

BIOINFORMATICA

INFORMATIESYSTE MEN MET PYTHON_

SQL IN PYTHON

CURRICULUM OVERZICHT

	Periode 1	Periode 2	Periode 3	Periode 4
1e jaar	Programmeren in Python (1)	Programmeren in python (2)	Informatiesystemen met Python	Geautomatiseerde identificatie van eiwitten via sequentievergelijking
	Metabolisme	Opsporen van genetische mutaties bij erfelijke ziektes	Vergelijkende genomanalyse: evolutie van virussen	Sequentie alignment (BLAST) en functionele eiwit analyse
2e jaar	Programmeren in Java	Datastructuren en algoritmen in Java	Analyse en ontwerptechnieken	Webtechnologie en textmining
	Proteomics: eiwitstructuren, eiwitfuncties, scheidingstechnieken en data analyse	Transcriptomics: data analyse mbt regulatie van metabole routes	Genomics: annoteren van genomisch DNA	Moleculaire fylogenie: evolutie, multiple sequence alignment mbt signaaltransductie
3e jaar	Stage of minor		Data Mining en grid computing	RNA seq, biostatistiek,
			Eiwitten: carcinogenese en eiwitstructuren	Workflows
4e jaar	Stage, afstudeerstage of minor		Minor of afstudeerstage	

STUDIEWIJZER

Les	Onderwerp	Theorie
1	Introductie tot Informatie Systemen	H1 Introduction to Database Development H2 Entity Relationships H3 Complex Relationships H4 Logical Database Design
2	Database ontwerp Constraints en normaliseren	H5 Normalisation H6 Introduction to Oracle SQL H7 Foreign Keys
3	SQL, operatoren en functies	H8 Selecting data from a Table
4	Rijen ophalen uit meerdere tabellen	H9 Selecting data from multiple tables
5	Subquery's en groepfuncties	H10 Subqueries and Group Functions
6	SQL Embedden in Python Set operatoren	
7	Python en SQL: mogelijkheden en risico's	

AGENDA

Databases

SQL embedden in Python

Set Operatoren in SQL

DATABASES

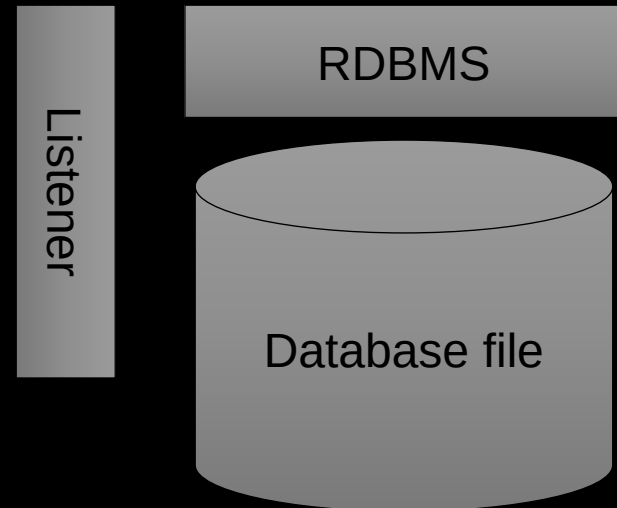
SQL is een *universele* vraagtaal die door ISO is gestandaardiseerd

De *interface* naar de database is echter niet universeel maar database afhankelijk

