

Opgave: Prakticumbeheer

T.b.v. de praktica in een studieprogramma wordt een database ontworpen. Een practicum heeft een nummer, een naam en een verantwoordelijk docent. Voor elk practicum is vastgelegd hoeveel opgaven uit de betreffende verzameling door een practicant moeten worden uitgevoerd. Een opgave heeft een nummer en omschrijving; het nummer is uniek binnen een bepaald practicum.

Wanneer een student zich voor een practicum 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 practicant 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 practicum. Van een student-assistent zijn verder opgenomen: studienummer, naam, datum van in diensttreding en ervaring.

Opdracht:

Ontwerp voor deze database een relationeel model dat voldoet aan BCNF. Geef alle primary keys en foreign keys aan. Extra aannames moeten gemotiveerd worden.

Antwoordindicatie:

practicum	(<u>pr#</u> , naam, docent, aantal_opgaven)
opgave	(<u>pr#</u> , <u>opg#</u> , omschrijving, verslag_indicatie)
practicant	(<u>stud#</u> , <u>pr#</u> , naam, inschr_datum)
opdracht	(<u>stud#</u> , <u>pr#</u> , <u>opg#</u>)
uitwerking	(<u>stud#</u> , <u>pr#</u> , <u>opg#</u> , <u>assistent_stud#</u> , datum)
verslag	(<u>stud#</u> , <u>pr#</u> , <u>opg#</u> , <u>assistent_stud#</u> , in_datum, uit_datum, in2_datum, acceptatie_datum, cijfer)
assistent	(<u>stud#</u> , naam, datum, ervaring)
bevoegdheid	(<u>stud#</u> , <u>pr#</u>)

Opgave: 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	(<u>d#</u> , dnaam, adres, plaats, telefoon, bevoegdheid, hoogste_opleiding, indienst_datum, salaris)
vak	(v#, vnaam)
klas	(<u>kl#</u> , soort, schooljaar, <u>d#-mentor</u>)
leerling	(<u>l#</u> , naam, adres, plaats, telefoon, geboorte_datum)
plaatsing	(<u>l#</u> , <u>kl#</u> , <u>soort</u> , <u>schooljaar</u>)
cursus	(<u>d#</u> , <u>kl#</u> , <u>soort</u> , <u>schooljaar</u> , <u>v#</u> , 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.

Opdracht: Geef van elk van de gegeven alternatieven aan of het juist of onjuist is.

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 AS 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'
```

Antwoordindicatie:

- | | | | | |
|-----|------------|----------|------------|------------|
| 3.1 | A: Onjuist | B: Juist | C: Juist | D: Onjuist |
| 3.2 | A: Onjuist | B: Juist | C: Onjuist | D: Onjuist |
| 3.3 | A: Onjuist | B: Juist | C: Onjuist | D: Juist |

Opgave 1: Casus Ontwikkelingshulp

Het Ministerie van Ontwikkelingssamenwerking subsidieert een aantal projecten in ontwikkelingslanden. Men wil een database opzetten met gegevens over landen, projecten en vrijwilligers.

Elk land krijgt een unieke code. Het ministerie heeft in elk land een vertegenwoordiging of een ambassade waarvan het adres en de plaats wordt vastgelegd. Van vrijwilligers worden naam, geboortedatum, adres en hoogste genoten opleiding geregistreerd. Aan iedere vrijwilliger wordt een identificatie toegekend. Vrijwilligers kunnen interesse hebben voor uitzending voor een of meer landen. De mate van voorkeur voor de betreffende landen wordt vastgelegd. De voorkeur voor een land wordt per vrijwilliger geregistreerd, ongeacht de in dat land lopende projecten.

Een vrijwilliger kan achtereenvolgend aan verschillende projecten deelnemen. De deelname van een vrijwilliger aan een project wordt vastgelegd, evenals de begin- en einddatum. Een vrijwilliger kan verschillende keren deelnemen aan hetzelfde project. Van projecten, die een uniek nummer hebben, wordt bijgehouden de naam, het land en de projectcoördinator.

Het kan voorkomen dat vrijwilligers nog een of meer stages moeten doorlopen als voorbereiding op de uitzending naar een ontwikkelingsland. Een stage heeft betrekking op een vakgebied, een vrijwilliger, heeft een begin- en einddatum en wordt begeleid door een begeleider die ook een beoordeling van de stage geeft. Elk vakgebied heeft een nummer, naam en een stagecoördinator. Van coördinatoren worden identificatie, naam en adres geregistreerd. Coördinatoren behoren tot de vaste kern van medewerkers, stagebegeleiders daarentegen niet.

Een stagebegeleider is dat voor slechts een vakgebied en heeft een eigen identificatie. Daarnaast worden van een stagebegeleider naam en adres bijgehouden. Er kunnen per vakgebied verschillende stagebegeleiders zijn. Vrijwilligers treden niet op als coördinator of als begeleider.

Opdracht:

Ontwerp voor deze database een relationeel model dat voldoet aan BCNF. Geef alle primary keys en foreign keys aan. Extra aannames moeten gemotiveerd worden.

Antwoordindicatie:

land	(<u>land#</u> , ambassade_adres, plaats, landnaam)
vrijwilliger	(<u>vrijwilliger#</u> , naam, geboortedatum, adres, opleiding)
medewerker	(<u>medewerker#</u> , naam, adres)
project	(<u>project#</u> , <u>land#</u> , projectnaam, <u>coördinator</u>)
interesse	(<u>vrijwilliger#</u> , <u>land#</u> , mate)
vakgebied	(<u>vakgebied#</u> , vakgebiednaam, <u>coördinator</u>)
deelname	(<u>project#</u> , <u>vrijwilliger#</u> , <u>begindatum</u> , einddatum)
stage	(<u>vrijwilliger#</u> , <u>begeleider#</u> , begindatum, einddatum)
begeleider	(<u>begeleider#</u> , naam, adres, <u>vakgebied#</u>)

Opgave 3: Casus Warenhuis

Een warenhuis gebruikt een database waarvan het relationele model luidt:

afdeling (afd#, afdnaam, chef, etage)

artikel (art#, artnaam, type)

firma (f#, fnaam, adres)

bestelling (f#, afd#, art#, datum, hoeveelheid)

De betekenis van de relaties afdeling, artikel en firma spreekt voor zichzelf. Relatie bestelling geeft aan dat een afdeling op een bepaalde datum een artikel in een zekere hoeveelheid bij een firma heeft besteld.

Voor een aantal vraagstellingen worden denkbare SQL-formuleringen gegeven. Bij elke vraag geldt dat het aantal goede alternatieven **nul of meer** kan zijn.

Opdracht: Geef van elk van de gegeven alternatieven aan of het juist of onjuist is.

3.1. Geef namen van de firma's waarbij alle artikelen van het type speelgoed zijn besteld.

- A.**
- ```
SELECT fnaam
FROM firma
WHERE NOT EXISTS SELECT *
 FROM artikel
 WHERE type = 'Speelgoed'
 AND NOT EXISTS SELECT art#
 FROM bestelling
 WHERE f# = firma.f#;
```
- B.**

