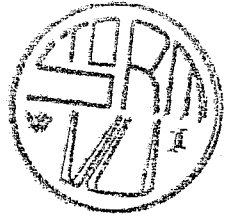


Tentamen Databases I (4 juni 2002)



Opgave 1:

- a) Geef een duidelijke omschrijving van de volgende begrippen:
- * superkey
 - * key
 - * alternate key
- b) Geef voor (1) het EER model en (2) het relationele model aan tot welk *type* model het behoort. Kies hierbij uit (i) high level model, (ii) low level model en (iii) representational model.
- c) Het bedrijf *Filminfo* is uitgever van een database op CDROM die als een zo compleet mogelijk naslagwerk dient wat betreft de films die in de 20^e eeuw zijn uitgebracht. De database – die onder andere informatie bevat over films, acteurs en studio's – is bedoeld voor *read-only* gebruik (de gebruikers voeren geen updates uit). Voor het ontwerp van de database dient een decompositie te worden gemaakt in één of meerdere tabellen. Deze decompositie kan voldoen aan (één of meerdere van) de volgende eigenschappen:
- (1) een zo hoog mogelijke normaalvorm
 - (2) de lossless-join eigenschap
 - (3) de dependency preserving eigenschap
- Geef aan wat voor de betreffende database de *minst* belangrijke eigenschap is. Geef tevens aan waarom het de minst belangrijke eigenschap is.

Opgave 2:

Classico Records is een platenmaatschappij op het gebied van klassieke muziek. Voor haar bedrijfsvoering is de volgende informatie van belang:

- * Een componist heeft een naam en geboortedatum, waarbij de naam uniek wordt geacht
- * Een componist heeft tijdens zijn leven één of meerdere composities geschreven; bij een compositie hoort echter slechts één componist. Composities hebben een naam, waarbij het kan voorkomen dat verschillende composities van verschillende componisten dezelfde naam hebben ("De Vier Jaargetijden" van Haydn is bijvoorbeeld iets anders dan "De Vier Jaargetijden" van Vivaldi). Composities van dezelfde componist hebben echter altijd verschillende namen.
- * Een opname, is gerelateerd aan een compositie en een uitvoerende. Een opname heeft een lengte (= tijdsduur) alsmede een datum van opname en wordt uniek geïdentificeerd door een opname nummer. Een opname betreft precies één compositie en heeft precies één uitvoerende. Deze uitvoerende is ofwel een orkest, ofwel een solist. In het eerste geval wordt de naam van het orkest, de naam van de dirigent en de namen van de overige orkestleden bijgehouden, in het tweede geval wordt de naam van de solist bijgehouden. Twee verschillende orkesten hebben altijd verschillende namen; hetzelfde geldt voor solisten. Iedere uitvoerende heeft tevens één bank- of gironummer (deze is voor iedere uitvoerende verschillend) en een commercieel manager.
- * Een verzamel-CD bevat één of meerdere opnames, waarbij een opname eventueel voor meerdere verzamel-CD's kan worden gebruikt. Een verzamel-CD heeft een adviesprijs, een unieke bestelcode en een unieke EAN-code (= "streepjescode")

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

Opgave 3:

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

- * Acteur(naam, jaar)
- * Film(titel, regisseur, jaar)
- * SpeeltIn(naam, titel) (naam) FK naar Acteur, (titel) FK naar Film

De relatie *Acteur* geeft bij een bepaalde acteur (naam) aan wat zijn of haar geboortejaar (jaar) is. De relatie *Film* geeft bij een bepaalde film (titel) aan wat de naam van de regisseur is (regisseur) alsmede het jaar waarin de film gemaakt is (jaar).

De relatie *SpeeltIn* geeft voor elke acteur aan in welke films hij of zij een rol heeft gespeeld.

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

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

- a) Geef van elke acteur die een rol speelt in een film van regisseur "Steven Spielberg" zijn of haar naam en geboortejaar (algebra, SQL).
- b) Geef de titels van films waar zowel "Tom Cruise" als "Nicole Kidman" in meespelen (tupelcalculus, SQL)
- c) Geef de titels van films waarin iedere acteur voor 1920 is geboren (algebra, domeincalculus)

Opgave 4:

Ga met behulp van het daartoe bedoelde algoritme (uit boek en dictaat) na of de lossless join eigenschap geldt voor de decompositie (AE, AC, BCD, DE) van het relatie-schema ABCDE met de functionele afhankelijkheden: $\{D \rightarrow E, A \rightarrow B, C \rightarrow B, BC \rightarrow D\}$

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

Doe dit door:

- (i) de beginsituatie weer te geven
- (ii) bij elke stap aan te geven of er iets verandert en zo ja, *wat* er verandert (verwerk de verandering niet alleen in de tabel, maar beschrijf ook in woorden wat er verandert)

Je hoeft niet bij iedere stap een nieuwe tabel op te schrijven (schrijf alleen een nieuwe tabel op wanneer de oude tabel door de vele wijzigingen onduidelijk dreigt te worden).

Om het beoordelen te vergemakkelijken **moet** (!) je bij deze opgave gebruik maken van de volgorde waarin de decompositie en functionele afhankelijkheden gegeven zijn.

Opgave 5

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'. Nullwaarden 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 modify: restricted)

Relatie X(C, D) heeft

- C als PK
- D als FK naar Y (on delete: cascade, on modify: 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 modify: 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 en ook niet gewenst; je moet alleen aangeven 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 in X set X.D = j where X.C = f
- Modify in X set X.C = r where X.C = e
- Modify in Y set Y.E = r where Y.E = g

Normering

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

- opgave 1: 17 punten (resp. 8 en 9 punten)
- opgave 2: 24 punten (resp. 8, 8 en 8 punten)
- opgave 3: 21 punten
- opgave 4: 20 punten
- opgave 5: 18 punten (resp. 3, 3, 3, 3, 3 en 3 punten)

