

Naam en voorletters:

Ident.nr.:

TECHNISCHE UNIVERSITEIT EINDHOVEN

Faculteit Wiskunde en Informatica

Tentamen Databases 1, 2M400, 15 augustus 2003.

Alle uitwerkingen van de opgaven moeten worden ingevuld in de daarvoor bestemde vrije ruimte bij de tentamenvragen. **Alleen wat op deze formulieren is ingevuld wordt nagekeken. Andere papieren worden meteen vernietigd. Ze worden in geen geval nagekeken.** De ruimte die is voorzien is in elk geval voldoende voor het correcte antwoord, in een normaal leesbare letter-grootte. Alles wat onvoldoende leesbaar is wordt fout gerekend! Elk antwoord moet staan in de ruimte bij de vraag, tenzij verwezen is naar de laatste (witte) pagina.

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, wel 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! Als je geen vrijstelling hebt en je slaat het pre-tentamen over wordt je hoofd-tentamen niet nagekeken en scoor je een 1.

Hier is een recept voor het met succes afronden van dit tentamen:

1. Bestudeer eerst alle vragen, en stel een lijstje op met de volgorde waarin je de vragen wil beantwoorden.
2. Begin met het beantwoorden van de vragen waarvan je zelf denkt dat je ze het beste kan. Werk zorgvuldig. Het is niet erg om wegens tijdgebrek aan het eind een vraag open te moeten laten.
3. Denk bij elke vraag eerst goed na over het “type” van de vraag: welk deel van het collegemateriaal is van belang bij het beantwoorden van de vraag? Bij een query: wat voor soort query is het? Bij afleidingsregels: welke “richting” moet ik bewijzen? Etc.
4. Werk elke vraag eerst op een kladblaadje uit en schrijf het antwoord dan over op het invulblad. **Wacht hiermee niet tot het einde want losse bijgevoegde papieren worden niet nagekeken! Sterker: ze worden door de surveillant geweigerd, en als hij/zij ze toch zou aannemen worden ze door de corrector meteen vernietigd!**
5. Als je nog 15 a 20 minuten hebt, begin dan niet meer aan een nieuwe vraag (die je niet ligt en daarom hebt uitgesteld tot het einde) maar kijk de vragen na die je al hebt beantwoord. Bij queries: vertaal ze zonder naar de opgave te kijken eens terug, en kijk dan of wat uit die vertaling komt ook de opgave was. Bij de afhankelijkheden of decompositie, ga na of je antwoord wel het antwoord is op de vraag die is gesteld: heb je de juiste “richting” bewezen? Heb je geen veronderstellingen gemaakt die de algemeenheid van je uitwerking schaden, voldoet bij een decompositie je resultaat wel aan de gevraagde normaalvorm, etc.
6. Verifieer dat je op elke pagina je naam en identiteitsnummer hebt geschreven.

Naam en voorletters:

Ident.nr.:

De bibliotheek-database

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, exjaar, 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 in een bepaald jaar (exjaar) 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 zeker op de volgende "valstrikken":

- 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. Zorg dat je altijd goed in de gaten hebt of er een "boek" als generiek object wordt bedoeld of een "exemplaar" als concreet item dat kan worden geleend.
- Wie alleen nog maar een boek gereserveerd heeft maar nog nooit iets heeft geleend is nog geen "lener". Een "lener" is iemand die al eens een boek heeft geleend.

Naam en voorletters:

Ident.nr.:

Pre-tentamendeel

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. Het antwoord MOET in de voorziene ruimte passen. Aparte vellen worden niet nagekeken!**

1. Beschrijf de volgende vraag over de bibliotheek-database in de relationele algebra:
“Geef de uitgevers van wie er ooit (1 of meer) boeken gereserveerd zijn.”.

