

Tentamen Gegevensbanken (211074) — 30 januari 2009

CONTROLEER EERST OF ALLE BLADZIJDEN T/M BLZ. 14 AANWEZIG ZIJN!

Vul het tentamenbriefje volledig in, zódanig dat BEIDE DOORSLAGEN goed leesbaar zijn.

NAAM, VOORLETTERS: _____

STUDENTNUMMER: _____

OPLEIDING: _____

De uitwerkingen moeten op deze opgavenformulieren worden genoteerd in de daarvoor bestemde vakken. Alle overige ruimte 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 Teletop 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	9bonus	10	11
-------------	---	---	---	---	---	---	---	---	---	--------	----	----

10p.

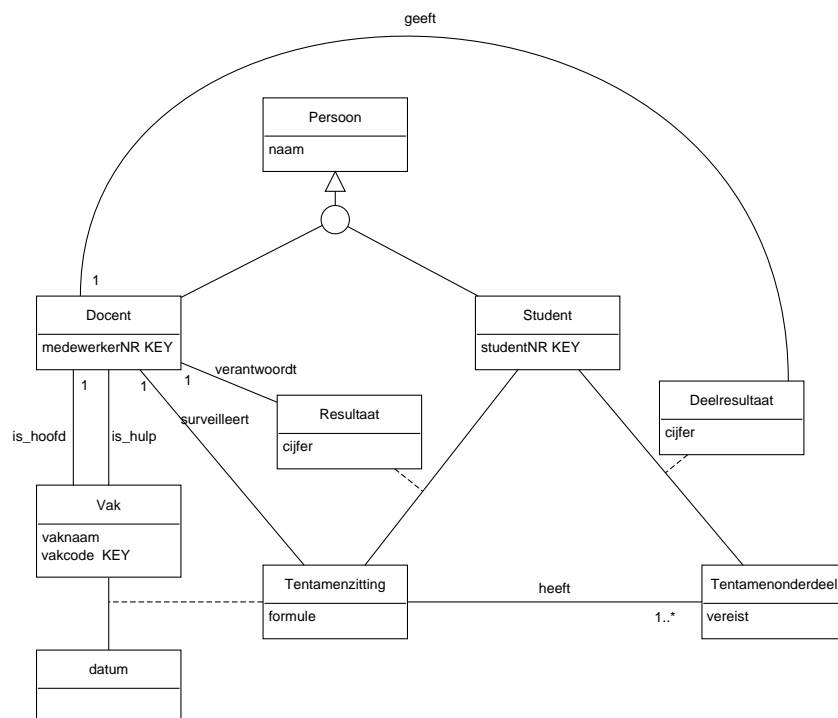
Opgave 1. Voor vakken met grote aantallen studenten hebben docenten behoefte aan computer-ondersteuning bij de cijferadministratie. Alvorens een *Cijfer Administratie Systeem Twente* (CAST) te bouwen, worden eerst de relevante entiteiten en relaties in kaart gebracht. Hier is een beschrijving in woorden van een enigszins vereenvoudigde werkelijkheid:

- Er zijn docenten en studenten; beiden hebben een nummer (respectievelijk medewerkernummer en studentnummer) en naam, enzovoort.
- Er zijn vakken; zij hebben een naam en worden uniek geïdentificeerd door een vakcode.
- Bij elk vak is er precies één hoofddocent aangewezen en nul of meer hulpdocenten.
- Een “tentamenzitting” is een combinatie van vak en datum; per tentamenzitting fungeert een docent als surveillant en is er een formule bekend. De formule bepaalt uit de cijfers voor een stel onderdelen een totaalcijfer.
- Een tentamenzitting bestaat uit één of meer tentamenonderdelen (zoals: opgave, practicum, huiswerkopgave). Van ieder onderdeel is bekend of het vereist is of optioneel; de formule van de tentamenzitting zal hiervan gebruik maken.
- Per combinatie van student en tentamenonderdeel geeft een docent een cijfer, ook wel genoemd: deelcijfer.
- Per combinatie van student en tentamenzitting wordt een eindcijfer bepaald, volgens de formule van de tentamenzitting, en precies één docent is aanspreekbaar op dit eindcijfer (die docent kan dit eindcijfer toelichten of eventueel wijzigen).

(De vraag staat op de volgende pagina! Lees die eerst!)

Onleesbare tekst wordt fout gerekend.

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



(Zie Toelichting.)

