

# Oefeningenreeks 9: herhalingsoefening

## Databank HBO5School

Studenten (studnr, voornaam, familienaam, geboortedatum, geslacht, betaald)

Cursussen (cursusnr, cursusnaam, inschrijvingsgeld)

Studenten\_Cursussen (studnr, cursusnr)

1. Geef een lijst van de mannelijke studenten waarvan de familienaam begint met de letter B. Sorteer op familienaam en voornaam.

```
SELECT *  
FROM studenten  
WHERE geslacht = 'M'  
      AND familienaam LIKE 'B%'  
ORDER BY familienaam, voornaam
```

2. Geef een lijst van de studenten die geboren zijn in de huidige maand. Sorteer op het geboortjaar. Indien het geboortjaar hetzelfde is, sorteer dan op geboortedag.

```
SELECT *  
FROM studenten  
WHERE MONTH(geboortedatum) = MONTH(GETDATE())  
ORDER BY YEAR(geboortedatum), DAY(geboortedatum)
```

3. Geef een overzicht van de studenten met de cursussen die ze volgen. Toon per student: studnr, familienaam, voornaam, cursusnr en cursusnaam. Sorteer stijgend op familienaam, voornaam en cursusnr.

```
SELECT studenten.studnr, familienaam, voornaam,  
       cursussen.cursusnr, cursusnaam  
FROM studenten  
JOIN studenten_cursussen ON studenten.studnr = studenten_cursussen.studnr  
JOIN cursussen ON studenten_cursussen.cursusnr = cursussen.cursusnr  
ORDER BY familienaam, voornaam, cursussen.cursusnr
```

4. Idem vorige vraag, maar zorg ervoor dat ook de studenten getoond worden die geen enkele cursus volgen.

```
SELECT studenten.studnr, familienaam, voornaam,  
       cursussen.cursusnr, cursusnaam  
FROM studenten  
LEFT JOIN studenten_cursussen ON studenten.studnr = studenten_cursussen.studnr  
LEFT JOIN cursussen ON studenten_cursussen.cursusnr = cursussen.cursusnr  
ORDER BY familienaam, voornaam, cursussen.cursusnr
```

5. Hoeveel studenten volgen de cursus C#?

```
SELECT COUNT(*) AS aantal
FROM studenten_cursussen
WHERE cursusnr = (SELECT cursusnr
                  FROM cursussen
                  WHERE cursusnaam = 'C#')

--Ook
SELECT COUNT(*) AS aantal
FROM studenten_cursussen sc
JOIN cursussen c ON sc.cursusnr = c.cursusnr
WHERE c.cursusnaam = 'C#'
```

6. Hoeveel mannelijke studenten zijn er en hoeveel vrouwelijke? Er worden 2 kolommen weergegeven: een kolom geslacht en een kolom aantal.

```
SELECT geslacht, COUNT(*) AS aantal
FROM studenten
GROUP BY geslacht
```

7. Geef alle gegevens van elke student (studenten.\*) en geef per student ook wat zou moeten betaald worden volgens de cursussen die gevolgd worden (voor elke cursus moet INSCHRIJVINGSGELD betaald worden). Sorteer alfabetisch op familienaam en voornaam.

```
SELECT studenten.*,
       (SELECT SUM(c.inschrijvingsgeld)
        FROM cursussen c
        JOIN studenten_cursussen sc ON c.cursusnr = sc.cursusnr
        WHERE sc.studnr = studenten.studnr) AS somtebetalen
FROM studenten
ORDER BY familienaam, voornaam
```

8. Geef de studenten (studenten.\*) die een naamgenoot hebben in de school. Een naamgenoot is een persoon met dezelfde familienaam. Sorteer de naamgenoten alfabetisch op familienaam en voornaam. Zorg dat geen 2 (of meer) keer dezelfde informatie getoond wordt.

```
SELECT s1.*
FROM studenten s1
WHERE familienaam IN
      (SELECT familienaam
       FROM studenten s2
       WHERE s2.familienaam = s1.familienaam
       AND s2.studnr != s1.studnr)
ORDER BY s1.familienaam, s1.voornaam

--Ook
SELECT DISTINCT s1.*
FROM studenten s1
JOIN studenten s2 ON s1.familienaam = s2.familienaam
WHERE s1.studnr != s2.studnr
ORDER BY s1.familienaam, s1.voornaam
```

