



Informatie & Databases

Informatie

Wat is informatie en waaruit het bestaat?

Stel op een kaart staat het getal 37 geschreven. Wat kun je dan zeggen van het cijfer 37? Niets bijzonders, toch? Alleen dat het een getal is. Gaat het om een patiëntenkaart dan geeft 37 de lichaamstemperatuur van de patiënt weer. Als op de kaart staat gefeliciteerd met je 37^{ste} verjaardag. Wat geeft de 37 nu aan? In deze situatie is het de leeftijd van een persoon.

We hebben in twee verschillende situaties naar het getal 37 gekeken. Op een patiëntenkaart en op een verjaardagskaart. In beide situaties had het getal 37 een andere betekenis.

We kunnen concluderen uit het bovenstaand voorbeeld dat het getal 37 alleen een **feit** is. Onder feiten verstaan we namen, concrete voorwerpen, toestanden etc.

De *betekenis* van het getal 37 is in de eerste situatie de lichaamstemperatuur en in de tweede betekent het de leeftijd. De betekenissen van een feit noemt men het **gegeven**.

De twee situaties worden de **context** genoemd waarin we naar de feiten kijken.

Informatie bestaat dus uit drie elementen, feiten, gegevens en een context. Waarbij gezegd kan worden dat: "Gegevens zijn de betekenissen van feiten in één bepaalde context."

Over informatie kunnen we dus het volgende zeggen:

**Informatie bestaat uit feiten,
gegevens en een context.**

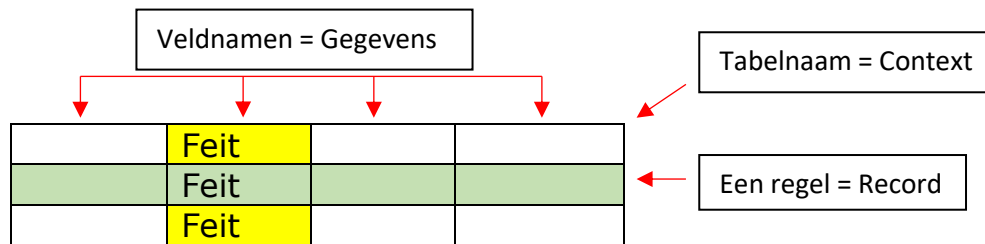
**Gegevens zijn de betekenissen van
feiten in één bepaalde context.**

Tabellen



Informatie & Databases

Computers worden gebruikt om informatie op te slaan. De manier waarop dit in computers gebeurt is met tabellen.



In de afbeelding hierboven zie je hoe de drie onderdelen van informatie worden toegepast op een tabel. De dingen die worden ingevuld in de hokjes zijn de feiten, de (*veld*)namen die we aan elke kolom geven zijn de gegevens. In database termen worden gegevens **attributen** genoemd. de naam van de tabel bepaalt de context. De tabel(naam) wordt een **entiteit** genoemd. Eén regel *unieke* informatie in een tabel is een **record**.

Binnen de wereld van automatisering wordt voor informatie het begrip **Data** gebruikt. Daarom ook de naam Database voor dit soort programma's.

Databases

Database Management Systemen (DBMS) zijn applicaties waarmee informatie, ofwel data, verwerkt wordt door middel van computers. Databases komen tegenwoordig overal voor in mobiele telefoons (contactlijsten), in videospellen, administraties van scholen, banken, doctoren, etc.

Voordelen van het werken met database programma's zijn:

- Grote hoeveelheden data kunnen worden opgeslagen en worden beheerd met deze applicaties.
- Opvragen, zoeken en muteren (aanpassen of verwijderen) van informatie is snel, moet eenvoudig en nauwkeurig gedaan worden.
- Informatie is op alle mogelijke manieren te ordenen.
- Er kunnen op relatief eenvoudige wijze selecties worden gemaakt uit de data.

Voor we echter verder kijken naar hoe een database werkt moeten we eerst definiëren wat nu precies informatie is en waaruit het bestaat.



Informatie & Databases

Basis functies

Alle Database Management Systemen (DBMS) hebben vier basis functies.

1. Tabellen

De tabellen bevatten alle data in de database en vormen de structuur van de informatie. Elke DBMS heeft de mogelijkheid om tabellen aan te maken en te beheren en relaties te kunnen leggen tussen de tabellen.