Geef in het antwoordblok (vorige pagina) een Entity-Relationship diagram dat de gegevens voor één tijdstip *zo precies mogelijk* modelleert en gebruik daarbij *zo geschikt mogelijke* constructies. 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 onbedoeld of met opzet *minder* eigenschappen modelleert dan de casus aangeeft. Om dat gebrek enigszins te herstellen is er deze vraag: Geef de eigenschappen (zoveel mogelijk maar hógstens twee) die *wel* uit de casustekst volgen maar *niet* in uw ER-diagram opgenomen zijn.

Onleesbare tekst wordt fout gerekend.

Het eindcijfer wordt volgens de formule bepaald uit de deelcijfers; dit is een relatie die uitdrukbaar is in de anderen.

De formule maakt gebruik van het attribuut “vereist/optioneel”.

De docent die aanspreekbaar is op een eindcijfer, is degene die dat cijfer kan toelichten of eventueel wijzigen.

Covering of *Disjoint* bij de generalisatie/specialisatie; beide lijken goed.

De typering van *Tentamenonderdeel* in *opgave*, *practicum*, *huiswerkopgave*.

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 de eigenschappen (zoveel mogelijk maar hógstens twee) die *wel* in het diagram staan maar *niet* uit de letterlijke tekst van de casus volgen.

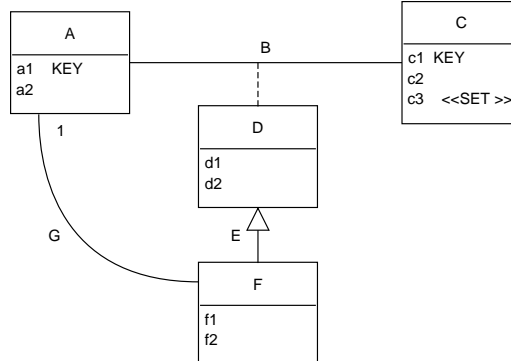
Onleesbare tekst wordt fout gerekend.

Er staat in de casus niet echt gegeven dat medewerkernummer, studentnummer en vakcode een sleutel zijn, maar dat is wel in het diagram aangegeven.

De generalisatie van *Docent* en *Student* tot *Persoon* staat niet letterlijk in de casus gegeven.

10p.

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



Het ERD kan op verschillende manieren vertaald worden naar een databaseschema dat geschikt is om informatie die past in het ERD op te slaan. U dient een manier te kiezen waarbij: (1) er zo *weinig mogelijk* relatieschema's in het databaseschema zijn, maar met de beperkingen dat (2) er geen NULLs nodig zijn vanwege de vertaling, (3) er geen redundantie geïntroduceerd wordt door de vertaling, en (4) 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:

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

Onleesbare tekst wordt fout gerekend.

$A(a1, a2, \text{primary key}(a1))$
$C(c1, c2, \text{primary key}(c1))$
$C'(c1, c3lid, \text{primary key}(c1, c3lid), \text{foreign key}(c1) \text{ references } C(c1))$
$D(a1, c1, d1, d2, \text{primary key}(a1, c1),$
foreign key $(a1)$ references $A(a1)$,
foreign key $(c1)$ references $C(c1)$)
$F(a1, c1, f1, f2, a1G, \text{primary key}(a1, c1),$
foreign key $(a1, c1)$ references $D1(a1, c1)$,
foreign key $(a1G)$ references $A(a1)$)
Eigenlijk moet de eigenschap not null toegevoegd worden aan iedere foreign key (en in instanties van het ERD mogen de key-attributen niet de waarde 'onbekend/null' hebben).
<i>(Zie Toelichting.)</i>

10p.

Opgave 3. Beschouw een universitaire omgeving met entiteiten zoals student, docent, vak, et cetera. Deze gegevens samen vormen als volgt een relatie \mathcal{R} :

Een tuple (V, O, H, S, C, D, A, W) zit op zeker tijdstip t in relatie \mathcal{R} precies wanneer al het volgende geldt op tijdstip t :

1. V is een V ak.
2. O is één van de *Ontwikkelaars* van vak V .
3. H is de (enige!) *Hoofddocent* van vak V .
4. S is één van de *Studenten* die vak V volgen.
6. C is een *Cijfer* dat student S in het verleden heeft behaald voor vak V .
5. D is de *Datum* waarop student S cijfer C heeft behaald voor vak V .
7. A is een *studentAssistent* bij vak V .
8. W is de indicatie of *studentassistent* A *Werkzaam* is in een studentenvereniging.

