



H1

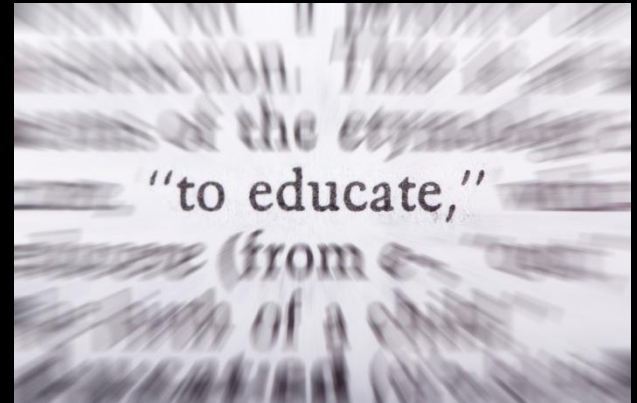
Databanken gekaderd

Gegevensmanagement
DBMS
Historiek

**HO
GENT**

Definities

1. Database
2. Databasemanagementsysteem
3. Databaseprogramma's



Definities

1. Database (databank)

- Een databank bestaat uit **verzamelingen van persistente gegevens** die gebruikt worden door de softwareapplicaties van een bedrijf en **beheerd** worden door een **DBMS** (J. Date).
- Een gedeelde verzameling van **logisch met elkaar verbonden gegevens** en hun **beschrijving**, ontworpen om aan de informatienoden van een organisatie te voldoen (O. Connolly).

We onthouden:

- verzameling van gegevens
- digitaal opgeslagen
- met onderlinge relaties
- met bepaalde betekenis
- voor een bepaalde groep van gebruikers

Definities

2. Databasemanagementsysteem (DBMS)

= verzameling van programma's waarmee:

- een DB wordt gecreëerd en beheerd
- gegevens in de DB worden geladen, gewijzigd en opgevraagd



PostgreSQL

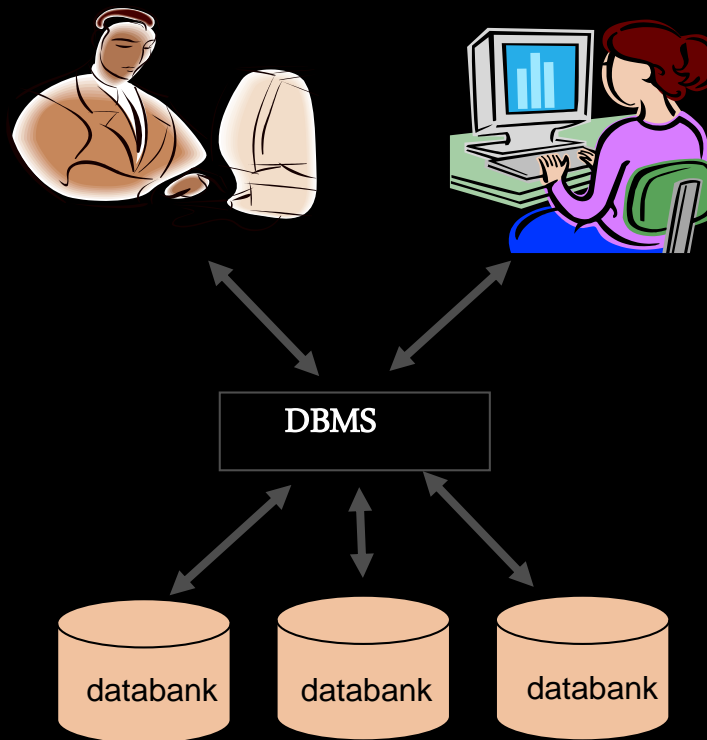


HO
GENT

Definities

2. Databasemanagementsysteem (DBMS)

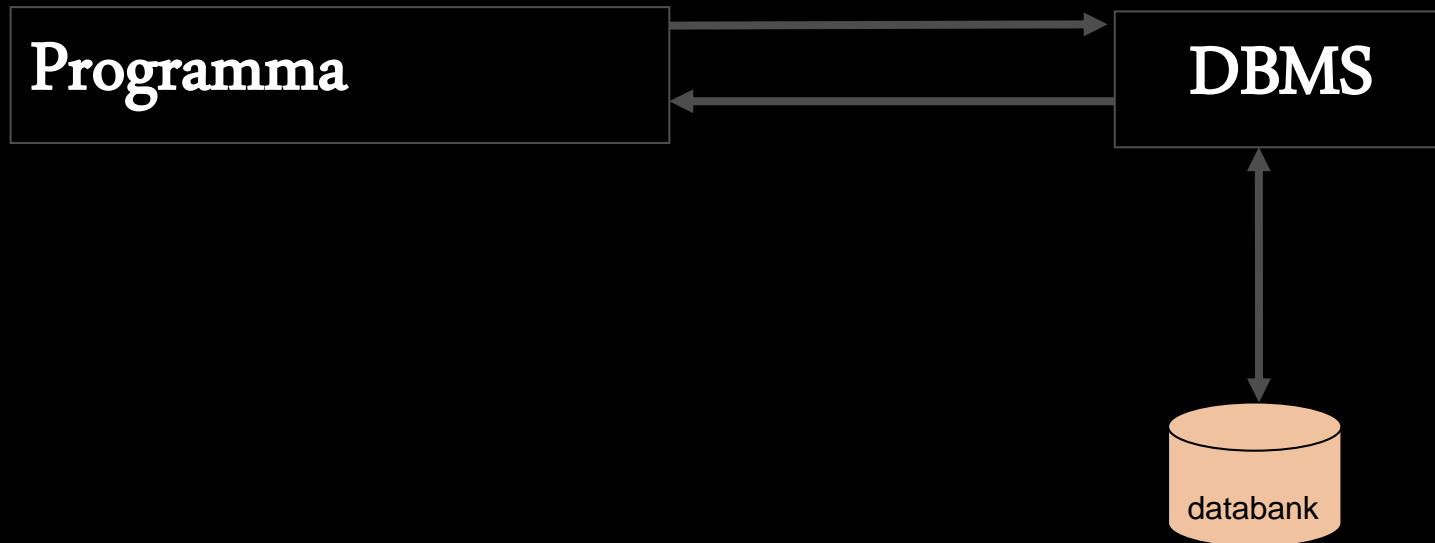
= interface tussen gebruikersprogramma's en de databank



Definities

3. Databaseprogramma

= **computerprogramma** dat gegevens uitwisselt met de databank
(maar nooit rechtstreeks!).



Toepassingen

- Databanken zijn niet meer weg te denken uit ons dagelijks leven.
- Wat je tegenwoordig ook doet, ergens kom je rechtstreeks of onrechtstreeks in aanraking met een databank.

