

Tentamen Gegevensbanken (19.211074.1) — 3 november 2011

CONTROLEER EERST OF ALLE BLADZIJDEN T/M BLZ. 16 AANWEZIG ZIJN!

NAAM, VOORLETTERS: _____

STUDENTNUMMER: _____

OPLEIDING: _____

De uitwerkingen moeten op deze opgavenformulieren worden genoteerd in de daarvoor bestemde vakken. Alle overige ruimte kun je zo nodig als **kladpapier** gebruiken en wordt niet bekeken en niet beoordeeld.

[Reeds op Blackboard aangekondigd:] Bij het tentamen mogen geen boeken en dergelijke gebruikt worden behoudens **één dubbelzijdig gebruikt** vel van A4-formaat met daarop eigen aantekeningen of kopieën van delen van het boek; **kopieën van tentamenuitwerkingen en ander materiaal zijn niet toegestaan** (dat moet dan maar in eigen aantekeningen verwerkt worden).

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

Na afloop moet de *volledige* set opgavenformulieren worden ingeleverd; het kladpapier niet. De tentamenopgaven zijn niet geheim en worden voorzien van modeluitwerkingen op Blackboard gepubliceerd. (Die modeluitwerkingen moet je op papier of elektronisch bij je hebben wanneer je je tentamen komt inzien.)

Niet invullen

5 gratis	1	2	3	4	5	6	7	8	9	10	bonus
-------------	---	---	---	---	---	---	---	---	---	----	-------

10p. **Opgave 1.** De volgende casus gaat over het tentamineren van vakken aan een universiteit.

Om voor studenten, per vak, een eindcijfer te bepalen, worden er tentamenzittingen gehouden:

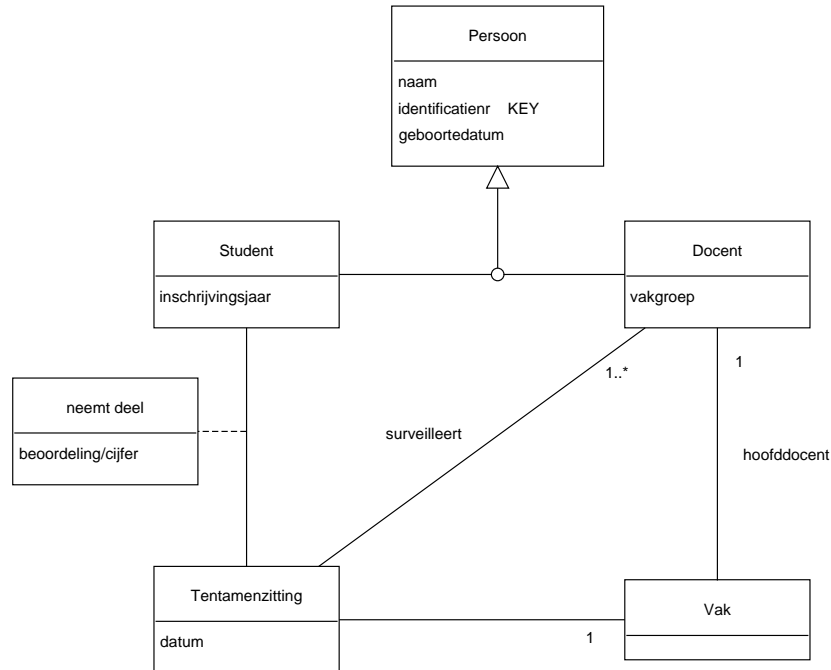
- Een tentamenzitting hoort bij precies één vak; per vak kunnen er meerdere tentamenzittingen (maar op verschillende data) gehouden worden.
- Een student die deelneemt aan een tentamenzitting krijgt een beoordeling (bij verschillende zittingen mogelijk verschillende beoordelingen).
- De beoordeling die een student gekregen heeft op de meest recente tentamenzitting, geldt als eindcijfer voor die student voor dat vak.
- Van een vak is bekend wie de hoofddocent ervan is.
- Bij een tentamenzitting zijn er één of méér docenten die surveilleren.
- Van personen (docenten en studenten) is de naam bekend, als ook het identificatienummer (studentnummer of medewerkernummer), en het geboortejaar. Van een student is het jaar van eerste inschrijving bekend. Van een docent is de vakgroep bekend.

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

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

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

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



(Zie Toelichting.)

Hier is ruimte voor eventuele toelichting:

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

In *Persoon* is *identificatienr* een key (het alternatief van keys *Snummer* en *Mnummer* in *Student* en *Docent* is ook toegestaan); van *Tentamenzitting* en *Vak* zijn de keys niet gegeven en niet duidelijk uit de casustekst.

