

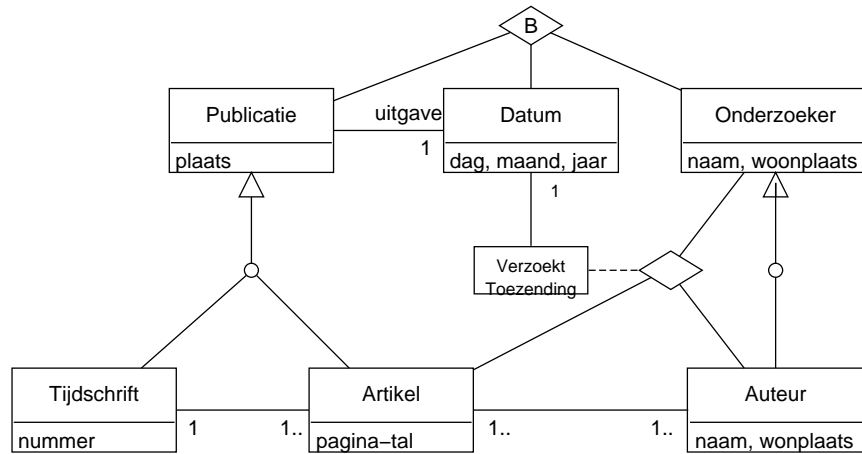
10p.

Opgave 2. Een uitgever van wetenschappelijke publicaties wil een en ander automatiseren. Daarvoor is het nodig om de relevante aspecten van de werkelijkheid in kaart te brengen. Hier is een beschrijving van een vereenvoudiging van die werkelijkheid:

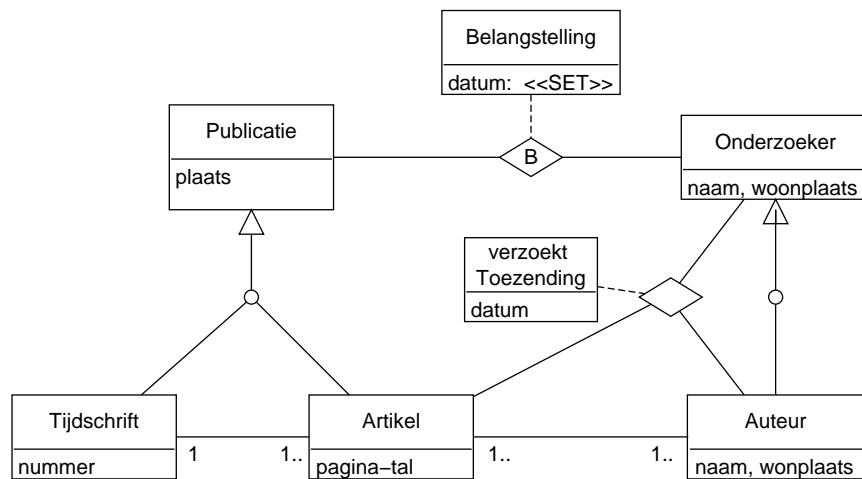
- De uitgever onderscheidt de publicaties in twee soorten: artikel en tijdschrift. Het verschil is dat het begrip auteur niet van toepassing is op tijdschrift maar wel op artikel. Bovendien is elk tijdschrift opgebouwd uit één of meer artikelen (en misschien nog wel andere delen, zoals vaste rubrieken), en in het wereldbeeld van de uitgever verschijnt ieder artikel in precies één tijdschrift.
- Naast ‘auteur’ kent de uitgever ook nog het begrip ‘onderzoeker’. Niet alle onderzoekers publiceren, maar gemakshalve beschouwt de uitgever iedere auteur als een onderzoeker. Iedere auteur is bij tenminste één artikel betrokken, en bij ieder artikel zijn één of meer personen als auteur betrokken.
- De uitgever realiseert zich bovendien dat het nogal eens voorkomt dat een onderzoeker een auteur a verzoekt om toezending van een artikel (waarvan a auteur is). De datum waarop dit gebeurt, is relevant voor de uitgever. In het wereldbeeld van de uitgever komt het niet voor dat een onderzoeker op verschillende datums toezending verzoekt van eenzelfde artikel.
- Voor promotionele activiteiten houdt de uitgever bij welke onderzoeker belangstelling getoond heeft op welke datum voor welke publicatie. (Een onderzoeker kan op verschillende datums belangstelling tonen voor eenzelfde publicatie.)
- Van een publicatie is de datum en plaats van verschijnen bekend, van een artikel het aantal pagina’s, van een tijdschrift het nummer, van een onderzoeker en auteur de naam en woonplaats. Een datum bestaat uit dag, maand en jaar.