Toepassingen

- Betalen aan kassa:
 - ➔ producten worden ingescand
 - ➔ prijs wordt 'gelezen'
 - ➔ deze informatie komt uit een databank

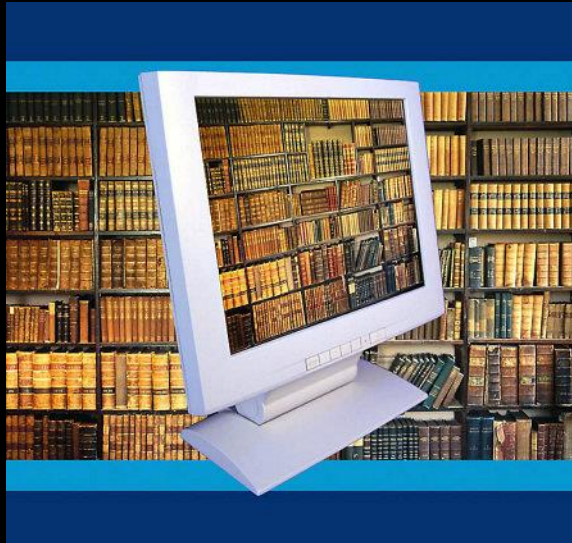


Toepassingen

- Bibliotheek:

opvragen of een bepaald boek aanwezig is of niet

➔ informatie van alle boeken zit in een databank



Toepassingen

- Online een vlucht boeken:

opvragen beschikbaarheid vlucht

➔ informatie van alle vluchten zit in een databank



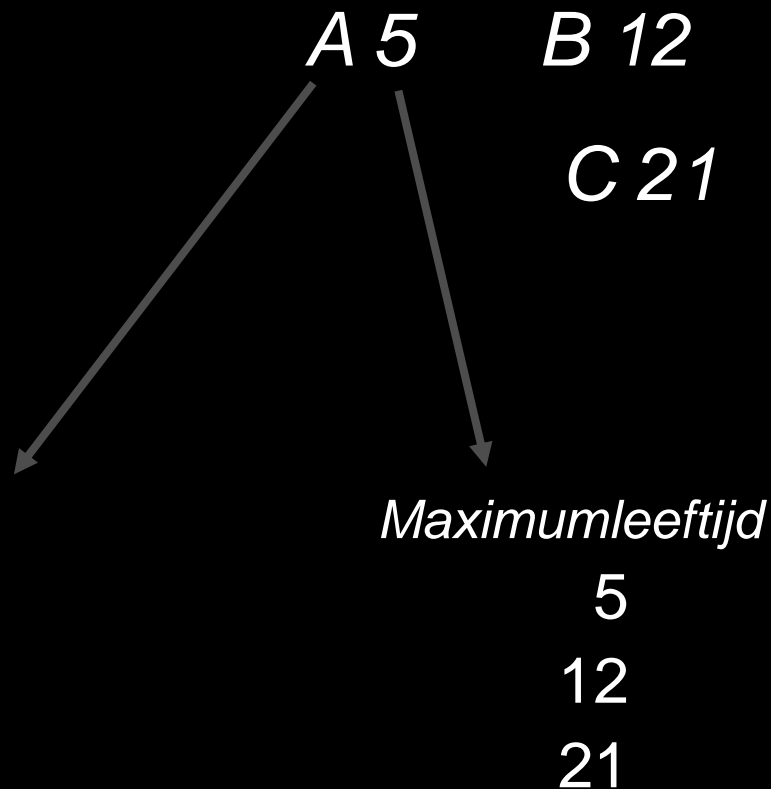
Toepassingen

Tallose andere:

- Voorraad checken
- Websites / Internet
- Betalingen met je bankkaart
- ...



Gegevens worden informatie



door er een betekenis aan te geven

Gegevensmanagement

Gegevensmanagement =

- verzamelen
 - opslaan
 - terugvinden
- } van gegevens



**HO
GENT**

Gegevensmanagement

Gegevensmanagement **vroeger** :

- Stand-alone applicaties
 - elke toepassing definieert en gebruikt zijn eigen bestanden
- ➔ Verschillende **bestanden** worden gedefinieerd en geïmplementeerd **zonder relaties** tussen beide. Het is in de applicatieprogramma's dat de relaties tussen de data uit de verschillende bestanden moeten worden gelegd/gevonden.

Gegevensmanagement

Enkele nadelen van stand-alone applicaties :

1. Verspreiding en isolatie van data
2. Redundantie
3. Incompatibiliteit
4. Fixed queries

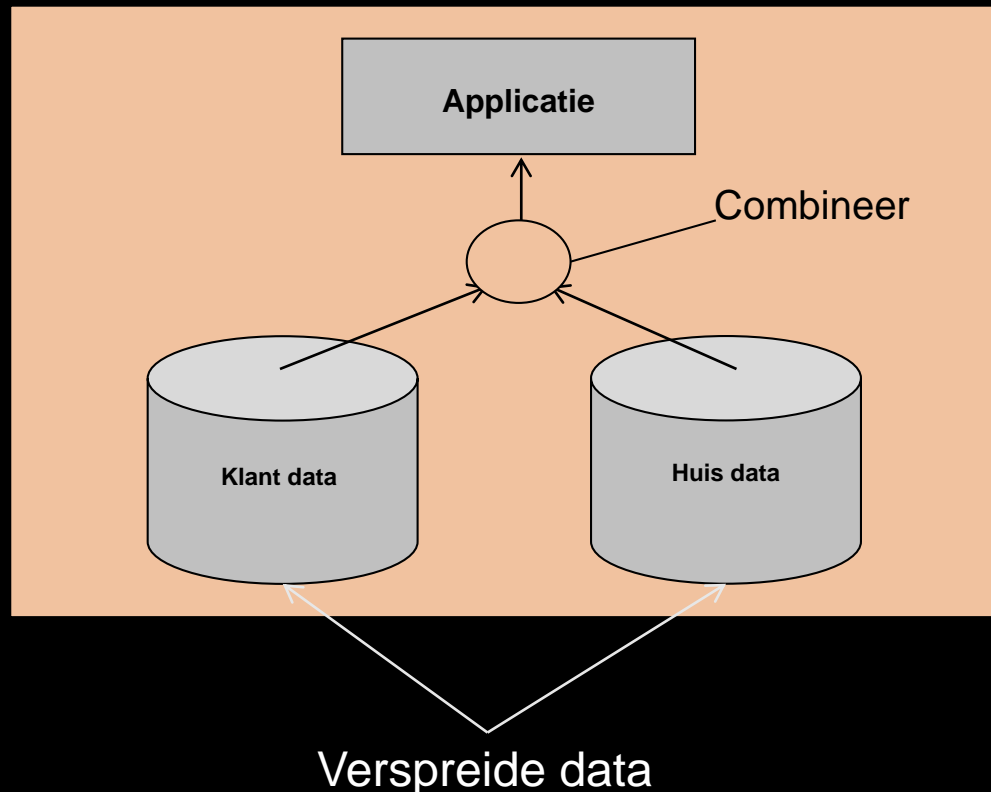
➔ **Inconsistentie en inefficiëntie !**

Gegevensmanagement

Nadeel 1: Verspreiding en isolatie van data

Voorbeeld: applicatie voor immobureau

Doel: Per klant de huizen opvragen die bij zijn wensen passen



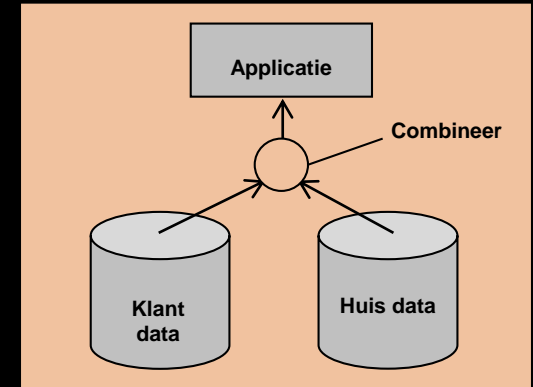
Gegevensmanagement

Nadeel 1: Verspreiding en isolatie van data

Hoe oplossen?

Een mogelijkheid:

1. Per klant een tijdelijk bestand aanmaken.
2. Voor elke klant: record per record de huisdata overlopen.
3. Elk passend record wegschrijven in het tijdelijk bestand van de betreffende klant.



Gegevensmanagement

Nadeel 1: Verspreiding en isolatie van data

Gevolgen?