Over *vakgroep* wordt kennelijk geen informatie bijgehouden (in de casustekst worden er geen attributen van gegeven), en daarom is *vakgroep* als attribuut gemodelleerd niet als entiteit.

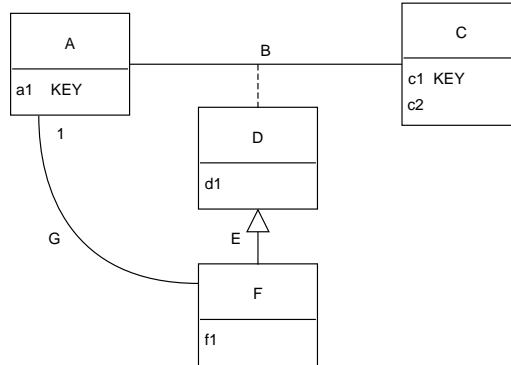
Een relatie *doceert* tussen *Vak* en *Docent* is niet gegeven in de casus en wordt niet gevraagd.

Wel in de casus gegeven maar niet in het diagram opgenomen: (1) Tentamenzittingen voor eenzelfde vak hebben verschillende data. (2) Het eindcijfer voor een vak is het meest recente tentamencijfer. (Beide aspecten worden besproken in de *Toelichting* op het ERD.)

10p.



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



Vertaal het ER-diagram naar een databaseschema dat *precies* de informatie op kan slaan die past in het ER-diagram, en voldoet aan de volgende eigenschappen:

- er zijn geen NULLs nodig vanwege de vertaling,
 - er wordt geen redundantie geïntroduceerd door de vertaling,
 - alle attributen hebben atomaire waarden,
- en verder, zoveel als mogelijk na vervulling van voorgaande eisen,
- er zijn zo *weinig mogelijk* tabelschema's.

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

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

Gebruik afkortingen, zoals "PK" voor "primary key", etc., en gebruik zonodig ook CHECKs.

Onleesbare tekst wordt fout gerekend.

A(a1, primary key(a1))
C(c1, c2, primary key(c1))
D(a1, c1, d1 primary key(a1, c1), foreign key(a1) references A(a1), foreign key(c1) references C(c1))
F(a1, c1, f1, f2, aG, primary key(a1, c1), foreign key(a1, c1) references D1(a1, c1), foreign key(aG) references A(a1))
Eigenlijk moet de eigenschap <code>not null</code> toegevoegd worden aan iedere foreign key (en verondersteld is dat in instanties van het ERD de key-attributen niet de waarde 'null' hebben).
(Zie Toelichting.)

10p.

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

$$\bar{R} = ABCDE$$

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

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

Onleesbare tekst wordt fout gerekend.

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

(Zie Toelichting.)

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

$$\mathcal{F} = \{AB \rightarrow CD, BC \rightarrow E, GH \rightarrow F, F \rightarrow E\}$$

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 een sleutel, namelijk: $ABGH$. 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.

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

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

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

$$\mathbf{R}_1 = (ABCD, \{AB \rightarrow CD\})$$

$$\mathbf{R}_2 = (ABEFGH, \{AB \rightarrow E, GH \rightarrow F, F \rightarrow E\})$$

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

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

NB3. \mathbf{R}_1 staat in BCNF (de key is AB) en \mathbf{R}_2 bevat precies drie schendingen van de BCNF-conditie; samen dus minder dan het aantal schendingen in \mathbf{R} .

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

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

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

We passen het BCNF-algoritme toe.

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

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

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

- We bekijken nu $\mathbf{R}_2 = (ABEFGH, \{AB \rightarrow E, GH \rightarrow F, F \rightarrow E\})$.

In \mathbf{R}_2 zijn $AB \rightarrow E$ en $GH \rightarrow F$ en $F \rightarrow E$ alledrie een schending van de BCNF-conditie; voor $GH \rightarrow F$ is de reden dat die niet-triviaal is en GH geen sleutel is in \mathbf{R}_2 (want $GH_{\mathcal{F}_2}^+ = EFGH \neq ABEFGH$). We kiezen (zomaar) $GH \rightarrow F$ ter eliminatie. Dus splitsen we \bar{R}_2 in $\bar{R}_{2a} = GHF$ en $\bar{R}_{2b} = ABE\cancel{F}GH = ABEGH$. Dit levert schema's $\mathbf{R}_{2j} = (\bar{R}_{2j}, \mathcal{F}_{2j})$, waarbij \mathcal{F}_{2j} (een basis voor) de inperking is van \mathcal{F}^+ (of \mathcal{F}_2^+) tot \bar{R}_{2j} . Dus $\mathbf{R}_{2a} = (GHF, \{GH \rightarrow F\})$ en $\mathbf{R}_{2b} = (ABEGH, \{AB \rightarrow E, GH \rightarrow E\})$. Let op: de $GH \rightarrow E$ zit weliswaar niet in \mathcal{F}_2 , maar toch in \mathcal{F}_{2b} zoals aan het begin is uitgelegd.