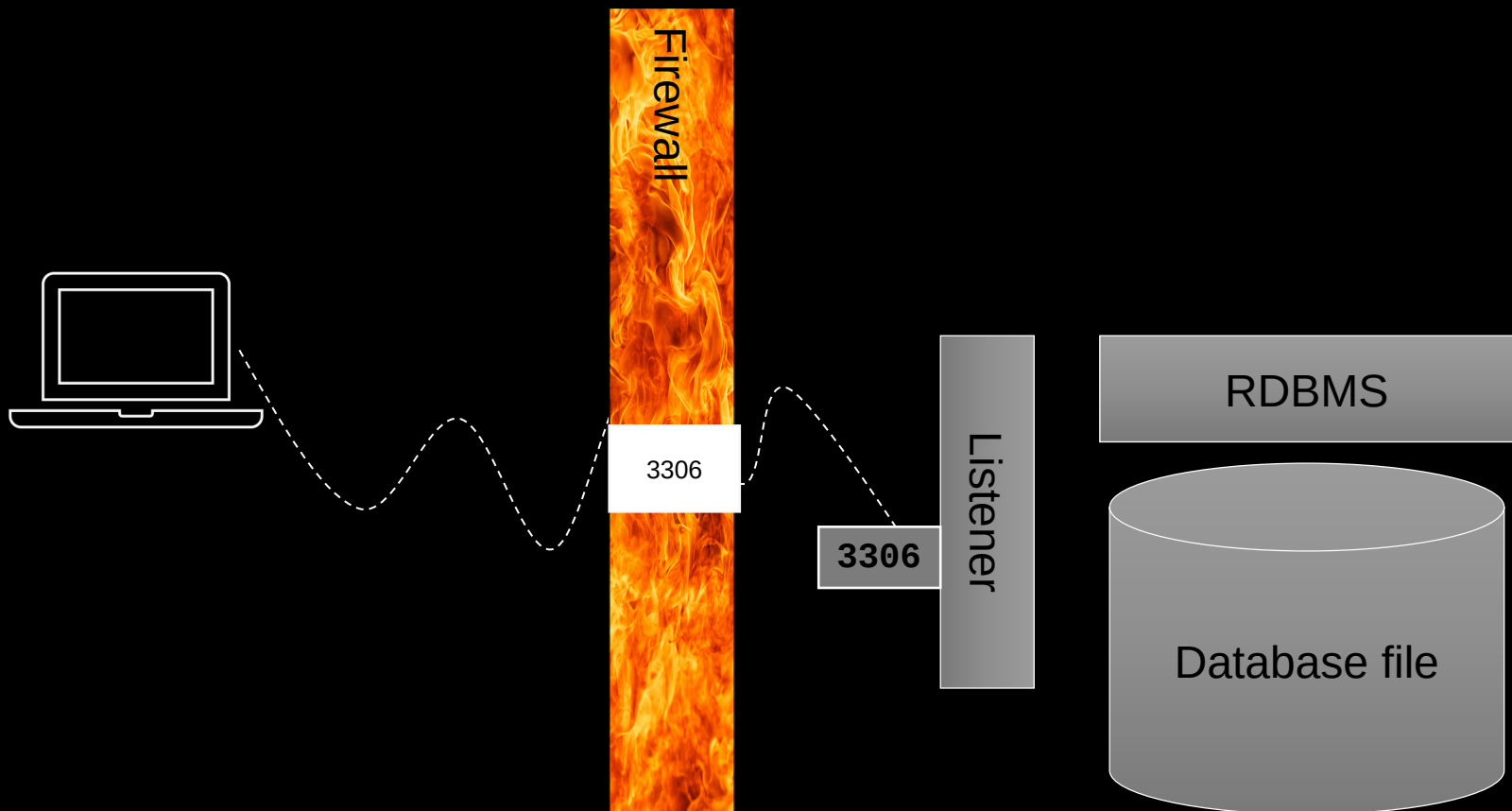
DATABASES

Een database is toegankelijk via een **protocol** in combinatie met een **poort**