1. Veel programmeerwerk
2. Foutgevoelig
3. Het kost veel tijd om dit uit te voeren (execution time)



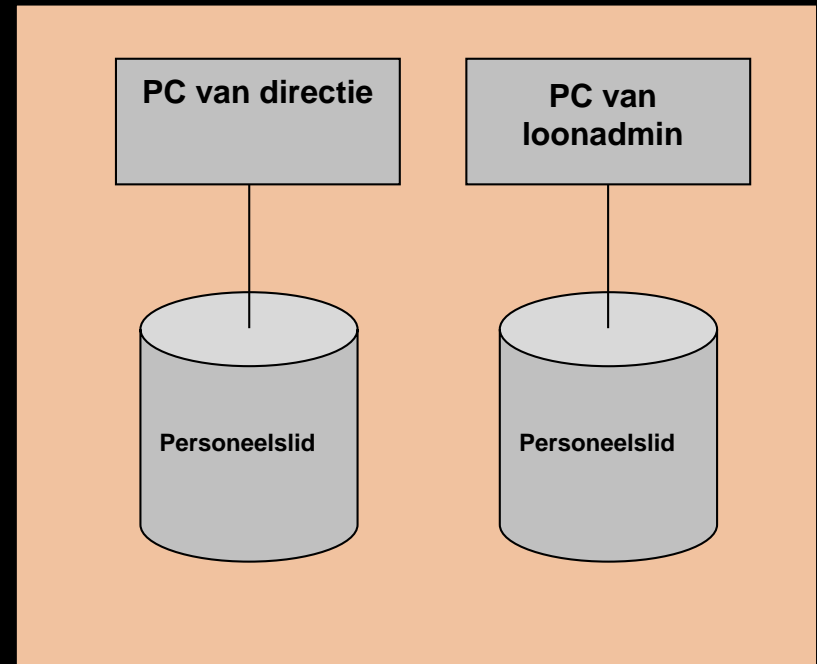
Gegevensmanagement

Nadeel 2: Redundantie = duplicatie van data

Voorbeeld:

➔ PC van de directie heeft
een bestand Personeelslid

➔ PC van de loonadministratie
heeft een kopie van het
bestand Personeelslid

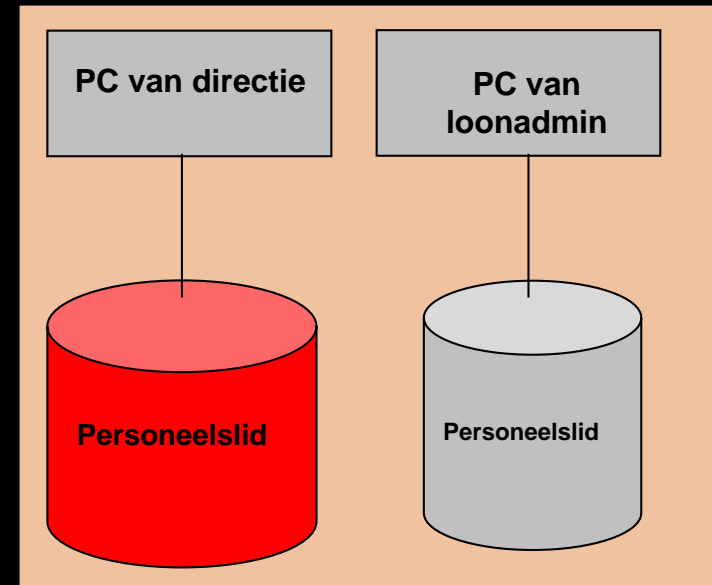


Zelfde gegevens verspreid over meerdere bestanden!

Gegevensmanagement

Nadeel 2: Redundantie = duplicatie van data

- ➔ Wat indien de directeur aan personeelslid Pieter Jansens 20 € opslag geeft?
- ➔ Loonsopslag zal opgenomen zijn in bestand van de directie maar niet in bestand van de loonadministratie.



Inconsistentie !

Gegevensmanagement

Nadeel 2: Redundantie = duplicatie van data

Hoe oplossen?

Mogelijkheid 1: Manueel

➔ Directie geeft elke wijziging door aan de loonadministratie,
zodanig dat zij dit in hun bestand kunnen aanpassen en vice
versa.

Grote kans op fouten (mensen zijn geen machines!)

Gegevensmanagement

Nadeel 2: Redundantie = duplicatie van data

Hoe oplossen?

Mogelijkheid 2: Kopie

➔ Het bestand Personeelslid op de pc van de directie wordt dagelijks gekopieerd naar de pc van de loonadministratie.

**Geen up-to-date informatie bij de loonadministratie!
Wat indien de loonadministratie gegevens wijzigt?**

Gegevensmanagement

Nadeel 2: Redundantie = duplicatie van data

Hoe oplossen?

Mogelijkheid 3: Synchroniseren

➔ Synchronisatieprogramma schrijven om gegevens van beide bestanden te vergelijken en up te daten.

1) Geen up-to-date informatie in beide bestanden tot de synchronisatie opgestart wordt.

Gegevensmanagement

Nadeel 2: Redundantie = duplicatie van data

Hoe oplossen?

Mogelijkheid 3: Synchroniseren

➔ Synchronisatieprogramma schrijven om gegevens van beide bestanden te vergelijken en up te daten.

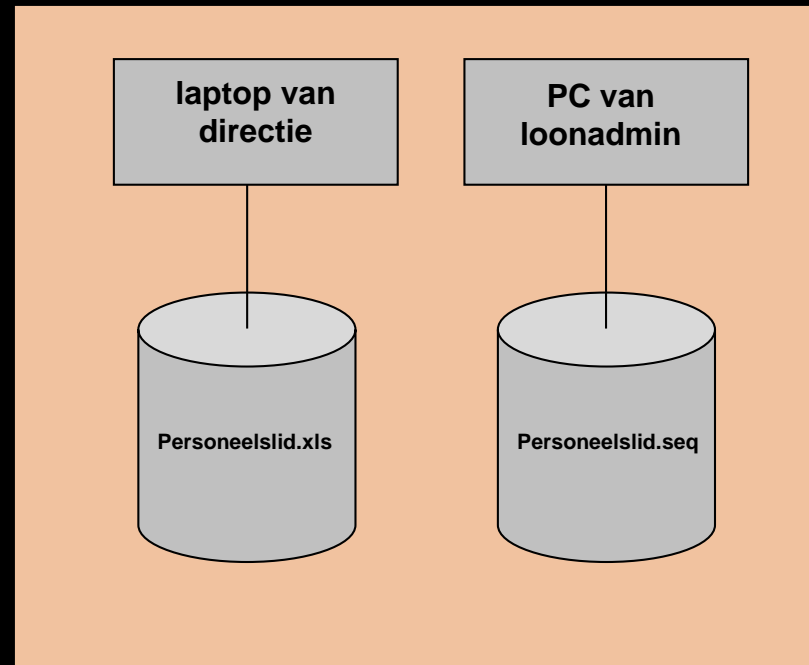
2) Wat indien een gegeven in beide bestanden een verschillende waarde heeft ➔ wat is de juiste waarde?

Gegevensmanagement

Nadeel 3: Incompatibiliteit

Voorbeeld:

- PC van de loonadministratie heeft een programma dat bestand Personeelslid.seq gebruikt (sequentieel bestand)
- Laptop van de directeur heeft enkel office-programma's. De gegevens van de personeelsleden moeten van het sequentieel bestand omgezet worden naar een excel-file



**HO
GENT**

Conversieprogramma nodig!

Gegevensmanagement

Nadeel 4: Fixed queries

- De manier om de gegevens op te vragen ligt vast in een programma.
- Indien een gebruiker de data op een andere manier wil opvragen moet er een nieuw programma geschreven worden.

Weinig flexibiliteit!

Gegevensmanagement

Conclusie?

- De traditionele bestandsgebaseerde oplossing is verre van perfect.
- Het brengt veel extra problemen en werk met zich mee.



