

BIOINFORMATICA

# INFORMATIESYSTE MEN MET PYTHON\_

BASIS VAN  
INFORMATIESYSTEMEN EN  
MODELLEREN



HAN\_ UNIVERSITY  
OF APPLIED SCIENCES

# EVEN VOORSTELLEN

[Thijn.vankempen@han.nl](mailto:Thijn.vankempen@han.nl)

- Voormalig wetenschapper Bioinformatica
- Docent Data Science for Life Sciences/Informatica
- Msc Bioinformatics

# JAAR 1

Basis van programmeren in Python

Complexe structuren

Databases voor opslag van data

|         | 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 genoomanalyse: evolutie van virussen | Sequentie alignment (BLAST) en functionele eiwit analyse              |

Ontwikkelen van applicaties voor biologisch analyse en visualisatie op het web

# STUDIEWIJZER

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

# UIT HET OPLEIDINGSSTATUUT

INFORB12 Informatiesystemen

7,5

Afvinkopdrachten (*weging 0*)

Thematoets (*weging 2*)

Statistiek R (*weging 1*)

Wiskunde 1 (*weging 0*)

# TOETSING INFORMATICA

- Eindopdracht
  - **Ontwerp een database op papier (35%)**
  - **Schrijf queries met de computer (65%)**

# GEBRUIKTE LITERATUUR



## VOOR WE BEGINNEN...

- Download eduvpn en log hier in.
- Dit is nodig om dit blok een verbinding te maken met de database.



# DOELEN

Aan het eind van deze week heb je kennis over

- Informatie Systemen
- Relationele database
- Het ontwerpen van een conceptueel ERD

# AGENDA

- **INFORMATIE SYSTEMEN**
- Relationele Databases
- SQL
- Data modelling: conceptueel ERD

**Waar denk jij aan bij  
informatiesystemen?**

# VOORBEELDEN VAN INFORMATIE SYSTEMEN

- Alluris/OSIRIS
- Onderwijsonline
- Facebook
- Snapchat
- Youtube
- Google Search
- Mail
- Google Maps
- Signal
- EPD
- Kahoot!
- WhatsApp
- Wikipedia
- TomTom
- Bankrekeningen
- NetFlix
- Spotify
- Bol.com
- PayPal
- Twitter
- Nu.nl

# VOORBEELDEN VAN INFORMATIESYSTEMEN

- Maar ook de blockchain technologie die aan de basis ligt van cryptocurrency zoals Bitcoin, Ethereum is een (gedistribueerde) database: een informatiesysteem.

# OPSLAG VAN DATA

Het merendeel van de informatiesystemen heeft een database voor opslag en ophalen van data

# INFORMATIESYSTEEM

Systeem waarmee over objecten of personen beheerd kan worden

Beheren

- Verzamelen
- Bewerken
- Analyseren
- Integreren
- Representeren (visualiseren)

24 januari 2020

# INFORMATIESYSTEMEN

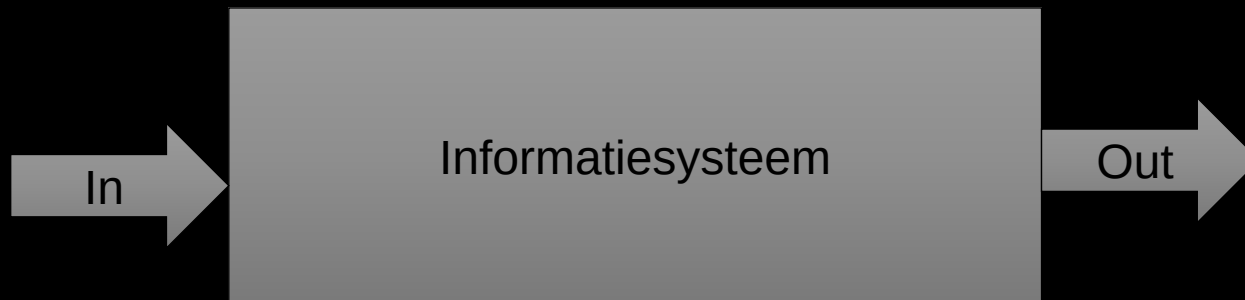
- TikTok
  - Facebook
  - Instagram
  - Twitter
  - WhatsApp
  - Signal
  - Telegram
  - Tinder
  - Snapchat
  - AirBnB
- 
- Informatiesystemen die niets anders doen dan data van jou en anderen opslaan en op een andere moment, andere plaats of andere manier weer tonen.
  - Maar hoe slaan ze die data op?

# SAMENVATTING

- Een informatiesysteem is in staat om data op te slaan en deze data te gebruiken om informatie te geven
- Bijna alle systemen die we gebruiken zijn vormen van informatie systemen

# SAMENVATTING

Een informatiesysteem betreft een systeem (**software**) dat input en output (I/O) heeft (**data**)



# AGENDA

- Informatie Systemen
- **RELATIONELE DATABASES**
- SQL
- Data modelling: conceptueel ERD

# SOORTEN DATA

## Ongestructureerde data



## Gestructureerde data

- Betekenis blijkt uit de **context**
- Geordend om toegankelijk te zijn voor mensen
- Voorbeelden
  - Tekst bestanden
  - Email
  - Social media
  - Media: foto/video

