

Tentamen Databases 1, 2M400, 30 juni 2003.

De uitwerkingen van de opgaven dienen duidelijk (ook duidelijk leesbaar) en bondig geformuleerd te worden, en overzichtelijk opgeschreven. Alles wat onvoldoende leesbaar is wordt fout gerekend!

Dit is een “open boek” tentamen. Je mag gebruik maken van boeken en aantekeningen, op papier of op de computer. Alle niet door jezelf geschreven materiaal dient te dateren van vóór de aanvang van dit tentamen. Het gebruik van communicatieapparatuur is niet toegestaan. (Geen mobiele telefoons, alleen laptops zonder netwerk voor het lezen van elektronisch materiaal.)

Dit tentamen bestaat uit twee delen. Er is een “pre”-tentamendeel en een “hoofd”-tentamendeel. Wanneer je geen vrijstelling voor het pre-tentamendeel hebt, dien je dit deel het eerst te maken. Het bestaat uit drie vragen, die alle drie correct beantwoord dienen te worden, voordat het hoofd-tentamendeel wordt nagekeken. Wees dus zorgvuldig in het beantwoorden van het eerste deel. Daarna kun je verder gaan met de overige vragen. Kandidaten met een vrijstelling kunnen meteen aan het hoofddeel beginnen. Wie vrijstelling heeft en wie niet staat al lange tijd op de website vermeld. Mocht je twijfelen of je vrijstelling hebt, maak dan het pretentamen!

In de query opgaven wordt een (deeltje van een) universiteitsbibliotheek-database gebruikt met de volgende tabellen en attributen:

boek : { ISBN, titel, uitgever, jaar }
auteur : { ISBN, voorletters, naam }
exemplaar : { barcode, ISBN, faculteit, exdatum, aanwezig }
reservering : { naam, faculteit, ISBN, datum, geannuleerd }
uitlening : { naam, faculteit, barcode, van, tot }

Korte beschrijving:

Elk boek heeft een uniek ISBN nummer. Het heeft een titel, uitgever, jaar van uitgifte en een aantal auteurs waarvan we de voorletters (samen als 1 string) en de achternaam bijhouden. Verschillende edities van eenzelfde boek hebben een verschillend ISBN nummer en zijn dus voor ons verschillende boeken. De database bevat alle auteurs van in de database aanwezige boeken, en geen auteurs die niet (met ISBN nummer) overeenkomen met een boek..

De universiteit heeft van elk boek 1 of meer exemplaren. Elk exemplaar is op een bepaalde datum (exdatum) aangekocht, heeft een sticker met een unieke barcode en is toegewezen aan een faculteit. (Boeken kunnen worden uitgeleend aan mensen van andere faculteiten.) Een exemplaar kan afwezig zijn omdat het nog niet geleverd is, wordt hersteld, of omdat het is uitgeleend.

Personen worden geïdentificeerd door hun achternaam en faculteit. Ze kunnen een boek reserveren op een bepaalde datum. (Het type van “datum” is zodanig dat data die niet NULL zijn met elkaar vergeleken kunnen worden, bijvoorbeeld met $<$ en \leq .) Personen kunnen een reservering op elk willekeurig moment weer annuleren. De datum blijft dan die van de reservering, dus de datum waarop de reservering is geannuleerd wordt niet bijgehouden. (Het “geannuleerd” attribuut heeft altijd een waarde “ja” of “nee”.)

Ze kunnen een exemplaar van een boek lenen op een bepaalde “van” datum. (Ze zijn dan “lener”.) Een eventuele reservering van dat boek (op hun naam) wordt dan automatisch geannuleerd. De “tot” datum blijft NULL tot het boek is teruggebracht. De “tot” datum is minstens 1 dag later dan

de “van” datum. Wanneer een boek is teruggebracht wordt de “uitlening” in de database bewaard en wanneer een reservering wordt geannuleerd (bijvoorbeeld omdat het boek wordt geleend maar het kan ook om andere redenen zijn) wordt die reservering ook bewaard.

De bibliotheek bestaat al een hele tijd. Bijgevolg heeft elke faculteit wel wat (exemplaren van) boeken, heeft er uit elke faculteit wel eens iemand een boek gereserveerd en iemand een exemplaar geleend, en is er van elke faculteit wel eens een boek gereserveerd en wel eens een (exemplaar van) een boek geleend.

Let op: De attributen “naam” en “faculteit” worden in deze database in twee verschillende betekenissen gebruikt. Zorg dat je die niet per ongeluk door elkaar gebruikt of in een join mee neemt. Let ook op voor de verschillen tussen “boeken” en “exemplaren”. In een vraag schrijven we misschien wel eens dat een “boek” wordt geleend, maar we bedoelen dan dat een “exemplaar” (eender welk!) van dat “boek” wordt geleend. Twee uitleningen van eenzelfde boek kunnen uitleningen van verschillende exemplaren zijn.

Let ook op: In oude tentamens of het proeftentamen werd ook een bibliotheekdatabase gebruikt, maar er kunnen verschillen zijn in de definities van die databases.

Pre-tentamendeel

Deze vragen worden helemaal correct of helemaal fout gerekend.

