



Hoofdstuk 1

Inleiding

Dit hoofdstuk biedt een inleiding in databaseverwerking. We nemen aan dat je op zijn minst enige kennis hebt van het werken met databases. We gaan ervan uit dat je met een product als Microsoft Access hebt gewerkt voor het invoeren van gegevens in formulieren, voor het maken van rapporten en misschien voor het uitvoeren van zoekopdrachten. Als je nog nooit dit soort dingen hebt gedaan, raden we je aan Microsoft Access 2007 aan te schaffen en de handleiding in bijlage A van dit boek door te werken.

1.1 De kenmerken van databases

Het doel van een database is mensen helpen gegevens bij te houden. Het meest gebruikte soort database is een **relationele database**. We gaan in hoofdstuk 3 dieper in op relationele databases, voor dit moment is het genoeg om een paar elementaire dingen over databases te weten.

Om te beginnen bergt een relationele database data op in tabellen. **Data** zijn gegevens zoals naam en adres, en getallen. Een **tabel** heeft rijen en kolommen, net als een spreadsheet. Een database bevat in principe meer tabellen. Elke tabel bevat gegevens over iets anders. Je ziet bijvoorbeeld een database met twee tabellen in Figuur 1-1. De tabel STUDENT bevat gegevens over studenten, de tabel VAK bevat gegevens over schoolvakken.

STUDENT			
Studentnummer	Achternaam	Voornaam	E-mailadres
100	Kok	Sam	Kok@OnzeU.edu
200	Lerik	Marcia	Lerik@OnzeU.edu
300	Holman	Louis	Holman@OnzeU.edu
400	Groen	Gerda	Groen@OnzeU.edu

VAK			
Vaknummer	Vaknaam	Trimester	Sectie
10	Scheikunde 1	E04	1
20	Scheikunde 1	E04	2
30	Scheikunde 1	T05	1
40	Accounting 1	E04	1
50	Accounting 2	T05	1

Figuur 1-1 De tabellen STUDENT en VAK

Elke rij van de tabel STUDENT bevat gegevens over één student: Kok, Lerik, Holman en Groen. Elke rij van de tabel VAK bevat op een soortgelijke manier gegevens over een specifiek vak. Een specifieke student of een bepaald vak heet ook wel een *instantie*. De gegevens van een bepaalde instantie die in een rij staan, noemen we ook wel een **record**. Elke **kolom** van een tabel bevat een bepaald kenmerk dat de instanties met elkaar gemeen hebben. Zo bevat de eerste kolom van STUDENT het Studentnummer, de tweede kolom de Achternaam, enzovoort.

1.1.1 Iets over naamconventies

Namen van tabellen schrijven we in dit boek in hoofdletters. Deze conventie maakt het je makkelijk namen van tabellen te herkennen. Je bent echter niet verplicht namen van tabellen in hoofdletters te schrijven. Microsoft Access en soortgelijke programma's staan je toe de naam van een tabel te schrijven zoals jij dat wilt, dus als STUDENT, student, Student, stuDent, of op elke andere gewenste manier.

Namen van kolommen beginnen in dit boek altijd met een hoofdletter. Ook dit is niet verplicht en heeft alleen als doel om de kolommen beter herkenbaar te maken. We zullen soms samengestelde kolomnamen gebruiken, waarbij we elk element met een hoofdletter beginnen om de leesbaarheid te bevorderen. Zoals AantalInBestelling of KostenKoekjes. Deze schrijfwijze is ook alleen weer een handige conventie. Consequent gebruik ervan versimpelt het interpreteren van databasestructuren. Zo weet je bijvoorbeeld altijd dat STUDENT de naam is van een tabel, en dat Student de naam is van een kolom van een tabel.

1.1.2 Een database omvat gegevens en relaties

Figuur 1-1 laat zien hoe gegevens in de tabellen van een database zijn opgeslagen, maar een database is niet compleet als de relaties tussen de rijen van de verschillende tabellen niet duidelijk zijn.