- Betekenis is gekoppeld aan de **opslag**
- Geordend om toegankelijk te zijn voor computers/applicaties
- Voorbeelden
  - Data van sensoren
  - Wetenschappelijke data

# GESTRUCTUREERDE DATA

- In de informatica spreken we over **gestructureerde** data wanneer deze zijn opgeslagen met de betekenis erbij

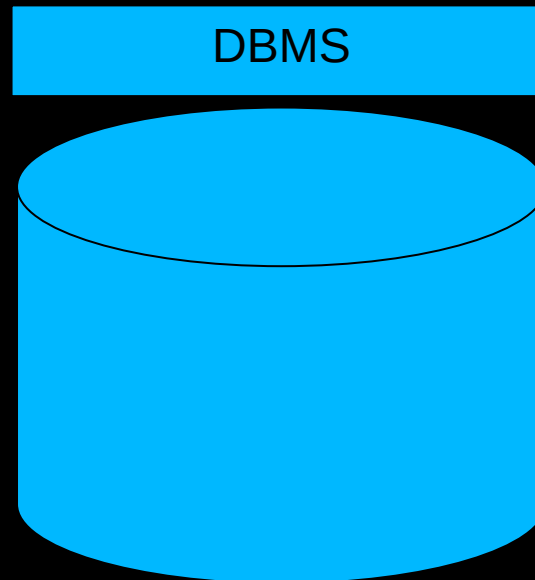
Voorbeelden:

- Excel bestand
  - Database
- 
- Data zoals deze in boeken of op webpagina's zijn te vinden beschouwen we als **ongestructureerde** data

# DBMS

De **opslag** van data vindt plaats in databases

Het **beheer** van data vindt plaats door software:  
DBMS = Database Management System



# DATABASES ZIJN OVERAL

Deze morgen heb je al gebruik gemaakt van meerdere databases:

- Spotify
- Whatsapp
- Rooster bekeken
- ...

# WAT IS EEN DATABASE?

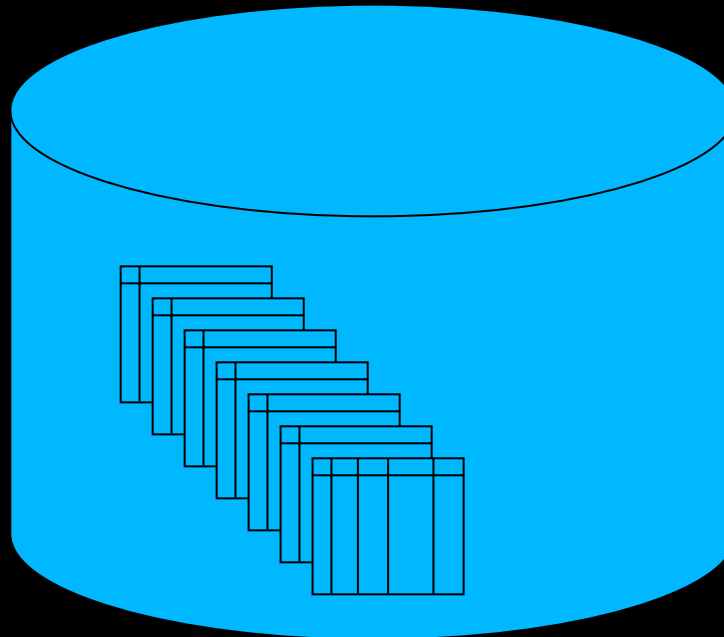
Een database is een verzameling van gegevens

# WAT IS EEN RELATIONELE DATABASE?

Een relationele database is een verzameling van gegevens  
geordend in tabellen

# RELATIONELE DATABASE

RDBMS



# WAT IS EEN RDBMS?

RDBMS = Relational Database Management System

Software programma (op server) dat relationele databases kan beheren

# WAT IS RELATIONEEL?

Relationeel geeft aan dat er **tabellen** in de database liggen

Boek maakt onderscheid tussen:

- Relation = tabel
- Relationship = relatie tussen tabellen

**Welke bedrijven produceren de software  
voor relationele databases?**

**Op welke manier wordt de data opgeslagen?**

**Welke bedrijven/organisaties maken gebruik van relationele databases?**

# SAMENVATTING

Een database is een verzameling van geordende gegevens

De gegevens worden beheerd door een **database management system**

Wanneer de data in tabellen zijn opgeslagen spreken we van een **relationeel database management systeem**

# AGENDA

- Informatie Systemen
- Relationale Databases
- SQL
- Data modelling: conceptueel ERD