Databases hebben een *listener* die staat te luisteren op een specifieke poort



DATABASE ARCHITECTUUR



POORT

Iedere computer heeft 256^2 (65536) poorten

Dit is een toegang tot de computer waarop processen kunnen reageren

Vaak is een poort gebonden aan een specifiek proces

CONTROLE VAN POORTEN

netstat -an toont alle in gebruik zijnde poorten met IP-adressen

```
martijn@jarmin:~$ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 127.0.0.1:63342         0.0.0.0:*                LISTEN
tcp      0      0 0.0.0.0:42415          0.0.0.0:*                LISTEN
tcp      0      0 127.0.0.1:63343         0.0.0.0:*                LISTEN
tcp      0      0 127.0.0.53:53          0.0.0.0:*                LISTEN
tcp      0      0 127.0.0.1:631          0.0.0.0:*                LISTEN
tcp      0      0 127.0.0.1:6942         0.0.0.0:*                LISTEN
tcp      0      0 127.0.0.1:6943         0.0.0.0:*                LISTEN
tcp      0      0 145.102.220.12:60750    54.229.113.239:443      ESTABLISHED
tcp      0      0 145.102.220.12:53884    40.101.80.194:443      ESTABLISHED
tcp      0      0 145.102.220.12:49884    88.221.165.61:443      ESTABLISHED
```

STANDAARD POORTEN

HTTP 80

HTTPS 443

FTP 21

MySQL 3306

HOST

Een host is een verwijzing naar de machine waar software draait

Deze machine is bereikbaar via een **naam (alias)** maar wordt gelokaliseerd via zijn **IP adres**

OPZOEKEN VAN IP ADRES

```
martijn@LT181743:~$ nslookup ensemblpdb.ensembl.org
```

```
Server:          192.168.1.1  
Address:         192.168.1.1#53
```

```
Non-authoritative answer:
```

```
ensemblpdb.ensembl.org  canonical name = ensemblpdb.ebi.ac.uk.
```

```
Name:   ensemblpdb.ebi.ac.uk
```

```
Address: 193.62.193.10
```

```
martijn@LT181743:~$ ping ensemblpdb.ensembl.org
```

```
PING ensemblpdb.ebi.ac.uk (193.62.193.10) 56(84) bytes of data.
```

```
64 bytes from ensemblpdb.ebi.ac.uk (193.62.193.10): icmp_seq=1 ttl=236 time=32.2 ms
```

```
64 bytes from ensemblpdb.ebi.ac.uk (193.62.193.10): icmp_seq=2 ttl=236 time=25.8 ms
```

```
64 bytes from ensemblpdb.ebi.ac.uk (193.62.193.10): icmp_seq=3 ttl=236 time=64.6 ms
```

FIREWALL

Een firewall is een bescherming tegen het binnendringen

Een firewall geeft specifieke **poorten** en **protocollen** en eventueel **machines** toegang

CONNECTIE GEGEVENS

The image shows a screenshot of a database connection configuration window. The window has several tabs: General, Options, SSH/SSL, Schemas, and Advanced. The 'General' tab is selected. The configuration includes the following fields:

- Connection type: default
- Driver: MySQL
- Host: 145.74.104.145
- User: dummy
- Password: <hidden>
- Save: Forever
- Database: dummy
- URL: jdbc:mysql://145.74.104.145:3306/dummy

Below the URL field, there is a note: "Overrides settings above". At the bottom of the window, there is a "Test Connection" button.

Callouts from the image:

- Server IP (points to Host)
- lettercode (points to User)
- Aa[studnr]! (points to Password)
- Lettercode HAN (points to URL)

MYSQL CONNECTOR

1. Ubuntu

- `sudo apt install python3-mysql.connector`

2. Conda

- `conda install -c anaconda mysql-connector-python`

3. pip

- `pip3 install mysql-connector-python`

AGENDA

Databases

SQL embedden in Python

Set Operatoren in SQL

MYSQL CONNECTOR IN PYCHARM

MySQL enablen voor een project

The screenshot shows the PyCharm Settings dialog for a project. The 'Project Interpreter' section is selected, showing the current interpreter as Python 3.7. Below it, a table lists installed packages. An 'Available Packages' dialog is open, showing a search for 'MySQL' and listing various packages, with 'mysql-connector' selected. The details for 'mysql-connector' are shown on the right.

