

Extra Tentamen Databases I (19 januari 2005)

Opgave 1

Flut Bedrijfsevenementen is een onderneming die gespecialiseerd is in het organiseren van bedrijfsfeestjes. Hiertoe heeft zij een groot aantal artiesten (met vaardigheden zoals mime, acrobatiek, goochelen, jongleren, ...) in haar bestand en ook (o.a. t.b.v. het sturen van uitnodigingen) informatie over werknemers van de bedrijven die klant zijn van Flut. Een deel van de artiesten werkt in loondienst, een ander deel werkt als freelancer. Een boeking wordt gedaan door een bedrijf en omvat een of meerdere artiesten die voor een bepaalde tijdsduur worden gereserveerd. Een bedrijfsfeest wordt primair georganiseerd voor de eigen werknemers, maar vaak zijn kinderen en echtgenotes ook welkom.

De informatiebehoefte omtrent Flut's bedrijfsvoering is uitgedrukt in het bijgevoegde EER-diagram (zie de laatste pagina).

Zet dit EER-diagram om naar een relationeel database ontwerp, i.e. een conceptueel schema van een relationele database.

Geef hierbij een zo volledig mogelijke specificatie, dus o.a. van de gebruikte relatienamen, attributen, sleutel(s) en eventuele verwijssleutels. Geef bij elke verwijssleutel ook expliciet de bijbehorende 'gedragsregel' (= referential triggered action) aan. Omdat ook constraints tot het conceptuele schema (en RDB ontwerp) behoren, moeten alle constraints uit het EER-diagram dus ook op een of andere manier tot uitdrukking komen in jouw conceptuele schema. Als een constraint niet 'impliciet' tot uitdrukking komt, moet je hem expliciet formuleren. Voor de expliciete formuleringen mag je je eigen 'syntax' kiezen, d.w.z. je mag gewoon Nederlands gebruiken en, indien gewenst, ook gebruik maken van symbolen en/of notaties uit de logica (zoals bijvoorbeeld: $\forall$, $\exists$ en $\in$). De enige eis is dat e.e.a. duidelijk is.

Opgave 2

Gegeven is een database uit de filmwereld met daarin de volgende drie relaties:

- Acteur (naam, jaar)
- Film (titel, regisseur, jaar)
- SpeeltIn (naam₁, titel₁) N.B. 'naam' is een FK naar 'Acteur', 'titel' is een FK naar 'Film'.

De relatie *Acteur* geeft de naam en het geboortjaar van filmacteurs.

De relatie *Film* geeft bij een film (geïdentificeerd door titel) aan wat de naam van de regisseur is en in welk jaar de film gemaakt is.

De relatie *SpeeltIn* geeft voor elke film in 'Film' aan welke acteurs er allemaal een rol in spelen. Verder is gegeven dat voor geen enkel attribuut null-waarden zijn toegestaan.

Formuleer de onderstaande opdrachten in **elk** van de aangegeven relationele talen:

- a) In SQL en in domeincalculus: "Geef de namen van regisseurs die wel eens in hun eigen film gespeeld hebben".
- b) In relationele algebra en in tupelcalculus: "Geef de naam en het geboortjaar van elke acteur die een rol speelt in een film van regisseur Lee Tomahori".
- c) In SQL en in tupelcalculus: "Geef de namen van alle acteurs die meespelen in één of meerdere films gemaakt vóór 1950, maar in geen enkele latere film".

Opgave 3

Gegeven een relatieschema $R = ABCDEF$ met bijbehorende verzameling functionele afhankelijkheden:

$$G = \{ AB \rightarrow C, AD \rightarrow F, B \rightarrow A, BE \rightarrow F, C \rightarrow AD \}.$$

Bepaal een minimal cover voor G . Geef als antwoord het (tussen)resultaat na elk van de hoofdstappen van de berekening.

Opgave 4

N.B.: O.a. om het beoordelen te vergemakkelijken **moet** (!) je bij deze opgave gebruik maken van exact de **volgorde** waarin de decompositie en functionele afhankelijkheden gegeven zijn!

Ga met behulp van het daartoe bedoelde algoritme (uit boek en dictaat) na of de lossless join eigenschap wel of niet geldt voor de decompositie (**AB**, **AC**, **BD**, **CE**) van het relatieschema $ABCDE$ met de zes functionele afhankelijkheden:

$$1) A \rightarrow D, \quad 2) DE \rightarrow C, \quad 3) E \rightarrow D, \quad 4) AD \rightarrow E, \quad 5) C \rightarrow E, \quad 6) B \rightarrow C.$$

Geef niet alleen het eindantwoord maar laat tevens zien welke stappen worden doorlopen.

Doe dit door:

- (i) de beginsituatie weer te geven (als stap 1)
- (ii) alle volgende stappen oplopend te nummeren
- (iii) bij elke genummerde vervolgstap duidelijk zichtbaar aan te geven *of* er iets verandert en zo ja, *wat* er in die stap allemaal veranderd is.

Beperk s.v.p. jullie schrijfwerk (en ons nakijkwerk) door het kiezen van een geschikte conventie. Alles mag, zolang het maar duidelijk is, d.w.z. eenduidig en snel leesbaar.

Denk om de gegeven volgorde van de decompositie en van de afhankelijkheden!

Opgave 5

- a) Geef kort aan welke verschillende gevolgen beschrijvende en voorschrijvende beperkingsregels (descriptive vs. prescriptive constraints) hebben voor de implementatie van een informatiesysteem.
- b) Als een foreign key bestaat uit (de combinatie van) twee attributen, zeg attributen A en B, wat voor beperking legt de referentiële integriteits-eis dan op?
- c) De ANSI/SPARC architectuur bestaat uit drie niveaus: het externe niveau, het conceptuele niveau en het interne niveau. Geef voor ieder van de navolgende 10 termen aan op welk niveau die term het meest van toepassing is:
 - 1) storage definition language, 2) logical database design, 3) hash, 4) base table, 5) derived table, 6) non-dense index, 7) physical database design, 8) view definition language, 9) partial index, 10) view updatability.

Normering

(Nagekeken werk komt ter inzage bij het onderwijsbureau)

cijfer := aantal punten / 10 (N.B. cijfers tussen 5 en 6 worden afgerond naar 5,0 of 6,0)

- opgave 1: 20 punten
- opgave 2: 28 punten (8, 10 en 10 punten)
- opgave 3: 20 punten
- opgave 4: 15 punten
- opgave 5: 17 punten (resp. 8, 4 en 5 punten)