Geef in het antwoordblok een Entity-Relationship diagram dat bovenstaande situatie *zo precies mogelijk* modelleert en gebruik daarbij *zo geschikt mogelijke* constructies. Zowel de ERD-notatie uit het boek als ook de UML notatie (class diagram) is toegestaan, maar een mengeling van beide niet.

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



of



(Zie Toelichting.)

Het zou kunnen zijn (maar het is niet de bedoeling) dat uw diagram ten onrechte of met opzet *méer* eigenschappen modelleert dan de casus aangeeft. Om zo'n "fout" enigszins te herstellen is er deze vraag: Noem en motiveer hoogstens twee eigenschappen die *wel* in het diagram staan maar *niet* uit de letterlijke tekst van de casus volgen (indien zulke eigenschappen bestaan):

Onleesbare tekst wordt fout gerekend.

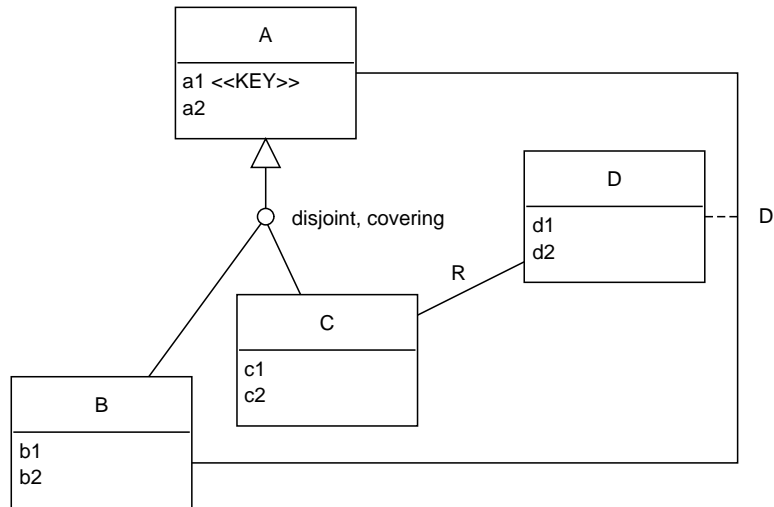
Geen

Het zou kunnen zijn (maar het is niet de bedoeling) dat uw diagram onbedoeld of met opzet *minder* eigenschappen modelleert dan de casus aangeeft. Om dat gebrek enigszins te herstellen is er deze vraag: Geef, in precieze bewoordingen of formules, hoogstens twee eigenschappen die *wel* uit de casustekst volgen maar *niet* in uw ER-diagram opgenomen zijn (indien zulke eigenschappen bestaan):

Onleesbare tekst wordt fout gerekend.

Wanneer $(p, o, a) \in \textit{Toezending}$, dan $(p, a) \in \textit{Auteur}$; in woorden: het artikel p waarvoor o toezending verzoekt aan a , is geschreven door a .
Niet expliciet gegeven in de casus maar wel realistisch: wanneer $(p, d, o) \in B$ en $p.witgave = d'$, dan $d' \leq d$; in woorden, een onderzoeker vertoont belangstelling voor een publicatie slechts nadat die publicatie verschenen is.
Niet expliciet gegeven in de casus maar wel realistisch: wanneer $(p, o, a) \in \textit{Toezending}$, dan $(p, o) \notin \textit{Auteur}$; in woorden, een onderzoeker die om toezending verzoekt van een artikel is zelf niet een auteur van dat artikel.

10p. **Opgave 3.** Beschouw het volgende ERD in de notatie van de UML:



(niet-geschreven multipliciteiten staan voor 0..*)

Het ER-diagram kan op verschillende manieren vertaald worden naar een databaseschema dat geschikt is om informatie die past in het ER-diagram, op te slaan. U dient een manier te kiezen waarbij er zo weinig mogelijk relatieschema's in het databaseschema zijn, maar met de beperkingen dat er geen NULLs nodig zijn vanwege de vertaling, er geen redundantie geïntroduceerd wordt door de vertaling, en alle attributen atomaire waarden hebben.