- We bekijken nu $\mathbf{R}_{2a} = (FGH, \{GH \rightarrow F\})$. Deze staat in BCNF.

- We bekijken nu $\mathbf{R}_{2b} = (ABEGH, \{AB \rightarrow E, GH \rightarrow E\})$. Zowel $AB \rightarrow E$ als ook $GH \rightarrow E$ vormen een schending van de BCNF-conditie in \mathbf{R}_{2b} ; bijvoorbeeld voor $AB \rightarrow E$: die FD is niet triviaal en het linkerlid is geen sleutel (want $AB_{\mathcal{F}_{2b}}^+ = ABE \neq ABEGH$). We kiezen (zomaar) $AB \rightarrow E$ ter eliminatie. Dus splitsen we \bar{R}_{2b} in $\bar{R}_{2b1} = ABE$ en $\bar{R}_{2b2} = AB\cancel{E}GH = ABGH$. Dit levert schema's $\mathbf{R}_{2bx} = (\bar{R}_{2bx}, \mathcal{F}_{2bx})$, waarbij \mathcal{F}_{2bx} (een basis voor) de inperking is van \mathcal{F}^+ (of \mathcal{F}_{2b}^+) tot \bar{R}_{2bx} . Dus $\mathbf{R}_{2b1} = (ABE, \{AB \rightarrow E\})$ en $\mathbf{R}_{2b2} = (ABGH, \{\})$. Beide staan in BCNF.

- Dus $\{\mathbf{R}_1, \mathbf{R}_{2a}, \mathbf{R}_{2b1}, \mathbf{R}_{2b2}\}$ is een decompositie van \mathbf{R} waarvan alle componenten in BCNF staan.

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

$$"ABCDEF GH = ABCD \bowtie FGH \bowtie ABE \bowtie ABGH \text{ geldt in } \mathbf{R}."$$

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

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

$BC \rightarrow E$ is niet behouden gebleven, al in de eerste stap.

10p.

Opgave 5. Voor een bedrijf dat producten verkoopt aan klanten, spelen (bestel)nummer, klantnaam, rekeningnummer, productnummer, omschrijving, beoordeling een rol in de bedrijfsvoering.

We definiëren een relatie \mathcal{R} die deze entiteiten aan elkaar relateert. Een tuple (N, K, R, P, O, B) zit op tijdstip t in relatie \mathcal{R} precies wanneer op tijdstip t het volgende geldt:

1. N : is het *N*ummer van een bestelling (N identificeert een bestelling)
2. K : is de *K*lantnaam van een klant die bestelling N doet
3. R : is het *R*ekeningnummer dat klant K bij bestelling N heeft opgegeven
4. P : is het *P*roductnummer van een product besteld door klant K in bestelling N
5. O : is de *O*mschrijving van product met productnummer P
6. B : is de *B*oordeling door klant K van product met productnummer P

Toelichting:

7. Een bestelling wordt door precies één klant gedaan.
8. Verschillende klanten hebben verschillende klantnamen.
9. Verschillende producten hebben verschillende productnummers.
10. Een klant kan een product in één bestelling hooguit éénmaal bestellen.
11. Een product kan door meerdere klanten besteld worden.
12. Een klant kan een product hooguit één beoordeling gegeven hebben. Heeft een klant voor een product nog geen beoordeling gegeven, dan heeft "de beoordeling van dat product door die klant" een *default* waarde.

Geef hieronder voor ieder van de functionele afhankelijkheden (FD) en multi-valued dependencies (MVD), met een letter W of O aan of die *W*aar of *O*nwaar is in relatie \mathcal{R} , en motiveer kort uw keuze zo *mogelijk* met verwijzing naar gegevens 1..12. De motivatie telt even zwaar mee in de beoordeling als het antwoord *Waar/Onwaar*.

Onleesbare tekst wordt fout gerekend.