Voorts gelden er de volgende eigenschappen op ieder tijdstip:

- a. Een student kan per datum en vak hooguit één cijfer halen.
- b. De hoofddocent van een vak kan één van de ontwikkelaars van dat vak zijn.
- c. De hoofddocent van een vak kan niet een ontwikkelaar van een ander vak zijn.

Geef voor ieder van de functionele afhankelijkheden hieronder, met een letter W aan of die volgens bovenstaande definitie altijd *Waar* is in relatie \mathcal{R} , en met een letter O of die afhankelijkheid volgens bovenstaande definitie *Onwaar* kan zijn in relatie \mathcal{R} , en motiveer kort uw keuze (de motivatie telt mee in de beoordeling).

Onleesbare tekst wordt fout gerekend.

FD	W/O	Motivatie voor uw keuze
$O \rightarrow V$	O	iemand kan twee vakken hebben ontwikkeld
$V \rightarrow O$	O	een vak kan meerdere ontwikkelaars hebben
$H \rightarrow O$	O	een vak met hoofddocent H kan meerdere ontwikkelaars hebben
$O \rightarrow H$	O	iemand kan vakken ontwikkelen met verschillende hoofddocenten
$S, D \rightarrow C$	O	een student kan op een dag (voor 2 vakken) verschill. cijfers halen
$S, C, D \rightarrow V$	O	een stud. kan op een dag één cijfer halen voor verschillende vakken
$W \rightarrow A$	O	verschillende assistenten kunnen eenzelfde werkzaamheid hebben
$OHSDCAW \rightarrow VO$		bij gegeven O, H, S, D, C, A, W zijn verschillende vakken mogelijk

10p. **Opgave 4.** In het databaseschema van de afdeling personeelszaken van een bedrijf staan de volgende SQL statements:

```
create table Werknemer
  (sofi ..., naam ..., primary key (sofi));
create table Kind
  (sofi ..., naam ...,
   primary key (sofi, naam),
   foreign key (sofi) references Werknemer (sofi));
```

(Verklaring: Verschillende werknemers hebben verschillende SOFI-nummers. Van kinderen van werknemers is alleen de naam van belang voor het bedrijf; er wordt aangenomen dat verschillende kinderen van een werknemer verschillende namen hebben. Maar twee verschillende werknemers kunnen beide een kind “Jan” hebben. Daarom wordt bij ieder kind ook het SOFI-nummer van zijn ouder opgeslagen.)

Welke toevoegingen moeten gedaan worden aan welke `create table` statements opdat het verwijderen van een werknemer uit de tabel `Werknemer` tot gevolg heeft dat automatisch ook al zijn kinderen worden verwijderd? (Wanneer je de SQL syntaxis niet precies kent, beschrijf dan de kern van de oplossing zo goed mogelijk.)

Onleesbare tekst wordt fout gerekend.

Breid de foreign key declaratie in `Kind` uit:

```
create table Kind
  (sofi ..., naam ...,
   primary key (sofi, naam),
   foreign key (sofi) references Werknemer (sofi) on delete cascade);
```

Hoe kan de automatische verwijdering van kinderen ook gerealiseerd worden zonder de `create table` statements te wijzigen? (Wanneer je de SQL syntaxis niet precies kent, beschrijf dan de kern van de oplossing zo goed mogelijk.)

Onleesbare tekst wordt fout gerekend.

Twee mogelijkheden:

```
create trigger AutomatischeVerwijderingVanWezen
  after delete on Werknemer
  referencing old as W          | referencing old table as Ws
  for each row                 | for each statement
  (delete from Kind           | (delete from Kind
  where sofi = W.sofi);       | where sofi in (select sofi from Ws);
```

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

$$\mathcal{F} = \{A \rightarrow C, \quad BCD \rightarrow E, \quad EF \rightarrow D, \quad E \rightarrow A, \quad F \rightarrow B\}$$

- (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-conditie voor \mathbf{R} .

Onleesbare tekst wordt fout gerekend.