Is er een betere oplossing?

Gegevensmanagement

Bemerging!



Bijna alle problemen komen voort uit :

- Datadefinitie wordt bewaard in de applicaties.
- Alle controle van de datatoegang en datamodificatie bevindt zich in de applicaties.

Een nieuwe oplossing moet minstens deze problemen aanpakken!

**HO
GENT**

Gegevensmanagement

Een oplossing voor de problemen bij een bestandsgebaseerde oplossing vond men in een **DBMS (DataBase Management System)**.

Deze systemen worden vandaag
overal gebruikt.



**HO
GENT**

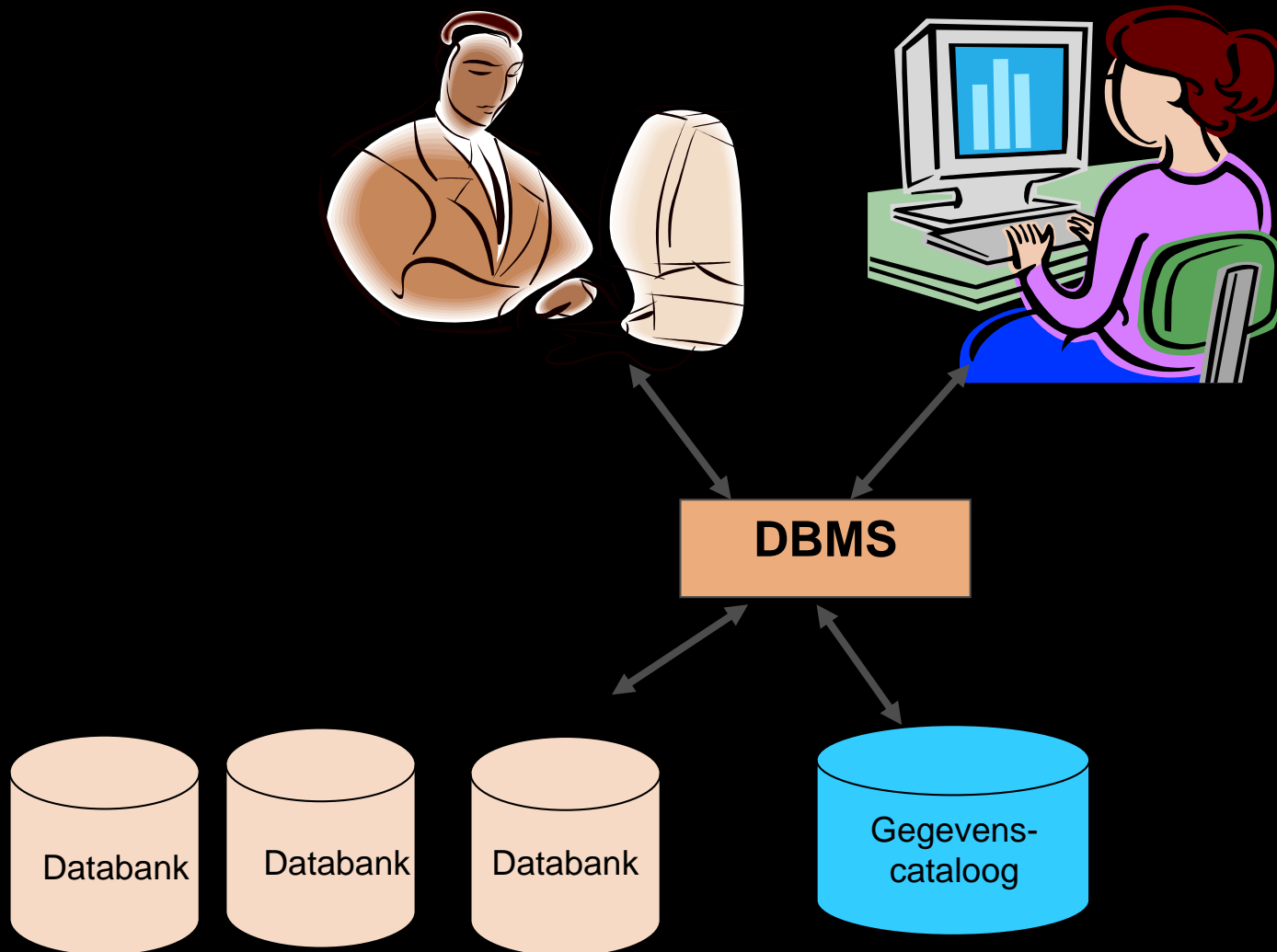
DBMS

In volgende delen zullen we verduidelijken waaruit een DBMS bestaat en hoe het de problemen van bestandsgebaseerde/stand-alone applicaties oplost:

1. Elementen van een DB-Systeem
2. Modellen
3. Rollen binnen een DBMS
4. Voor-en nadelen

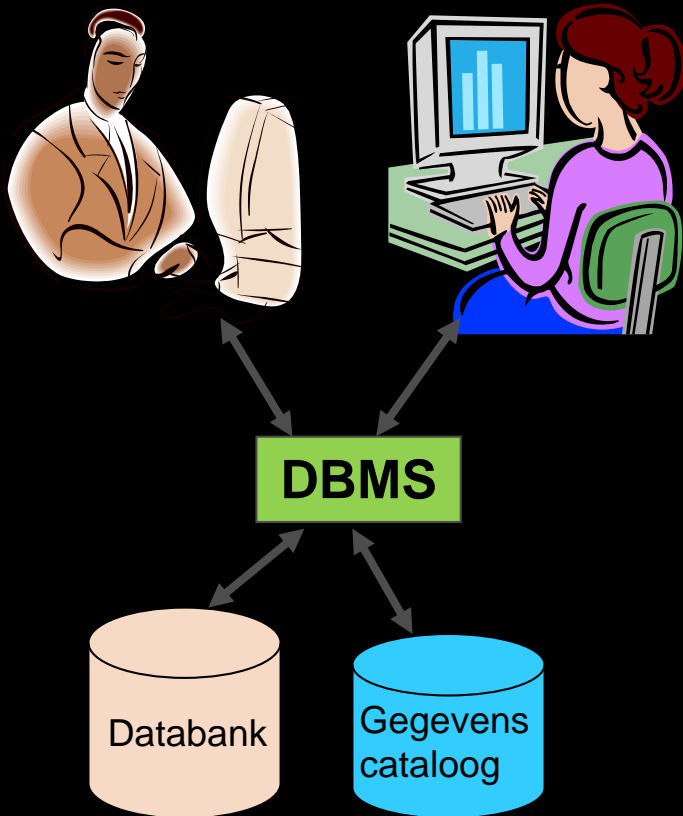


DBMS: Onderdelen van een DBMS-systeem



DBMS: Onderdelen van een DBMS-systeem

DBMS

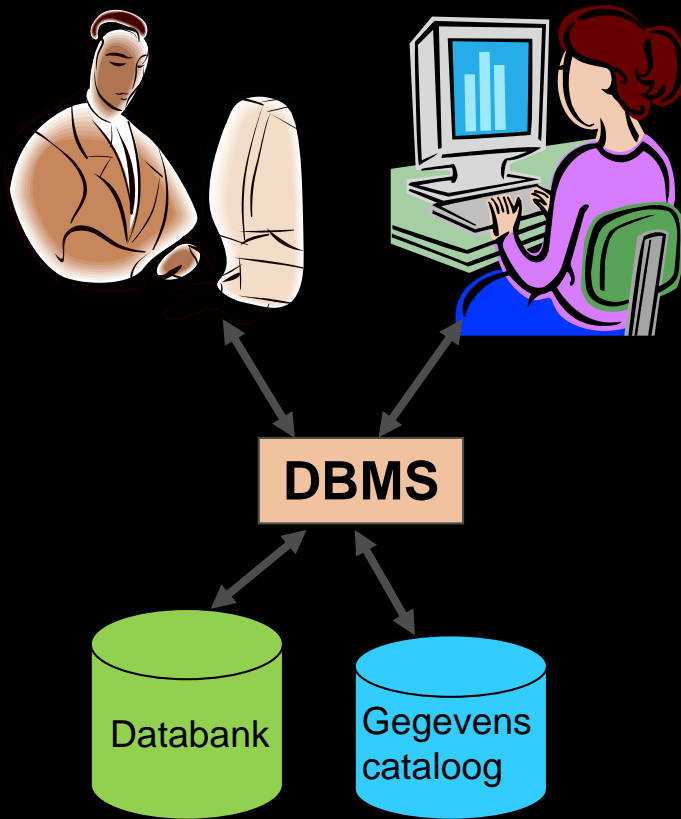


DataBase Management System biedt databasetalen en interfaces aan

- voor het definiëren van gegevens (definities uit de 3 modellen)
- om de gegevens te manipuleren
- voor het bewaken van de integriteit
- voor beveiliging, back-up en recovery
- beheerstools: statistieken omtrent efficiëntie en effectiviteit

DBMS: Onderdelen van een DBMS-systeem

Databank

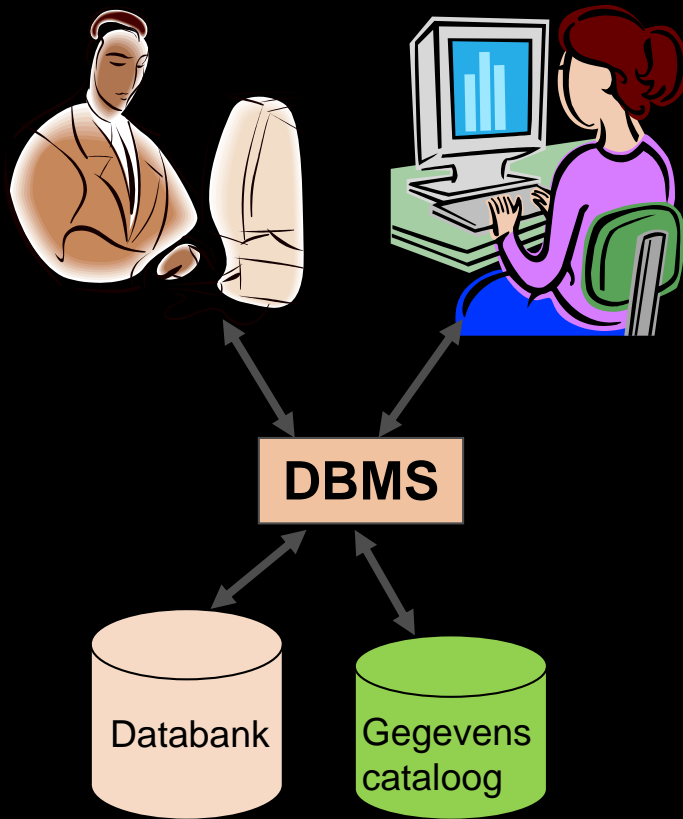


bevat de eigenlijke gegevens die kunnen opgevraagd worden via :

- queries
- DB-programma's

DBMS: Onderdelen van een DBMS-systeem

Gegevenscatalogoog

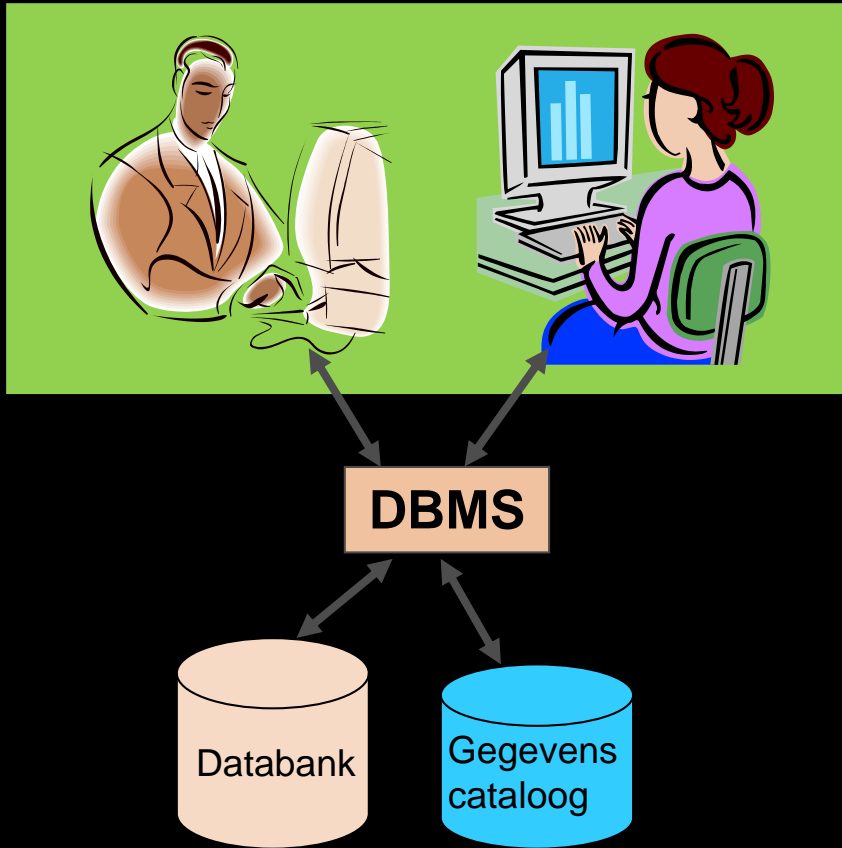


bevat definities van:

- **Logisch model**
= datadefinities van alle gegevens.
- **Fysiek model**
= hoe de gegevens fysiek zijn opgeslagen.
- **Externe modellen**
= view van gebruikers op de gegevens.

DBMS: Onderdelen van een DBMS-systeem

Databaseprogramma



- Computerprogramma dat gegevens uitwisselt met de databank.
- Gebruikers/programma's communiceren nooit rechtstreeks met de databank.
- Gebruikers/programma's passeren steeds via het DBMS.

DBMS: Modellen

- Logisch model

beschrijving van alle gegevens in de DB:

- objecten
- verbanden tussen de objecten
- integriteitsregels

Geen implementatie- of opslagdetails!

- Fysiek model

beschrijving van hoe de gegevens
fysiek opgeslagen zijn.

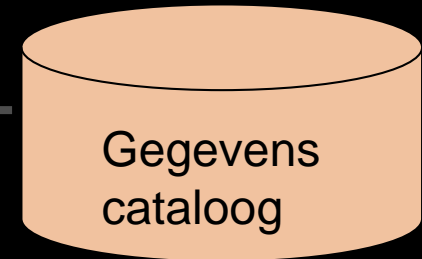
- Externe modellen

deelverzameling van het logisch model voor
één gebruikersgroep of voor één toepassing.