9. Geef per geslacht de studenten (studenten.\*) die het minst betalen. Sorteer volgens geslacht en daarbinnen op familienaam en voornaam.

```
SELECT *
FROM studenten s
WHERE betaald =
      (SELECT MIN(betaald)
       FROM studenten
       WHERE geslacht = s.geslacht)
ORDER BY geslacht, familienaam, voornaam
--Ook
SELECT *
FROM studenten s1
WHERE NOT EXISTS
      (SELECT *
       FROM studenten s2
       WHERE s2.geslacht = s1.geslacht
       AND s2.betaald < s1.betaald)
ORDER BY geslacht, familienaam, voornaam
```

10. Geef per cursus (cursussen.\*) het aantal studenten dat de cursus volgt. Sorteer op cursusnr.

```
SELECT cursussen.*,
      (SELECT COUNT(*) AS aantalStudenten
       FROM studenten_cursussen
       WHERE cursusnr = cursussen.cursusnr) AS aantalStudenten
FROM cursussen
ORDER BY cursusnr
```

11. Geef per cursus (cursussen.\*) het aantal studenten dat de cursus volgt. Toon enkel de cursussen met meer dan 3 studenten. Sorteer op cursusnr.

```
SELECT cursussen.*,
      (SELECT COUNT(*) AS aantalStudenten
       FROM studenten_cursussen
       WHERE cursusnr = cursussen.cursusnr) AS aantalStudenten
FROM cursussen
WHERE (SELECT COUNT(*) AS aantalStudenten
       FROM studenten_cursussen
       WHERE cursusnr = cursussen.cursusnr) > 3
ORDER BY cursusnr
```

12. Geef de cursussen waarvoor geen vrouwen zijn ingeschreven. Sorteer op cursusnr.

```
SELECT *
FROM cursussen
WHERE cursusnr NOT IN
      (SELECT cursusnr
       FROM studenten_cursussen sc
       JOIN studenten s ON sc.studnr = s.studnr
       WHERE s.geslacht = 'v')
```

```
WHERE geslacht = 'V')  
ORDER BY cursusnr
```

13. Voeg de werknemers uit Computerland uit de Fima-databank toe aan de studententabel.

```
INSERT INTO studenten(familienaam)  
SELECT wnaam  
FROM firma.dbo.werknemers  
WHERE vesnaam = 'Computerland'
```

14. Vul voor deze nieuwe studenten de voornaam in. Kies zelf voornamen.

```
/*Meerdere UPDATE-statements zoals onderstaande*/  
UPDATE studenten  
SET voornaam = 'Karel'  
WHERE studnr = 15
```

15. Verwijder de student met nummer 10. Indien deze student inschrijvingen heeft, worden deze ook verwijderd.

```
DELETE FROM studenten_cursussen  
WHERE studnr = 10  
  
DELETE FROM studenten  
WHERE studnr = 10
```

16. Verwijder de studenten die geen enkele cursus volgen.

```
DELETE FROM studenten  
WHERE studnr NOT IN (SELECT studnr FROM studenten_cursussen)
```

17. Voeg de nieuwe cursus MySQL toe (inschrijvingsgeld 120 Euro). Schrijf de student met nummer 4 in voor deze cursus.

```
INSERT INTO cursussen(cursusnaam, inschrijvingsgeld)  
VALUES('MySQL', 120)  
  
INSERT INTO studenten_cursussen(studnr, cursusnr)  
VALUES (4, @@IDENTITY)
```

18. Voeg de nieuwe student Peter Decorte, geboren op 22 november 1997, toe. Deze student schrijft zich in voor de cursus met nummer 6 en heeft 120 Euro betaald.

```
INSERT INTO studenten(voornaam, familienaam, geboortedatum, geslacht, betaald)  
VALUES ('Peter', 'Decorte', '1997-11-22', 'M', 120)  
  
INSERT INTO studenten_cursussen(studnr, cursusnr)  
VALUES (@@IDENTITY, 6)
```

19. Wijzig de kolom betaald voor de student met nummer 4. Deze student heeft het inschrijvingsgeld voor alle cursussen waarvoor hij ingeschreven is, betaald.

```
UPDATE studenten
SET betaald =
    (SELECT SUM(c.inschrijvingsgeld)
     FROM cursussen c
     JOIN studenten_cursussen sc ON c.cursusnr = sc.cursusnr
     WHERE sc.studnr = 4)
WHERE studnr = 4
```