# VRAAGTAAL

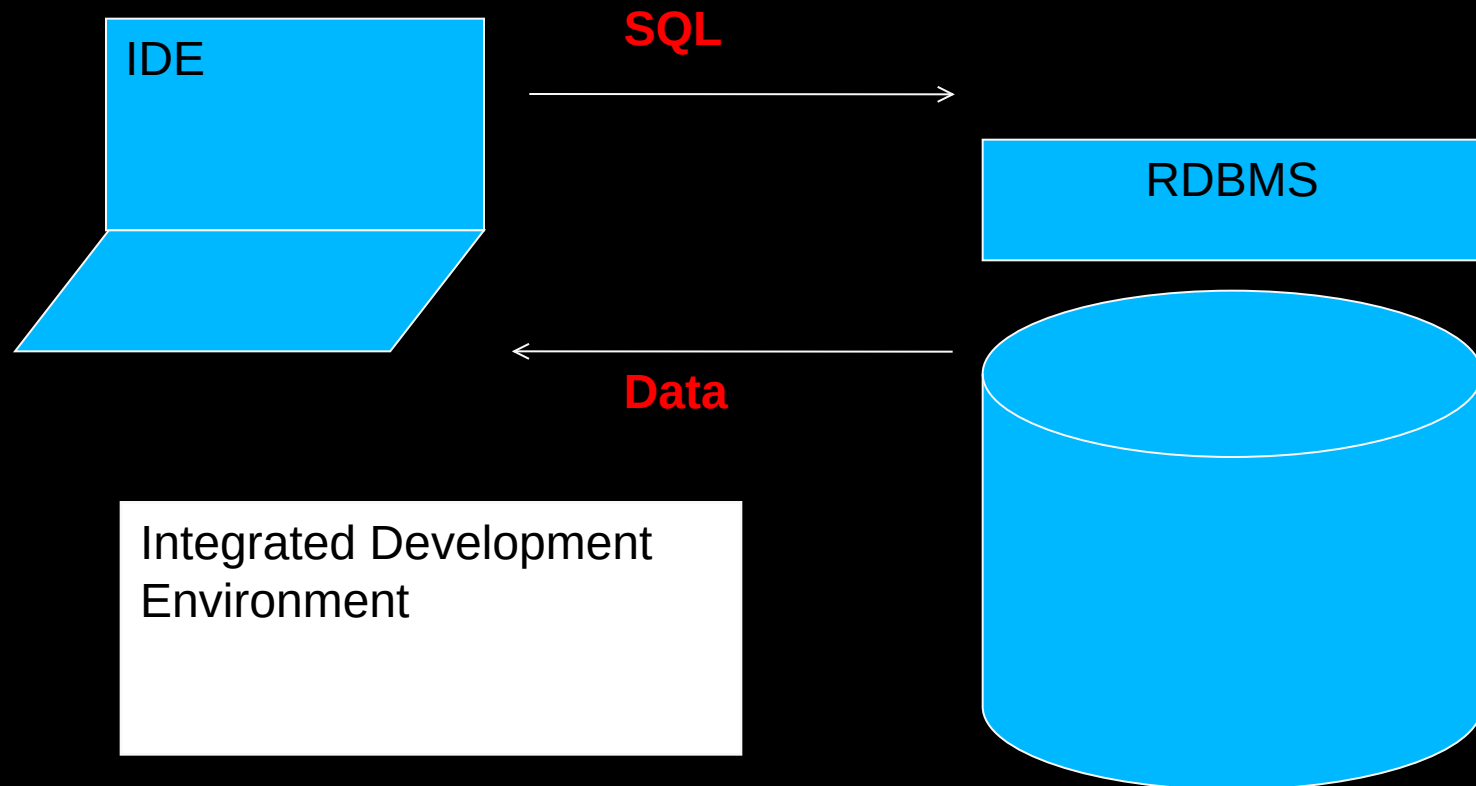
- Om een relationele database te bevragen is een **universele taal** ontwikkeld: SQL

**Heb je al eerder van SQL gehoord?**

# WAT IS SQL?

- SQL staat voor **Structured Query Language**
- Spreek uit als *siekwel* of S-Q-L
- Het is een wiskundige manier om data op te halen

# WERKEN MET EEN RELATIONELE DATABASE



# TABELLEN

|     |  |  |       |  |
|-----|--|--|-------|--|
|     |  |  | kolom |  |
|     |  |  |       |  |
| rij |  |  | cel   |  |
|     |  |  |       |  |
|     |  |  |       |  |
|     |  |  |       |  |
|     |  |  |       |  |
|     |  |  |       |  |
|     |  |  |       |  |
|     |  |  |       |  |

# SELECT STATEMENT

- Select
  - Meest gebruikt
  - Data retrieval
- Meest complexe statement
- Het begint altijd met select

# SYNTAX VAN HET SELECT STATEMENT

```
select kolommen  
from tabel
```

# PYTHON VS. SQL

## Python



## SQL

- Programmeertaal
- **Procedureel**
  - variabelen
  - iteratie
  - conditionele controle

```
bestand = open  
("stud.txt")  
for regel in bestand:  
    print (regel.split()[2])
```

Beschrijving **hoe** je een  
bestand wilt lezen

- **Declaratieve** taal: beschrijving  
van wat je wilt

```
select * from student
```

zal alle studenten tonen uit de  
tabel studenten

Beschrijving **wat** je uit de  
database wilt halen

# SAMENVATTING

- Data Retrieval statements beginnen altijd met select
- Het select statement is het meest uitgebreide statement in SQL

# AGENDA

- Informatie Systemen
- Relationale Databases
- SQL
- Data modelling: conceptueel ERD

# OVERZICHT ORGANISMEN: REDUNDANTIE

| Kingdom  | Phylum   | Class    | Order    | Family  | Scientific name |
|----------|----------|----------|----------|---------|-----------------|
| ANIMALIA | CHORDATA | MAMMALIA | RODENTIA | MURIDAE | Mus cypriacus   |
| ANIMALIA | CHORDATA | MAMMALIA | RODENTIA | MURIDAE | Mus macedonicus |
| ANIMALIA | CHORDATA | MAMMALIA | RODENTIA | MURIDAE | Mus musculus    |
| ANIMALIA | CHORDATA | MAMMALIA | RODENTIA | MURIDAE | Mus spicilegus  |
| ANIMALIA | CHORDATA | MAMMALIA | RODENTIA | MURIDAE | Mus spretus     |
| ANIMALIA | CHORDATA | MAMMALIA | RODENTIA | MURIDAE | Rattus rattus   |