```
SELECT fnaam
FROM firma
WHERE NOT EXISTS SELECT art#
 FROM artikel
 WHERE type = 'Speelgoed'
 AND art# NOT IN SELECT art#
 FROM bestelling
 WHERE f# = firma.f#;
```

**C.**

```
SELECT fnaam
FROM firma
WHERE NOT EXISTS SELECT *
 FROM artikel, bestelling
 WHERE type = 'Speelgoed'
 AND artikel.art# = bestelling.art#
 AND f# NOT IN SELECT f#
 FROM bestelling;
```

**D.**

```
SELECT fnaam
FROM firma
WHERE NOT EXISTS SELECT *
 FROM artikel
 WHERE type = 'Speelgoed'
 AND art# NOT IN SELECT art#
 FROM bestelling
 WHERE f# = firma.f#;
```

**3.2.** Geef namen van de afdelingen die geen bestelling bij de firma Krop hebben geplaatst.

- A.**

```
SELECT afdnaam
FROM afdeling
WHERE afd# NOT IN SELECT bestelling.afd#
 FROM bestelling, firma
 WHERE bestelling.f# = firma.f#
 AND fnaam = 'Krop';
```
- B.**

```
SELECT afdnaam
FROM afdeling, bestelling
WHERE afdeling.afd# = bestelling.afd#
AND NOT EXISTS SELECT bestelling.afd#
 FROM bestelling, firma
 WHERE bestelling.f# = firma.f#
 AND fnaam = 'Krop';
```
- C.**

```
SELECT afdnaam
FROM afdeling, bestelling
WHERE afdeling.afd# = bestelling.afd#
AND NOT EXISTS SELECT *
 FROM bestelling, firma
 WHERE bestelling.f# = firma.f#
 AND firma.afd# = afdeling.f#
 AND fnaam = 'Krop';
```
- D.**

```
SELECT afdnaam
FROM afdeling, bestelling
WHERE afdeling.afd# = bestelling.afd#
AND afdeling.afd# NOT IN SELECT afd#
 FROM bestelling
 WHERE fnaam = 'Krop';
```

**3.3.** Geef firma-nummers van de firma's waar in totaal meer dan 1000 exemplaren van eenzelfde artikel besteld zijn.

- A.**

```
SELECT f#
FROM firma, bestelling
WHERE firma.f# = bestelling.f#
AND SUM(hoeveelheid) > 1000
GROUP BY firma.f#, art#;
```
- B.**

```
SELECT f#
FROM bestelling
GROUP BY f#, art#
HAVING SUM(hoeveelheid) > 1000;
```
- C.**

```
SELECT firma.f#
FROM firma, bestelling
WHERE firma.f# = bestelling.f#
GROUP BY firma.f#, art#
HAVING SUM(hoeveelheid) > 1000;
```
- D.**

```
SELECT firma.f#
FROM firma, bestelling
GROUP BY firma.f#, art#
HAVING SUM(hoeveelheid) > 1000;
```

**Antwoordindicatie:**

- 3.1 A: Onjuist B: Juist C: Onjuist D: Juist  
3.2 A: Juist B: Onjuist C: Juist D: Onjuist  
3.3 A: Onjuist B: Juist C: Juist D: Onjuist