bullet b.n\}$.

De gelijkwaardigheid van deze formulering met de gegeven formulering is voor ervaren logici niet moeilijk te bewijzen (onder de aanname dat *Ship* niet leeg is).