# OVERZICHT ORGANISMEN

| Kingdom  | Phylum       | Class      | Order       | Family  | Scientific name |
|----------|--------------|------------|-------------|---------|-----------------|
| ANIMALIA | CHORDATA     | MAMMALIA   | RODENTIA    | MURIDAE | Mus cypriacus   |
| ANIMALIA | CHORDATA     | MAMMALIA   | KNAAGDIEREN | MURIDAE | Mus macedonicus |
| ANIMALIA | CHORDATA     | MAMMALIA   | RODENTIA    | MURIDAE | Mus musculus    |
| DIEREN   | CHORDATA     | MAMMALIA   | RODENTIA    | MURIDAE | Mus spicilegus  |
| ANIMALIA | CHORDATA     | MAMMALIA   | RODENTIA    | MURIDAE | Mus spretus     |
| ANIMALIA | CHORDADIEREN | ZOOGDIEREN | RODENTIA    | MURIDAE | Rattus rattus   |

# INFORMATIE SYSTEEM

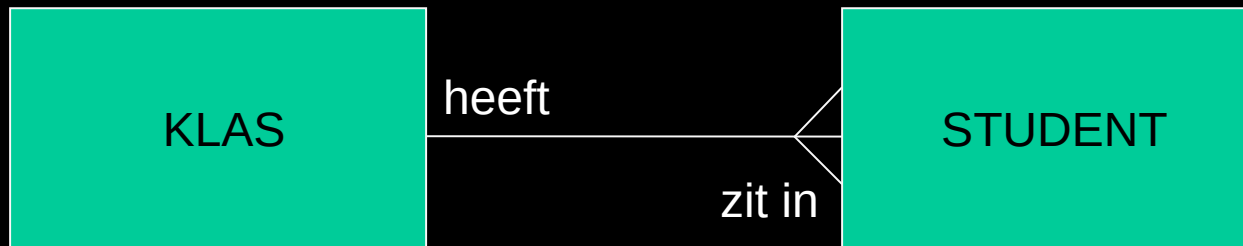
- Een relationele database bouwen vereist een gedegen voorbereiding
- Er worden verschillende stappen doorlopen voordat de database wordt gemaakt

# ERD

- Samenhang tussen entiteiten geef je schematisch weer in een ERD
  - ERD = **E**ntity **R**elationship **D**iagram
- ERD's gebruik je tijdens de **informatieanalyse**

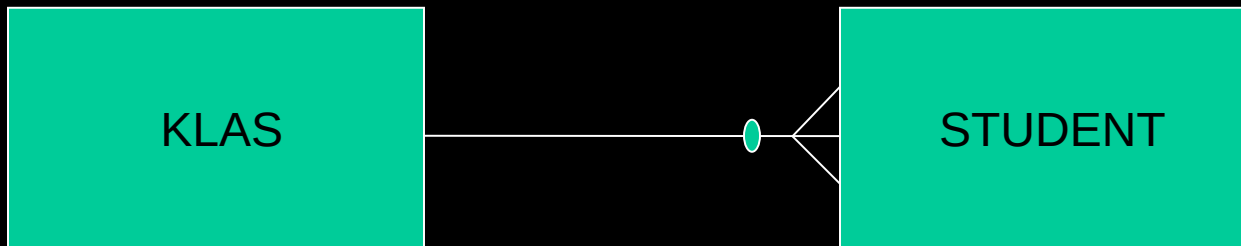
# ERD BEGRIPPEN CROWS FEET

- Relatie
  - verbinding tussen entiteiten op basis van aanwezigheid



# ERD BEGRIPPEN CROWS FEET

- Optioneel
  - aanwezigheid niet verplicht



# ERD BEGRIPPEN CROWS FEET

- Verplicht
  - bij de aanwezigheid van de ene is aanwezigheid van de ander verplicht



# ERD BEGRIPPEN CROWS FEET

- 1:N
  - Als de ene aanwezig is, zijn er van de andere 0 of meer



# ERD BEGRIPPEN CROWS FEET

- N:M
  - In beide richtingen van de relatie kunnen bij de aanwezigheid van de ene, 0 of meer van de andere aanwezig zijn. Probleem hier is herleidbaarheid