Project: untitled > Project Interpreter For current project

Project Interpreter: Python 3.7 C:\Users\Martijn\AppData\Local\Programs\Python\Python37\python.exe

Package	Version	Latest version
Click	7.0	7.0
Flask	1.0.2	1.0.2
Jinja2	2.10	2.10
MarkupSafe	1.1.0	▲ 1.1.1
Werkzeug	0.14.1	0.14.1

Available Packages

Search: MySQL

- mysql
- mysql-autodoc
- mysql-autotrbackup
- mysql-batch
- mysql-binlog-explorer
- mysql-connector**
- mysql-connector-async-dd
- mysql-connector-python
- mysql-connector-python-dd
- mysql-connector-python-ri

Description

MySQL driver written in Python

Version

2.1.6

Author

Oracle and/or its affiliates

<https://github.com/sanpingz/mysql-connector>

STAPPEN

```
import mysql.connector
conn = mysql.connector.co
    host = "ensemldb.e
"anonymous", db="h
cursor = conn.cu
cursor.execute (
rows = cursor.fe
print (rows)
cursor.cl
conn.cl
```

(1) Importeren van de mysql connector package

(2) Connectie maken aan een database

(2) host is de naam van de server waar de database draait

(2) Naam van de database (schema) maar dit kan ook in de query worden

(3) Openen van een cursor (geheugen) om rijen op te vragen en

(4) Geef de cursor een query opdracht en plaats de rijen in de cursor

(5) Haal de rijen op en plaats deze allemaal in een lijst (rows)

(6) Doe iets met de rijen in de list

(7) Sluit de cursor

(8) Verbreek de connectie met de database

```
import mysql.connector
conn = mysql.connector.connect(
    host = "ensembldb.ensembl.org",
    user = "anonymous",
    db="homo_sapiens_core_95_38"
cursor = conn.cursor ()
cursor.execute ("SELECT * FROM gene")
rows = cursor.fetchall ()
print (rows)
cursor.close ()
conn.close ()
```

AGENDA

Databases

SQL embedden in Python

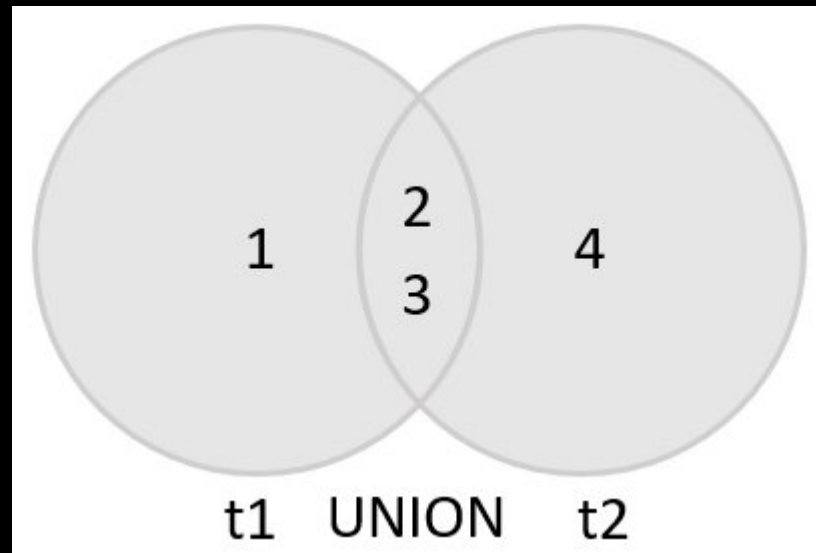
Set Operatoren in SQL

INLEIDING

Set operatoren combineren rijen en koppelen kolommen aan elkaar
Dit hebben we ook gezien met sets in Python

UNION

Alle rijen die in een van beide of beide tabellen voorkomen



UNION

Een union van alle namen

```
select voornaam, tussenvoegsels, achternaam
from student
union
select voornaam, tussenvoegsels, achternaam
from docent
order by achternaam
```