Belangrijke hint (niet alleen voor dit deel trouwens): **Alle vragen kunnen redelijk eenvoudig en bondig beantwoord worden. Als je denkt dat het antwoord heel ingewikkeld en lang wordt dan is er veel kans dat je verkeerd bezig bent.**

1. Bestudeer grondig de beschrijving van de bibliotheek-database.
Beschrijf dan de volgende vraag in de relationele algebra:
“Geef de naam en voorletters van de auteurs van boeken waarvan minstens één faculteit een exemplaar in bezit heeft.”
2. Beschrijf de volgende vraag over de bibliotheek-database in SQL:
“Geef de naam en faculteit van personen die wel eens een boek hebben geleend dat ze nooit hebben gereserveerd.”
3. Beschouw een relationeel schema R met attributen $\{A, B, C\}$, elk met een groot domein zoals integers of strings, en de verzameling functionele afhankelijkheden $F = \{A \rightarrow B, B \rightarrow C\}$. Is dit schema in BCNF? Leg uit waarom dit wel of niet zo is. (Zonder uitleg wordt deze vraag fout gerekend.)

Einde pre-tentamendeel. Wanneer je deze drie vragen niet alle drie goed hebt wordt de rest van het tentamen niet nagekeken. Je krijgt dan een 1 als dank voor je deelname. Je kunt voor jezelf bepalen of je nog verder wilt gaan met het hoofdtentamen!

Hoofd-tentamendeel

Je hoeft de vragen niet te beantwoorden in de onderstaande volgorde. Begin met wat je het beste kan!

1. Gegeven is het volgende database schema (tabelnaam met attribuutnamen en onderlijning van de primaire sleutelattributen):

bemanningslid(bnr, bnaam)

kwalificatie(bnr, fnaam, ttype)

functie(fnaam, ttype, minaanal, maxaanal)

vliegtuigtype(ttype, aantalspers)

toewijzing(bnr, vnr, fnaam) (hier was per ongeluk ttype ook vermeld in de gebruikte opgave)

vlucht(vnr, ttype, van, naar)

Het schema beschrijft de toewijzing van bemanningsleden aan vluchten. Daar spelen de vier andere tabellen nog een rol bij. Een vlucht wordt gevlogen met een bepaald type vliegtuig (bijvoorbeeld Boeing 747) met een aantal stoelen. Voor de bediening van elk vliegtuigtype zijn een aantal specifieke functies vereist. Een functie als “eerste piloot” op een “Boeing 747” is een andere functie dan “eerste piloot” op een “Cessna Citation Mustang”. Niet alle functies komen voor bij elk vliegtuig. De Boeing 747 kent wel de functie “stewardess”, maar de Cessna niet. Bovendien is er van een bepaalde functie op een bepaald toesteltype een minimum aantal en een maximum aantal bemanningsleden met een bepaalde functie vereist. Een bemanningslid is voor een aantal specifieke functies (functie met vliegtuigtype) gekwalificeerd en mag alleen werken op een vlucht in een functie, waarvoor hij of zij gekwalificeerd is.

Het bovenstaande database schema is in BCNF. (Dat mag je aannemen.)

Veronderstel nu dat de beperkingen (de beschrijving) in de database als volgt worden veranderd:

- De beperking “Een bemanningslid is voor een aantal specifieke functies (functie met vliegtuigtype) gekwalificeerd” vervalt.
- Elk bemanningslid is gekwalificeerd voor precies 1 functie (vb. piloot, stewardess...).
- Elk bemanningslid dat gekwalificeerd is voor een functie op een bepaald toestel (b.v. piloot), is automatisch gekwalificeerd voor dezelfde functie op alle toestellen, waarop deze functie voorkomt.

Pas nu het database schema aan, zodat het weer in BCNF is. Beargumenteer de benodigde aanpassingen (of het achterwege laten daarvan), in functie van het opnieuw in BCNF brengen van het schema.

2. Druk de volgende vragen over de bibliotheek uit in één van de bestudeerde querytalen (relationele algebra, tupel calculus of SQL) naar keuze. Hint: sommige vragen zijn in sommige talen gemakkelijker dan in andere. Je mag het jezelf dus gemakkelijk maken door een verstandige keuze van query taal. Je mag bij elke vraag een andere querytaal gebruiken als je wilt, maar je mag ze ook allemaal in eenzelfde taal uitwerken.
 - a. Geef de naam en faculteit van leners en het ISBN-nummer van de boeken waarvan zij de eerste lener waren.
 - b. Geeft de naam en faculteit van leners die nog nooit het meest recent aangekochte exemplaar van een boek geleend hebben. (meest recent op tijdstip van uitlening)
 - c. Geef de naam en faculteit van leners die geen enkel boek meer dan twee keer geleend hebben.