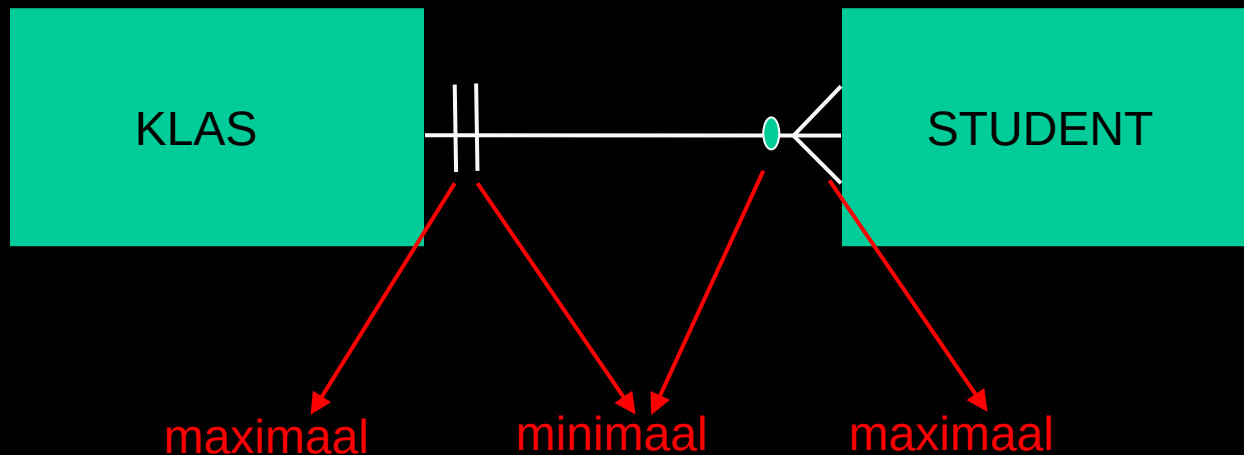
# ERD BEGRIPPEN CROWS FEET

- 1 op veel waarbij de || aangeeft dat het exact 1 moet zijn



# ERD BEGRIPPEN CROWS FEET

- 1 op veel waarbij de || aangeeft dat het exact 1 moet zijn



# RELATIE

- In database jargon wordt gesproken over **kardinaliteit** bij het soort relaties tussen entiteiten

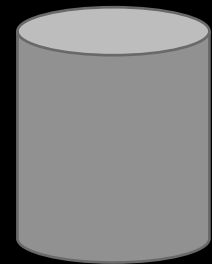
# SAMENGEVAT

|                                                                                      |                    |
|--------------------------------------------------------------------------------------|--------------------|
|    | One                |
|    | Many               |
|    | One (and only one) |
|    | Zero or one        |
|  | One or many        |
|  | Zero or many       |

# STAPPEN IN DATABASE MODELLERING



# STAPPEN IN ONTWIKKELING DATABASE



# STAP 1: CONCEPTUEEL MODEL

- Het model zoals dat door de gebruikers wordt gezien
  - Entiteiten
  - Relaties
  - Kardinaliteiten
  - Attributen (zonder datatypes)
- Zonder verwijzingen (foreign keys)

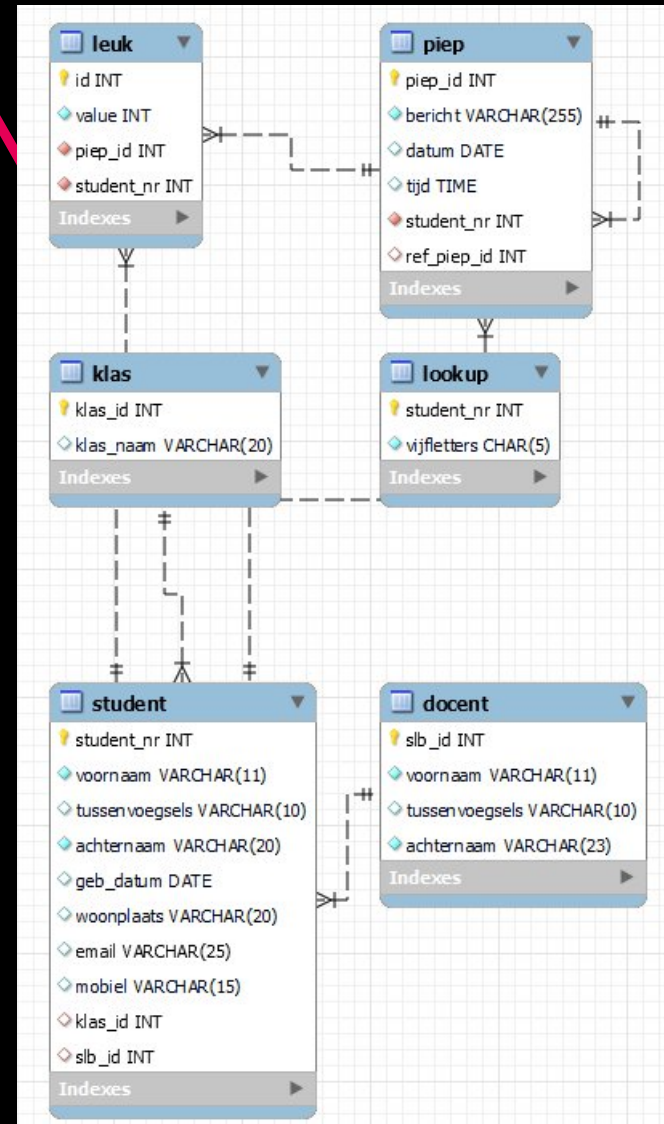
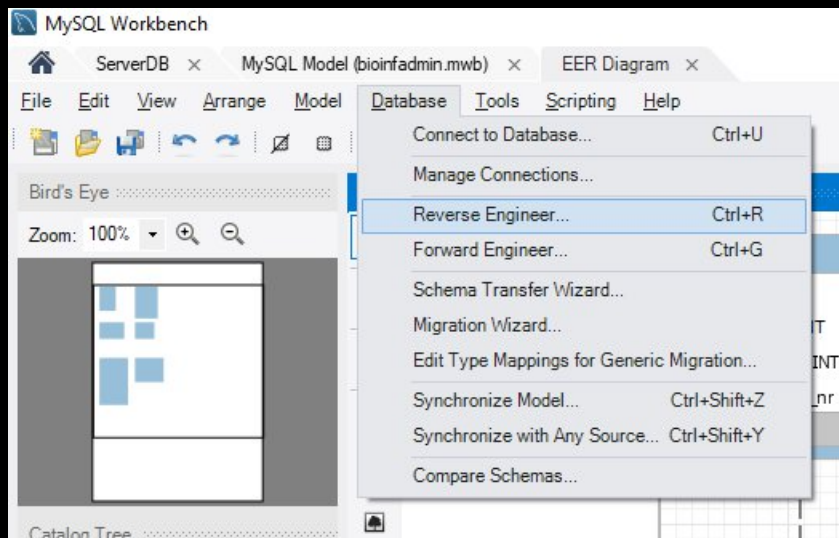
## STAP 2: LOGISCH MODEL