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

$X(x_1, \dots, \text{primary key } (x_i, x_j \dots), \text{foreign key } (x_m, x_n \dots) \text{ references } (Y(y), Z(z) \dots), \dots);$

Onleesbare tekst wordt fout gerekend.

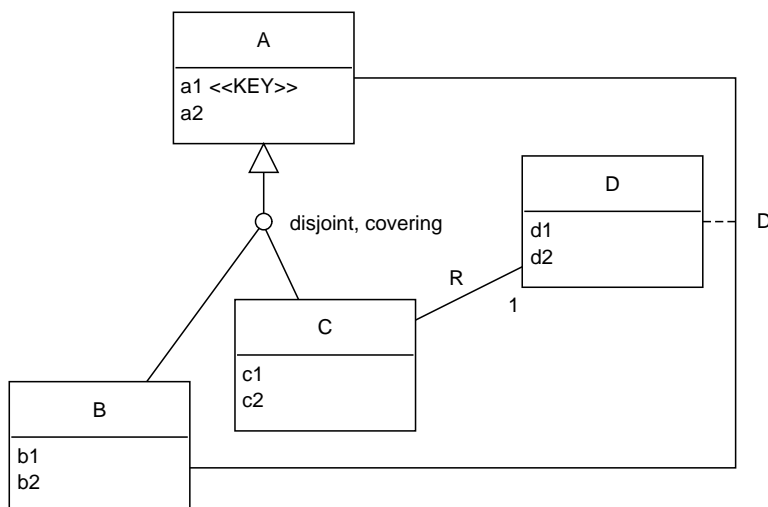
$A(a1, a2, \text{primary key } (a1));$
$B(a1, b1, b2, \text{primary key } (a1),$
foreign key (a1) references (A(a1)));
$C(a1, c1, c2, \text{primary key } (a1),$
foreign key (a1) references (A(a1)));
$D(Ba1, Aa1, d1, d2, \text{primary key } (Ba1, Aa1),$
foreign key (Ba1) references (B(a1)),
foreign key (Aa1) references (A(a1)));
$R(Ca1, Ba1, Aa1, \text{primary key } (Ca1, Ba1, Aa1),$
foreign key (Ca1) references (C(a1)),
foreign key (Ba1, Aa1) references (D(Ba1), D(Aa1)));
<i>(Zie Toelichting.)</i>

Zijn er eigenschappen die wel in het ERD staan maar niet in uw databaseschema, of omgekeerd? Zo ja, geef die eigenschappen met een maximum van drie.

Onleesbare tekst wordt fout gerekend.

Ja, de disjoint en covering eigenschap.

Het ERD wordt nu gewijzigd doordat bij de R -associatie aan de D -kant multipliciteit 1 komt te staan:



(niet-geschreven multipliciteiten staan voor 0..*)

Geef nu weer het gevraagde databaseschema of beschrijf de benodigde wijziging:

Onleesbare tekst wordt fout gerekend.

Een correct databaseschema wordt verkregen door alleen $Ca1$ tot primary key van R te nemen; met andere woorden, vervang R door deze R' :

$R'(Ca1, Ba1, Aa1, \text{primary key } (Ca1),$

foreign key $(Ca1)$ references $(C(a1)),$

foreign key $(Ba1, Aa1)$ references $(D(Ba1), D(Aa1))$);

Een correct schema met minder tabellen (zoals gevraagd wordt) verkrijgt je nu als volgt.

Verwijder tabel R' en voeg toe aan tabel C al de attributen van R' behalve de key:

$C(a1, c1, c2, Ba1, Aa1, \text{primary key } (a1),$

foreign key $(a1)$ references $(A(a1)),$

foreign key $(Ba1, Aa1)$ references $(D(Ba1), D(Aa1))$);

10p.

Opgave 4. Stichting *IAPC* is een ICT winkel en adviescentrum “voor studenten door studenten”. Je kunt bij *IAPC* terecht voor producten uiteenlopend van kabels tot complete PC’s en voor al je technische vragen. (Vanwege een tekort aan bestuursleden is de winkel (anno 2007) voorlopig slechts beperkt open.) Bij de stichting *IAPC* spelen verscheidene entiteiten een rol, zo als: order, cliënt, vrijwilliger, bestuurslid, product, leverancier, kontaktpersoon. We definiëren een relatie \mathcal{R} die die entiteiten aan elkaar relateert, op de volgende manier.