3. Druk de volgende drie vragen over de bibliotheek uit in de relationele algebra, SQL of tupel calculus. Je mag elke taal slechts bij één vraag gebruiken, dus één vraag beantwoorden in SQL, één vraag in tupel calculus en één vraag in de relationele algebra.
- Geef een lijst met de titels van boeken, de naam van de auteur en het aantal keer geleend.
 - Geef de titels van boeken waarvan een exemplaar afwezig is, hoewel er nog nooit een exemplaar van het boek is uitgeleend.
 - Geef de titels van aanwezige boeken, die zijn gereserveerd, maar nog nooit uitgeleend.
4. Leg in normaal klinkend Nederlands uit wat er in de volgende queries wordt gevraagd.
- ```
SELECT b.titel
FROM boek b
WHERE
 (SELECT COUNT(*)
 FROM reservering r
 WHERE r.ISBN = b.ISBN AND r.geannuleerd = "nee")
>
(SELECT COUNT(*)
 FROM uitlening u, exemplaar e
 WHERE u.barcode = e.barcode and e.ISBN = b.ISBN)
```
  - $\prod_{\text{naam, faculteit}} ( \sigma_{\text{datum} > \text{tot}} ( \prod_{\text{barcode, ISBN}} (\text{exemplaar}) \bowtie \text{uitlening} \bowtie \text{reservering} ) )$
5. Veronderstel dat we een database willen construeren uit een verzameling data items  $\{A, B, C, D, E, F, G\}$  (die later attributen van tabellen zullen worden) en een verzameling  $F$  van functionele afhankelijkheden:
- $$F = \{BCD \rightarrow A, BC \rightarrow E, A \rightarrow F, F \rightarrow G, C \rightarrow D, A \rightarrow G\}$$
- Bereken een kanonieke afdekking (canonical cover)  $G$  voor deze verzameling afhankelijkheden.
  - Begin met een tabel  $T$  met als attributen  $ABCDEFG$  en decomposeer deze met behoud van afhankelijkheden en verliesloos (dependency-preserving en lossless) tot een database in 3NF.
  - Is deze decompositie ook in BCNF? Beargumenteer je antwoord. (Zonder uitleg wordt het antwoord fout gerekend.)
6. Neem een relatieschema  $R = ( \Omega, \Delta, \text{dom}, \text{SC} )$ . Gegeven zijn de volgende inferentieregels voor functionele afhankelijkheden, waarbij  $V, W, X, Y, Z$  deelverzamelingen van  $\Omega$  zijn:
- R1:** Strikte Reflectiviteitsregel: Er geldt altijd dat  $X \rightarrow X$ .
- R2:** Projectieregel: Als  $X \rightarrow YZ$ , dan  $X \rightarrow Y$ .
- R3:** Accumulatieregels: Als  $X \rightarrow YZ$  en  $Z \rightarrow VW$ , dan  $X \rightarrow VYZ$ .
- Bewijs, dat R1, R2 en R3 correct zijn.
- Bij deze vraag mag je aannemen, dat F1, F2 en F3 (reflexivity, augmentation, transitivity, pag 265 in Silberschatz) correct en volledig zijn (maar je hoeft dit niet te gebruiken als je niet wil).

Waardering: vraag 1: 3 punten, vraag 2, 3 en 4: 2 punten per query, vraag 5: 2 punten per antwoord (a, b en c), vraag 6: 3 punten. Een punt komt overeen met een verwachte inspanning van 6 minuten. Samen is dit 168 minuten of 28 punten. 2 punten krijg je kado voor aanwezigheid. (De 12 minuten die overblijven kun je indien nodig aan het pretentamen besteden.) Het totaal wordt herleid tot 10 door te delen door drie, met afronding naar boven (2/3 van een punt) of naar beneden (1/3 van een punt). Je moet dus 15 van de 28 punten halen. 14 is echt een onvoldoende want geeft 16 op 30 en wordt een 5.

## Uitwerking Pretentamen

Vraag 1:  $\Pi_{\text{naam, voorletters}}$  (auteur).

Opmerking: De join met exemplaar:  $\Pi_{\text{naam, voorletters}}$  (auteur  $\bowtie$  exemplaar ) is niet fout, maar negeert het gegeven dat de universiteit van elk boek 1 of meer exemplaren heeft, die toegewezen zijn aan faculteiten. Daardoor is deze uitwerking nodeloos ingewikkeld.

Opmerking: de bijzin “waarvan minstens ...” is overbodig, omdat in de natural join alleen aanwezige tupels worden meegenomen.

Vraag 2:

Select u.naam, u.faculteit

From uitlening as u, exemplaar as e

Where u.barcode = e.barcode and e.ISBN not in

( Select r.ISBN

From reservering as r

Where u.naam = r.naam and u.faculteit = r.faculteit )

Opmerking: Er werd niet gevraagd dat de reservering eerder was dan de uitleendatum, dus het vergelijken van de data is niet nodig. Had er gestaan: “... nooit eerder hebben gereserveerd” dan zou de uitwerking worden:

Select u.naam, u.faculteit

From uitlening as u, exemplaar as e

Where u.barcode = e.barcode and e.ISBN not in

( Select r.ISBN

From reservering as r

Where u.naam = r.naam and u.faculteit = r.faculteit  
and u.van < r.datum)

Hoewel dit voor de gegeven vraag dus fout is hebben we dit niet beschouwd als een fout die het nakijken van de rest van het tentamen verhinderde.

Vraag 3:

Dit schema is niet in BCNF. B is geen superkey, maar komt wel voor aan de linkerkant van de afhankelijkheid  $B \rightarrow C$  en schendt dus de eis voor BCNF.

(Mocht het domein van attribuut A uit niet meer dan 1 waarde bestaan dan gold ook  $B \rightarrow A$  en was het schema wel in BCNF, vandaar de opmerking dat de domeinen groot genoeg moeten zijn.)