	X	\rightarrow	Y
20	<u>DE</u>	\rightarrow	AC
21	<u>DF</u>	\rightarrow	B
22	<u>EF</u>	\rightarrow	ABCD
23	ABC	\rightarrow	
24	ABD	\rightarrow	CE
27	ACD	\rightarrow	
28	ACE	\rightarrow	
29	ACF	\rightarrow	B
30	ADE	\rightarrow	C
31	<u>ADF</u>	\rightarrow	BCE
32	<u>AEF</u>	\rightarrow	BCD
33	<u>BCD</u>	\rightarrow	AE
35	BCF	\rightarrow	
37	BDF	\rightarrow	
38	<u>BEF</u>	\rightarrow	ACD
39	CDE	\rightarrow	A
40	<u>CDF</u>	\rightarrow	ABE
41	<u>CEF</u>	\rightarrow	ABD
42	<u>DEF</u>	\rightarrow	ABC
61	<u>ABDEF</u>	\rightarrow	C
62	<u>ACDEF</u>	\rightarrow	B
63	<u>BCDEF</u>	\rightarrow	A
64	<u>ABCDEF</u>	\rightarrow	

(Zie Toelichting.)

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

$$\mathcal{F} = \{AB \rightarrow C, \quad CD \rightarrow E, \quad EF \rightarrow A\}$$

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

Onleesbare tekst wordt fout gerekend.

Alle drie FDs uit \mathcal{F} vormen een schending; bijvoorbeeld voor $AB \rightarrow C$: deze is niet triviaal (dwz, het rechterlid C is niet deel van het linkerlid AB) en het linkerlid AB omvat geen sleutel. De sleutels van \mathbf{R} zijn $BDFx$ voor iedere keuze van x uit A, C , en E .

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 $F = AB \rightarrow C$. Splits \bar{R} in $\bar{R}_1 = ABC$ (alle attributen van F) en $\bar{R}_2 = ABDEF$ (alles zonder de attributen uit het rechterlid van F). 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:

$$\mathbf{R}_1 = (ABC, \{AB \rightarrow C\})$$

$$\mathbf{R}_2 = (ABDEF, \{ABD \rightarrow E, EF \rightarrow A\}).$$

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

Voer nu de volgende opdrachten uit:

- 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.
- Geef aan of de functionele afhankelijkheden behouden blijven onder de decompositie.

Onleesbare tekst wordt fout gerekend.

Let op: $ABD \rightarrow E$ volgt uit \mathcal{F} en zal dus (bij een correcte redenering) een FD worden van een component met attributen $ABD\dots E\dots$, ook al zit C daar niet bij. Net zo: $CDF \rightarrow A$ volgt uit \mathcal{F} en zal dus (bij een correcte redenering) een FD worden van een component met attributen $A\dots CD\dots F$, ook al zit E daar niet bij.

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.

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

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

- We bekijken nu $\mathbf{R}_2 = (ABDEF, \{ABD \rightarrow E, EF \rightarrow A\})$.

In \mathbf{R}_2 zijn $ABD \rightarrow E$ en $EF \rightarrow A$ beide een schending van de BCNF-conditie; voor $ABD \rightarrow E$ is de reden dat die niet-triviaal is en ABD geen sleutel is in \mathbf{R}_2 (want $ABD_{\mathcal{F}_2}^+ = ABDE \neq ABDEF$). We kiezen (zomaar) $ABD \rightarrow E$ ter eliminatie. Dus splitsen we $\bar{\mathbf{R}}_2$ in $\bar{\mathbf{R}}_{2a} = ABDE$ en $\bar{\mathbf{R}}_{2b} = ABDE \setminus F = ABDF$. Dit levert schema's $\mathbf{R}_{2j} = (\bar{\mathbf{R}}_{2j}, \mathcal{F}_{2j})$, waarbij \mathcal{F}_{2j} (een basis voor) de inperking is van \mathcal{F}^+ (of \mathcal{F}_2^+) tot $\bar{\mathbf{R}}_{2j}$. Dus

$$\mathbf{R}_{2a} = (ABDE, \{ABD \rightarrow E\}) \text{ en}$$

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

Beide schema's staan in BCNF (ze hebben geen niet-triviale FDs). De FDs van \mathbf{R}_2 zijn verloren gegaan.

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

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

$$\text{“}ABCDEF = ABC \bowtie ABDE \bowtie ABDF \text{ geldt in } \mathbf{R}.\text{”}$$

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

In de volgende opgavenserie wordt het volgende databaseschema gebruikt:

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

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

Battle (*name*, *date*)

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

De attributen die tot de sleutel behoren zijn onderstreept.

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

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

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

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

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

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

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

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

B (*n*, *d*)

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

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



Geef ieder schip dat met een *ander* schip van dezelfde klasse deelneemt aan eenzelfde zeeslag.

Geef voor deze vraag een *afleiding in kleine stappen* in Verzamelingsnotatie.