Een tuple (O, C, V, B, P, L, K) zit in relatie \mathcal{R} precies wanneer het volgende geldt:

1. O is een *Order*.
2. C is de *Cliënt* die order O heeft uitstaan.
3. V is de *Vrijwilliger* die bij order O betrokken is.
4. B is een *Bestuurslid* van *IAPC*.
5. P is een *Product* dat besteld wordt in order O .
6. L is de *Leverancier* van product P .
7. K is de *Kontaktpersoon* uit het bestuur voor vrijwilliger V .

Neem het volgende aan:

- a. Relatie \mathcal{R} beschrijft de situatie van één tijdstip (namelijk: 1 februari 2007).
- b. Een kontaktpersoon is een bestuurslid.
- c. In een order kunnen meerdere producten besteld worden.
- d. Ieder product heeft precies één leverancier.
- e. Er zijn minstens twee bestuursleden.

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. (De motivatie telt mee in de beoordeling.)

Onleesbare tekst wordt fout gerekend.

FD	W/O	Motivatie voor uw keuze
$C \rightarrow O$	O	Een cliënt kan meerdere orders hebben uit staan.
$V \rightarrow O$	O	Een vrijwilliger kan bij meerdere orders betrokken zijn.
$V \rightarrow B$	O	Ook al is V gegeven, er zijn minstens 2 bestuursleden (e).
$V, K \rightarrow B$	O	idem (e).
$V, O, C, P, L, K \rightarrow B$	O	idem (e).
$O \rightarrow P$	O	Eigenschap (c).
$O \rightarrow P, V$	O	Eigenschap (c). (Er geldt wel $O \rightarrow V$ volgens 3.)

Geef voor ieder van de volgende MultiValued Dependencies aan of die *Waar* of *Onwaar* is in relatie \mathcal{R} :

Onleesbare tekst wordt fout gerekend.

MVD	W/O	
$OCVBPLK = O \bowtie OCVBPLK$	W	doorsnede O is key van linkercomponent.
$OCVBPLK = OCVBPLK \bowtie K$	W	doorsnede K is key van rechtercomponent.
$OCVBPLK = OCVPLK \bowtie B$	W	volgens de definitie van \mathcal{R} neemt B per $OCVPLK$ alle waarden aan!

10p. **Opgave 5.** 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.

	$X \rightarrow Y$
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	$\underline{BE} \rightarrow ACD$
14	$CD \rightarrow B$
15	$\underline{CE} \rightarrow ABD$
16	$DE \rightarrow$
17	$ABC \rightarrow D$
18	$ABD \rightarrow C$
19	$\underline{ABE} \rightarrow CD$
20	$ACD \rightarrow B$
21	$\underline{ACE} \rightarrow BD$
22	$ADE \rightarrow$

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

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

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

We passen het BCNF-algoritme toe.

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

Van de gegeven \mathcal{F} zijn alle leden een schending van de BCNF-eis voor \mathbf{R} , want die FDs zijn niet-triviaal en hun linkerleden bevatten geen sleutel van \mathbf{R} — de sleutels van \mathbf{R} zijn AD en BD en CD . We kiezen (zomaar) de eerste, $A \rightarrow B$, ter eliminatie. Dus splitsen we $\bar{R} = ABCD$ in $\bar{R}_1 = AB$ en $\bar{R}_2 = A\cancel{B}CD = ACD$.

Dit levert schema's $\mathbf{R}_i = (\bar{R}_i, \mathcal{F}_i)$, waarbij \mathcal{F}_i (een basis voor) de inperking is van \mathcal{F}^+ tot \bar{R}_i . (Let op: vergeet de closure \dots^+ bij \mathcal{F} niet!)

Dus, **LET OP**, $\mathbf{R}_1 = (AB, \{A \rightarrow B, B \rightarrow A\})$ en $\mathbf{R}_2 = (ACD, \{A \rightarrow C, C \rightarrow A\})$. De $B \rightarrow A$ en $A \rightarrow C$ zitten weliswaar niet in \mathcal{F} , maar toch in \mathcal{F}_2 omdat ze in \mathcal{F}^+ zitten en dus ook, per definitie, in \mathcal{F}_2 . (Merk ook op dat de afhankelijkheid $B \rightarrow C$ NIET verloren is gegaan, want die volgt nog uit de aanwezige $B \rightarrow A$ en $A \rightarrow C$!)