## Uitwerking Hoofdtentamen

Vraag 1:<sup>1</sup>

In de beschrijving van de database komen geen functionele of multivalued dependencies naar voor. Er zijn dus alleen de via de onderlijning aangegeven sleutelafhankelijkheden. Merk op dat de beperking in het eerste streepje “Een bemanningslid is voor een aantal specifieke functies (functie met vliegtuigtype) gekwalificeerd” niet echt een beperking is maar aangeeft dat er geen fd  $\{bnr\} \rightarrow \{fnaam\}$  of  $\{bnr\} \rightarrow \{ttype\}$  of zelfs  $\{bnr\} \rightarrow \{fnaam, ttype\}$  geldt. Het vervallen van deze “beperking” heeft dus geen invloed op het schema omdat ze geen echte beperking is.

De eis “Elk bemanningslid is gekwalificeerd voor precies één functie” kan geformuleerd worden als een FD:  $\{bnr\} \rightarrow \{fnaam\}$  in de tabellen “kwalificatie” en “toewijzing”. Deze tabellen zijn dus niet meer in BCNF. We zullen deze FD gebruiken voor decompositie.

De tweede eis is onder andere een MVD in de tabel “kwalificatie”:  $\{fnaam\} \twoheadrightarrow \{bnr\}$  of  $\{fnaam\} \twoheadrightarrow \{ttype\}$ . Door deze MVD is de tabel “kwalificatie” niet meer in 4NF. Voor BCNF is een MVD geen probleem. We hoeven met deze MVD dus niets te doen, maar het mag wel.

Er zijn nu twee mogelijkheden: we doen eerst iets met de FD of we doen eerst iets met de MVD. Het eerste ligt voor de hand omdat we met de MVD niets hoeven te doen voor BCNF.

Met de FD  $\{bnr\} \rightarrow \{fnaam\}$  splitsen we de tabel “kwalificatie” op in “kwalificatie\_1” met attributen (bnr, fnaam) en “kwalificatie\_2” met (bnr, ttype). Met deze fd splitsen we ook de tabel “toewijzing” op in “toewijzing\_1” met attributen (bnr, fnaam) en “toewijzing\_2” met attributen (bnr, vnr). We krijgen dan:

bemanningslid(bnr, bnaam)  
kwalificatie\_1(bnr, fnaam)  
kwalificatie\_2(bnr, ttype)  
functie(fnaam, ttype, minaantal, maxaantal)  
vliegtuigtype(ttype, aantalpers)  
toewijzing\_1(bnr, fnaam)  
toewijzing\_2(bnr, vnr)  
vlucht(vnr, ttype, van, naar)

Als we dit gedaan hebben zijn we voor wat betreft het beantwoorden van de vraag klaar, want het schema is nu terug in BCNF.

**Hier eindigt dus het gevraagde antwoord op vraag 1.** Wat volgt mag er nog bij, maar dat is dus niet gevraagd. We hebben de vraag wel goedgekeurd als er slechts weinig van de nu volgende argumentatie is gegeven om te komen tot het eindresultaat (zie verderop). Wel hebben we punten afgetrokken als in de karige gegeven argumentatie grove onwaarheden stonden.

Hier is een correcte argumentatie voor het vervolg van de mogelijke aanpassingen aan het schema:

Het gegeven schema heeft nog een beperking, die niet verandert: “Een bemanningslid mag alleen werken op een vlucht in een functie waarvoor hij of zij gekwalificeerd is.”

---

<sup>1</sup> Er stond een klein foutje in de opgave: In de tabel “toewijzing” mocht geen attribuut “ttype” voorkomen want anders was het gegeven schema niet in BCNF. We gaan er hier even van uit dat het attribuut er niet stond. Dit heeft echter niet tot problemen geleid in uitwerkingen door studenten. Het heeft alleen invloed op het extraatje.

Deze beperking is uit te drukken als:

$$\prod_{\{bnr, fnaam, ttype\}}(\text{toewijzing} \triangleright \triangleleft \text{vlucht}) \subseteq \text{kwalificatie} \quad (\text{opmerking 0})$$

Een interessant gevolg hiervan is:

$$\prod_{\{bnr, fnaam\}}(\text{toewijzing}) \subseteq \prod_{\{bnr, fnaam\}}(\text{kwalificatie}) \quad (\text{opmerking 1})$$

De opmerking (opmerking 1), kunnen we na de decompositie vertalen in:

$$\text{toewijzing}_1 \subseteq \text{kwalificatie}_1 \quad (\text{opmerking 1'})$$

We kunnen daarom overwegen of we misschien de tabel “toewijzing\_1” weg mogen laten. Immers, het enige verschil (vanwege de nieuwe FD  $\{bnr\} \rightarrow \{fnaam\}$ ) is dat in toewijzing\_1 de bemanningsleden staan die helemaal nergens aan toegewezen zijn.

Een handig truukje (verzamelingenleer) vertelt ons dat opmerking 1' impliceert dat:

$$\text{toewijzing}_1 = \text{kwalificatie}_1 - (\text{kwalificatie}_1 - \text{toewijzing}_1)$$

Laten we even joinen met toewijzing\_2:

$$\begin{aligned} \text{toewijzing}_1 \triangleright \triangleleft \text{toewijzing}_2 = \\ (\text{kwalificatie}_1 - (\text{kwalificatie}_1 - \text{toewijzing}_1)) \triangleright \triangleleft \text{toewijzing}_2 \end{aligned}$$

