

Naam en voorletters:

Ident.nr.:

TECHNISCHE UNIVERSITEIT EINDHOVEN

Faculteit Wiskunde en Informatica

Extra Tentamen Databases 1, 2M400, 8 oktober 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.** De ruimte die is voorzien is in elk geval voldoende voor het correcte antwoord, in een normaal leesbare letter-grootte. Elk antwoord moet staan in de ruimte bij de vraag, tenzij verwezen is naar de laatste (witte) pagina voor een “tweede poging”.

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.)

Hier is het recept voor dit tentamen:

1. Zoek een vraag die je denkt te kunnen beantwoorden.
2. Werk het antwoord uit.
3. Kom het antwoord tonen en indien nodig toelichten.
4. Ga na beoordeling/toelichting verder met een volgende vraag, of met een tweede (en tevens laatste) poging voor dezelfde vraag.
5. Herhaal dit proces tot het eerste van de volgende drie situaties zich voordoet: je hebt alle vragen beantwoord, het is 12 uur is, of je geeft het op.

Beoordeling: elke vraag of onderdeel van een vraag staat op 2 punten. Is de vraag meteen goed beantwoord dan krijg je 2 punten. Is de vraag bij een herkansing goed krijg je nog 1 punt. Is de vraag na een herkansing nog steeds niet (helemaal) goed krijg je 0 punten.

Alles telt op tot 20 en wordt gedeeld door 2, met afronding naar boven.

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 ≤.) 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.:

1. Druk de volgende vragen over de bibliotheek uit in één van de drie bestudeerde querytalen (relationele algebra, tupel calculus of SQL) naar keuze. Als je een aggregatie wil gebruiken dan moet je SQL als taal kiezen. Er zijn 5 vragen, en in de 5 antwoorden moet minstens 1 keer de relationele algebra worden gebruikt, 1 keer de tupel calculus en 1 keer SQL.
 - a. **Geef de titels van alle boeken, van auteurs waarvan de bibliotheek meer dan één boek bezit.** (Verschillende exemplaren van eenzelfde boek tellen niet als meer dan één boek, verschillende edities van eenzelfde boek wel.)

Uitwerking:

Voor de eenvoud maar even in tupel calculus gedaan.

We zoeken twee boeken (b en b1, met verschillend ISBN nummer want anders is het hetzelfde boek) en met dezelfde auteur (a en a1 zijn dezelfde persoon)..

$$\{ t \mid \exists b \in \text{boek} (b[\text{titel}] = t[\text{titel}] \wedge \\ \exists a \in \text{auteur} (b[\text{ISBN}] = a[\text{ISBN}] \wedge \\ \exists b1 \in \text{boek} (a[\text{ISBN}] \neq b1[\text{ISBN}] \wedge \\ \exists a1 \in \text{auteur} (a1[\text{ISBN}] = b1[\text{ISBN}] \wedge \\ a1[\text{naam}] = a[\text{naam}] \wedge a1[\text{voorletters}] = a[\text{voorletters}]))))) \}$$

- b. **Geef de titel van de boeken die al vaker zijn gereserveerd dan uitgeleend.**

Uitwerking:

Tellen, dus SQL gebruiken. Opletten we de nullen meetellen.

```
SELECT titels
FROM boek as b
WHERE ( SELECT COUNT(*)
        FROM reservering as r
        WHERE b.ISBN = r.ISBN )
>
( SELECT COUNT(*)
  FROM uitlening as u, exemplaar as e
  WHERE u.barcode = e.barcode
        AND u.ISBN = b.ISBN )
```

Naam en voorletters:

Ident.nr.:

- c. **Geef de titels van boeken die al door leners van elke faculteit zijn geleend.**
(Dus uit elke faculteit een lener. Leners die uit alle faculteiten tegelijk komen bestaan niet.)

Uitwerking:

“elke faculteit”, dus division in relationele algebra, want we vergelijken met een vaste verzameling.

$$\Pi_{\text{titel}} (\Pi_{\text{titel, ISBN, faculteit}} (\text{uitlening} \bowtie \Pi_{\text{barcode, ISBN(exemplaar)}} \bowtie \text{uitlening}) \div \Pi_{\text{faculteit}} (\text{exemplaar}))$$

- d. **Geef de naam en faculteit van elke lener en het aantal openstaande leningen van die lener.** (Een lening staat open als het boek nog niet is teruggebracht.)

Uitwerking:

Niet vergeten dat er leners zijn die geen openstaande leningen hebben (maar wel leningen waarbij de boeken zijn teruggebracht).

```
SELECT u.naam, u.faculteit, COUNT(*) AS aantal
FROM uitlening AS u
WHERE u.tot IS NULL
GROUP BY u.naam, u.faculteit
```

UNION

```
SELECT u.naam, u.faculteit, 0 AS aantal
FROM uitlening AS u
WHERE NOT EXISTS
( SELECT *
  FROM uitlening AS uu
  WHERE u.naam = uu.naam
        AND u.faculteit = uu.faculteit
        AND u.tot IS NULL )
```

Naam en voorletters:

Ident.nr.:

- e. **Geef de faculteiten die met elke andere faculteit een boek gemeenschappelijk hebben.** (Gemeenschappelijk betekent dat beide faculteiten een exemplaar hebben van eenzelfde boek.)