FD	W/O	motivatie
$N \rightarrow K$	W	volgens 7 (dus: <i>de</i> klant bij N) en 2 of 8 (<i>de</i> klantnaam van K)
$N \rightarrow R$	W	volgens voorgaande $N \rightarrow K$ en 3 (<i>het</i> rekeningnummer bij N, K)
$N \rightarrow P$	O	(ondanks 10); per N kunnen meerdere producten besteld worden
$K \rightarrow R$	O	(ondanks 3); een K kan meerdere rekeningen hebben
$NK \rightarrow P$	O	(ondanks 10); per N kan een K meerdere producten bestellen (vgl. 4)
$KP \rightarrow B$	W	volgens 6 (dus: <i>de</i> beoordeling door K voor P) en ook volgens 12
$P \rightarrow O$	W	volgens 5 en 9 (dus: <i>de</i> omschrijving van <i>het</i> product met nr P)
$PB \rightarrow K$	O	een B voor P kan van meerdere klanten afkomen (bv de default)
$NKRPOB = NKRP \bowtie NOB$, O,		$\left\{ \begin{array}{l} \text{Kies voor } i=1,2 \text{ rijen } (n, \dots, p_i, \dots, o_i) \in \mathcal{R} \text{ waarbij} \\ (n, \dots, p_i, \dots, o_j) \notin \mathcal{R} \text{ voor } i \neq j. \text{ Decompositie} \\ \text{en dan joinen levert toch zo'n rij die niet in } \mathcal{R} \text{ zit.} \end{array} \right.$
$NKRPOB = NK \bowtie NRPOB$, W, $NK \cap NRPOB = N$ is een key in NK .		

10p. **Opgave 6.** Beschouw de twee volgende transacties:

$$T_1 = r_1(x); w_1(y); w_1(x)$$
$$T_2 = r_2(y); r_2(x); w_2(y)$$

Hierbij staan x, y voor verschillende data items, en $r_i(\)$, $w_i(\)$ voor een lees- en schrijfactie.

Met deze abstractie van transacties is er geen verschil tussen gelijktijdige executie indien van iedere transactie het isolatieniveau REPEATABLE READ of SERIALIZABLE is; met andere woorden, de verzameling van schedules die kunnen ontstaan bij gelijktijdige executie van de transacties, is onafhankelijk van het isolatieniveau dat voor T_1 en voor T_2 geeist wordt, indien beide maar REPEATABLE READ of SERIALIZABLE zijn. Geef een verklaring hiervoor:

Onleesbare tekst wordt fout gerekend.

Het verschil tussen REPEATABLE READ en SERIALIZABLE uit zich alleen door phantoms. Phantoms zijn alleen mogelijk bij row locking (in plaats van table locking). Bij de gegeven transacties is geabstraheerd van het verschil tussen rijen en tabellen: er is alleen sprake van "data items".

De isolatieniveaus van transacties T_1 en T_2 worden op SERIALIZABLE gesteld, en in de implementatie wordt dit met het locking protocol gerealiseerd.

De transacties worden tegelijkertijd uitgevoerd. Geef hiervoor alle schedules die beginnen met $r_1(x); r_2(y); r_2(x); \dots$ en waarin iedere transactie met een "commit" eindigt:

Onleesbare tekst wordt fout gerekend.

$r_1(x); r_2(y); r_2(x); w_2(y); c_2; w_1(y); w_1(x); c_1$

NB1. Omdat $r_1(x)$ commuteert met ieder van $r_2(y), r_2(x), w_2(y)$, is snel in te zien dat het schedule equivalent is met $T_2; T_1$.

NB2. Net nadat het gegeven beginstuk $r_1(x); r_2(y); r_2(x)$ is uitgevoerd, geldt het volgende.

(1) T_2 heeft een readlock op x en y , dus kan T_1 de writelock op y niet verkrijgen (en moet dus wachten), en dat blijft zo totdat T_2 bij zijn commit de lock (inmiddels een writelock) op y weer vrij geeft. (2) T_1 heeft een readlock op x (tegelijk met $T_2!$), dus kan T_2 de writelock op y verkrijgen en zijn executie voltooien.

Geef nu, voor het oorspronkelijk gegeven schema, een of meer triggers waardoor de gewenste acties (a) en (b) automatisch door het databasesysteem worden uitgevoerd.

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

Onleesbare tekst wordt fout gerekend.