2. Forms

Elke DBMS heeft de mogelijkheid om formulieren/schermen aan te maken die verbonden worden met de tabellen. De gebruikers van de database kunnen dan doormiddel van deze schermen data verwerken in de tabellen. De gebruikers hebben geen directe toegang tot de tabellen en de data in de tabellen.

3. Query's

Met behulp van een query (vraag) is het mogelijk om een selectie (deelverzameling) van gegevens uit de database te selecteren en aan de gebruiker te tonen. Met een query kun je informatie uit meerdere tabellen samenvoegen.

4. Rapporten

Met behulp van de rapport functie kunnen er rapporten gemaakt worden die dan kunnen worden afgedrukt.

Informatie in Access

In Access net als alle andere database programma's wordt er gebruik gemaakt van tabellen. Hiernaast zie je een weergave van zo'n tabel in Access, in de *Ontwerpweergave (Design view)*.

Leerling	
Field Name	Data Type
Stamboeknummer	AutoNumber
Voornaam	Short Text
Achternaam	Short Text
Adres	Short Text
EMail	Short Text

De naam van de tabel leerling is de context van de informatie die deze tabel bevat/zal bevatten.

De veldnamen zijn de gegevens waaruit de informatie zal bestaan binnen de context van deze tabel.

De feiten zijn de informatie die in de tabel zijn ingevoerd.

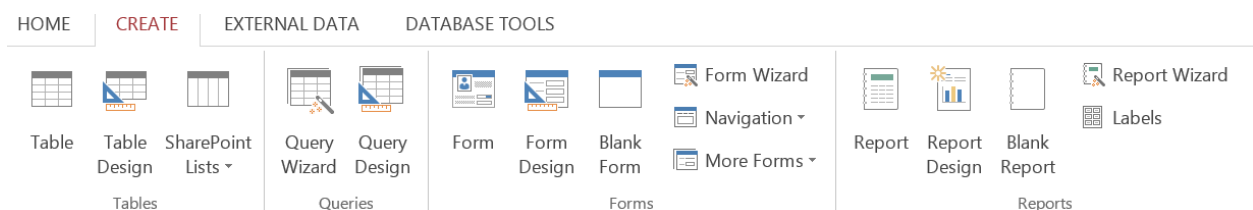


Informatie & Databases

Leerling				
Stamboeknu	Voornaam	Achternaam	Adres	EMail
1	Jan	Jansen	Pita straat 5	JJ@setarnet.aw
2	Juancho	Ras	Ponton 135A	JR@hotmail.com
3	Maria	Croes	Wayaca 7B	SweetMaria@yahoo.com

Diagram illustrating the concept of a record (Feiten) in a table. A blue box labeled 'Feiten' has arrows pointing to the first three rows of the table, indicating that each row represents a single record.

Eén regel in een tabel wordt een **record** genoemd.



In de afbeelding hier boven zie je de functies in het *Lint* van Access.

Relaties

De eenvoudigste vorm van een database is een platte tabel, ofwel een *flat table*. In deze vorm wordt alle data in één tabel bewaard, dat als voordeel heeft dat het zeer eenvoudig is.

Echter zo'n platte tabel heeft meerdere nadelen. Wanneer je van leerlingen bijvoorbeeld de behaalde cijfers wil gaan bijhouden dan merk je al gauw dat je veel data herhaaldelijk moet opschrijven om alles precies bij te houden.

Stamboek	Voornaam	Achternaam	Schooljaar	Datum	Vak	Type	Cijfer
00000001	Juancho	Ras	2013-2014	12/08	WI	CP	5.7
00000001	Juancho	Ras	2013-2014	12/08	NE	CP	7.7
00000001	Juancho	Ras	2014-2015	10/10	WI	CP	6.4
00000002	Maria	Croes	2013-2014	12/08	WI	CP	5.7

Je ziet in de bovenstaande tabel dat de feiten in de velden Stamboek Voornaam Achternaam steeds herhaald moeten worden om een cijfer per leerling te identificeren.

Daar de middelen (resources) van een computer beperkt zijn, is er gezocht naar een efficiëntere manier om data in tabellen op te slaan. Er wordt niet meer van één, maar van meerdere tabellen gebruik gemaakt waar bij de tabellen met elkaar in relatie staan. Deze vorm van databases wordt een **relatieve database** genoemd.