Nu even de join over de twee stukken verdelen (we gebruiken hier de distributiviteit van join en verschil, die we natuurlijk eenvoudig met verzamelingenleer kunnen bewijzen):

$$\begin{aligned} \text{toewijzing}_1 \triangleright \triangleleft \text{toewijzing}_2 = (\text{kwalificatie}_1 \triangleright \triangleleft \text{toewijzing}_2) \\ - ((\text{kwalificatie}_1 - \text{toewijzing}_1) \triangleright \triangleleft \text{toewijzing}_2) \end{aligned}$$

Het tweede deel is leeg, want in “kwalificatie\_1 - toewijzing\_1” komen alleen niet-toegewezen bemanningsnummers voor en in toewijzing\_2 alleen de wel toegewezen bemanningsnummers. Dus hebben we:

$$\text{toewijzing}_1 \triangleright \triangleleft \text{toewijzing}_2 = \text{kwalificatie}_1 \triangleright \triangleleft \text{toewijzing}_2$$

Het linkerdeel kennen we nog uit de decompositie:

$$\text{toewijzing} = (\text{kwalificatie}_1 \triangleright \triangleleft \text{toewijzing}_2)$$

Laten we even projecteren op  $\{bnr, fnaam\}$ :

$$\prod_{\{bnr, fnaam\}}(\text{toewijzing}) = \prod_{\{bnr, fnaam\}}(\text{kwalificatie}_1 \triangleright \triangleleft \text{toewijzing}_2)$$

Maar dat linkerdeel kennen we natuurlijk ook nog:

$$\text{toewijzing}_1 = \prod_{\{bnr, fnaam\}}(\text{kwalificatie}_1 \triangleright \triangleleft \text{toewijzing}_2)$$

We kunnen dus niet alleen overwegen om “toewijzing\_1” weg te laten. We mogen dat ook doen want we kunnen toewijzing\_1 uit de overige tabellen berekenen.

We willen ook nog van tabel kwalificatie\_2 af die zegt voor welke toestellen een bemanningslid een kwalificatie heeft.

Als we wat “boerenverstand” toepassen op de beschrijving dan weten we ook dat iemand alleen kan gekwalificeerd zijn voor een functie op een bepaald toestel als die functie op dat bepaald toestel bestaat. We nemen voor de eenvoud ook even aan dat in de “functie” tabel geen flauwekul staat zoals een maximum van 0. Deze elementaire veronderstellingen betekenen dat we mogen aannemen dat:

$$\prod_{\{fnaam, ttype\}}(\text{kwalificatie}) \subseteq \prod_{\{fnaam, ttype\}}(\text{functie})$$

Elementaire verzamelingenleer leert ons dan dat:

$$\prod_{\{fnaam, ttype\}}(kwalificatie) = \prod_{\{fnaam, ttype\}}(\text{functie}) - (\prod_{\{fnaam, ttype\}}(\text{functie}) - \prod_{\{fnaam, ttype\}}(kwalificatie))$$

De MVD (even als join dependency genoteerd) zegt:

$$kwalificatie = \prod_{\{fnaam, ttype\}}(kwalificatie) \bowtie \prod_{\{fnaam, bnr\}}(kwalificatie)$$

Met het verzamelingenleer-truukje wordt dat herschreven als:

$$kwalificatie = (\prod_{\{fnaam, ttype\}}(\text{functie}) - (\prod_{\{fnaam, ttype\}}(\text{functie}) - \prod_{\{fnaam, ttype\}}(kwalificatie))) \bowtie \prod_{\{fnaam, bnr\}}(kwalificatie)$$

Nu weer even de join over de twee stukken verdelen (we gebruiken hier opnieuw de distributiviteit van join en verschil)

$$kwalificatie = \prod_{\{fnaam, ttype\}}(\text{functie}) \bowtie \prod_{\{fnaam, bnr\}}(kwalificatie) - ((\prod_{\{fnaam, ttype\}}(\text{functie}) - \prod_{\{fnaam, ttype\}}(kwalificatie)) \bowtie \prod_{\{fnaam, ttype\}}(kwalificatie))$$

In het tweede stuk zien we in het join-attribuut fnaam weer dat we de functies nemen die niet in kwalificatie voorkomen en joinen met de functies die wel in kwalificatie voorkomen en die join is dus altijd leeg. Bijgevolg wordt de formule eenvoudiger:

$$kwalificatie = \prod_{\{fnaam, ttype\}}(\text{functie}) \bowtie \prod_{\{fnaam, bnr\}}(kwalificatie)$$

Even projecteren op {fnaam,ttype} en we krijgen:

$$\prod_{\{fnaam, ttype\}}(kwalificatie) = \prod_{\{fnaam, bnr\}}(\prod_{\{fnaam, ttype\}}(\text{functie}) \bowtie \prod_{\{fnaam, bnr\}}(kwalificatie))$$

of

$$kwalificatie\_2 = \prod_{\{fnaam, bnr\}}(\prod_{\{fnaam, ttype\}}(\text{functie}) \bowtie kwalificatie\_1)$$

De tabel kwalificatie\_2 is dus inderdaad overbodig (mits we een vanzelfsprekende maar niet expliciet vermelde aanname maken).