DBMS: Modellen

Logisch model



Student : Tabel

Veldnaam	Gegevenstype	Beschrijving
stamnummer	Numeriek	
voornaam	Tekst	
familienaam	Tekst	
klas	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: Integer

Notatie:

Aantal decimalen: Automatisch

Invoermasker:

Bijschrift:

Standaardwaarde: 0

Validatieregel:

Validatietekst:

Vereist: Nee

Geïndexeerd: Ja (Geen duplicaten)

Infolabels:

Klas : Tabel

Veldnaam	Gegevenstype	Beschrijving
klasid	Tekst	
klastitularis	Numeriek	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: 5

Notatie:

Invoermasker:

Bijschrift:

Standaardwaarde:

Validatieregel:

Validatietekst:

Vereist: Nee

Lengte nul toestaan: Ja

Geïndexeerd: Ja (Geen duplicaten)

Unicode-compressie: Nee

IME-modus: Geen besturingselement

IME-zinmodus: Geen

Infolabels:

Student-Vak : Tabel

Veldnaam	Gegevenstype	Beschrijving
stamn	Numeriek	
vakcode	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: Integer

Notatie:

Aantal decimalen: Automatisch

Invoermasker:

Bijschrift:

Standaardwaarde: 0

Validatieregel:

Validatietekst:

Vereist: Nee

Geïndexeerd: Nee

Infolabels:

Vak : Tabel

Veldnaam	Gegevenstype	Beschrijving
Vakcode	Tekst	
omschrijving	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: 5

Notatie:

Invoermasker:

Bijschrift:

Standaardwaarde:

Validatieregel:

Validatietekst:

Vereist: Nee

Lengte nul toestaan: Ja

Geïndexeerd: Ja (Geen duplicaten)

Unicode-compressie: Ja

IME-modus: Geen besturingselement

IME-zinmodus: Geen

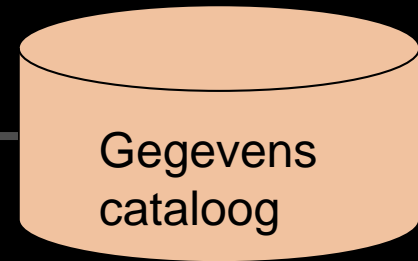
Infolabels:

Beschrijving van alle gegevens in de DB:

- objecten
- verbanden
- integriteitsregels

DBMS: Modellen

Logisch model



Four screenshots of a database design tool showing table definitions for 'Student', 'Klas', 'Student-Vak', and 'Vak'.

Student : Tabel

Veldnaam	Gegevenstype	Beschrijving
stamnummer	Numeriek	
voornaam	Tekst	
familienaam	Tekst	
klas	Tekst	

Veldeigenschappen

Algemeen | Opzoeken

Veldlengte: Integer
Notatie: Automatisch
Aantal decimalen: Automatisch
Invoermasker:
Bijschrift:
Standaardwaarde: 0
Validatiereg:
Validatietekst:
Vereist: Nee
Geïndexeerd: Ja (Geen duplicaten)
Infolabels:

Klas : Tabel

Veldnaam	Gegevenstype	Beschrijving
klasid	Tekst	
klastitularis	Numeriek	

Veldeigenschappen

Algemeen | Opzoeken

Veldlengte: 5
Notatie:
Invoermasker:
Bijschrift:
Standaardwaarde:
Validatiereg:
Validatietekst:
Vereist: Nee
Lengte nul toestaan: Ja
Geïndexeerd: Ja (Geen duplicaten)
Unicode-compressie: Nee
IME-modus: Geen besturingselement
IME-zinmodus:
Infolabels:

Student-Vak : Tabel

Veldnaam	Gegevenstype	Beschrijving
stamn	Numeriek	
vakcode	Tekst	

Veldeigenschappen

Algemeen | Opzoeken

Veldlengte: Integer
Notatie: Automatisch
Aantal decimalen: Automatisch
Invoermasker:
Bijschrift:
Standaardwaarde: 0
Validatiereg:
Validatietekst:
Vereist: Nee
Geïndexeerd: Nee
Infolabels:

Vak : Tabel

Veldnaam	Gegevenstype	Beschrijving
Vakcode	Tekst	
omschrijving	Tekst	

Veldeigenschappen

Algemeen | Opzoeken

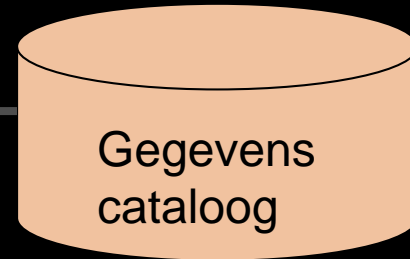
Veldlengte: 5
Notatie:
Invoermasker:
Bijschrift:
Standaardwaarde:
Validatiereg:
Validatietekst:
Vereist: Nee
Lengte nul toestaan: Ja
Geïndexeerd: Ja (Geen duplicaten)
Unicode-compressie: Ja
IME-modus: Geen besturingselement
IME-zinmodus:
Infolabels:

Beschrijving van alle gegevens in de DB:

- objecten
- verbanden
- integriteitsregels

DBMS: Modellen

Logisch model



Beschrijving van alle gegevens in de DB:

- objecten
- verbanden
- integriteitsregels

The screenshot shows a database design tool with four tables: "Student : Tabel", "Klas : Tabel", "Student-Vak : Tabel", and "Vak : Tabel". Each table has a "Veldnaam" (Field Name), "Gegevenstype" (Data Type), and "Beschrijving" (Description) column. Below each table is a "Veldeigenschappen" (Field Properties) section with tabs for "Algemeen" (General) and "Opzoeken" (Search). Arrows indicate relationships between fields in different tables: from "stamnummer" in "Student" to "klasid" in "Klas", from "voornaam" in "Student" to "klastitularis" in "Klas", from "familiennaam" in "Student" to "klastitularis" in "Klas", from "klas" in "Student" to "klasid" in "Klas", from "stamn" in "Student-Vak" to "vakcode" in "Vak", and from "vakcode" in "Student-Vak" to "vakcode" in "Vak".

Veldnaam	Gegevenstype	Beschrijving
stamnummer	Numeriek	
voornaam	Tekst	
familiennaam	Tekst	
klas	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: Integer
Notatie: Automatisch
Aantal decimalen: Automatisch
Invoermasker:
Bijlschrift:
Standaardwaarde: 0
Validatieregels:
Validatietekst:
Vereist: Nee
Geïndexeerd: Ja (Geen duplicaten)
Infolabels:

Veldnaam	Gegevenstype	Beschrijving
klasid	Tekst	
klastitularis	Numeriek	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: 5
Notatie:
Invoermasker:
Bijlschrift:
Standaardwaarde:
Validatieregels:
Validatietekst:
Vereist: Nee
Lengte nul toestaan: Ja
Geïndexeerd: Ja (Geen duplicaten)
Unicode-compressie: Nee
IME-modus: Geen besturingselement
IMF-zinmodus:
Infolabels:

Veldnaam	Gegevenstype	Beschrijving
stamn	Numeriek	
vakcode	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: Integer
Notatie: Automatisch
Aantal decimalen: Automatisch
Invoermasker:
Bijlschrift:
Standaardwaarde: 0
Validatieregels:
Validatietekst:
Vereist: Nee
Geïndexeerd: Nee
Infolabels:

Veldnaam	Gegevenstype	Beschrijving
vakcode	Tekst	
omschrijving	Tekst	

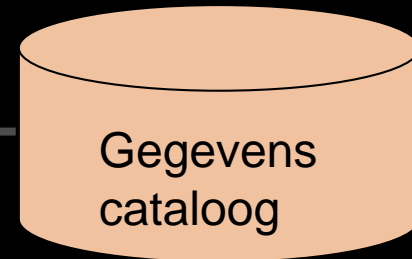
Veldeigenschappen

Algemeen Opzoeken

Veldlengte: 5
Notatie:
Invoermasker:
Bijlschrift:
Standaardwaarde:
Validatieregels:
Validatietekst:
Vereist: Nee
Lengte nul toestaan: Ja
Geïndexeerd: Ja (Geen duplicaten)
Unicode-compressie: Ja
IME-modus: Geen besturingselement
IMF-zinmodus:
Infolabels:

DBMS: Modellen

Logisch model



The image displays four windows from a database design tool, showing the configuration of tables and fields in a logical model.

