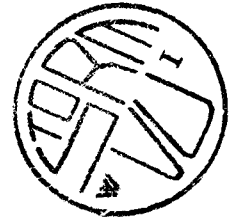


Uitwerking tentamen Databases I d.d. 25 mei 2004



Antwoord opgave 1

BOEKING(boekingsnr, datum_feest, KvK_nr)

boekingsnr PK

KvK_nr FK naar BEDRIJF: NOT NULL [on delete: cascade/restricted, on modify: cascade]

ARTIEST(artiest_naam, netnr, abonneenr, prijs)

artiest_naam PK

LOONDIENSTARTIEST(artiest_naam, maandsalaris)

artiest_naam PK

artiest_naam FK naar ARTIEST [on delete: cascade/restricted, on modify: cascade]

FREELANCEARTIEST(artiest_naam, uurvergoeding)

artiest_naam PK

artiest_naam FK naar ARTIEST [on delete: cascade/restricted, on modify: cascade]

ADRES(huisnummer, straat, woonplaats, postcode)

(huisnummer, straat, woonplaats) PK

(huisnummer, postcode) AK

BEDRIJF(KvK_nr, bankgironr, huisnummer, straat, woonplaats)

KvK_nr PK

(huisnummer, straat, woonplaats) FK naar ADRES: NOT NULL [on delete: restricted, on modify: cascade]

PERSONEELSLID(sofi_nr, voornaam, achternaam, huisnummer, straat, woonplaats)

sofi_nr PK

(huisnummer, straat, woonplaats) FK naar ADRES: NOT NULL [on delete: restricted, on modify: cascade]

ECHTGENOOT(voornaam, sofi_personeelslid)

sofi_personeelslid PK

sofi_personeelslid FK naar PERSONEELSLID [on delete: cascade/restricted, on modify: cascade]

KIND(voornaam, sofi_personeelslid, geboortedatum_kind)

(voornaam, sofi_personeelslid) PK

sofi_personeelslid FK naar PERSONEELSLID [on delete: cascade/restricted, on modify: cascade]

[**BoekingVanArtiest**(boekingsnr, artiest_naam, tijdstip_vanaf, tijdstip_tot)

(boekingsnr, artiest_naam) PK

boekingsnr FK naar Boeking [on delete: cascade/restricted, on modify: cascade]

artiest_naam FK naar ARTIEST [on delete: cascade/restricted, on modify: cascade]

[**WerktVoor**(sofi_personeelslid, KvK_nr_bedrijf)

(sofi_personeelslid, KvK_nr_bedrijf) PK

sofi_personeelslid FK naar PERSONEELSLID [on delete: cascade/restricted, on modify: cascade]

KvK_nr_bedrijf FK naar BEDRIJF [on delete: cascade/restricted, on modify: cascade]

[**Vaardigheden**(artiest_naam, vaardigheid)

(artiest_naam, vaardigheid) PK

artiest_naam FK naar ARTIEST [on delete: cascade, on modify: cascade]

✓ PK ∈ PERSONEELSLID: PK komt minstens 1 keer voor als FK in WerktVoor;

✓ PK ∈ BEDRIJF: PK komt minstens 1 keer voor als FK in WerktVoor;

✓ PK ∈ BEDRIJF: PK komt minstens 1 keer voor als FK in BOEKING;

✓ PK ∈ Boeking: PK komt minstens 1 keer voor als FK in BoekingVanArtiest;
✓ PK ∈ Artiest: (PK ∈ LoondienstArtiest ∨ PK ∈ FreelanceArtiest) ∨
(PK ∈ LoondienstArtiest ∨ PK ∈ FreelanceArtiest);

opdrachten
+
tegenwoordig < tijdperk

$\forall PK \in \text{Boeking: PK komt minstens 1 keer voor als FK in BoekingVanArtiest;}$

$\forall PK \in \text{ARTIEST: (PK} \in \text{Docent} \wedge PK \notin \text{Student_assistent}) \vee (PK \notin \text{Docent} \wedge PK \in \text{Student_assistent}) \Rightarrow \text{bondleus} \text{ artist}$

Antwoord opgave 2

N.B.: op de antwoorden zoals hier gegeven zijn vele varianten mogelijk.

a) Algebra:

$\pi_{\text{bestemming}}((\sigma_{\text{reisbureau}=\text{"Gladjes"}} \text{Levert}) * \text{Biedt})$

SQL:

```
SELECT [DISTINCT] B.bestemming
FROM      Biedt AS B, Levert AS L
WHERE     B.touroperator = L.touroperator AND L.reisbureau = "Gladjes";
```

b) Tupelcalculus:

$\{ k.\text{reisbureau} \mid \text{Klant}(k) \wedge$
 $(\exists g1: (\text{Geïnteresseerd}(g1) \wedge (g1.\text{persoon} = k.\text{persoon} \wedge g1.\text{bestemming} = \text{"Mallorca"})) \wedge$
 $(\neg \exists g2: (\text{Geïnteresseerd}(g2) \wedge (g2.\text{persoon} = k.\text{persoon} \wedge g2.\text{bestemming} = \text{"Ibiza"})) \}$

SQL:

```
SELECT [DISTINCT] K.reisbureau
FROM      Klant AS K
WHERE     EXISTS (SELECT      G1.persoon
                     FROM      Geïnteresseerd AS G1
                     WHERE     G1.persoon = K.persoon AND
                               G1.bestemming = "Mallorca")
          AND NOT EXISTS
            (SELECT      G2.persoon
             FROM      Geïnteresseerd AS G2
             WHERE     G2.persoon = K.persoon AND
                       G2.bestemming = "Ibiza");
```

Alternatieve oplossing SQL:

```
SELECT [DISTINCT] K.reisbureau
FROM      Klant AS K
WHERE     K.persoon IN (SELECT      G1.persoon
                       FROM      Geïnteresseerd AS G1
                       WHERE     G1.bestemming = "Mallorca")
          AND NOT ( K.persoon IN (SELECT      G2.persoon
                                   FROM      Geïnteresseerd AS G2
                                   WHERE     G2.bestemming = "Ibiza") );
```

c) Algebra:

$\text{Biedt} \div \pi_{\text{bestemming}}(\sigma_{\text{persoon}=\text{"Piet"}} \text{Geïnteresseerd})$

Domein calculus:

$\{ t \mid \text{Biedt}(\text{touroperator: } t) \wedge \forall b (\text{Geïnteresseerd}(\text{persoon: "Piet", bestemming: } b) \rightarrow \text{Biedt}(\text{touroperator: } t, \text{bestemming: } b)) \}$

Antwoord opgave 3

Stap 1 (maak enkelvoudige rechterkanten):

$\{ AB \rightarrow C, AD \rightarrow F, B \rightarrow A, BE \rightarrow F, C \rightarrow A, C \rightarrow D \}$

Stap 2 (verwijder overbodige attributen uit de linkerkant):

$\{ B \rightarrow C, AD \rightarrow F, B \rightarrow A, B \rightarrow F, C \rightarrow A, C \rightarrow D \}$

Stap 3 (verwijder overbodige afhankelijkheden):

$\{ B \rightarrow C, AD \rightarrow F, C \rightarrow A, C \rightarrow D \}$

Toelichting:

ad 1) Stap 1 is triviaal.

ad 2) In stap 2 moeten we van elke afhankelijkheid nagaan of de linkerkant minimaal is en als dat niet het geval is, dan moeten we die linkerkant alsnog minimaliseren.

Neem $G = \{ AB \rightarrow C, AD \rightarrow F, B \rightarrow A, BE \rightarrow F, C \rightarrow A, C \rightarrow D \}$ (resultaat stap 1).

Bekijk eerst $AB \rightarrow C$.

We proberen eerst A weg te laten. Dan moeten we nagaan of $C \in (AB - A)^+_G$.

Er geldt: $(B)^+_G = ABCDF$ en $C \in ABCDF$. Volgens het algoritme mogen we $AB \rightarrow C$ dus vervangen door $B \rightarrow C$.

Dus neem nu $G := G - \{ AB \rightarrow C \} \cup \{ B \rightarrow C \}$.

Nu moeten we nagaan of deze nieuwe $B \rightarrow C$ wel een minimale linkerkant heeft. Dat is triviaal.

Vervolgens bekijken we $AD \rightarrow F$.

We proberen eerst A weg te laten. $(D)^+_G = D$ en $F \notin D$. Dus A kunnen we niet weglaten.

Vervolgens proberen we D weg te laten. Er geldt: $(A)^+_G = A$ en $F \notin A$. Dus D kunnen we ook niet weglaten.

Kortom, $AD \rightarrow F$ heeft een minimale linkerkant.

De afhankelijkheid $B \rightarrow A$ heeft overduidelijk al een minimale linkerkant.

Dus bekijken we nu $BE \rightarrow F$.

We proberen eerst B weg te laten. Er geldt dan: $(E)^+_G = E$ en $F \notin E$. Dus B kunnen we niet weglaten.

Vervolgens proberen we E weg te laten. Er geldt: $(B)^+_G = ABCDF$ en $F \in ABCDF$. Dus heeft $BE \rightarrow F$ geen minimale linkerkant en mogen we nu $BE \rightarrow F$ vervangen door $B \rightarrow F$.