- Model zoals het conceptuele model maar nu ook
  - Attributen (met datatypes)
  - Entiteiten
  - Relaties
  - Kardinaliteiten

# STAP 3: FYSIEK MODEL

- Fysiek, technisch of implementatie model
  - Verwijzingen (foreign keys)
  - Genormaliseerd
  - Attributen (met datatypes)
  - Entiteiten
  - Relaties
  - Kardinaliteiten
- Afgeleid van conceptueel model
- Vorm van de database zoals die geïmplementeerd gaat/kan worden
- Reverse engineering

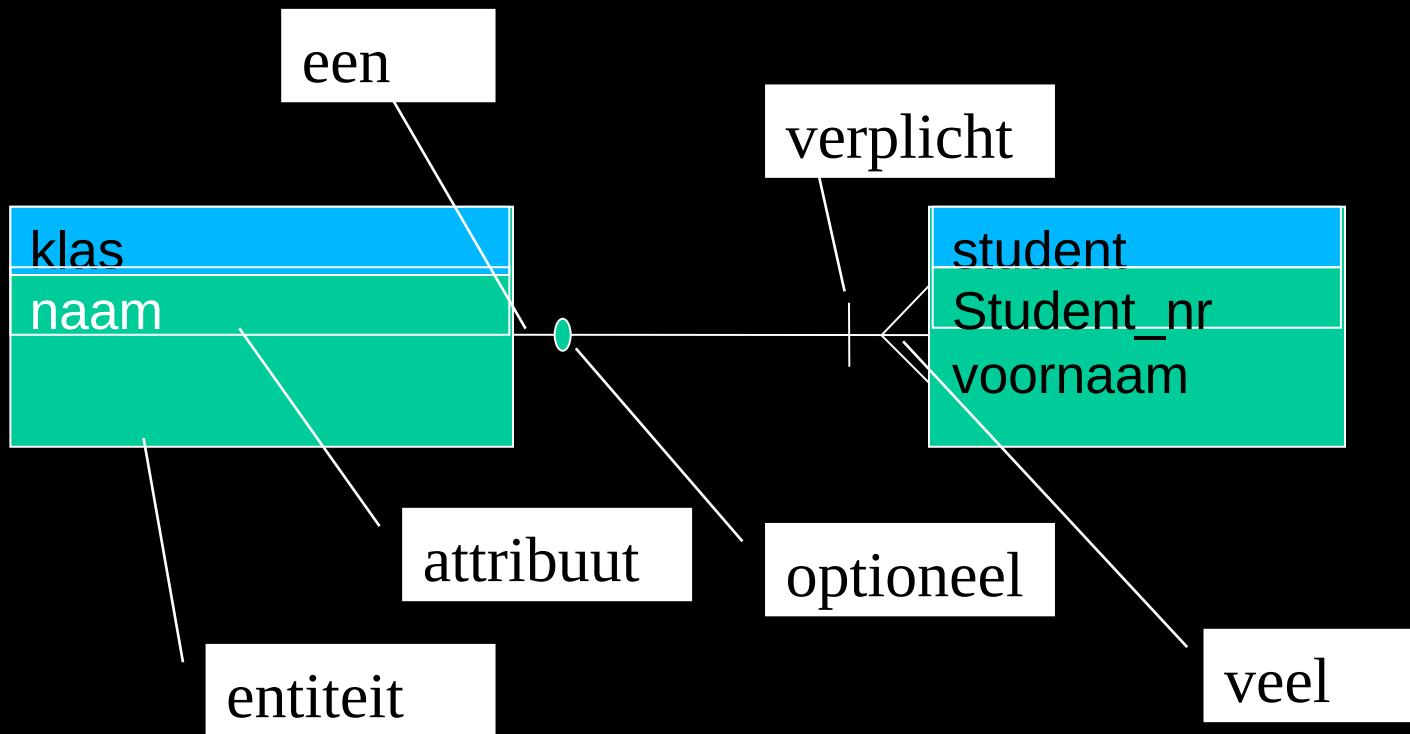
# REVERSE ENGINEERING



# CONCEPTUEEL, LOGISCH EN FYSIEK ERD

|              | Conceptueel     | Logisch  | Fysiek      |
|--------------|-----------------|----------|-------------|
| Datatypes    | niet            | aanwezig | aanwezig    |
| Veel-op-veel |                 |          | tussentabel |
| Keys         | niet            | niet     | aanwezig    |
| Voor wie     | Bedrijfsanalist |          | Beheerder   |

# SAMENVATTING



# OPENMRS

- De basis van iedere applicatie is het ontwerp van de opslag van data
- De data die erachter liggen zijn zichtbaar via een interface

# LIVE DEMO

- Op [https://www.han.nl/medisch](#) kunnen we een live demo bekijken van een Medical Record System
- Login met admin en Admin123

# OPENMRS

- Bespreek met elkaar
  - Hoeveel tabellen gaan hier achter schuil?
  - Welke tabellen zijn er in gebruik?



