TECHNISCHE UNIVERSITEIT DELFT
Sectie Database Systemen
Zuidplantsoen 4
2628 BZ DELFT

Deeltoets Databases voor tb141
21 mei 1999
tijdsduur: 50 min.

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

Opgave 1: Casus Prakticumbeheer

T.b.v. de praktica in een studieprogramma is een database ontworpen. Een prakticum heeft een nummer, een naam en een verantwoordelijk docent. Voor elk prakticum is vastgelegd hoeveel opgaven uit de betreffende verzameling door een prakticant moeten worden uitgevoerd. Een opgave heeft een nummer en omschrijving; het nummer is uniek binnen een bepaald prakticum.
Wanneer een student zich voor een prakticum inschrijft, worden de naam en het studienummer van de student genoteerd. Tevens wordt de inschrijfdatum vastgelegd. Alle opdrachten die de student moet uitvoeren worden in één keer uitgereikt. Geregistreerd wordt welke opgaven dit zijn.
Sommige opgaven vereisen een uitwerking van de student. Als een student zo'n opgave klaar heeft, legt hij de uitwerking voor ter beoordeling aan een student-assistent. Bij afkeuring wordt niets vastgelegd. Bij goedkeuring wordt behalve de datum ook genoteerd welke assistent de betreffende uitwerking heeft goedgekeurd.
Andere opgaven bestaan uit het maken van een verslag. Als een verslag door een student wordt ingeleverd, wordt de inleverdatum vastgelegd en wordt ook genoteerd bij welke assistent de inlevering plaatsvond. Als de assistent het verslag heeft nagekeken en het als onvoldoende beoordeelt, stuurt hij het naar de prakticant terug, die het na verbetering weer bij dezelfde assistent kan inleveren. Zowel de datum van terugzenden als van opnieuw inleveren worden vastgelegd. Als het verslag (evt. na verbetering) wordt geaccepteerd, wordt de acceptatiedatum genoteerd, een cijfer vastgelegd (kan een onvoldoende zijn) en wordt het verslag ingehouden.
Er is voor iedere assistent vastgelegd voor welke praktica hij/zij deze bevoegdheid bezit. Een dergelijke bevoegdheid geldt dan voor het gehele prakticum. Van een student-assistent zijn verder opgenomen: studienummer, naam, datum van in diensttreding en ervaring.

Voor deze database is het volgende relationele model ontworpen:

prakticum	(pr#, naam, docent, aantal_opgaven)
opgave	(pr#, opg#, omschrijving, verslag_indicatie)
prakticant	(stud#, pr#, naam, inschr_datum)
opdracht	(stud#, pr#, opg#)
uitwerking	(stud#, pr#, opg#, assistent_stud#, datum)
verslag	(stud#, pr#, opg#, assistent_stud#, in_datum, uit_datum, in2_datum, acceptatie_datum, cijfer)
assistent	(stud#, naam, datum, ervaring)
bevoegdheid	(stud#, pr#)

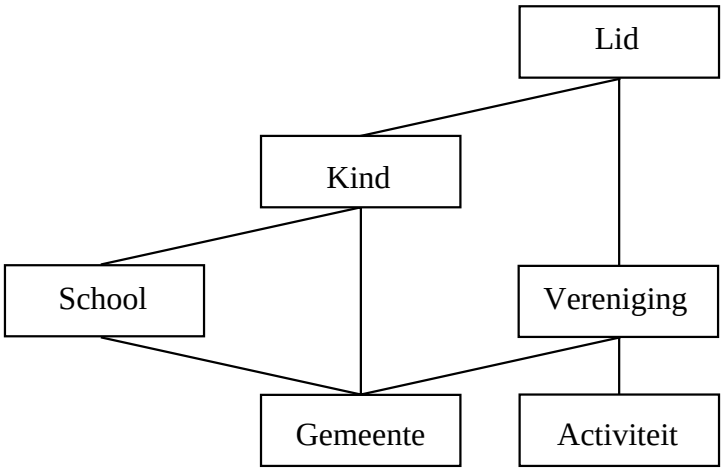
Opdracht: Voeg alle primary keys en foreign keys toe overeenkomstig bovenstaande beschrijving zodat het model voldoet aan de eisen van BCNF.

Opgave 2: Casus Gemeentes

Er is een registratie van gemeentes en hun schoolgaande inwoners. Kinderen wonen in een gemeente en gaan naar school. Scholen liggen ook in een van de geregistreerde gemeentes. Kinderen zijn lid van verenigingen die bepaalde activiteiten ontplooien (bijv. voetbal, scouting). Verenigingen horen ook bij een bepaalde gemeente. Het volgende relationele model in BCNF is voor deze situatie ontworpen. Hierin zijn primary keys onderstreept en foreign keys onderstippeld.

Gemeente	(Gnr, gemeentenaam, provincie)
School	(Snr, schoolnaam, adres, postcode, gemeentenaam, Gnr)
Kind	(Knr, naam, adres, Gnr, Snr)
Vereniging	(Vnr, verenigingsnaam, adres, Gnr, Anr)
Activiteit	(Anr, omschrijving)
Lid	(Knr, Vnr)

Met dit relationele model correspondeert de volgende abstractie-hierarchie.