Π_{uitgever} (boek \bowtie reservering)

2. Beschrijf de volgende vraag over de bibliotheek-database in SQL:
“Geef de (van) data waarop (1 of meer) uitleningen zijn geregistreerd van niet-bestaande exemplaren.” (dus met een barcode die niet bij een exemplaar hoort)

```
SELECT u.van
FROM uitlening as u
WHERE u.barcode NOT IN (
    SELECT e.barcode
    FROM exemplaar as e )
```

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 $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$. Is dit schema in BCNF? Leg uit waarom dit wel of niet zo is. (Zonder uitleg wordt deze vraag fout gerekend.)

Met de afleidingsregels kunnen we $A \rightarrow ABC$, $B \rightarrow ABC$ en $C \rightarrow ABC$ afleiden. Bijgevolg zijn alle drie de attributen kandidaat sleutels (en zijn alle andere niet-lege verzamelingen attributen supersleutels).

Voor elke functionele afhankelijkheid uit F^* geldt dat:

- Links staat A, B, C, AB, BC, AC of ABC en dit zijn allemaal supersleutels;
of
- Links staat \emptyset , maar dan is de afhankelijkheid $\emptyset \rightarrow \emptyset$ (want $\emptyset \rightarrow$ iets anders geldt niet in R) en dan is de afhankelijkheid triviaal.

Bijgevolg is het schema in BCNF.

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!

Naam en voorletters:

Ident.nr.:

Hoofd-tentamendeel

1. Gegeven is het volgende database schema voor een school (tabelnaam met attributnamen en onderlijning van de primaire sleutelattributen):

docent(dnr, dnaam, aantal-uur)

bevoegdheid(dnr, vaknaam, klas)

programma(klas, vaknaam, aantal-uur)

doceert(dnr, vaknaam, klas)

Het schema beschrijft de toewijzing van docenten aan vakken en klassen. Elke docent (met een nummer en een naam) werkt een aantal uur, en is bevoegd om bepaalde vakken te verzorgen voor bepaalde klassen. (Er zijn docenten die wiskunde alleen aan klassen 1, 2 en 3 mogen geven en niet aan 4, 5 en 6 bijvoorbeeld.) Elke klas krijgt een bepaald aantal uur van een vak. (Als dat aantal 0 is komt het vak niet voor bij die klas. De programma tabel bevat dus geen nullen.) Een docent die een bepaald vak geeft aan een bepaalde klas geeft alle uren van dat vak aan die klas. Docenten geven alleen vakken aan klassen als ze daarvoor bevoegd zijn.

Dit schema is in BCNF. (Dat mag je aannemen.) Er zijn dus geen verborgen afhankelijkheden die we niet hebben vermeld.

We voeren nu twee nieuwe regels in (nog iets scherper gesteld t.o.v. het tentamen):

1. Elke docent mag in de school maar 1 vak doceren (ook al blijft hij natuurlijk voor meer vakken bevoegd).
2. Een docent is bevoegd voor een aantal vakken en een aantal klassen en is bevoegd om elk van die vakken in elk van die klassen te doceren.

Pas het bovenstaande database schema aan zodat het opnieuw in BCNF is.

Leg alle stappen die je maakt uit met behulp van de database theorie. Een eindresultaat alleen, of een “jan-boere-fluitjes” uitleg is niet voldoende.

Doe niet méér dan er wordt gevraagd. Er wordt een punt afgetrokken als je teveel doet.

Regel 1 zegt dat de functionele afhankelijkheid $\{dnr\} \rightarrow \{vaknaam\}$ geldt in de tabel “doceert”. Deze is daardoor niet meer in BCNF omdat $\{dnr\} \rightarrow \{vaknaam\}$ geldt terwijl $\{dnr\}$ geen supersleutel is want $\{dnr\} \rightarrow \{klas\}$ geldt in het algemeen niet in “doceert”.

We kunnen de tabel “doceert” volgens het algoritme van figuur 7.13 decomponeren. Dat levert meteen in 1 stap het volgende op:

doceert_1(dnr, vaknaam)

doceert_2(dnr, klas)

Merk op dat regel 1 geen invloed heeft op de tabel “bevoegdheid”: de docent blijft voor dezelfde vakken bevoegd. Hij mag alleen niet meer dan 1 vak in de school doceren.