UNION ALL

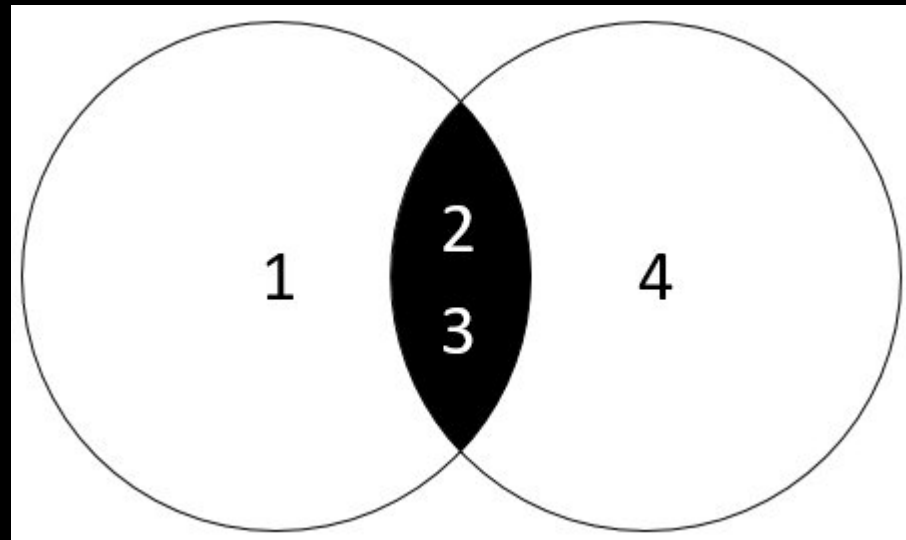
Set operatoren werken met unieke voorkomens in verzamelingen (Python)

Met **union all** krijg je ook dubbele waardes terug

```
select voornaam
from student
union all
select voornaam
from docent
order by achternaam
```

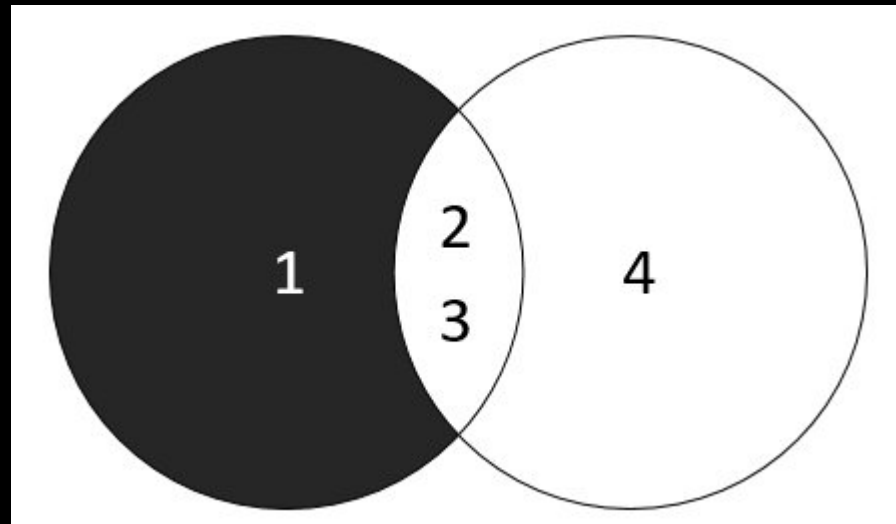
INTERSECT

Alle rijen die in beide tabellen voorkomen (niet MySQL)