Twee mogelijkheden voor (a):

create trigger VerwijderingVanWezen	
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);

Mogelijkheid voor (b):

create trigger InvoeringVanWezen	
before insert on Kind	
referencing new as K	referencing new table as Ks
for each row	for each statement
when K.sofi not in	when exist (select sofi from Ks except
(select sofi from Werknemer)	select sofi from Werknemer)
rollback	rollback

In het rechterdeel worden *alle* toevoegingen van de 'insert into Kind' statement ongedaan gemaakt; dat is méér dan gevraagd.

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 identificers 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:

Vind ieder schip dat twee (of meer) zeeslagen heeft doorstaan met verschillend resultaat.

Geef voor deze vraag een afleiding in kleine stappen in verzamelingsnotatie naar een vorm die dicht aansluit bij een SQL query die geen subqueries heeft en zo weinig mogelijk tabellen in de from clause. Het begin is al gegeven. Kort identifiers tot hun eerste letter af.

Onleesbare tekst wordt fout gerekend.

<p>“ieder schip dat twee (of meer) zeeslagen <i>verschillend</i> heeft doorstaan”</p>
<p>= $\{s : S \mid \text{“}s \text{ heeft twee (of meer) zeeslagen } \textit{verschillend} \text{ doorstaan”} \bullet s.n\}$</p>
<p>= $\{s : S \mid (\exists b_1, b_2 : B \bullet \text{“}s \text{ heeft } b_1, b_2 \text{ op verschillende wijze doorstaan”}) \bullet s.n\}$</p>
<p>= $\{s : S \mid (\exists b_1, b_2 : B \bullet \exists o_1, o_2 : O \bullet$</p>
<p style="padding-left: 40px;">“s heeft volgens o_1, o_2 deelgenomen aan b_1, b_2 met verschillende resultaten”) $\bullet s.n\}$</p>
<p>= $\{s : S \mid (\exists b_1, b_2 : B \bullet \exists o_1, o_2 : O \bullet$</p>
<p style="padding-left: 80px;">$o_1.s = s.n = o_2.s \wedge o_1.b = b_1.n \wedge o_2.b = b_2.n \wedge o_1.r \neq o_2.r) \bullet s.n\}$</p>
<p>= $\{s : S \mid (\exists o_1, o_2 : O \bullet \exists b_1, b_2 : B \bullet$</p>
<p style="padding-left: 80px;">$o_1.s = s.n = o_2.s \wedge o_1.b = b_1.n \wedge o_2.b = b_2.n \wedge o_1.r \neq o_2.r) \bullet s.n\}$</p>
<p>= $\{s : S \mid (\exists o_1, o_2 : O \bullet o_1.s = s.n = o_2.s \wedge$</p>
<p style="padding-left: 80px;">$(\exists b_1 : B \bullet o_1.b = b_1.n) \wedge (\exists b_2 : B \bullet o_2.b = b_2.n) \wedge o_1.r \neq o_2.r) \bullet s.n\}$</p>
<p>= [in O is b een foreign key referencing $B(n)$, dus voor all o geldt: $\exists b : B \bullet o.b = b.n$]</p>
<p>$\{s : S \mid (\exists o_1, o_2 : O \bullet o_1.s = s.n = o_2.s \wedge true \wedge true \wedge o_1.r \neq o_2.r) \bullet s.n\}$</p>
<p>= $\{s : S \mid (\exists o_1, o_2 : O \bullet o_1.s = s.n = o_2.s \wedge o_1.r \neq o_2.r) \bullet s.n\}$</p>
<p>= $\{s : S; o_1, o_2 : O \mid o_1.s = s.n = o_2.s \wedge o_1.r \neq o_2.r \bullet s.n\}$</p>
<p>= $\{s : S; o_1, o_2 : O \mid o_1.s = s.n \wedge o_1.s = o_2.s \wedge o_1.r \neq o_2.r \bullet o_1.s\}$</p>
<p>= $\{o_1, o_2 : O \mid (\exists s : S \bullet o_1.s = s.n) \wedge o_1.s = o_2.s \wedge o_1.r \neq o_2.r \bullet o_1.s\}$</p>
<p>= [in O is s een foreign key referencing $S(n)$, dus voor all o geldt: $\exists s : S \bullet o.s = s.n$]</p>
<p>$\{o_1, o_2 : O \mid o_1.s = o_2.s \wedge o_1.r \neq o_2.r \bullet o_1.s\}$ (Zie Toelichting.)</p>

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.

<pre>select distinct o1.s</pre>
<pre>from O o1, O o2</pre>
<pre>where o1.s=o2.s and o1.r<>o2.r</pre>
<hr/>

Geef een formulering van de vraag in Relationale Algebra, zonder het Cartesisch product \times te gebruiken [toegevoegd dd 2011-11-07:] maar in plaats daarvan alleen de natural join \bowtie (die koppelt rijen precies wanneer gelijknamige attributen gelijke waarden hebben):

Onleesbare tekst wordt fout gerekend.

