TECHNISCHE UNIVERSITEIT DELFT
Basiseenheid Data en Kennis Systemen
Mekelweg 4
2628 CD DELFT

Deeltoets Databases voor tb141
8 juni 2001
tijdsduur: 50 min.

Gewichtsfactoren: Opgave 1: 3; Opgave 2: 2; Opgave 3: 3.

Opgave 1: Assemblagefabriek

Een assemblagefabriek gebruikt ten behoeve van de productie van verschillende apparaten een database. De typen apparaten worden van elkaar onderscheiden door een typenummer. Daarnaast wordt van ieder type de typenaam en de modelnaam vastgelegd.

Van elk type apparaat worden keer op keer in een serie een aantal exemplaren geassembleerd. Alle exemplaren van een serie worden voorzien van hetzelfde serienummer en dezelfde productiedatum. Een serienummer wordt daarna niet meer gebruikt bij andere series van hetzelfde type apparaat; eenzelfde serienummer kan echter wel bij verschillende typen apparaten voorkomen.

Om een serie te kunnen fabriceren is het nodig te weten uit welke standaardonderdelen en in welke aantallen een type apparaat zou moeten bestaan. Van ieder onderdeel zijn de vaste leverancier en voorraad van belang. Daarnaast wordt vastgelegd in welke typen het onderdeel als fractie kan voorkomen. Hierbij wordt het benodigde aantal ook vastgelegd.

Indien bij de fabricage van een serie een bepaald onderdeel inmiddels verouderd is of niet voorradig is, kan een ander onderdeel worden gebruikt. In de database wordt vastgelegd welk onderdeel als vervanging van het oorspronkelijke onderdeel in een type dienst kan doen.

Uiteindelijk zijn alle exemplaren van een serie samengesteld uit dezelfde onderdelen in dezelfde aantallen. De database bevat deze gegevens over de samenstelling van een apparaat.

Voor deze database is het volgende relationele model ontworpen:

type	(type#, typenaam, modelnaam)
serie	(serie#, type#, produktie_datum, aantal)
onderdeel	(onderdeel#, leverancier, vooraad)
fractie	(onderdeel#, type#, aantal)
vervanging	(onderdeel#, type#, nieuw_onderdeel#)
samenstelling	(serie#, type#, onderdeel#, aantal)

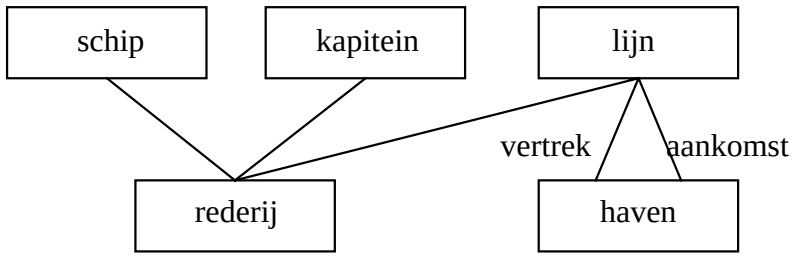
Opdracht: Voeg alle primary keys en foreign keys toe overeenkomstig bovenstaande beschrijving zodat het model voldoet aan de eisen van BCNF.

Opgave 2: Lijnvaart

Bij een overkoepelende organisatie van lijnvaartrederijen wordt van elke rederij bijgehouden welke schepen zij bezit, op welke lijnen zij vaart en welke personeelsleden als kapitein optreden. Een heenvaart is een andere lijn dan de terugvaart. Daartoe is het volgende relationele model ontworpen, waarin iedere candidate key tevens primary key is:

rederij (rederijnaam, vestigingsplaats)
schip (schipnaam, rederijnaam, capaciteit)
lijn (lijnnaam, rederijnaam, vertrek_havennaam, aankomst_havennaam)
kapitein (familienaam, geboortedatum, rederijnaam)
haven (havennaam, landnaam, liggeld)

Hiermee correspondeert de volgende abstractie-hierarchie:



Beantwoord de volgende vier vragen over dit model:

2.1. Bij elke rederij behoren één of meer schepen.
Deze situatie wordt in het model:
A. afgedwongen.
B. toegestaan, maar niet afgedwongen.
C. uitgesloten.

2.2. Eenzelfde lijnnaam mag door één rederij gebruikt worden.
Deze situatie wordt in het model:
A. afgedwongen.
B. toegestaan, maar niet afgedwongen.
C. uitgesloten.

2.3. Vertrekhaven en aankomsthaven van een lijn zijn verschillende havens.
Deze situatie wordt in het model:
A. afgedwongen.
B. toegestaan, maar niet afgedwongen.
C. uitgesloten.