Regel 2 is een multivalued afhankelijkheid in de tabel “bevoegdheid”. We hoeven hier voor BCNF niets mee te doen, en omdat teveel doen leidt tot punten-afrek doen we er ook niets mee.

Het eindresultaat wordt dan:

docent(dnr, dnaam, aantal-uur)

bevoegdheid(dnr, vaknaam, klas)

programma(klas, vaknaam, aantal-uur)

doceert_1(dnr, vaknaam)

doceert_2(dnr, klas)

Naam en voorletters:

Ident.nr.:

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 titels van boeken waarvan de bibliotheek geen exemplaar heeft aangekocht in het jaar van uitgifte.**

Welk formalisme (query taal) kies je, en waarom? Herken je in deze vraag een “type” dat je bij de keuze van het formalisme of de aard van de uitwerking gebruikt?

Dit is een query met een verschil. Dat kan makkelijk in de tupel calculus en in SQL. In de relationele algebra is het even iets lastiger omdat we ervoor moeten zorgen dat de “tabellen” die we van elkaar aftrekken dezelfde structuur hebben. Dat kost een extra join en projectie. We geven hier alle drie de formalismen voor de volledigheid.

Uitwerking:

$$\Pi_{\text{titel}} (\Pi_{\text{titel,ISBN}}(\text{boek}) - \Pi_{\text{titel,ISBN}}(\sigma_{\text{jaar} = \text{exjaar}} (\text{boek} \bowtie \text{exemplaar})))$$
$$\{ t \mid \exists b \in \text{boek} (b[\text{titel}] = t[\text{titel}] \wedge \neg \exists e \in \text{exemplaar} (b[\text{ISBN}] = e[\text{ISBN}] \wedge b[\text{jaar}] = e[\text{exjaar}])) \}$$

```
SELECT b.titel
FROM boek as b
WHERE NOT EXISTS
( SELECT *
  FROM exemplaar as e
  WHERE b.jaar = e.exjaar
    AND b.ISBN = e.ISBN)
```

Zitten er in deze opgave 1 of meer addertjes onder het gras? Zoja welke en wat heb je gedaan om het addertje (of de addertjes) te vermijden?

Je moet bij deze vraag opletten dat je geen titels van titels aftrekt want boeken worden geïdentificeerd door het ISBN nummer en niet door de titel. In de algebra is deze fout al snel gemaakt. Bij SQL maak je de fout gemakkelijk als je EXCEPT gebruikt in plaats van NOT EXISTS.

Bij de relationele algebra is er nog een ander klein addertje dat “jaar = exjaar” niet automatisch uit de join volgt. Bij SQL is er een heel klein addertje dat je niet mag vergeten de b.ISBN = e.ISBN te schrijven.

Naam en voorletters:

Ident.nr.:

b. Geef de titel van de boeken die al werden gereserveerd nog vóór de eerste uitlening (van dat boek).

Welk formalisme kies je, en waarom? Herken je in deze vraag een “type” dat je bij de keuze van het formalisme of de aard van de uitwerking gebruikt?

Dit is andermaal een verschil-vraag. In de algebra is ze een beetje lastig, maar het gaat wel. Het aanbrenge van de geschikte structuur voor de tabellen die van elkaar afgetrokken moeten worden is de moeilijkelijkheid. De tupel calculus en SQL liggen dus meer voor de hand. Voor de volledigheid geven we de drie uitwerkingen.

Uitwerking:

$$\prod_{\text{titel}} \left(\prod_{\text{titel, ISBN, datum}} (\text{boek} \bowtie \prod_{\text{ISBN, datum}} (\text{reservering})) - \prod_{\text{titel, ISBN, datum}} (\sigma_{\text{van} \leq \text{datum}} (\text{boek} \bowtie \text{exemplaar} \bowtie \prod_{\text{ISBN, datum}} (\text{reservering}) \bowtie \prod_{\text{barcode, van}} (\text{uitlening}))) \right)$$