$\pi_s (\sigma_{r_1 \neq r_2} (O[s, b_1, r_1] \bowtie O[s, b_2, r_2]))$

iets omslachtiger: $\pi_s (\sigma_{r_1 \neq r_2} (O' \bowtie O''))$ met $O' = O[s, b_1, r_1]$ en $O'' = O[n, b_2, r_2]$.

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

Geef voor ieder schip dat met minstens acht andere schepen aan zeeslagen heeft deelgenomen, het aantal zeeslagen waaraan het heeft deelgenomen.

Onleesbare tekst wordt fout gerekend.

select $o.s$, count (distinct $o.b$)

from $O o, O o1$

where $o.b = o1.b$ and $o.s \neq o1.s$

group by $o.s$

having count (distinct $o1.s$) ≥ 8

(Zie Toelichting.)

5p. **Opgave 10.** (Goede beantwoording levert 5 bonuspunten boven op de 5 punten die voor deze opgave gegeven worden. Daardoor kan het totaal aantal behaalde punten op 105 uitkomen.) Beschouw de volgende vraag:

Geef ieder schip s waarvoor geldt dat in iedere zeeslag waarin het deel neemt, er een schip s' is dat ook aan die zeeslag deel neemt maar met een ander resultaat dan s .

Formuleer de vraag geheel in verzamelingsnotatie op een manier die zo direct mogelijk aansluit bij de gegeven formulering van de vraag:

Onleesbare tekst wordt fout gerekend.

We korten $b : B$ af tot b , et cetera. Alleen de laatste regel wordt gevraagd.
$\{s \mid (\forall b \mid \text{“}s \text{ in } b\text{”} \bullet (\exists s' \mid \text{“}s' \text{ in } b\text{”} \bullet \text{“resultaten van } s \text{ en } s' \text{ in } b \text{ verschillen”})) \bullet s.n\}$
= $\{s \mid (\forall b \mid (\exists o \bullet s.n = o.s \wedge o.b = b.n) \bullet (\exists s' \mid (\exists o' \bullet s'.n = o'.s \wedge o'.b = b.n) \bullet s \neq_{result,b} s')) \bullet s.n\}$
= [shunting, tweemaal]
$\{s \mid (\forall b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists s', o' \mid s'.n = o'.s \wedge o'.b = b.n \bullet s \neq_{result,b} s')) \bullet s.n\}$
= $\{s \mid (\forall b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists s', o' \mid s'.n = o'.s \wedge o'.b = b.n \bullet o.r \neq o'.r)) \bullet s.n\}$

Formuleer de vraag in verzamelingsnotatie op een manier die zo direct mogelijk aansluit bij de SQL query die u zo dadelijk gaat geven:

Onleesbare tekst wordt fout gerekend.

= [shunting]
$\{s \mid (\forall b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists o' \mid (\exists s' \bullet s'.n = o'.s) \wedge o'.b = b.n \bullet o.r \neq o'.r)) \bullet s.n\}$
= [in O is s foreign key verwijzend naar $Ship(n)$]
$\{s \mid (\forall b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists o' \mid o'.b = b.n \bullet o.r \neq o'.r)) \bullet s.n\}$
= $\{s \mid (\forall b, o \mid s.n = o.s \wedge o.b = b.n \bullet (\exists o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$
= [shunting]
$\{s \mid (\forall o \mid s.n = o.s \wedge (\exists b \bullet o.b = b.n) \bullet (\exists o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$
= [in O is b foreign key verwijzend naar $B(n)$]
$\{s \mid (\forall o \mid s.n = o.s \bullet (\exists o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$
= $\{s \mid \neg (\exists o \mid s.n = o.s \bullet \neg (\exists o' \mid o'.b = o.b \bullet o.r \neq o'.r)) \bullet s.n\}$

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

Onleesbare tekst wordt fout gerekend.