- Student : Tabel**

Veldnaam	Gegevenstype	Beschrijving
stamnummer	Numeriek	
voornaam	Tekst	
familienaam	Tekst	
klas	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: Integer
Notatie: Automatisch
Aantal decimalen: Automatisch
Invoermasker:
Bijlschrift:
Standaardwaarde: 0
Validatiereg:
Validatietekst:
Vereist: Nee
Geindexeerd: Ja (Geen duplicaten)
Infolabels:
- Klas : Tabel**

Veldnaam	Gegevenstype	Beschrijving
klasid	Tekst	
klastitularis	Numeriek	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: 5
Notatie:
Invoermasker:
Bijlschrift:
Standaardwaarde:
Validatiereg:
Validatietekst:
Vereist: Nee
Lengte nul toestaan: Ja
Geindexeerd: Ja (Geen duplicaten)
Unicode-compressie: Nee
IME-modus: Geen besturingselement
IME-zinmodus: Geen
Infolabels:
- Student-Vak : Tabel**

Veldnaam	Gegevenstype	Beschrijving
stamn	Numeriek	
vakcode	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: Integer
Notatie: Automatisch
Aantal decimalen: Automatisch
Invoermasker:
Bijlschrift:
Standaardwaarde: 0
Validatiereg:
Validatietekst:
Vereist: Nee
Geindexeerd: Nee
Infolabels:
- Vak : Tabel**

Veldnaam	Gegevenstype	Beschrijving
Vakcode	Tekst	
omschrijving	Tekst	

Veldeigenschappen

Algemeen Opzoeken

Veldlengte: 5
Notatie:
Invoermasker:
Bijlschrift:
Standaardwaarde:
Validatiereg:
Validatietekst:
Vereist: Nee
Lengte nul toestaan: Ja
Geindexeerd: Ja (Geen duplicaten)
Unicode-compressie: Ja
IME-modus: Geen besturingselement
IME-zinmodus: Geen
Infolabels:

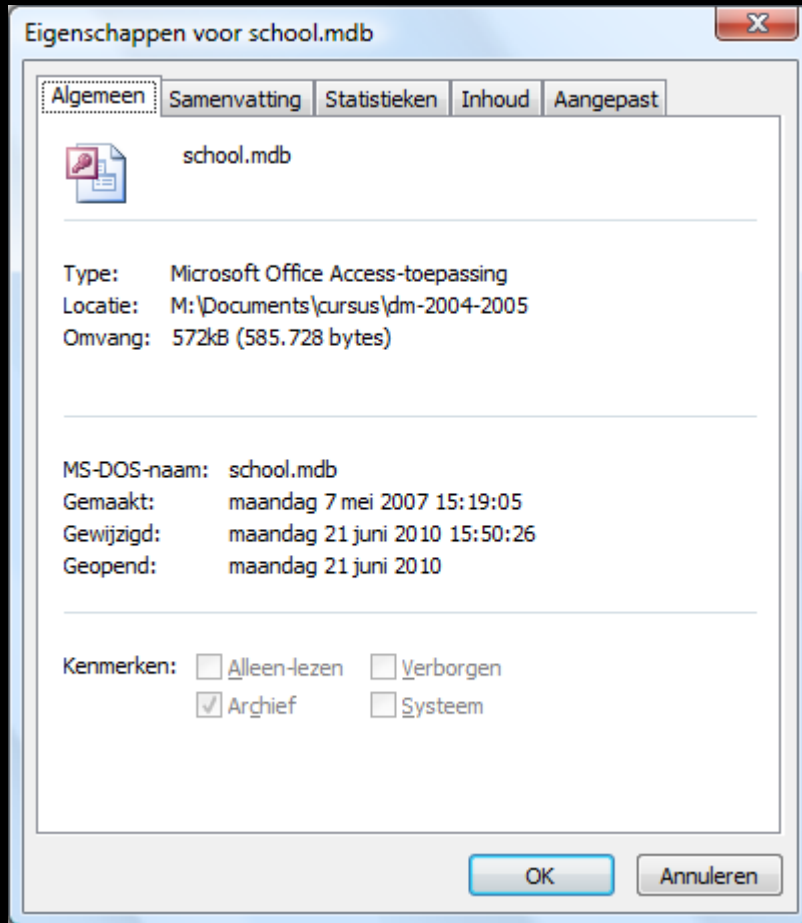
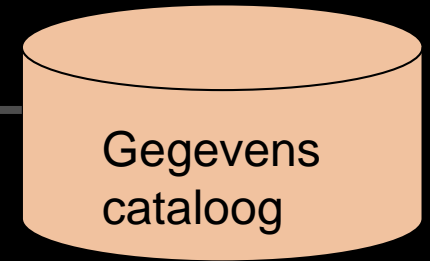
Beschrijving van alle gegevens in de DB:

- objecten
- verbanden
- integriteitsregels

HO
GENT

DBMS: Modellen

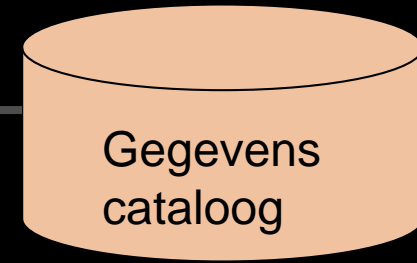
Fysiek model



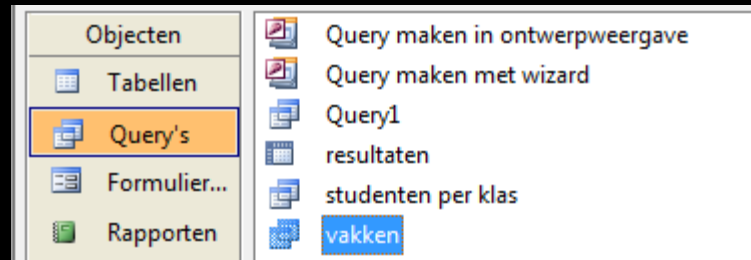
Beschrijving van hoe de gegevens fysiek opgeslagen zijn.