De verkregen eindvorm moet dicht aansluiten bij SQL en, na omzetting naar SQL, geen subqueries hebben en geen overbodige tabellen gebruiken en geen group-by clauses bevatten. Kort tabel- en attribuutnamen af tot hun eerste letter. Het begin is al gegeven.

Onleesbare tekst wordt fout gerekend.

<p>“ieder schip dat met een <i>ander</i> schip van dezelfde klasse deelneemt aan eenzelfde zeeslag”</p>
<p>= $\{s \mid (\exists s' \mid s' \neq s \wedge \text{“klasse van } s' \text{ is die van } s\text{”} \bullet (\exists b \bullet \text{“}s' \text{ en } s \text{ nemen deel aan } b\text{”})) \bullet s.n\}$</p>
<p>= $\{s \mid (\exists s' \mid s' \neq s \wedge s'.c=s.c \bullet (\exists b \bullet \text{“}s' \text{ en } s \text{ nemen deel aan } b\text{”})) \bullet s.n\}$</p>
<p>= $\{s \mid (\exists s' \mid s' \neq s \wedge s'.c=s.c \bullet (\exists b \bullet (\exists o, o' \mid o'.s=s'.n \wedge o.s=s.n \bullet o'.b=o.b=b.n))) \bullet s.n\}$</p>
<p>= $\{s \mid (\exists s' \mid s' \neq s \wedge s'.c=s.c \bullet (\exists b, o, o' \mid o'.s=s'.n \wedge o.s=s.n \bullet o'.b=o.b=b.n)) \bullet s.n\}$</p>
<p>= [shunting]</p>
<p>$\{s \mid (\exists s' \mid s' \neq s \wedge s'.c=s.c \bullet (\exists o, o' \mid o'.s=s'.n \wedge o.s=s.n \bullet o'.b=o.b \wedge (\exists b \bullet o.b=b.n))) \bullet s.n\}$</p>
<p>= [in O is b een foreign key naar $B(n)$; dus voor o geldt: $\exists b \bullet o.b = b.n$]</p>
<p>$\{s \mid (\exists s' \mid s' \neq s \wedge s'.c=s.c \bullet (\exists o, o' \mid o'.s=s'.n \wedge o.s=s.n \bullet o'.b=o.b)) \bullet s.n\}$</p>
<p>= [shunting]</p>
<p>$\{s, s' \mid s' \neq s \wedge s'.c=s.c \wedge (\exists o, o' \mid o'.s=s'.n \wedge o.s=s.n \bullet o'.b=o.b) \bullet s.n\}$</p>
<p>= [shunting]</p>
<p>$\{s, s', o, o' \mid s' \neq s \wedge s'.c=s.c \wedge o'.s=s'.n \wedge o.s=s.n \wedge o'.b=o.b \bullet s.n\}$</p>
<p> </p>
<p> </p>
<p>We hebben korthedshalve vaak $s : S$ afgekort tot s, en net zo bij $c:C$, $b:B$, $o:O$.</p>
<p> </p>
<p> </p>

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

Onleesbare tekst wordt fout gerekend.

<p>select distinct $s.name$</p>
<p>from $S s, S s', O o, O o'$</p>
<p>where $s' \neq s$ and $s'.c=s.c$ and $o'.s=s'.n$ and $o.s=s.n$ and $o'.b=o.b$</p>
<p> </p>
<p> </p>

Geef ook een formulering van de vraag in Relationele Algebra waarin zoveel mogelijk gebruik gemaakt wordt van de natural join (\bowtie) in plaats van andere vormen van joins of het Cartesische product.

Onleesbare tekst wordt fout gerekend.

$\pi_{sn} (\sigma_{sn' \neq sn''} (S' \bowtie S'' \bowtie O' \bowtie O''))$
Hierbij zijn met renaming nieuwe relaties gedefinieerd:
$S' = S[sn', cn, sl']$ en $O' = O[sn', ob, or']$ en $S'' = S[sn'', cn, sl'']$ en $O'' = O[sn'', ob, or'']$.
(Let op de gelijkheid en ongelijkheid van de nieuwe attribuutnamen.)

<i>Class</i> (<u>name</u> , type, country, guns, bore, displacement)	C (<u>n</u> , t, c, g, b, d)
<i>Ship</i> (<u>name</u> , classname, launched)	S (<u>n</u> , c, l)
<i>Battle</i> (<u>name</u> , date)	B (<u>n</u> , d)
<i>Outcome</i> (<u>shipname</u> , battlename, result)	O (<u>s</u> , b, r)

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

Geef voor iedere combinatie van g en b het *aantal* (*mits* > 0) *van zeeslagen* waarin minstens één schip van een ‘klasse met g en b als waarde voor *guns* en *bore*’ gezonken is, én de *datum* van de eerste van die zeeslagen.

