

Tentamen Databases I (29 augustus 2001)

Opgave 1:

- a) Geef een duidelijke omschrijving van de volgende begrippen:
 - entity integrity constraint
 - referential integrity constraint
 - intensie van een relatie
 - extensie van een relatie
- b) Geef van de volgende constraints aan of ze statisch dan wel dynamisch zijn, en of ze analytisch, empirisch of deontisch zijn:
 - i) De prijs van eengezinswoningen in Amsterdam-centrum ligt boven de 1 miljoen gulden.
 - ii) Huizen gekocht in IJburg dienen minstens 3 jaar in bezit van de koper te zijn, alvorens sprake kan zijn van doorverkoop. (Anti-speculatieclausule koopcontract.)
 - iii) Het tuinoppervlak van een woning is altijd niet-negatief.
- c) Gegeven een relatieschema $R = ABCDE$ met een minimal cover F , waarbij F geen enkele functionele afhankelijkheid bevat van de vorm " $X \rightarrow C$ " (met $X \subseteq ABCDE$). Welke conclusie kunnen we hieruit trekken wat betreft de key(s) van R ?
- d) Aan welke 4 gewenste eigenschappen dient een transactie te voldoen? Geef van iedere eigenschap zowel naam als omschrijving.

Opgave 2:

Fut Vliegzeizen is een touroperator gespecialiseerd in vliegarrangementen. De informatie die nodig is voor haar bedrijfsvoering kan als volgt worden omschreven:

- Een boeking (geïdentificeerd door een boekingsnummer) is gedaan op een bepaalde datum door een bepaalde klant (geïdentificeerd met een klantnummer) met bijbehorend bank- of girorekeningnummer. *Fut Vliegzeizen* heeft uitsluitend particulieren als klant.
- Iedere boeking heeft één of meerdere reisdeelnemers, waarbij elk van hen één of meerdere speciale wensen kan hebben (vegetariër, roker, kosher eten, ...) Degene die de reis geboekt heeft is niet noodzakelijk reisdeelnemer. Iedere reisdeelnemer heeft precies één contactpersoon die in geval van ongevallen of calamiteiten gewaarschuwd kan worden.
- Een boeking wordt gedaan voor één arrangement, geïdentificeerd met een catalogusnummer. Naast een catalogusnummer heeft ieder arrangement ook nog een bestemming, een begindatum, een einddatum, een luchtvaartmaatschappij en een hotel. Een uitzondering hierop is een *fly-drive* arrangement; deze heeft geen hotel maar wel een type auto (beschreven door merk en model). Voor ieder arrangement wordt tevens bijgehouden hoeveel vrije plaatsen er nog zijn.

Verder is gegeven dat van iedere relevante persoon naam, adres en telefoonnummer worden geregistreerd, waarbij de combinatie (naam, adres) uniek wordt geacht.

Maak een EER diagram waarin de bovenstaande informatiebehoefte is uitgedrukt. Gebruik hierbij de (min, max) notatie voor het aangeven van structurele constraints.

Opgave 3:

Gegeven is een database m.b.t. de reiswereld. De database bestaat uit de volgende 4 relaties:

- *Geïnteresseerd*(persoon, bestemming)
- *Klant*(persoon, reisbureau)
- *Levert*(reisbureau, touroperator)
- *Biedt*(touroperator, bestemming)

De relatie *Geïnteresseerd* geeft voor iedere persoon aan in welke bestemmingen hij/zij geïnteresseerd is.

De relatie *Klant* geeft aan bij welke reisbureaus een persoon klant is.

De relatie *Levert* geeft voor elk reisbureau aan bij welke touroperators er reizen kunnen worden geboekt.

De relatie *Biedt* geeft aan welke bestemmingen een touroperator aanbiedt.

Er is gegeven dat voor geen enkel attribuut null-waarden zijn toegestaan.