DBMS: Modellen

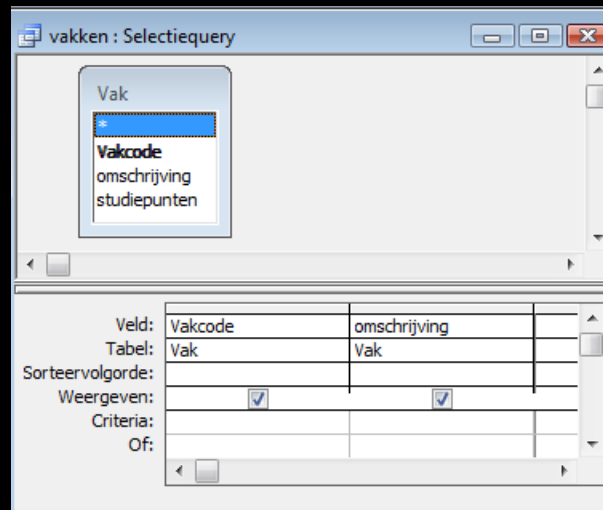
Externe modellen



Views van een welbepaalde groep van gebruikers

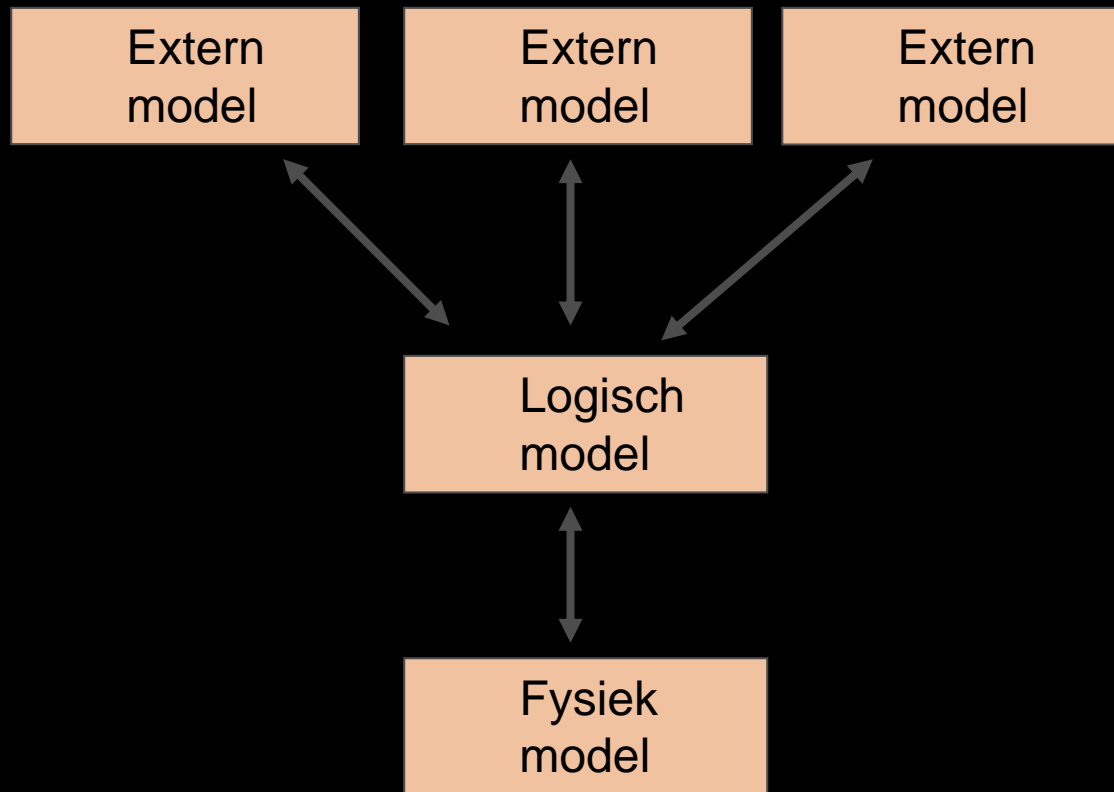


= deelverzameling van logisch model:



DBMS: Modellen

3 lagen



Een wijziging in één laag mag geen gevolgen hebben voor de andere lagen.

DBMS: Rollen binnen een DBMS



Databaseontwerpers

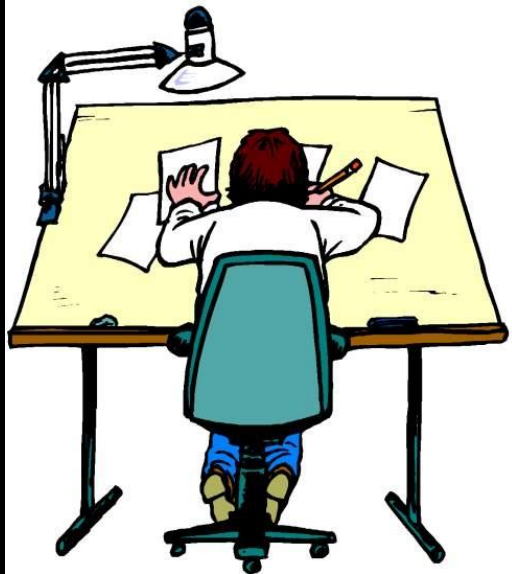


Databasebeheerders



Eindgebruikers

DBMS: Rollen binnen een DBMS



Databaseontwerpers

Ontwerpen van de databank:

- ontwikkelen van de modellen
- bepalen van de constraints
- bepalen van de relaties
-

Wij !

DBMS: Rollen binnen een DBMS



Databasebeheerders

Beheren de database:
optimaliseren van de DB

DBMS: Rollen binnen een DBMS



Eindgebruikers

Gebruiken gegevens uit de databank:

toevoegen,

opvragen,

wijzigen,

verwijderen

DBMS: Voordelen & Nadelen



DBMS: Voordelen

Beheersen van data-redundantie

Stand-alone bestandsstructuren

zelfde informatie in verschillende bestanden omdat verschillende gebruikers
verschillende data-behoeften hebben



(Gedeelde) Databank

views (externe modellen) voor elke groep van gebruikers → geen dubbele
gegevens



DBMS: Voordelen

Beheersen van data-redundantie

Opmerking!

Soms doelbewuste replicatie:

- Verhogen van performantie
- Backups
- Gedistribueerde systemen



DBMS: Voordelen

Consistente data



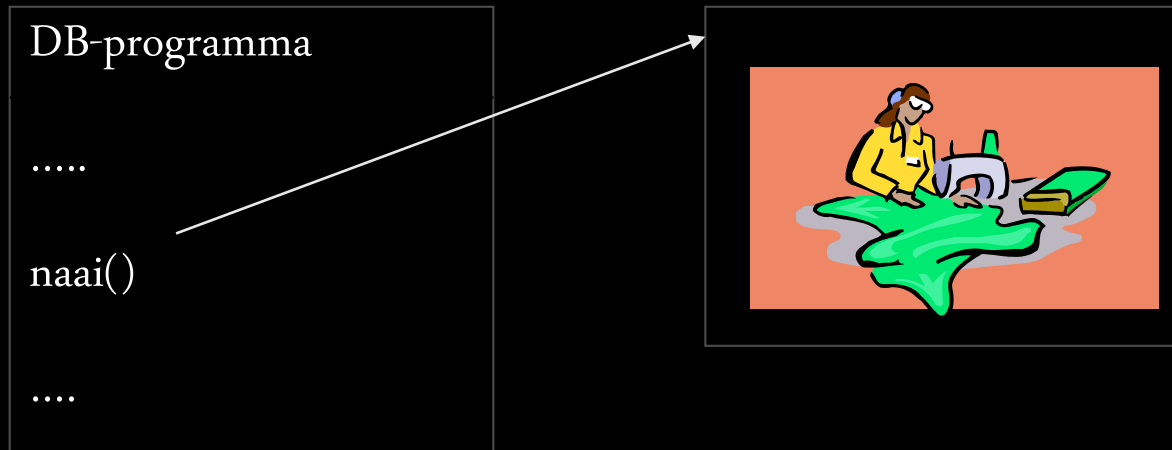
Doordat gegevens slechts éénmaal opgeslagen worden, moet een update van de data slechts één keer gebeuren.

➔ Nieuwe waarde is onmiddellijk beschikbaar voor alle gebruikers

DBMS: Voordelen

Opdrachtonafhankelijkheid

Programma's moeten niet gewijzigd worden als de implementatie van een opdracht wordt veranderd.



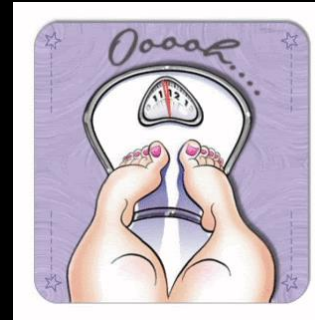
DBMS: Voordelen

Flexibiliteit

Verschillende views (externe modellen) op dezelfde data zijn mogelijk.



Zij



Hij



DBMS: Voordelen

Performantie



- beste toegangspad bepalen
- intelligent verdelen van de gegevens over de opslagmedia
- invoeren van indexen (heeft een index ook nadelen?)
- taak van database-administrator



DBMS: Nadelen

Niks is perfect, ook een DBMS niet ...

- Complex
- Kosten (hardware / software)
- Grote gevolgen bij defect (failsave systeem nodig)