- We bekijken nu $\mathbf{R}_1 = (AB, \{A \rightarrow B, B \rightarrow A\})$.

Omdat $A \rightarrow B$ niet van de vorm $\emptyset \rightarrow Y$ is, en \mathbf{R}_1 slechts twee attributen heeft, staat \mathbf{R}_1 in BCNF.

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

Nu zijn $A \rightarrow C$ en $C \rightarrow A$ een schending van de BCNF-eis voor \mathbf{R}_2 , want die FD is niet-triviaal en de linkerleden bevatten geen sleutel van \mathbf{R}_2 (de sleutels in \mathbf{R}_2 zijn AD en CD). We kiezen $A \rightarrow C$ ter eliminatie. Dus splitsen we $\bar{R}_2 = ACD$ in $\bar{R}_{2a} = AC$ en $\bar{R}_{2b} = A\cancel{C}D = AD$.

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} . (Let op: vergeet de closure \dots^+ bij \mathcal{F} niet!)

- We bekijken nu $\mathbf{R}_{2a} = (AC, \{A \rightarrow C, C \rightarrow A\})$ en $\mathbf{R}_{2b} = (AD, \{\})$. Omdat in \mathbf{R}_{2a} zowel A als ook C een sleutel is, zijn $A \rightarrow C$ en $C \rightarrow A$ geen schending van de BCNF-conditie voor \mathbf{R}_{2a} , en is \mathbf{R}_{2a} in BCNF.

Omdat er in \mathbf{R}_{2b} alleen maar triviale functionele afhankelijkheden zijn, is \mathbf{R}_{2b} in BCNF.

- Dus $\{\mathbf{R}_{2a}, \mathbf{R}_{2b}\}$ is een decompositie van \mathbf{R}_2 , en $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b}\}$ van \mathbf{R} .

- De functionele afhankelijkheden zijn wel behouden; met name geldt dit voor $B \rightarrow C$, die nog steeds afleidbaar is.

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

$$"ABCD = AB \bowtie AC \bowtie AD \text{ geldt in } \mathbf{R}."$$

Wanneer je met een andere schending begint kun je mogelijk een andere decompositie bereiken.

- NB. In de volgende lossless decompositie staan ook alle componenten in BCNF. Deze decompositie kan verkregen worden door met de schending $A \rightarrow BC$ te starten (die staat niet in de basis \mathcal{F} maar is wel lid van de closure \mathcal{F}^+):

$$"ABCD = ABC \bowtie AD \text{ geldt in } \mathbf{R}."$$

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 *class* zijn: het land (*country*), het aantal kanonnen (*guns*), de diameter in centimeters van de kanonsloop (*bore*), en de waterverplaatsing (*displacement*, gemeten in tonnen). Alle schepen van een klasse komen uit één land: het land dat in de klasse genoemd staat. Van een *ship* 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).

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

```
C (n, t, c, g, b, d)
S (n, c, l)
B (n, d)
O (s, b, r)
```

10p. **Opgave 7.** Beschouw de volgende zoekopdracht:



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

(Een schip komt uit land NL als de klasse van dat schip uit land NL komt: $country = NL$.) Geef voor deze vraag een afleiding in Verzamelingsnotatie 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 $o.b$]
$\{b : B; s : S; c : C; o : O \mid c.n = s.c \wedge c.c = NL \wedge s.n = o.s \wedge o.b = b.n \bullet o.b\}$
= --[“ $b : B$ naar binnen”]
$\{s : S; c : C; o : O \mid c.n = s.c \wedge c.c = NL \wedge s.n = o.s \wedge (\exists b : B \bullet o.b = b.n) \bullet o.b\}$
= --[$O.b$ is FK naar $B.n$, dus voor elke o geldt $\exists b : B \mid o.b = b.n$]
$\{s : S; c : C; o : O \mid c.n = s.c \wedge c.c = NL \wedge s.n = o.s \bullet o.b\}$

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:

Onleesbare tekst wordt fout gerekend.