Onleesbare tekst wordt fout gerekend.

select <i>c.guns</i> , <i>c.bore</i> , count(<i>distinct b.name</i>), min(<i>b.date</i>)
from C <i>c</i> , S <i>s</i> , O <i>o</i> , B <i>b</i>
where <i>c.name</i> = <i>s.class</i> and <i>s.name</i> = <i>o.ship</i> and <i>o.result</i> = sunk and <i>o.battle</i> = <i>b.name</i>
group by <i>c.guns</i> , <i>c.bore</i>

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 iedere klasse waarvoor er een zeeslag is aan te wijzen zódanig dat *elk* schip van die klasse dat in die zeeslag zinkt, een andere zeeslag overleeft.

Formuleer de vraag in Verzamelingsnotatie op een manier die zo *direct mogelijk* aansluit bij de gegeven formulering van de vraag.

Onleesbare tekst wordt fout gerekend.

Kortheidshalve laten we de typering achterwege (en korten dus $b : B$ af tot b , et cetera).

$$\{c \mid (\exists b \bullet (\forall s \mid s.c=c.n \wedge (\exists o \bullet o.s=s.n \wedge o.b=b.n \wedge o.r=sunk)$$

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

$$\bullet c.n\}$$

(Zie Toelichting.)

Geef hier de formulering in Verzamelingsnotatie van de SQL query die u zo dadelijk gaat geven. (Een afleiding op kladpapier zou u hierbij kunnen helpen!)

Onleesbare tekst wordt fout gerekend.

$$\{c, b \mid \neg (\exists s, o \mid s.c=c.n \wedge o.s=s.n \wedge o.b=b.n \wedge o.r=sunk$$

$$\bullet \neg (\exists o' \mid o'.s=s.n \bullet o'.r \neq sunk))$$

$$\bullet c.n\}$$

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

Onleesbare tekst wordt fout gerekend.

select distinct $c.n$ from C c , B b where not exists (

select * from S s , O o where

$s.c=c.n$ and $o.s=s.n$ and $o.b=b.n$ and $o.r=sunk$ and not exists (

select * from O o' where $o'.s=s.n$ and $o'.r \neq sunk$))

5p. **Opgave 10.** In sommige databasesystemen (DBMSs) wordt automatisch een *index* aangemaakt voor de *primary key* van een tabel. Wat is het nut hiervan?

Onleesbare tekst wordt fout gerekend.

Bij een index voor attribuut A zijn de rijen met gegeven waarde voor A efficiënt te vinden. De rijen (hooguit één) met gegeven waarde voor de *primary key* moeten vaak gevonden worden: om bij toevoegingen en wijzigingen te controleren of er geen rijen met gelijke *primary keys* ontstaan, en bij joins met ander tabellen. Dus een index op de *primary key* verhoogt de efficiëntie van zowel deze controle, als ook de joins.

5p. **Opgave 11.** Zet in onderstaande tabel een merkteken ‘✓’ bij iedere combinatie van isolatieniveaus voor transacties T_1 en T_2 waarbij transactie T_1 géén phantoms ervaart bij gelijktijdige executie van beide transacties.

Onleesbare tekst wordt fout gerekend.

Op de met ‘✓’ gemerkte niveaus ervaart T_1 geen phantoms:

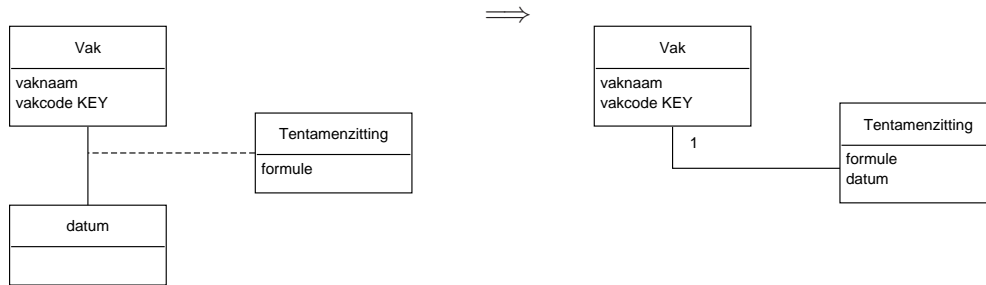
Isolatieniveau voor T_2 : ↓	Isolatieniveau voor T_1 :			
	read uncommitted	read committed	repeatable read	serializable
read uncommitted				✓
read committed				✓
repeatable read				✓
serializable				✓

Het niveau voor T_2 doet niet terzake!