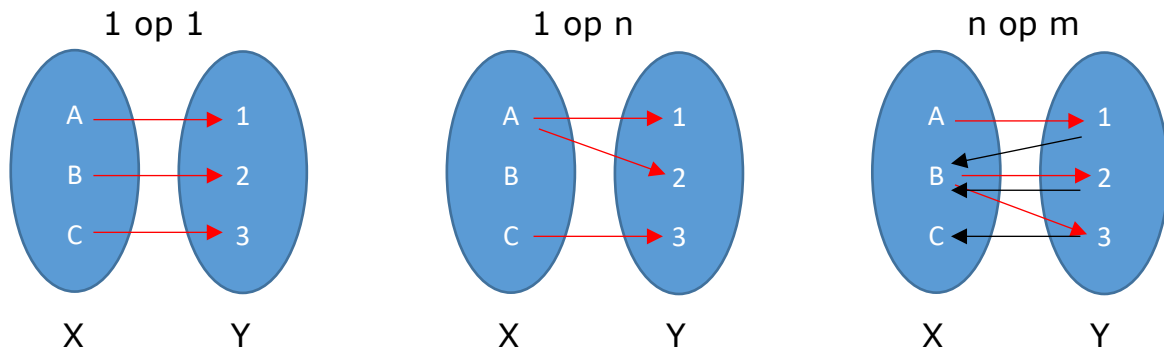


Informatie & Databases

Bij relationele databases (RDB) wordt de informatie opgesplitst in twee of meerder tabellen om zo de efficiëntie van het opslaan van informatie te verbeteren, maar ook het kunnen aanpassen van informatie, **maken** genoemd, gaat veel eenvoudiger.

Cardinaliteit, ofwel Relatie typen

Er worden drie relatie typen onderscheiden:



1 op 1 relatie:

Elk element in verzameling X heeft één of geen koppeling met één element in verzameling Y. Of wel elk element in verzameling Y is afhankelijk van een element in verzameling X.

1 op n relatie:

Elk element in verzameling X heeft nul, één of meerdere koppelingen met verschillende elementen in verzameling Y.

n op m relatie:

Elk element in verzameling X heeft nul, één of meerdere koppelingen met verschillende elementen in verzameling Y. Tevens heeft elk element in verzameling Y nul, één of meerdere koppelingen met verschillende elementen in verzameling X.

Normaliseren

Om de informatie op een goede manier in verschillende tabellen te brengen zijn er verschillende methodes ontwikkeld. Een daar van is het normaliseren.

Er zijn in totaal 5 normaalvormen bij het normaliseren, maar de eerste drie worden het meest gebruikt.

Een reden om alleen de eerste drie regels te gebruiken is dat als er verder genormaliseerd wordt de database te gefragmenteerd wordt (te veel losse tabellen). Hierdoor zal het functioneren van de database niet sneller en efficiënter worden maar juist trager.



Informatie & Databases

SQL

Om een database te maken en te bewerken wordt er een speciale programmeertaal gebruikt, deze heet **Structured Query Language**, ofwel **SQL**.

De taal SQL kan worden opgedeeld in drie componenten **Data Definition Language** (DDL), **Data Manipulation Language** (DML), en **Data Control Language** (DCL).

De tabel hieronder geeft een overzicht van de instructies uit elk deel van SQL.

DDL	DML	DCL
CREATE TABLE	INSERT INTO	ALTER DATABASE
DROP TABLE	SELECT INTO	CREATE GROUP
ALTER TABLE	UPDATE	DROP GROUP
CREATE INDEX	DELETE	CREATE USER
	SELECT	ALTER USER
	UNION	DROP USER
	TRANSFORM	ADD USER
	PARAMETER	GRANT PRIVILEGE
		REVOKE PRIVILEGE

DDL wordt gebruikt om tabellen te creëren en relaties te maken tussen de tabellen.

DML wordt gebruikt om operaties uit te voeren op de data. Zoals het opvragen van, het verwijderen van, het wijzigen van en het navigeren door de data.

DCL wordt gebruikt om beveiligingen aan te brengen aan de data in een database.

In de volgende lessen zullen we aandacht besteden aan het DDL en DML deel van SQL en we zullen deze gaan toepassen met behulp van het programma Microsoft Access. Binnen Access is het niet mogelijk gebruik te maken van DCL, dit kan echter wel via programmeertalen als Visual Basic of C#.

Tevens zullen we zien hoe de tabellen gekoppeld kunnen worden met formulieren die voor de gebruikers van de database bedoeld zijn.