Een laatste optimalisering die we kunnen toepassen is de tabellen bemanningslid en kwalificatie\_1 te joinen. Vanwege {bnr} → {bnaam} en {bnr} → {fnaam} weten we dat dit een lossless join is (1 FD was al genoeg), en omdat {bnr} in de nieuwe tabel een sleutel is mag dit ook nog van de BCNF. Het eindresultaat wordt dan (na een kleine hernoemingsoperatie):

bemanningslid(bnr, bnaam, fnaam)  
 functie(fnaam, ttype, minaantal, maxaantal)  
 vliegtuigtype(ttype, aantalpers)  
 toewijzing(bnr, vnr)  
 vlucht(vnr, ttype, van, naar)

Aangezien dit schema op hernoeming en weglating van tabellen na hetzelfde is als de eerdere BCNF uitwerking is ook dit schema nog steeds in BCNF.



Vraag 2:

2a. De eerste vraag is handig te doen in zowel tuple-calculus als SQL: “eerste lener zijn” betekent dat er geen uitlening is van een exemplaar van het betreffende boek, die eerder start.

$$\{ t \mid \exists u1 \in \text{uitlening} ( t[\text{naam}] = u1[\text{naam}] \wedge t[\text{faculteit}] = u1[\text{faculteit}] \wedge \exists e1 \in \text{exemplaar} ( t[\text{ISBN}] = e1[\text{ISBN}] \wedge e1[\text{barcode}] = u1[\text{barcode}] \wedge \neg \exists e2 \in \text{exemplaar} ( e2[\text{ISBN}] = e1[\text{ISBN}] \wedge \exists u2 \in \text{uitlening} ( u2[\text{barcode}] = e2[\text{barcode}] \wedge u2[\text{van}] < u1[\text{van}] ) ) ) ) ) \}$$

Dit is rechttoe-rechtaan te vertalen naar SQL (zie recept colstructie).

Opmerking: Veel studenten hebben alleen gekeken of de lener de eerste lener van het exemplaar was in plaats van eerste lener van het boek (dus van eender welk exemplaar van het boek).

2b. Het meest recente exemplaar betekent hier het exemplaar met de meest recente aanschafdatum op het moment van lenen, d.w.z. de grootste waarde van het exdatum-attriboot kleiner dan de uitleendatum.

$$\{ t \mid \exists u1 \in \text{uitlening} ( t[\text{naam}] = u1[\text{naam}] \wedge t[\text{faculteit}] = u1[\text{faculteit}] \wedge \forall u2 \in \text{uitlening} ( u2[\text{naam}] = u1[\text{naam}] \wedge u1[\text{faculteit}] = u2[\text{faculteit}] \wedge \forall e1 \in \text{exemplaar} ( e1[\text{barcode}] = u2[\text{barcode}] \Rightarrow \exists e2 \in \text{exemplaar} ( e2[\text{ISBN}] = e1[\text{ISBN}] \wedge e2[\text{exdatum}] > e1[\text{exdatum}] \wedge u2[\text{van}] \geq e2[\text{exdatum}] ) ) ) ) ) \}$$

Opmerking: Veel studenten kijken maar naar 1 boek terwijl er gevraagd wordt naar leners die nog van geen enkel boek het meest recent aangekochte exemplaar hebben geleend. Sommigen vergeten ook om de aanschafdatum te relateren aan het tijdstip van uitlening. Dat is natuurlijk gewoon stom. Het stond expliciet met zoveel woorden in de opgave.

2c. In deze opgave moet tot meer dan 2 geteld worden. We gebruiken dus SQL:

```
Select naam, faculteit
From uitlening
Except
Select u.naam, u.faculteit
From uitlening as u, exemplaar as e
Where u.barcode = e.barcode and
2 < (Select count(*)
 From uitlening as u1, exemplaar as e1
 Where u1.barcode = e1.barcode and u1.naam= u.naam and u1.faculteit = u.faculteit
 and e1.ISBN = e.ISBN)
```

Merk op dat de “Except” operator automatisch dubbels elimineert, zodat we geen “Distinct” hoeven te schrijven, maar het is niet fout om dat toch te doen. (zie pag .146).

Opmerking: Het is uiteraard ook mogelijk om een ander formalisme te gebruiken waarin je dan niet kunt tellen, maar ja je moet maar drie uitleningen vinden om een lener uit te sluiten. Dat betekent wel dat je dan in de query (minstens) 4 keer de tabel “uitlening” nodig hebt. Er zijn ook enkele studenten die moeite hebben met het nederlands. Meer dan twee keer geleend betekent dat twee keer dus niet genoeg is.

Opmerking: Sommige studenten zoeken alleen naar leners die een of ander boek nog geen drie keer geleend hebben. Fout! Sommigen tellen de uitleningen van alle boeken samen, sommigen van alle leners, etc. Allemaal fout.

Vraag 3:

3a. Bij deze vraag moeten we tellen. Daarom gebruiken we SQL met een aggregatie. We hebben dit jaar de aggregatie alleen in SQL bestudeerd. We keuren het wel of niet gebruiken van de voorletters allebei goed. Omdat de informatie in de lijst anders is dan die waarop geteld wordt, kan het beste met een hulptabel gewerkt worden:

With tel(ISBN, aantal) as

```
Select b.ISBN, count(u.barcode)
From boek as b, exemplaar as e, uitlening as u
Where b.ISBN = e.ISBN and e.barcode = u.barcode
Group by b.ISBN
```

```
(Select b.titel, a.naam, a.voorletters, t.aantal
From boek as b, auteur as a, tel as t
where b.ISBN = a.ISBN and b.ISBN = t.ISBN
UNION
Select b.titel, a.naam, a.voorletters, 0
From boek as b, auteur as a
where b.ISBN = a.ISBN and
not b.ISBN in (Select ISBN From tel))
```

Opmerking: Omdat op uitlening gejoined wordt komen boeken, die nooit uitgeleend zijn niet aan bod en moeten via een UNION toegevoegd worden. Merk op dat het tellen van de u.barcode werkt omdat dubbels niet worden verwijderd.