Toelichtingen

Toelichting bij antwoord 1.

In plaats van *Datum* als aparte entiteit (die verder niet *als entiteit* interessant is), kun je *datum* als attribuut opnemen in *Tentamenzitting* en dan een relatie tussen *Vak* en *Tentamenzitting* postuleren:



Er zijn twee verschillen met de gegeven uitwerking: ten eerste kan *Vak* nu meermalen per dag getentamineerd worden en ten tweede “bestaan” nu alleen de datums van *Tentamenzittingen* en geen andere.

Alternatieve namen voor *Resultaat* en *Deelresultaat* zijn, bijvoorbeeld, *Deelname* en *Uitwerking*.

Toelichting bij antwoord 2.

De verzameling $c3$ van een C -instantie van het ERD, met primary key waarde (x) , wordt uit de tabelinhouden berekend als de verzameling $\{c' : C' \mid c'(c1) = (x) \bullet c'(c3lid)\}$.

Relatie B is per definitie identiek aan relatie D ; het zijn synoniemen. En er is expliciet gevraagd zo weinig mogelijk tabellen te gebruiken en daarom is er geen tabel B opgenomen.

De generalisatie/specialisatie E wordt gerepresenteerd door de foreign key eigenschap in tabel F die verwijst naar tabel D . De F -instanties worden niet opgenomen in de tabel voor de D -instanties omdat dan NULLs nodig zouden zijn (en dat mag niet volgens de opgave) om een D -instantie te representeren die geen F -instantie is. Het is fout om in tabel F in plaats van de foreign key $(a1, c1)$ naar $D(a1, c1)$ te declareren dat $(a1)$ een foreign key naar $A(a1)$ is en $(c1)$ naar $C(c1)$ (want dat garandeert niet dat elke $(a1, c1)$ -waarde in F ook voorkomt in een rij in D).

Er is geen aparte tabel nodig om de instanties van de G -relatie te representeren; vanwege de ‘1’-multipliciteit van relatie G kan een G -instantie met als sleutelwaarde (x, y) (d.w.z., x is de waarde van sleutel-van- A , en y is de waarde van sleutel-van- F) gerepresenteerd worden in tabel F door de rij met precies x en y als waarden van de attributen $a1G$ en sleutel-van- F (en in de gegeven uitwerking is $(a1, c1)$ de sleutel-van- F).

Toelichting bij antwoord 5.

Let op: iedere key is ook een superkey, omgekeerd is dit niet het geval.

Toelichting bij antwoord 9.

In het gegeven antwoord is het deel $\exists b'$ met $o'.b = b'.n$ overbodig zoals hieronder in de afleiding blijkt. De test $b \neq b'$ is overbodig omdat twee zeeslagen waarin s overleeft respectievelijk zinkt, niet gelijk kunnen zijn; formeel: in O is (s, b) een key, dus zowel een rij $(s_0, b_0, sunk)$

in O als ook een rij $(s_0, b_0, ok/damaged)$ in tabel O is onmogelijk.

Een (gedetailleerde) afleiding tot en met een SQL formulering luidt als volgt:

$$\begin{aligned}
& \{c \mid (\exists b \bullet \text{“elk schip van } c \text{ dat zinkt in } b \text{ overleeft een (andere) zeeslag”}) \bullet c.n\} \\
= & \quad [\text{vertaling}] \\
& \{c, b \mid \text{“elk schip van } c \text{ dat zinkt in } b \text{ overleeft een (andere) zeeslag”} \bullet c.n\} \\
= & \quad [\text{vertaling}] \\
& \{c, b \mid (\forall s \mid \text{“}s \text{ is van } c\text{”} \wedge \text{“}s \text{ zinkt in } b\text{”} \bullet \text{“}s \text{ overleeft een (andere) zeeslag”}) \bullet c.n\} \\
= & \quad [\text{vertaling}] \\
& \{c, b \mid (\forall s \mid s.c=c.n \wedge (\exists o \bullet o.s=s.n \wedge o.b=b.n \wedge o.r=sunk) \bullet (\exists b' \bullet \text{“}s \text{ overleeft } b'\text{”})) \bullet c.n\} \\
= & \quad [\text{shunting}] \\
& \{c, b \mid (\forall s, o \mid s.c=c.n \wedge o.s=s.n \wedge o.b=b.n \wedge o.r=sunk \bullet (\exists b' \bullet \text{“}s \text{ overleeft } b'\text{”})) \bullet c.n\} \\
= & \quad [\text{vertaling}] \\
& \{c, b \mid (\forall s, o \mid s.c=c.n \wedge o.s=s.n \wedge o.b=b.n \wedge o.r=sunk \\
& \quad \bullet (\exists b' \bullet (\exists o' \mid o'.s=s.n \wedge o'.b=b'.n \bullet o'.r \neq sunk))) \\
& \quad \bullet c.n\} \\
= & \quad [\text{predicaten logica: } \exists x \bullet \exists y \dots = \exists x, y \dots] \\
& \{c, b \mid (\forall s, o \mid s.c=c.n \wedge o.s=s.n \wedge o.b=b.n \wedge o.r=sunk \\
& \quad \bullet (\exists b', o' \mid o'.s=s.n \wedge o'.b=b'.n \bullet o'.r \neq sunk)) \\
& \quad \bullet c.n\} \\
= & \quad [\text{shunting}] \\
& \{c, b \mid (\forall s, o \mid s.c=c.n \wedge o.s=s.n \wedge o.b=b.n \wedge o.r=sunk \\
& \quad \bullet (\exists o' \mid o'.s=s.n \wedge (\exists b' \bullet o'.b=b'.n) \bullet o'.r \neq sunk)) \\
& \quad \bullet c.n\} \\
= & \quad [\text{In } O \text{ is } b \text{ een foreign key naar } B.n; \text{ dus voor } o' \text{ geldt: } (\exists b' \bullet o'.b=b'.n)] \\
& \{c, b \mid (\forall s, o \mid s.c=c.n \wedge o.s=s.n \wedge o.b=b.n \wedge o.r=sunk \\
& \quad \bullet (\exists o' \mid o'.s=s.n \bullet o'.r \neq sunk)) \\
& \quad \bullet c.n\} \\
= & \quad [\text{predicaten logica: } \neg \forall D \mid P \bullet Q = \exists D \mid P \bullet \neg Q] \\
& \{c, b \mid \neg (\exists s, o \mid s.c=c.n \wedge o.s=s.n \wedge o.b=b.n \wedge o.r=sunk \\
& \quad \bullet \neg (\exists o' \mid o'.s=s.n \bullet o'.r \neq sunk)) \\
& \quad \bullet c.n\} \\
= & \text{select distinct } c.n \text{ from } C \text{ } c, B \text{ } b \text{ where not exists (} \\
& \quad \text{select * from } S \text{ } s, O \text{ } o \text{ where} \\
& \quad \quad s.c=c.n \text{ and } o.s=s.n \text{ and } o.b=b.n \text{ and } o.r=sunk \text{ and not exists (} \\
& \quad \quad \text{select * from } O \text{ } o' \text{ where } o'.s=s.n \text{ and } o'.r \neq sunk \text{))}
\end{aligned}$$

Uit de een-na-laatste formulering in verzamelingsnotatie volgen onmiddellijk ook de volgende:

$$\begin{aligned}
& \{c, b \mid (\forall s, o \mid s.c=c.n \wedge \text{etc} \wedge o.r=sunk \bullet (\exists o' \mid o'.r \neq sunk \bullet o'.s=s.n)) \bullet c.n\} \\
= & \{c, b \mid (\forall s, o \mid s.c=c.n \wedge \text{etc} \wedge o.r=sunk \bullet s.n \in \{o' \mid o'.r \neq sunk \bullet o'.s\}) \bullet c.n\} \\
= & \{c, b \mid \{s, o \mid s.c=c.n \wedge \text{etc} \wedge o.r=sunk \bullet s.n\} \subseteq \{o' \mid o'.r \neq sunk \bullet o'.s\} \bullet c.n\} \\
= & \{c, b \mid \{s, o \mid s.c=c.n \wedge \text{etc} \wedge o.r=sunk \bullet s.n\} \setminus \{o' \mid o'.r \neq sunk \bullet o'.s\} = \emptyset \bullet c.n\}
\end{aligned}$$

De tweede en laatste hiervan kunnen ook direct in SQL omgezet worden (met behulp van *in* en *except*).