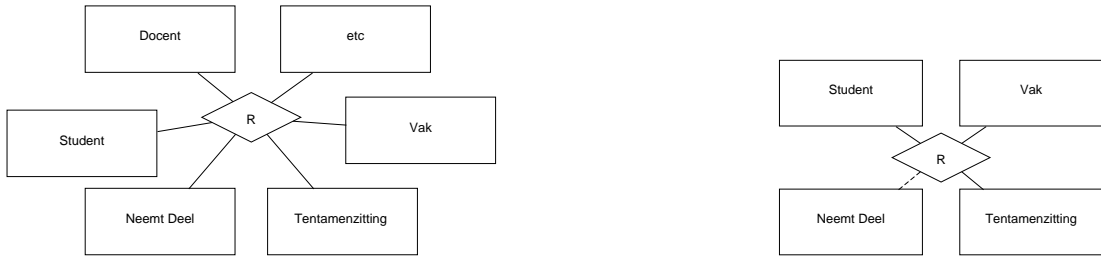
= select $s.n$ from S s where not exists (<div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> select * from O o where $s.n = o.s$ and not exists (<div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> select * from O o' where $o'.b = o.b$ and $o.r \neq o'.r$)) (Zie Toelichting.) </div> </div>

Toelichtingen

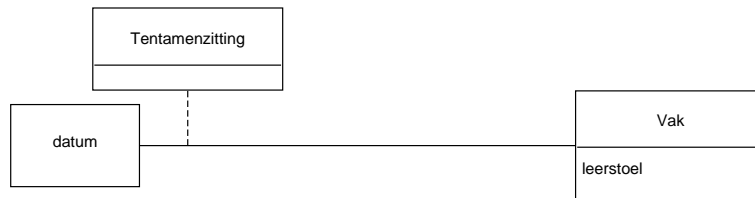
Toelichting bij antwoord 1.

Veel gemaakte fouten:

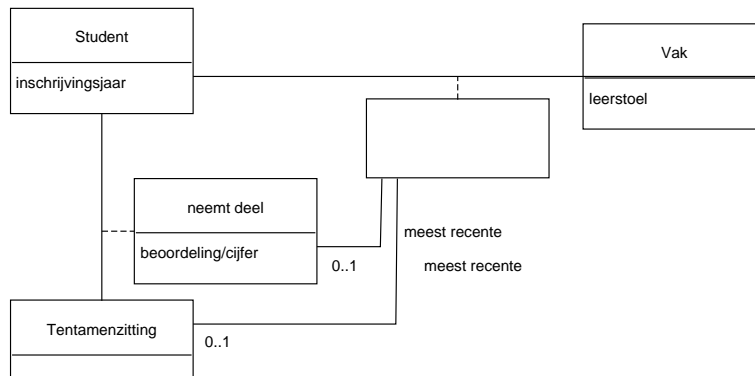
- Geen *Persoon*, geen *Vak*. *Hoofddocent* als specialisatie van *Docent*.
- Herhaling van attributen van *Persoon* in *Student* en *Docent*.
- Attribuit *hoofddocent* in *Vak*; in een ERD moet een “verwijzing” naar *Docent* gemodelleerd worden door een relatie met *Docent*. Net zo: een attribuit *surveillanten* in *Tentamenzitting*, een attribuit *vak* in *Tentamenzitting*.
- Te veel in één relatie; zeer extreem gedaan ziet dat er uit als links hieronder. Minder extreem, maar soms wél echt voorkomend, is zoiets als rechts hieronder.



De eigenschap “per vak hebben verschillende tentamenzittingen verschillende datums” kan als volgt in het ERD gemodelleerd worden:



Het begrip “eindcijfer” kan in het ERD gemodelleerd worden door, per student-vak combinatie, een beoordeling/tentamenzitting (van die student en dat vak) apart aan te wijzen, en daarvoor een extra constraint te formuleren, namelijk “de datum van de meest recente beoordeling/zitting, indien bestaand, is recenter dan de datum van iedere andere beoordeling/zitting van die student en dat vak”:



Toelichting bij antwoord 2.

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. Relatie B kan als view gedefinieerd worden: `select * from D`.

De generalisatie/specialisatie E wordt gerepresenteerd door de foreign key eigenschap in tabel F die verwijst naar tabel D . Met andere woorden, relatie E kan als view gedefinieerd worden: (informeel) “select $(a1, c1)$ as keyVanD, $(a1, c1)$ as keyVanF from F ”.

De F -instanties worden niet opgenomen in de tabel voor de D -instanties omdat dan NULLs nodig zouden zijn 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 kan relatie G kan als volgt als view gedefinieerd worden: `select aG, a1, c1 from F`. Wanneer G toch als aparte tabel opgenomen wordt in het database schema, dan luidt het schema ervan als volgt:

$G(aG, a1, c1, \text{primary key } (a1, c1),$
foreign key (aG) references $A(a1),$
foreign key $(a1, c1)$ references $F(a1, c1))$

Vanwege de multipliciteit zit aG niet in de key van G .