$$\{ t \mid \exists b \in \text{boek} (b[\text{titel}] = t[\text{titel}] \wedge \exists r \in \text{reservering} (r[\text{ISBN}] = b[\text{ISBN}] \wedge \neg \exists e \in \text{exemplaar} (\exists u \in \text{uitlening} (b[\text{ISBN}] = e[\text{ISBN}] \wedge e[\text{barcode}] = u[\text{barcode}] \wedge u[\text{van}] \leq r[\text{datum}])))) \}$$

```
SELECT b.titel
FROM boek as b, reservering as r
WHERE b.ISBN = r.ISBN AND b.ISBN NOT IN
( SELECT e.ISBN
  FROM exemplaar as e, uitlening as u
  WHERE e.ISBN = b.ISBN AND e.barcode = u.barcode
    AND u.van <= r.datum)
```

Zitten er in deze opgave 1 of meer valkuilen? Zoja welke en wat heb je gedaan om niet in die valkuil(en) te trappen?

Bij de relationele algebra moet je opletten dat de “datum” nog in de projecties zit die je gaat van elkaar aftrekken omdat je anders alle boeken wegfiltert waarvoor er een reservering is vóór een uitlening, maar niet noodzakelijk vóór elke uitlening. Bij de tupel calculus en SQL is die valkuil er niet. Maar bij de tupel calculus moet je goed opletten dat je eerst “ $\exists r \in \text{reservering}$ ” en dan “ $\neg \exists e \in \text{exemplaar} (\exists u \in \text{uitlening}$ ” doet, voor het geval er nog geen uitlening zou zijn. Dus, je kijkt of de reservering vóór de eerste uitlening is, als die bestaat, en niet of de eerste uitlening na de eerste reservering komt, want dan gaat het mis. Bij SQL zie je die valkuil als je in de eerste FROM ook uitlening gebruikt.

Naam en voorletters:

Ident.nr.:

- c. **Geef de titels van boeken waarvan elke faculteit een exemplaar heeft aangekocht in het jaar van uitgifte.**

Welk formalisme kies je, en waarom? Herken je in deze vraag een “type” dat je bij de keuze van het formalisme of de aard van de uitwerking gebruikt?

De uitdrukking “waarvan elke faculteit...” wordt zo gebruikt dat het om de vaste verzameling van alle faculteiten gaat, dus is dit een query om in de relationele algebra uit te werken, met de division operator. We geven alleen maar een uitwerking in de algebra.

Uitwerking:

$$\Pi_{\text{titel}}(\Pi_{\text{ISBN, titel, faculteit}}(\sigma_{\text{jaar} = \text{exjaar}}(\text{boek} \bowtie \text{exemplaar}))) \div \Pi_{\text{faculteit}}(\text{exemplaar})$$

Zitten er in deze opgave 1 of meer instinkers? Zoja welke en wat heb je gedaan om er niet in te stinken?

De instinker is dat je niet mag vergeten om de projectie op ISBN, titel en faculteit te maken omdat je anders nog de per exemplaar verschillende barcode hebt waardoor de deling fout gaat.

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. Aggregatie mag je alleen in SQL gebruiken (niet in de algebra).

- a. **Geef de naam en faculteit van elke lener en het aantal (lopende of oude) reserveringen van die lener.**

Welk formalisme kies je, en waarom? Herken je in deze vraag een “type” dat je bij de keuze van het formalisme of de aard van de uitwerking gebruikt?

Dit is een vraag met aggregatie dus dit moet in SQL. Het is ook nog een “union met een 0” vraag.

Uitwerking:

```
SELECT u.naam, u.faculteit, count(*) as aantal
FROM reservering as r
WHERE (r.naam, r.faculteit) IN
      (SELECT u.naam, u.faculteit
       FROM uitlening as u)
GROUP BY r.naam, r.faculteit
UNION
SELECT u.naam, u.faculteit, 0 as aantal
FROM uitlening as u
WHERE NOT EXISTS
      ( SELECT *
        FROM reservering as r
        WHERE u.naam = r.naam and u.faculteit = r.faculteit )
```

Naam en voorletters:

Ident.nr.:

Probeer deze vraag je om de tuin te leiden? Zoja hoe, en wat heb je gedaan om niet om de tuin geleid te worden?