$\pi_{ob} (\sigma_{cn=sc \wedge cc=NL \wedge sn=os} (S' \times C' \times O'))$	=	$\pi_b (\sigma_{cc=NL} (S'' \bowtie C'' \bowtie O''))$
Hierbij zijn met renaming nieuwe relaties gedefinieerd:		
$O' = O[os, ob, or]$ en $S' = S[sn, sc, sl]$ en $C' = C[cn, cc, \dots]$ en		
$O'' = O[s, b, or]$ 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> , b, r)

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



Geef voor ieder land dat tenminste twee schepen heeft (niet noodzakelijk van eenzelfde klasse) de vroegste en laatste datum waarop er een schip van dat land te water is gelaten.

(Het land van een schip is het land van de klasse van dat schip.) Veronderstel dat *datums* vergeleken kunnen worden met $<$ ('kleiner dan'), etc, en dus ook *min* en *max* daarop van toepassing zijn.

Onleesbare tekst wordt fout gerekend.

select <i>c.country</i> , <i>min(s.launched)</i> , <i>max(s.launched)</i>
from <i>Ship s</i> , <i>Class c</i>
where <i>s.class</i> = <i>c.name</i>
group by <i>c.country</i>
having <i>count(*)</i> > 1

10p. **Opgave 10.** Gegeven zijn twee transacties waarvan de executievolgorde van de individuele operaties er schematisch als volgt uit zien:

T_1 : $read_1(x)$; $read_1(y)$; $write_1(x)$; $write_1(y)$.

T_2 : $read_2(x)$; $write_2(x)$.

Hierbij staan x en y voor verschillende objecten uit de database. De *read*-operaties lezen de waarde van het object, en de *write*-operaties schrijven een nieuwe waarde in het object.

Geef een niet-serialiseerbare executievolgorde aan voor de operaties van deze transacties:

Onleesbare tekst wordt fout gerekend.

Een *interleaving* van T_1 en T_2 die er zó uit ziet:

... $read_2(x)$... $write_1(x)$... $write_2(x)$... of

... $read_1(x)$... $write_2(x)$... $write_1(x)$...

Geef een executievolgorde voor de operaties van deze transacties aan zó dat T_2 een *dirty read* doet:

Onleesbare tekst wordt fout gerekend.

$read_1(x)$; $read_1(y)$; $write_1(x)$; $read_2(x)$; ... abort/rollback van T_1 ...

Wanneer 'abort/rollback' wordt vervangen door 'commit', dan is er in feite nog steeds sprake van een *dirty read*, hoewel die in dat geval niet erg is omdat de executievolgorde dan serialiseerbaar is.

We willen de gelijktijdige executie van transacties T_1 en T_2 zo min mogelijk beperken, maar wel zó dat T_1 geen *dirty reads* zal doen. Welke transactie(s) moeten we dan op welk isolatieniveau zetten? (Herinner je dat de isolatieniveaus, in opklimmende sterkte van beperkingen, de volgende zijn: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE.)

Onleesbare tekst wordt fout gerekend.

Zet het isolatieniveau van T_1 op READ COMMITTED of sterker.

10p. **Opgave 11.** 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.

View, en het nut van views:

Onleesbare tekst wordt fout gerekend.

Een *view* is (het resultaat van) een select-query, en kan gebruikt worden in select-queries waar een tabel verwacht wordt. Views zijn nuttig als hulpmiddel bij de constructie van queries (modulariteit), als hulpmiddel om de database te presenteren in een vorm die begrijpelijk is voor een gebruiker die het databaseschema niet kent (abstractie), en als hulpmiddel bij het uitdelen van toegangsrechten op delen van de database (beveiliging).

Toelichtingen

Toelichting bij antwoord 2.

Bij de 3-plaatsige relatie *Toezending* is *Datum* uniek bepaald door het drietal; bij de 3-plaatsige relatie *Belangstelling* is *Datum* een van de drie participanten.

Toelichting bij antwoord 3.

NB. Tabel *A* weglaten en zijn attributen opnemen in tabel *B* en *C* gaat fout, want *A* is nodig in de relatie *D*.

Toelichting bij antwoord 9.

Uiteraard kan de verzameling direct voor de regel (*) ook gemakkelijk in SQL omgezet worden (maar die is dan nodeloos ingewikkeld).