Bedenk dat ‘foreign key (x, y) references $Z(x, y)$ ’ een *sterkere* eigenschap is dan ‘foreign key (x) references $Z(x)$, foreign key (y) references $Z(y)$ ’, en bovendien vereist de laatste uitdrukking dat zowel x als ook y een key is in Z (terwijl die eerste uitdrukking alleen maar vereist dat (x, y) een sleutel is in Z)!

Toelichting bij antwoord 3.

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

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

Toelichting bij antwoord 4.

Een andere decompositie wordt verkregen door een andere schending van \mathbf{R} te kiezen ter eliminatie. Je kunt ook een schending uit $\mathcal{F}^+ \setminus \mathcal{F}$ nemen, bijvoorbeeld $AB \rightarrow CDE$.

Toelichting bij antwoord 8.

Wanneer “doorstaan” impliceert “niet zinken”, dan moet $o_1.r \neq o_2.r$ overal vervangen worden door $\text{sunk} \neq o_1.r \neq o_2.r \neq \text{sunk}$ (of door $o_1.r = \text{damaged} \wedge \text{ok} = o_2.r$).

Merk op dat $o_1.b \neq o_2.b$ volgt uit $o_1.r \neq o_2.r \wedge o_1.s = o_2.s$ (want in O is (s, b) een key).

Toelichting bij antwoord 9.

Hier is een voorbeeld waarbij schip s met schepen s_1, s_2 heeft deelgenomen aan zeeslagen b_1, b_2, b_3 :

“from O o , O $o1$ where $o.b = o1.b$ and $o.s \neq o1.s$ group by $o.s$ ”:

$o.s$	$o.b = o1.b$	$o.r$	$o1.s (\neq o.s)$	$o1.r$
\vdots				
s'				
\vdots				
s	b_1	...	s_1	...
s	b_2	...	s_1	...
s	b_3	...	s_1	...
s	b_3	...	s_2	...
\vdots				
s''				
\vdots				

Het middenstuk in deze tabel geeft “de groep voor s ” weer; het is onmiddellijk duidelijk dat de select-expressie ‘count (distinct $o.b$)’ niet vervangen mag worden door ‘count ($o.b$)’ of ‘count (*)’.

In plaats van de having-conditie ‘count (distinct $o1.s$) ≥ 8 ’ kun je ook nemen ‘count (distinct $o1.s$) > 8 ’ indien je dan tegelijk de where-conditie ‘ $o.s \neq o1.s$ ’ schrapt.

Hier is een andere oplossing, waarbij de having-conditie voor de groep van $o.s$ nu geformuleerd wordt als een where-conditie voor $o.s$ zelf (namelijk: “ $o.s$ heeft met minstens acht andere schepen deelgenomen aan zeeslagen”):

```
select  $o.s$ , count (*)
from  $O$   $o$ 
where (select count (*) from  $O$   $o'$ 
      where  $o'.s \neq o.s$  and  $o'.b$  in (select  $o''.b$  from  $O$   $o''$  where  $o''.s = o.s$ ) )  $\geq 8$ 
                                alle zeeslagen van  $o.s$ 
group by  $o.s$ 
```

Het is fout om het deel “in (alle zeeslagen van $o.s$)” te vervangen door “= $o.b$ ” want ten eerste hoeven die andere acht schepen niet allemaal in eenzelfde zeeslag met $o.s$ hebben deelgenomen, en ten tweede, op die plek heeft $o.b$ geen betekenis (want in de groep van $o.s$ komen meerdere zeeslagen voor; in de select-expressie en where-expressie en de having-expressie mag $o.b$ alleen maar geaggregeerd gebruikt worden).

Veel-gemaakte fouten: (i) een attribuut dat niet in de group by clause staat, gebruiken zonder het te aggregeren, (ii) groeperen op $o.b$ in plaats van $o.s$ (de groep van “alle schepen die aan $o.b$ hebben deelgenomen” is irrelevant voor query).

Toelichting bij antwoord 10.

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

$$\{s \mid (\forall o \mid s.n = o.s \bullet o.b \in \{o' \mid o.r \neq o'.r \bullet o'.b\}) \bullet s.n\}$$

$$\{s \mid \{o \mid s.n = o.s \bullet o.b\} \subseteq \{o' \mid o.r \neq o'.r \bullet o'.b\} \bullet s.n\}$$

$$\{s \mid \{o \mid s.n = o.s \bullet o.b\} \setminus \{o' \mid o.r \neq o'.r \bullet o'.b\} = \emptyset \bullet s.n\}$$

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