In het eerste deel van de query moet je opletten dat je niet “FROM uitlening as u, reservering as r” doet, want dan tel je het aantal reserveringen maal het aantal uitleningen (zo werkt het cart. produkt dat je maakt). Met de subquery vermijdt je dit probleem. Bij de foutieve FROM wil je misschien ook count(r.*) zetten maar dat mag niet in SQL. Sommigen vergeten de subquery, maar dan krijg je ook mensen die nog nooit iets geleend hebben (alleen gereserveerd) en dat zijn geen “leners”.

Als je niet oplet vergeet je ook de union die nodig is om leners te geven die 0 reserveringen hebben.

- b. Geef de naam en voorletters van auteurs wiens “werk” door leners van alle faculteiten is geleend. (Ze hoeven niet allemaal hetzelfde boek te lenen.)**

Welk formalisme kies je, en waarom? Herken je in deze vraag een “type” dat je bij de keuze van het formalisme of de aard van de uitwerking gebruikt?

Hier komen we weer de verzameling van alle faculteiten tegen, dus gaan we de algebra gebruiken met de division operator. Merk op dat aangezien elke lener bij slechts 1 faculteit hoort dus moet “leners van alle faculteiten” betekenen “uit elke faculteit 1 of meer leners”.

Uitwerking:

$$\prod_{\text{naam, voorletters, faculteit}}(\text{auteur}) \bowtie \prod_{\text{ISBN, barcode}}(\text{exemplaar}) \bowtie \prod_{\text{barcode, faculteit}}(\text{uitlening}) \div \prod_{\text{faculteit}}(\text{exemplaar})$$

Zitten er in deze opgave 1 of meer voetangels en klemmen? Zoja welke en wat heb je gedaan om die te vermijden?

Een eerste probleem is dat de faculteit van het exemplaar niet mag gejoind worden met de faculteit van de uitlening. Een tweede probleem is dat de naam van de auteur niet mag gejoind worden met de naam van de lener. Een derde probleem is dat de overbodige attributen moeten weggeprojecteerd worden voordat de division plaats heeft.

- c. Geef naam en faculteit van de leners die nooit een boek hebben geleend bij een andere faculteit dat hun eigen faculteit op dit moment (nu dus) bezit.**

Welk formalisme kies je, en waarom? Herken je in deze vraag een “type” dat je bij de keuze van het formalisme of de aard van de uitwerking gebruikt?

De tupel calculus is over en is ook prima geschikt voor deze vraag. Er zit een verschil in wat in de tupel calculus meestal gemakkelijker is dan in de algebra.

Naam en voorletters:

Ident.nr.:

Uitwerking:

$$\{ t \mid \exists u \in \text{uitlening} (u[\text{naam}] = t[\text{naam}] \wedge u[\text{faculteit}] = t[\text{faculteit}] \wedge \\ \neg \exists uu \in \text{uitlening} (u[\text{naam}] = uu[\text{naam}] \wedge u[\text{faculteit}] = uu[\text{faculteit}] \wedge \\ \exists e \in \text{exemplaar} (\exists ue \in \text{exemplaar} (ue[\text{barcode}] = uu[\text{barcode}] \wedge \\ e[\text{ISBN}] = ue[\text{ISBN}] \wedge e[\text{faculteit}] = uu[\text{faculteit}] \wedge \\ ue[\text{faculteit}] \neq e[\text{faculteit}])))) \}$$

Zitten er in deze opgave 1 of meer valstrikken? Zoja welke en wat heb je gedaan om niet “gestrikt” te worden?

In deze vraag zitten geen echte valstrikken.