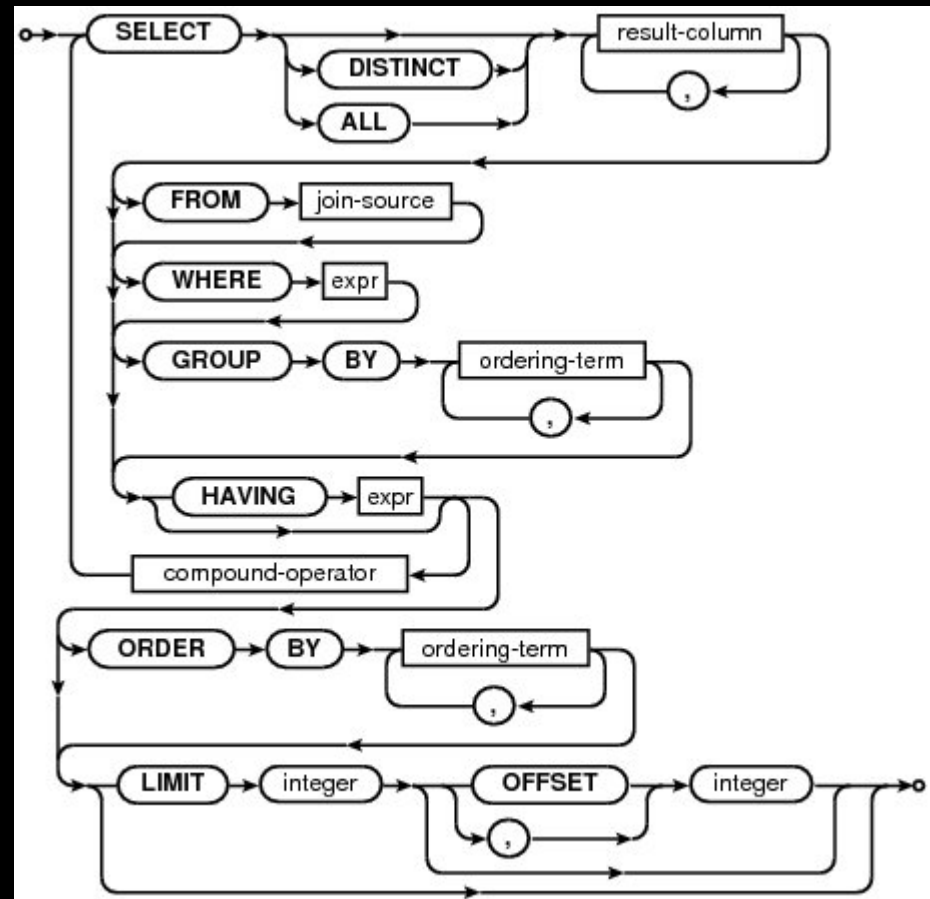
MINUS

Alle rijen die alleen in de eerste tabel voorkomen (niet MySQL)