# INTERFACE EN SQL

 OpenMRS  
MEDICAL RECORD SYSTEM

 admin ▾

 Laboratory ▾

Logout 

 > Find Patient Record

## Find Patient Record

Paul Walker 

| Identifier | Name        | Gender | Age | Birthdate   |
|------------|-------------|--------|-----|-------------|
| 10000X     | Paul Walker | M      | 47  | 11 Apr 1971 |

Showing 1 to 1 of 1 entries

[First](#) [Previous](#) 1 [Next](#) [Last](#)

```
select concat(given_name, ' ', family_name)
      , gender
      , birthdate
      , TIMESTAMPDIFF(YEAR, birthdate, CURDATE())
from person
      join person_name using (person_id)
where given_name = 'Paul' and family_name = 'Walker'
```

# NORMALISEREN

- Gestructureerd opslaan van gegevens is belangrijk
- Normaliseren is een hulpmiddel om de **structuur** te ontwikkelen
- Het is gebaseerd op **fexibiliteit, onderhoudbaarheid** en het voorkomen van **redundante** registratie van en in het model

# NORMALISEREN

- Normaliseren leidt tot een eenduidige beschrijving van een mogelijke informatie structuur
- Op basis van deze beschrijving kan een model (schema) getekend worden
- Dit model vormt vaak de basis van de implementatie van databases

# REDUNDANTIE

- Het meer dan benodigd (overvloedig) zijn van iets

# REDUNDANTIE

- Vaak is redundantie wenselijk:
  - In technische systemen
    - Vliegtuigen hebben systemen vaak drievoudig (Backups)
  - In de biologie
    - Er bestaan veel verschillende soorten die door verschillende soorten cellen geproduceerd worden, sommige hebben overlappende functies en ze werken in op verschillende andere soorten cellen.
    - We hebben twee nieren, waarbij 1 zou kunnen voldoen
  - ...

# REDUNDANTIE, OVERBODIGHEID

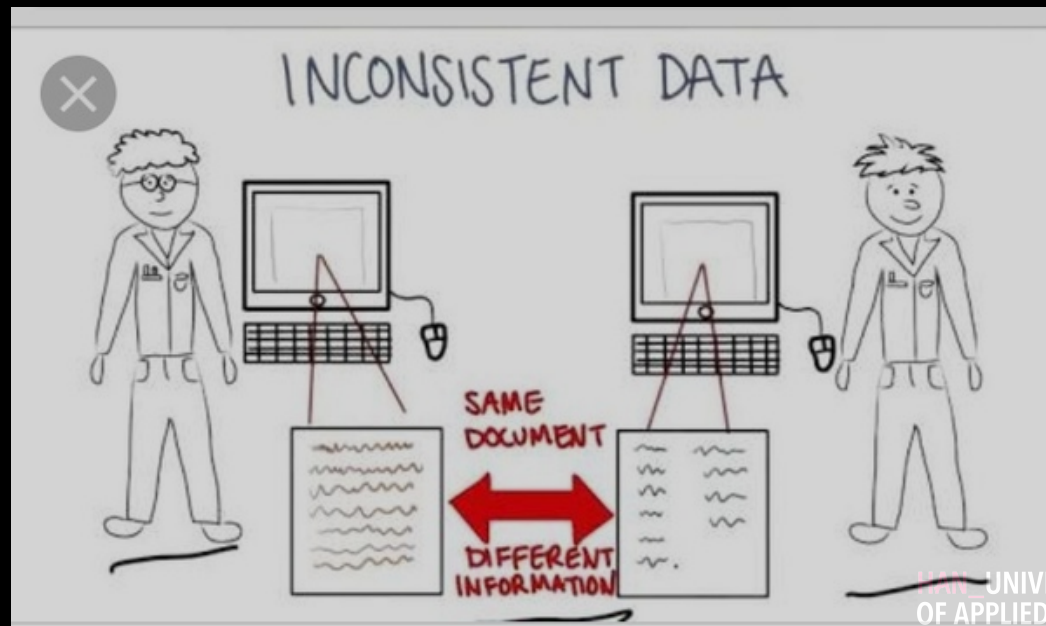
- De laatste dag van januari, 31 januari
- De huidige president van Amerika, Donald Trump
- Hij fietste met een fiets naar school
- Hij is geboren op 25 december 2001, eerste kerstdag  
en nu 18 jaar oud

# REDUNDANTIE LEIDT TOT INCONSISTENTIE

- De laatste dag van januari, 17 januari
- De huidige president van Amerika, Freek Vonk
- Hij fietste met een auto naar school
- Hij is geboren op 25 december 2001, eerste paasdag  
en nu 8 jaar