Uitwerking:

Faculteit f heeft een exemplaar e van een bepaald boek en faculteit ff heeft een exemplaar ee van datzelfde boek. Merk op dat het boek dat f met een faculteit gemeenschappelijk heeft verschillend kan zijn per faculteit. Merk ook op dat wanneer f en ff dezelfde faculteit zijn de conditie triviaal voldaan is door e en ee eenzelfde exemplaar van een boek in de faculteit f te laten zijn.

$$\{ t \mid \exists f \in \text{exemplaar} (f[\text{faculteit}] = t[\text{faculteit}] \wedge \\ \forall ff \in \text{exemplaar} (\exists ee \in \text{exemplaar} (ee[\text{faculteit}] = ff[\text{faculteit}] \wedge \\ \exists e \in \text{exemplaar} (e[\text{faculteit}] = f[\text{faculteit}] \wedge \\ e[\text{ISBN}] = ee[\text{ISBN}])))) \}$$

4. Leg in normaal klinkend Nederlands uit wat er in de volgende queries wordt gevraagd.
- a. $\prod_{\text{faculteit}} (\text{exemplaar}) - \prod_{\text{faculteit}} (\text{exemplaar} \bowtie \text{uitlening})$

Geef de faculteiten waarvan nog geen enkele medewerker ooit een boek heeft geleend van zijn eigen faculteit.

b. $\{ t \mid \exists e \in \text{exemplaar} (t[\text{faculteit}] = e[\text{faculteit}] \wedge \\ \exists b \in \text{boek} (b[\text{ISBN}] = e[\text{ISBN}] \wedge \\ \exists f \in \text{exemplaar} (\forall u \in \text{uitlening} (\forall ee \in \text{exemplaar} (\\ (u[\text{faculteit}] = f[\text{faculteit}] \wedge ee[\text{barcode}] = u[\text{barcode}] \wedge ee[\text{ISBN}] = b[\text{ISBN}]) \Rightarrow \\ ee[\text{faculteit}] \neq e[\text{faculteit}]))))) \}$

Geef de faculteiten die een boek hebben dat nog niet door leners van alle faculteiten bij die (eerste) faculteit is geleend.

Naam en voorletters:

Ident.nr.:

5. We hebben een relatieschema R , met attributen $\{A, B, C\}$ en functionele afhankelijkheden $\{A \rightarrow B, B \rightarrow C\}$.

- a. We kunnen dit schema decomponeren in R_1 met attributen $\{A, B\}$ en R_2 met attributen $\{A, C\}$. Is dit een lossless join decompositie? Is ze dependency preserving? In welke normaalvorm is deze decompositie? Motiveer elk onderdeel van je antwoord.

Uit $A \rightarrow B$ en $B \rightarrow C$ volgt $A \rightarrow ABC$, dus A is een supersleutel.

R_1 en R_2 hebben A als enige gemeenschappelijk attribuut.

Aangezien dit een supersleutel is is de decompositie een lossless-join decompositie.

In R_1 geldt $A \rightarrow B$ (en $A \rightarrow AB$) als enige niet-triviale beperking, en in R_2 geldt $A \rightarrow C$ (en $A \rightarrow AC$) als enige niet-triviale beperking.

Hieruit kunnen we niet $B \rightarrow C$ afleiden. Die kan alleen gecontroleerd worden in $R_1 \bowtie R_2$. Bijgevolg is de decompositie niet dependency-preserving.

R_1 en R_2 zijn in BCNF (en dus ook in 3NF) want elke relatie met maar twee attributen (en waarin \emptyset geen supersleutel is) is in BCNF.

- b. We kunnen dit schema ook decomponeren in R_2 met attributen $\{A, C\}$ en R_3 met attributen $\{B, C\}$. Is dit een lossless join decompositie? Is ze dependency preserving? In welke normaalvorm is deze decompositie? Motiveer elk onderdeel van je antwoord.

R_2 en R_3 hebben het attribuut C als (enige) gemeenschappelijke attribuut. Maar $C \rightarrow A$ geldt niet en $C \rightarrow B$ ook niet, dus deze decompositie is geen lossless-join decompositie.

in R_2 geldt $A \rightarrow C$ (en $A \rightarrow AC$) als enige niet-triviale beperking, en in R_3 geldt $B \rightarrow C$ (en $B \rightarrow BC$) als enige niet-triviale beperking. Hieruit kunnen we niet

$A \rightarrow B$ afleiden. Die kan alleen gecontroleerd worden in $R_1 \bowtie R_2$.

Bijgevolg is de decompositie niet dependency-preserving.

R_2 en R_3 zijn in BCNF (en dus ook in 3NF) want elke relatie met maar twee attributen (en waarin \emptyset geen supersleutel is) is in BCNF.

Naam en voorletters:

Ident.nr.:

6. Neem een relatieschema R met attributen $\{A, B, C\}$. In dit schema geldt de multivalued afhankelijkheid $A \twoheadrightarrow B$. Behalve logische gevolgen hiervan, inclusief triviale afhankelijkheden gelden er geen andere beperkingen in R .
Geldt in R dan ook $B \twoheadrightarrow C$? Bewijs de correctheid van je antwoord.

Neem de volgende tabel:

A	B	C
0	0	0
1	0	1

$A \twoheadrightarrow B$ geldt triviaal in deze tabel want opdat $A \twoheadrightarrow B$ niet zou gelden moeten we tenminste twee tupels met dezelfde A waarde vinden.

$B \twoheadrightarrow C$ geldt niet in deze tabel want daarvoor zouden we ook de tupels

1	0	0
0	0	1

moeten hebben.

We hebben dus een tegenvoorbeeld tegen de bewering dat als $A \twoheadrightarrow B$ geldt dan ook $B \twoheadrightarrow C$ zou gelden.