Opmerking: Sommige studenten hebben ineens geprobeerd een Group by te doen, met iets als

```
Select b.titel, a.naam, count(u.barcode)
From auteur as a, boek as b, exemplaar as e, uitlening as u
Where a.ISBN = b.ISBN and b.ISBN = e.ISBN and e.barcode = u.barcode
Group by b.ISBN, b.titel, a.naam
```

Hierin komen gemakkelijk verschillende fouten. Om te beginnen **moet** je alle attributen die in de Select voorkomen (buiten de aggregatie-functie) ook in de Group by zetten. Dus Group by b.ISBN is fout. Group by b.titel alleen is natuurlijk ook fout want een boek is geïdentificeerd door ISBN, niet titel. De a.naam niet mee in de Group by zetten geeft bovendien het probleem dat de count alles zoveel keer telt als er auteurs zijn van het boek.

3b. Deze vraag is geschikt voor een algebra aanpak met een verschil operator. We bepalen de titels van boeken, waar geen uitlening voor bestaat:

$$\Pi_{\text{boek.titel}}(\text{boek} \bowtie (\Pi_{\text{ISBN}}(\sigma_{\text{aanwezig}=\text{false}}(\text{exemplaar})) - \Pi_{\text{exemplaar.ISBN}}(\sigma_{\text{exemplaar.barcode} = \text{uitlening.barcode}}(\text{exemplaar} \times \text{uitlening})))$$

Opmerking: Veel studenten gingen hier in de fout door titels van titels af te trekken. Maar boeken worden geïdentificeerd door het ISBN nummer, niet de titel. De uitleg over verschillende edities van een boek, met dezelfde titel en verschillend ISBN nummer, staat niet voor niets in de beschrijving van de database!

3c. Deze opgave dient dus beantwoord te worden in tupel-calculus, want die hebben we over.

$$\{ t \mid \exists b \in \text{boek} (b[\text{titel}] = t[\text{titel}] \wedge \exists r \in \text{reservering} (r[\text{ISBN}] = b[\text{ISBN}] \wedge \exists e \in \text{exemplaar} (e[\text{ISBN}] = b[\text{ISBN}] \wedge e[\text{aanwezig}] = \text{"true"} \wedge \neg \exists u \in \text{uitlening} ( \exists ee \in \text{exemplaar} ( ee[\text{ISBN}] = b[\text{ISBN}] \wedge u[\text{barcode}] = ee[\text{barcode}] ) ) ) ) ) \}$$

Opmerking: Let op dat er tweemaal exemplaar nodig is: eenmaal voor het aanwezige exemplaar en een tweede maal om de niet bestaande uitleningen te koppelen aan alle mogelijke exemplaren van dat boek. Het volgende is dus fout omdat het titels geeft van boeken waarvan er een exemplaar a

$$\{ t \mid \exists b \in \text{boek} (b[\text{titel}] = t[\text{titel}] \wedge \exists r \in \text{reservering} (r[\text{ISBN}] = b[\text{ISBN}] \wedge \exists e \in \text{exemplaar} (e[\text{ISBN}] = b[\text{ISBN}] \wedge e[\text{aanwezig}] = \text{"true"} \wedge \neg \exists u \in \text{uitlening} (u[\text{barcode}] = ee[\text{barcode}]) ) ) ) \}$$

Vraag 4:

4a: Geef de titels van boeken waarvoor het aantal openstaande reserveringen groter is dan het totaal aantal uitleningen (in het verleden en heden).

4b: Geef de naam en faculteit van personen, die een boek geleend en teruggebracht hebben en het daarna nog eens gereserveerd.

Vraag 5:

5a. We passen het algoritme uit figuur 7.8 toe op

$$F = \{BCD \rightarrow A, BC \rightarrow E, A \rightarrow F, F \rightarrow G, C \rightarrow D, A \rightarrow G\}.$$

- unieregel:  $\{BCD \rightarrow A, BC \rightarrow E, A \rightarrow FG, F \rightarrow G, C \rightarrow D\}$
- D is overbodig in  $BCD \rightarrow A$ . Te bewijzen:  $F \vdash (F - \{BCD \rightarrow A\}) \cup \{BC \rightarrow A\}$ . Bewijs: er geldt:  $C \rightarrow D$ , dus ook (aug)  $C \rightarrow CD$ , en (aug)  $BC \rightarrow BCD$ . Dan geldt met  $BCD \rightarrow A$  via transitiviteit:  $BC \rightarrow A$ .
- G is overbodig in  $A \rightarrow FG$ . Te bewijzen:  $(F - \{A \rightarrow FG\}) \cup \{A \rightarrow F\} \vdash F$ . Bewijs: er geldt  $F \rightarrow G$ , dus (aug)  $F \rightarrow FG$ , en dan met  $A \rightarrow F$  en transitiviteit:  $A \rightarrow FG$ .