SAMENVATTING

Set operatoren geven de mogelijkheid om rijen verticaal te combineren



BELANGRIJKE COMMANDO'S IN MYSQL

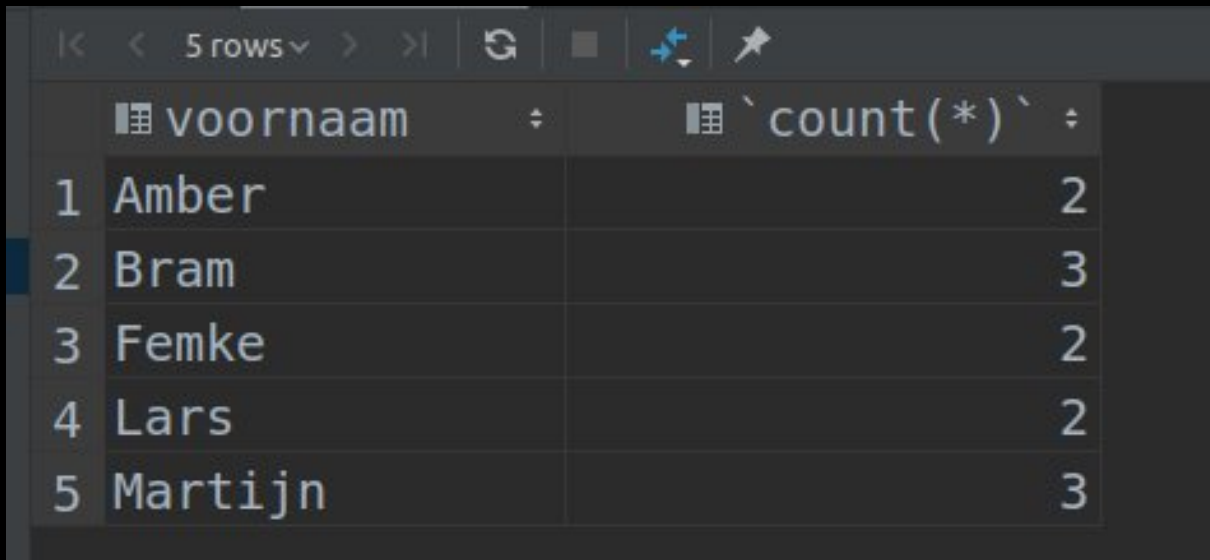
show tables

show databases

use <database>

VRAAG 1

Toon alle namen die meer dan 1 keer voorkomen, zowel voor studenten als docenten. Doe dit in 1 query.



The screenshot shows a database query result in a dark-themed interface. The table has two columns: 'voornaam' and 'count(*)'. The data is as follows:

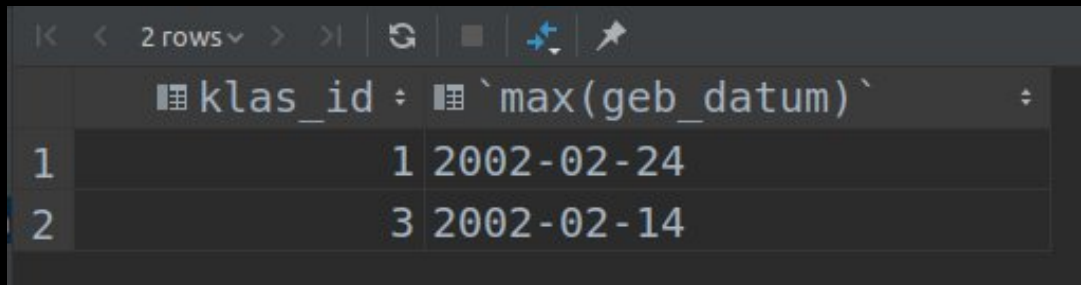
	voornaam	count(*)
1	Amber	2
2	Bram	3
3	Femke	2
4	Lars	2
5	Martijn	3

ANTWOORD 1: UNION

```
select voornaam, count(*)  
from (  
    select voornaam  
    from student  
    union all  
    select voornaam  
    from docent  
    ) namen  
group by voornaam  
having count(*)>1
```

VRAAG 2

Toon de geboorte datum van de jongste student uit klas 1 en klas 3.



The screenshot shows a database query result with two columns: 'klas_id' and 'max(geb_datum)'. The results are as follows:

	klas_id	max(geb_datum)
1	1	2002-02-24
2	3	2002-02-14

ANTWOORD 1: UNION

```
select 1, max(geb_datum)
from student
where klas_id = 1
union
select 3, max(geb_datum)
from student
where klas_id = 3
```

ANTWOORD 2: GROUP BY

```
select klas_id, max(geb_datum)
from student
where klas_id in (1,3)
group by klas_id
```

ANTWOORD 3: GECORRELEERDE SUBQUERY

```
select klas_id, geb_datum
from student hq
where geb_datum in (select max(geb_datum)
                    from student sq
                    where hq.klas_id = sq.klas_id
                    and klas_id in (1,3))
```

VERANTWOORDING

In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen

Alle teksten © Martijn van der Bruggen/HAN tenzij expliciet externe bronnen zijn aangegeven

Screenshots op basis van eigen werk auteur en/of vernoemde sites

Eventuele images zijn opgenomen met vermelding van bron