# INCONSISTENTIE

- Inconsistentie is het gevolg van redundantie
- Gegevens zijn met elkaar in tegenspraak
- Dat kunnen afleidbare gegevens zijn zoals een geboortedatum die niet overeenkomt met de leeftijd



# NORMALISEREN

- Om data “veilig” op te slaan in een relationele database normaliseren we data
- Normaliseren van data leidt ertoe dat data niet redundant worden opgeslagen
- Hiermee voorkom je inconsistenties

# NORMALISEREN

- In database jargon kunnen we spreken over de **normaalkorm** waarin data zich bevindt
- Als data zich in de eerste normaalvorm bevindt voldoet het aan de volgende **Eisen**:
  - elk attribuut bevat een atomaire waarde
  - geen enkel attribuut wordt herhaald
  - alle attributen blijven constant in de tijd

# EERSTE NORMAALVORM (1NF)

1. elk attribuut bevat een atomaire waarde
2. geen enkel attribuut wordt herhaald
3. alle attributen blijven constant in de tijd

# EERSTE NORMAALVORM?

| Stud_nr | Naam            | Cijfer           |
|---------|-----------------|------------------|
| 652163  | Pietje Puk      | 8 bioinformatica |
| 628491  | Thomas de Trein | 9 Java I         |
| 090912  | Leon Kennedy    | 9 LS 2           |
| 264820  | Peppa Pig       | 8 Labrekenen     |

## Eerste Normaal Vorm

1. elk attribuut bevat een atomaire waarde
2. geen enkel attribuut wordt herhaald
3. alle attributen blijven constant in de tijd

# EERSTE NORMAALVORM?

| Stud_nr | vnaam  | Tussen | achter  | Over x dagen<br>jarig |
|---------|--------|--------|---------|-----------------------|
| 652163  | Pietje |        | Puk     | 201                   |
| 628491  | Thomas | De     | Trein   | 368                   |
| 090912  | Leon   |        | Kennedy | 17                    |
| 264820  | Peppa  |        | Pig     | 154                   |

## Eerste Normaal Vorm

1. elk attribuut bevat een atomaire waarde
2. geen enkel attribuut wordt herhaald
3. alle attributen blijven constant in de tijd

# EERSTE NORMAALVORM?

| Stud_nr | vnaam  | tussenv | achter  | Studie          |
|---------|--------|---------|---------|-----------------|
| 652163  | Pietje |         | Puk     | Bio-Informatica |
| 628491  | Thomas | De      | Trein   | Bio-Informatica |
| 090912  | Leon   |         | Kennedy | Bio-Informatica |
| 264820  | Peppa  |         | Pig     | Bio-Informatica |

## Eerste Normaal Vorm

1. elk attribuut bevat een atomaire waarde
2. **geen enkel attribuut(waarde) wordt herhaald**
3. alle attributen blijven constant in de tijd

# EERSTE NORMAALVORM?

| Stud_<br>nr | vnaa<br>m  | tusse<br>nv | achter  | Vak 1 | Vak 2 | Vak 3 |
|-------------|------------|-------------|---------|-------|-------|-------|
| 652163      | Pietje     |             | Puk     | 7     | 8     | 9     |
| 628491      | Thoma<br>s | De          | Trein   | 8     | 9     | 6     |
| 090912      | Leon       |             | Kennedy | 8     | 6     | 9     |
| 264820      | Peppa      |             | Pig     | 9     | 9     | 7     |

1. elk attribuut bevat een atomaire waarde
2. **geen enkel attribuut wordt herhaald**
3. alle attributen blijven constant in de tijd

# EERSTE NORMAALVORM?

| Stud_<br>nr | vnaam  | tusse<br>nv | achter  | Python | Chemie | LS |
|-------------|--------|-------------|---------|--------|--------|----|
| 652163      | Pietje |             | Puk     | 7      | 8      | 9  |
| 628491      | Thomas | De          | Trein   | 8      | 9      | 6  |
| 090912      | Leon   |             | Kennedy | 8      | 6      | 9  |
| 264820      | Peppa  |             | Pig     | 9      | 9      | 7  |

# INTEGRITEIT

- Gegevens in een database zijn integer als ze overeenkomen met de werkelijkheid
- Een geboortedatum voor een student van 29-feb-2019 is een niet integere waarde

# SAMENVATTING

- Normaliseren geeft **flexibiliteit** en voorkomt **redundantie**
- Bij afwezigheid van redundantie voorkom je **inconsistentie**
- Data zijn **integer** indien ze overeenkomen met de werkelijkheid

# BELANGRIJKE BEGRIPPEN

- Informatiesysteem
- Gestructureerde data
- Ongestructureerde data
- Database
- Normaliseren
- Redundantie
- Inconsistentie
- Integriteit van data
- DBMS
- RDBMS
- ERD
  - Conceptueel ERD
  - Logisch ERD
  - Fysiek ERD
- Entiteit
- Attribuut
- Atomair
- Normaliseren
- Eerste normaalvorm

# OPDRACHT

- Ontwerp een ERD voor de opslag van data van studenten met hun huisdieren
- Wat je wilt kunnen ophalen uit de database is de naam en leeftijd van de student en de naam en soort van het huisdier.
- Niet iedere student heeft een huisdier, maar sommige hebben er wel meer dan 1
- Alle huisdieren zijn gekoppeld aan een student

# 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