4. Leg in normaal klinkend Nederlands uit wat er in de volgende queries wordt gevraagd.

a. SELECT u.naam, u.faculteit

FROM uitlening as u, exemplaar as e, exemplaar as ue, boek as b, boek as ub
WHERE u.barcode = ue.barcode AND u.faculteit = e.faculteit
AND u.faculteit <> ue.faculteit AND e.ISBN = b.ISBN
AND ue.ISBN = ub.ISBN AND b.titel = ub.titel

Geef de naam en faculteit van leners die (ooit) een (exemplaar van een) boek hebben geleend van een andere faculteit hoewel hun eigen faculteit (nu) een (exemplaar van een) boek bezit met dezelfde titel.

Zit er in deze vraag iets dat je tracht te misleiden? Zoja wat heb je gedaan om niet misleid te worden?

Het is even opletten dat een boek wordt geïdentificeerd door het ISBN nummer en niet door de titel. Dit is in de Nederlandse zin te merken (“dezelfde titel”).

b. $\prod_{\text{naam, faculteit}} (\sigma_{\text{datum} > \text{tot}} (\text{exemplaar} \bowtie \text{uitlening} \bowtie \text{reservering}))$

Geef de naam en faculteit van leners die een (exemplaar van een) boek van hun eigen faculteit hebben geleend en teruggebracht en die daarna dat boek hebben gereserveerd.

Probeert deze vraag je op het verkeerde been te zetten? Zoja wat heb je gedaan om op het juiste been te blijven staan?

Bij de join worden de verschillende betekenissen van “faculteit” mee-gejoind. Dit leidt tot het “van hun eigen faculteit” in de Nederlandse zin.

5. Veronderstel dat een domme ontwerper een relatieschema R heeft gemaakt met attributen {A, B, C, D, E, F, G} en functionele afhankelijkheden {BCD → A, BC → E, A → F, F → G, C → D, A → G}.

a. Geef een decompositie die in BCNF is. Beargumenteer je antwoord, m.a.w. geef beknopt aan hoe je tot die decompositie komt en waarom ze volgens jou in BCNF is. (Zonder uitleg wordt het antwoord fout gerekend.)

Naam en voorletters:

Ident.nr.:

In de uitwerking van het tentamen van 30 juni 2003 staat de berekening van een canonical cover en daarna van een decompositie in derde normaalvorm, die ook in BCNF bleek te zijn. Die decompositie is (A,B,C,E), (C,D), (A,F), (F,G).

Hiervoor worden de algoritmes van figuren 7.8 en 7.14 gebruikt.

Je moet gewoon die uitwerking overnemen.

- b. Is je decompositie ook “dependency preserving”? Zoja, beargumenteer dat, zonne, kun je dan een andere decompositie geven in BCNF die wel dependency preserving is? En als dit niet zou kunnen, waarom dan niet?

Omdat de BCNF decompositie het resultaat is van de algoritmes 7.8 en 7.14 is de decompositie gegarandeerd dependency preserving. (Je ziet elke functionele afhankelijkheid van het canonical cover in 1 van de tabellen terug.)

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 Ω 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 **volledig** 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).

Om te bewijzen dat R1, R2 en R3 volledig zijn gaan we uit deze regels de regels F1, F2 en F3 afleiden (waarvan we mogen aannemen dat ze volledig zijn)

F1: Stel dat $Y \subseteq X$, dan geldt volgens R1 $X \rightarrow X$, wat gelijk is aan $X \rightarrow Y(X - Y)$. Volgens R2 geldt dan $X \rightarrow Y$ waarmee F1 bewezen is.

F2: Stel dat $X \rightarrow Y$ geldt, en dat Z nog een verzameling attributen is. Volgens R1 geldt $XZ \rightarrow XZ$. Dit combineren we met $X \rightarrow Y$ in R3 tot $XZ \rightarrow XYZ$. Met R2 projecteren we X weg en krijgen $XZ \rightarrow YZ$.

F3: Als $X \rightarrow Y$ en $Y \rightarrow Z$ gelden dan geeft R3 $X \rightarrow YZ$, waaruit we met R2 $X \rightarrow Z$ afleiden.

Waardering: vraag 1: 3 punten, vraag 2, 3 en 4: 2 punten per query, vraag 5: 3 punten per antwoord (a en b), 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.