Formuleer de onderstaande vragen in de aangegeven relationele talen:

- a) Geef de bestemmingen van touroperators die geleverd worden door reisbureau "B-Reizen". (algebra, SQL)
- b) Geef de reisbureaus met één of meerdere klanten die wel naar IJsland, maar niet naar Ibiza willen. (tupelcalculus, SQL)
- c) Geef de touroperators die *alle* bestemmingen aanbieden waar persoon "Piet" in geïnteresseerd is. (algebra, domeincalculus)

Opgave 4:

Gegeven is een relationeel DBMS met daarin 4 relaties (W, X, Y en Z) met in totaal 8 attributen (A, B, C, D, E, F, G, H). Het domein van ieder attribuut bestaat uit de letters 'a' t/m 'z'. Null waarden zijn toegestaan voor attributen die niet tot de primary key behoren.

Relatie W(A, B) heeft

- (A, B) als PK (primary key)
- B als FK (foreign key) naar X (on delete: restricted, on update: restricted)

Relatie X(C, D) heeft

- C als PK
- D als FK naar Y (on delete: cascade, on update: cascade)

Relatie Y(E, F) heeft

- E als PK

Relatie Z(G, H) heeft

- G als PK en
- H als FK naar Y (on delete: nullify, on update: nullify)

We gaan uit van de volgende inhoud (extensie) van de database:

W: A B	X: C D	Y: E F	Z: G H
a d	d g	g l	o g
b e	e g	h l	p h
c e	f h	i m	q j
		j n	

Neem bij *elk* van de onderstaande opdrachten steeds weer opnieuw de hierboven gegeven database-inhoud als begintoestand. (m.a.w.: ga dus **niet** het antwoord van een opdracht gebruiken als invoer voor de volgende opdracht.)

Geef aan hoe de toestand van de database gewijzigd wordt als gevolg van de vermelde opdracht. Tabellen uitschrijven is niet nodig, je hoeft alleen aan te geven wat er verandert.

- Delete from X where X.C = f
- Delete from Y where Y.E = g
- Delete from Y where Y.E = h
- Modify from Y set Y.E = r where Y.E = h
- Modify from Y set Y.E = r where Y.E = g

Opgave 5:

Doe voor elk van de hieronder gegeven voorbeelden het volgende:

- Geef alle sleutels
- Geef (in de gegeven volgorde) voor elke functionele afhankelijkheid uit F aan of het een r_sleutelafhankelijkheid is (s), een r_partiële afhankelijkheid (p), een r_transitieve afhankelijkheid (t) of geen van deze drie (g).
NB: om schrijfwerk te voorkomen is het de bedoeling dat je de gegeven volgorde van de functionele afhankelijkheden in F aanhoudt en in die volgorde voor iedere afhankelijkheid een s, p, t of g opschrijft.
- Geef de hoogste normaalvorm waaraan het betreffende relatie-schema voldoet. Kies hierbij uit 1NF, 2NF, 3NF of BCNF.
- Indien het antwoord bij iii niet BCNF is, geef dan een functionele afhankelijkheid uit F die de eerstvolgende "hogere" normaalvorm schendt. (dus als je antwoord bij iii bijvoorbeeld 1NF is, geef dan een afhankelijkheid uit F die 2NF schendt; als je antwoord bij iii BCNF is, dan hoef je niets te doen.)

- R = ABCD en F = {AB → C, C → D}
- R = ABCD en F = {BC → A, BC → D, CD → B, ABC → D}
- R = ABCD en F = {BC → A, BC → D, D → B}
- R = ABCD en F = {A → D, B → C, C → D}

Normering

cijfer := aantal punten / 10

- opgave 1: 21 punten (resp. 4, 9, 4 en 4 punten)
- opgave 2: 20 punten
- opgave 3: 24 punten (resp. 8, 8 en 8 punten)
- opgave 4: 15 punten
- opgave 5: 20 punten (resp. 5, 5, 5 en 5 punten)