2.4 Een kapitein kan bij twee rederij werken.
Deze situatie wordt in het model:
A. afgedwongen.
B. toegestaan, maar niet afgedwongen.
C. uitgesloten.

Na enige tijd blijkt het systeem te moeten worden aangepast. Van elke lijn wordt bijgehouden wanneer een afvaart is, welk schip hiervoor gebruikt wordt, wie hierbij de kapitein is en hoe groot het aantal passagiers is. Schepen worden met wisselende kapiteins op verschillende lijnen van de eigen rederij ingezet. Een kapitein kan hoogstens eenmaal per dag bij twee afvaarten (de heen- en terugvaart) betrokken zijn. Schepen kunnen voor meer dan een afvaart per dag worden ingezet. Het relationele model wordt daartoe uitgebreid met de volgende onvolledige relatie:

Afvaart (....., Datum, Aantal_passagiers)

De ontbrekende attributen ..... hebben betrekking op bestaande relaties.

2.5 Ervan uitgaande dat de relatie Afvaart in BCNF staat, uit hoeveel attributen moet de primary key van Afvaart bestaan?

A. 1
B. 2
C. 3
D. 4
E. 5
F. 6

2.6 Ervan uitgaande dat de relatie Afvaart in BCNF staat, hoeveel foreign keys heeft de relatie Afvaart?

A. 1
B. 2
C. 3
D. 4
E. 5
F. 6

Opgave 3: Ziekenhuis

Een ziekenhuis gebruikt voor de administratie van poliklinische behandelingen van patienten een database. Een patient komt onder behandeling van een of meer specialisten. Gedurende de periode van behandeling kan de specialist verschillende onderzoeken voorschrijven. Een uitgevoerd onderzoek is altijd van een bepaalde soort. Tijdens een behandeling kunnen medicijnen worden voorgeschreven. Voor ieder medicijn afzonderlijk wordt een recept verstrekt.

Het model van de database bestaat uit de volgende relaties in BCNF:

specialist	(spec#, naam, specialisme)
patient	(patient#, naam, adres, plaats, geb_datum, geslacht)
behandeling	(spec#, patient#, begin_datum)
soort	(soort#, naam, beschrijving)
onderzoek	(spec#, patient#, soort#, datum)
recept	(spec#, patient#, medicijn, datum, dosis)

Voor een tweetal vraagstellingen worden denkbare SQL-formuleringen gegeven. Bij elke vraag geldt dat het aantal juiste alternatieven nul of meer kan zijn. Geef van elk van de gegeven alternatieven aan of het juist of onjuist is. Hierbij hoeft geen rekening te worden gehouden met het eventueel voorkomen van duplicaten in het query-resultaat.

3.1. Geef de nummers van de patienten die meer dan driemaal eenzelfde soort onderzoek hebben ondergaan.

A.
```
SELECT patient#
FROM onderzoek OX
WHERE 3 < SELECT COUNT(*)
          FROM onderzoek
          WHERE onderzoek.soort# = OX.soort#
          AND onderzoek.patient# = OX.patient#;
```

B.
```
SELECT patient#
FROM onderzoek OX
WHERE 3 < SELECT COUNT(*)
          FROM onderzoek
          WHERE onderzoek.patient# = OX.patient#
          GROUP BY soort#;
```

C.
```
SELECT patient#
FROM onderzoek
GROUP BY patient#, soort#
HAVING COUNT(soort#) > 3;
```

D.
```
SELECT patient#
FROM onderzoek
GROUP BY patient#
HAVING COUNT(soort#) > 3;
```

3.2 Geef de namen van de specialisten die nog nooit het medicijn xynol hebben voorgeschreven.

A.
```
SELECT naam
FROM specialist
WHERE spec# NOT IN SELECT spec#
                   FROM recept
                   WHERE medicijn = 'xynol';
```

B.
```
SELECT naam
FROM specialist
WHERE NOT EXISTS SELECT *
                 FROM recept
                 WHERE medicijn = 'xynol'
                 AND spec# = specialist.spec#;
```

C.
```
SELECT naam
FROM specialist
WHERE spec# IN SELECT spec#
               FROM recept RX
               WHERE 'xynol' NOT IN SELECT medicijn
                                    FROM recept
                                    WHERE spec# = RX.spec#;
```

D.
```
SELECT naam
FROM specialist, recept
WHERE specialist.spec# = recept.spec#
AND spec# NOT IN SELECT spec#
                 FROM recept
                 WHERE medicijn = 'xynol';
```

========== **EINDE DEELTOETS** ==========