We hebben nu:  $F_c = \{BC \rightarrow A, BC \rightarrow E, A \rightarrow F, F \rightarrow G, C \rightarrow D\}$  en via de unieregel:

$$F_c = \{BC \rightarrow AE, A \rightarrow F, F \rightarrow G, C \rightarrow D\}.$$

De afhankelijkheden in  $F_c$  zijn niet verder te vereenvoudigen.

Opmerking: In de vraag wordt alleen het canonical cover gevraagd en niet de methode om het te berekenen. Wie dus alleen het resultaat geeft krijgt de volle twee punten. Als het resultaat echter fout is en er is geen methode beschreven dan wordt het natuurlijk een nul, terwijl als de afleiding van het cover is gegeven en daar zit een foutje in dan kan soms nog een puntje behaald worden.

5b. Met behulp van het algoritme uit figuur 7.14 (3NF synthese algoritme) krijgen we

$$R1 = ABCE, R2 = AF \text{ (want niet besloten in } R1), R3 = FG \text{ en } R4 = CD.$$

We moeten nu eerst nog vaststellen of er een kandidaat-sleutel is om te zien of daarvoor nog een relatie voorzien moet worden. Er geldt:  $BC^+ = ABCDEFG$ ,  $A^+ = AFG$ ,  $F^+ = FG$  en  $C^+ = CD$ . Er is dus maar één kandidaatsleutel: BC en deze is bevat in R1. De vier relatieschema's zijn dus voldoende voor een dependency-preserving, lossless-join decompositie van T.

Opmerking: We beoordelen in dit onderdeel alleen maar of een correcte 3NF decompositie wordt gegeven. Daardoor kan dit onderdeel ook beoordeeld worden als 5a fout was.

5c. De 3NF decompositie is ook in BCNF, omdat er in elk van de tabellen alleen sleutelafhankelijkheden gelden.

Opmerking: We beoordelen in dit onderdeel alleen of de decompositie uit onderdeel 5b in BCNF is. Wanneer 5b fout is beoordelen we dus alleen of van het foute antwoord uit 5b correct wordt geanalyseerd of het in BCNF is.

Vraag 6.

Er zijn twee manieren om deze vraag te beantwoorden: je kunt gebruik maken van de definitie van functionele afhankelijkheden, en je kunt gebruik maken van de correctheid van F1, F2, F3.

Eerste methode:

R1: Neem een willekeurige relatieinstantie  $r$  van  $R$ , en een  $X \subseteq \Omega$ . Voor elk paar tupels  $t_1$  en  $t_2 \in r$  geldt dat wanneer  $t_1[X] = t_2[X]$  dan is ook  $t_1[X] = t_2[X]$  (triviaal). Volgend de definitie van functionele afhankelijkheid betekent dit dat  $X \rightarrow X$  geldt in  $r$  (en dus in  $R$  want  $r$  was willekeurig).

R2: Neem een willekeurige relatieinstantie  $r$  van  $R$  die voldoet aan:  $X \rightarrow YZ$ . Dit betekent dat er voor elk paar tupels  $t_1$  en  $t_2 \in r$  geldt dat wanneer  $t_1[X] = t_2[X]$ , er geldt dat  $t_1[YZ] = t_2[YZ]$  (definitie van functionele afhankelijkheid). Maar dan geldt ook dat  $t_1[Y] = t_2[Y]$ . Er geldt dus ook  $X \rightarrow Y$  op  $r$ .

R3: Veronderstel nu dat  $X \rightarrow YZ$  en  $Z \rightarrow VW$  gelden op  $r$ . Wanneer  $t_1[X] = t_2[X]$ , dan geldt ook (def. func. afh.)  $t_1[YZ] = t_2[YZ]$ , waaruit volgt dat  $t_1[Z] = t_2[Z]$ . Vanwege dit laatste geldt (def func. afh.) dat  $t_1[VW] = t_2[VW]$  en dus ook  $t_1[V] = t_2[V]$ . Wanneer  $t_1[X] = t_2[X]$ , dan geldt dus dat  $t_1[VYZ] = t_2[VYZ]$ .

Tweede methode:

De uitwerking gebruikt alleen F1, F2 en F3. Het kan eenvoudiger als je ook aanneemt dat je F4, F5 en F6 mag gebruiken maar dat doen we hier even niet.

R1: omdat  $X \subseteq X$  geldt  $X \rightarrow X$  volgens F1.

R2: stel  $X \rightarrow YZ$  geldt; uit F1 volgt dat  $YZ \rightarrow Y$  geldt (want  $Y \subseteq YZ$ ); we passen dan F3 toe op  $X \rightarrow YZ$  en  $YZ \rightarrow Y$  en krijgen  $X \rightarrow Y$ .

R3: uit  $X \rightarrow YZ$  volgen  $X \rightarrow Y$  en  $X \rightarrow Z$  met de inmiddels bewezen R2; uit  $Z \rightarrow VW$  volgt  $Z \rightarrow V$  volgens R2 en daaruit volgt  $Z \rightarrow VZ$  volgens F2 (aug. met  $Z$ ); uit  $X \rightarrow Z$  en  $Z \rightarrow VZ$  volgt  $X \rightarrow VZ$  volgens F3; augmentation (F2) met  $Y$  geeft  $XY \rightarrow VYZ$ ;  $X \rightarrow Y$  geeft  $X \rightarrow XY$  met F2 (aug. met  $X$ ) en uit  $X \rightarrow XY$  en  $XY \rightarrow VYZ$  volgt  $X \rightarrow VYZ$  met F3.