Dus neem nu $G := G - \{ BE \rightarrow F \} \cup \{ B \rightarrow F \}$.

Nu moeten we nagaan of deze nieuwe $B \rightarrow F$ wel een minimale linkerkant heeft. Dat is triviaal.

Ook de afhankelijkheden $C \rightarrow A$ en $C \rightarrow D$ hebben overduidelijk al een minimale linkerkant.

ad 3) In stap 3 moeten we van elke afhankelijkheid nagaan of deze overbodig is en als dat het geval is, dan moeten we deze verwijderen.

Neem $G = \{ B \rightarrow C, AD \rightarrow F, B \rightarrow A, B \rightarrow F, C \rightarrow A, C \rightarrow D \}$ (resultaat stap 2).

Bekijk eerst $B \rightarrow C$.

Ga na of $C \in (B)^+_H$ met $H = G - \{ B \rightarrow C \}$. Er geldt: $(B)^+_H = ABF$ en $C \notin ABF$, dus $B \rightarrow C$ is niet overbodig.

Antwoord opgave 4

- a) Tupel (f, h) wordt uit Y verwijderd.
- b) Er verandert niets. (Vanwege verwijzingen vanuit W met "on delete: restricted".) (via Y)
- c) Tupel (h, l) wordt uit X verwijderd.
Tupel (f, h) wordt uit Y verwijderd.
Tupel (p, h) in Z wordt (p, null)
- d) Tupel (f, h) in Y wordt (f, j)
- e) Er verandert niets. (Vanwege verwijzingen vanuit W met "on modify: restricted".)
- f) Tupel (g, l) in X wordt (r, l)
Tupel (d, g) in Y wordt (d, r)
Tupel (e, g) in Y wordt (e, r)
Tupel (o, g) in Z wordt (o, null)

Antwoord opgave 5

- a) De verschillende gevolgen voor de implementatie van een IS zijn dat schending van beschrijvende beperkingsregels niet geaccepteerd moet worden, terwijl schending van voorschrijvende beperkingsregels wel geaccepteerd moet kunnen worden. Toestanden waarin een voorschrift wordt overtreden, moeten door het IS dus wel degelijk gerepresenteerd kunnen worden! Bij een goede implementatie van een IS zal deze de overtreding niet alleen signaleren, maar ook corrigerende actie(s) op gang proberen te brengen.
- b) De referentiële integriteits-eis legt dan als beperking op dat de aanwezige foreign key waarden(combinaties):
 - ofwel 1) geheel null moeten zijn (d.w.z. elk attribuut van die foreign key moet de waarde null hebben),
 - ofwel 2) als sleutelwaarde voorkomen in een tupel van de relatie waarnaar de foreign key verwijst.

Anders geformuleerd:

De referentiële integriteits-eis legt dan als beperking op dat in de verwijzende relatie voor *elk* tupel geldt dat:

 - ofwel 1) de attributen A en B *beiden* de waarde **null** bevatten,
 - ofwel 2) de in de attributen A en B van dat tupel aanwezige waardencombinatie tevens moet voorkomen als sleutelwaarde van een tupel *aanwezig* in de verwezen relatie (m.a.w. de verwezen relatie moet dan tegelijkertijd een tupel bevatten met die waardencombinatie als waarde voor zijn sleutel).
- c) Extern niveau:
 - derived table, view definition language, view updatabilityConceptueel niveau:
 - logical database design, base tableIntern niveau:
 - storage definition language, hash, non-dense index, physical database design, partial index.

- 1) Intern
- 2) Conc.
- 3) Intern
- 4) Conc.
- 5) Ext.
- 6) Intern
- 7) Intern
- 8) Extern.
- 9) Intern
- 10) Ext.

1 2 3 4 5 6 7 8 9 10

Bekijk dan $AD \rightarrow F$.

Als $H = G - \{ AD \rightarrow F \}$, dan geldt: $(AD)^+_H = AD$ en $F \notin AD$. Dus $AD \rightarrow F$ is niet overbodig.

Bekijk nu $B \rightarrow A$.

Als $H = G - \{ B \rightarrow A \}$, dan geldt: $(B)^+_H = ABCDF$ en $A \in ABCDF$. Dus $B \rightarrow A$ is overbodig. Dus neem nu: $G := G - \{ B \rightarrow A \}$.

Het bekijken van de laatste 3 afhankelijkheden gaat analoog. Daarbij wordt alleen $B \rightarrow F$ weggelaten.

Dus eindresultaat: $\{ B \rightarrow C, AD \rightarrow F, C \rightarrow A, C \rightarrow D \}$

[Hier t.b.v. het nakijken ook nog eventuele alternatieve eindresultaten toevoegen!]