Beantwoord de volgende vier vragen over dit model:

2.1. Uit dit model volgt dat ieder kind in de eigen gemeente naar school gaat.
Is deze bewering juist?
A Ja **B** Nee

2.2. Het aantal kinderen dat lid is van een vereniging in de eigen gemeente is afleidbaar.
Is deze bewering juist?
A Ja **B** Nee

2.3. Uit dit model volgt dat in iedere gemeente tenminste één school ligt.
Is deze bewering juist?
A Ja **B** Nee

2.4 Het attribuut gemeentenaam in de relatie school is redundant.
Is deze bewering juist?
A Ja **B** Nee

Opgave 3: Casus Scholengemeenschap

Een scholengemeenschap verzorgt drie soorten onderwijs, nl. mavo, havo en vwo. Ten einde overzicht te kunnen houden over bepaalde gegevens wordt een relationeel DBMS gebruikt dat gebaseerd is op o.a. de volgende relaties:

docent	(d#, dnaam, adres, plaats, telefoon, bevoegdheid, hoogste_opleiding, indienst_datum, salaris)
vak	(v#, vnaam)
klas	(kl#, soort, schooljaar, d#-mentor)
leerling	(l#, naam, adres, plaats, telefoon, geboorte_datum)
plaatsing	(l#, kl#, soort, schooljaar)
cursus	(d#, kl#, soort, schooljaar, v#, uren)

Elke onderwijssoort (in de definitie hierboven: soort) heeft een aantal klassen. Per onderwijssoort komt een bepaald klasniveau (kl#) maar één keer voor. Een schooljaar wordt aangeduid met het jaartal waarin het schooljaar begint. Sommige docenten zijn mentor. Voor een aantal vraagstellingen worden denkbare SQL-formuleringen gegeven. Bij elke vraag geldt dat het aantal goede alternatieven **nul of meer** kan zijn.

Geef **alle juiste** alternatieven aan.

3.1. Geef naam, adres en plaats van de leerlingen die een klas gedubbeld hebben.

```
A.    SELECT naam, adres, plaats
      FROM leerling
      WHERE 2 = SELECT COUNT (*)
                FROM plaatsing
                WHERE plaatsing.l# = leerling.l#

B.    SELECT naam, adres, plaats
      FROM leerling
      WHERE l# IN SELECT l#
                  FROM plaatsing : plx
                  WHERE 2 = SELECT COUNT (*)
                            FROM plaatsing
                            WHERE plx.l# = plaatsing.l#
                            AND plx.soort = plaatsing.soort
                            AND plx.kl# = plaatsing.kl#

C.    SELECT naam, adres, plaats
      FROM leerling
      WHERE l# IN SELECT l#
                  FROM plaatsing
                  GROUP BY l#, kl#, soort
                  HAVING COUNT (*) = 2

D.    SELECT naam, adres, plaats
      FROM leerling, plaatsing
      WHERE leerling.l# = plaatsing.l#
      AND 2 = SELECT COUNT (*)
              FROM plaatsing, leerling, klas
              WHERE plaatsing.l# = leerling.l#
              AND plaatsing.kl# = klas.kl
              AND plaatsing.soort = klas.soort
```

3.2. Geef een overzicht van leerlingenaantallen per onderwijssoort voor het schooljaar 1997 en noem de onderwijssoort.

```
A.    SELECT COUNT (l#), soort
      FROM plaatsing
      GROUP BY soort
      HAVING schooljaar = 1997

B.    SELECT soort, COUNT (l#)
      FROM plaatsing
      WHERE schooljaar = 1997
      GROUP BY soort

C.    SELECT soort, COUNT (kl#)
      FROM plaatsing
      WHERE schooljaar = 1997
      GROUP BY soort

D.    SELECT soort, COUNT (*)
      FROM plaatsing
      WHERE schooljaar = 1997
      GROUP BY soort, kl#
```

3.3. Geef docentnaam van de docenten die alleen les geven aan vwo-klassen.

```
A.    SELECT dnaam
      FROM docent
      WHERE d# IN SELECT d#
                  FROM cursus
                  WHERE soort = 'vwo'

B.    SELECT dnaam
      FROM docent, cursus c1
      WHERE docent.d# = c1.d#
      AND soort = 'vwo'
      AND NOT EXISTS SELECT *
                     FROM cursus c2
                     WHERE soort <> 'vwo'
                     AND c2.d# = docent.d#

C.    SELECT dnaam
      FROM docent, cursus
      WHERE docent.d# = cursus.d#
      AND soort = 'vwo'
      AND soort <> 'havo'
      AND soort <> 'mavo'

D.    SELECT dnaam
      FROM docent
      WHERE d# IN SELECT d#
                  FROM cursus
                  WHERE soort = 'vwo'
                  AND d# NOT IN SELECT d#
                                FROM cursus
                                WHERE soort <> 'vwo'
```

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