Het belang hiervan is in Figuur 1-2 te zien. De tabel bevat een aantal cijfers maar de relaties ontbreken. De gegevens in CIJFER zijn nutteloos als je niet weet welk cijfer bij welke student hoort. Het is alsof je een sportverslaggever alleen de uitslagen hoort voorlezen: 1-1, 2-0, 2-3, 5-2. Dergelijke uitslagen zijn nutteloos als je niet weet over welke wedstrijden het gaat. Een database bevat daarom niet alleen gegevens maar ook de relaties tussen die gegevens.

CIJFER
Cijfer
7,7
7,5
7,7
7,1
7,9
7,5

Figuur 1-2 De tabel CIJFER

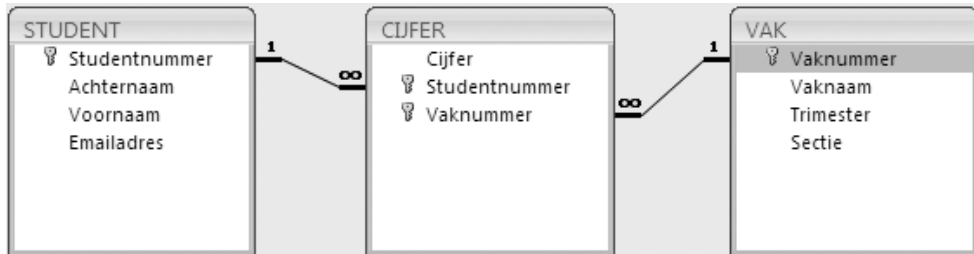
In Figuur 1-3 zie je de complete tabel met niet alleen cijfers, maar ook met de relaties naar de tabellen uit Figuur 1-1. Bijvoorbeeld Studentnummer 400 (Groen) heeft het cijfer 7,9 gekregen in Vaknummer 40 – en dat is ‘Accounting 1’. Hij heeft verder het cijfer 7,5 in Vaknummer 50 – dat is ‘Accounting 2’.

CIJFER		
Studentnummer	Vaknummer	Cijfer
100	10	7,7
100	40	7,5
200	20	7,7
300	30	7,1
400	40	7,9
400	50	7,5

Figuur 1-3 De tabel CIJFER met relaties

Figuur 1-1 en Figuur 1-3 laten nog een ander belangrijk aspect van databases zien: elke rij bevat een unieke waarde, de zogeheten *primaire sleutel*. Deze waarde gebruiken we om de relatie tussen de tabellen aan te geven. In de tabel STUDENT is Studentnummer de primaire sleutel. Elke waarde van Studentnummer is uniek, komt dus maar één keer voor, en hoort bij precies één student. Zo hoort Studentnummer 100 bij student Kok. Op dezelfde manier heeft elk vak in de tabel VAK een eigen Vaknummer.

Als je Figuur 1-3 en 1-4 vergelijkt zie je dat aan elke rij van CIJFER de primaire sleutel van STUDENT en van VAK is toegevoegd (Studentnummer en Vaknummer). De combinatie van Studentnummer en Vaknummer maken elke rij in CIJFER uniek, we noemen dit een *vreemde sleutel* (of externe sleutel of foreign key). De sleutel heet *vreemd* omdat hij afkomstig is uit een andere (= vreemde) tabel. De vreemde sleutel geeft de relatie tussen de tabellen aan.



Figuur 1-4 STUDENT, CIJFER EN VAK in Access 2007

Figuur 1-4 laat de manier zien waarop Microsoft Access 2007 tabellen en hun onderlinge relaties toont. Het sleutelsymbool geeft de primaire en vreemde sleutels aan. De symbolen bij de verbindingslijnen die de relatie aangeven (het cijfer 1 en het symbool voor oneindig ∞) geven bijvoorbeeld aan dat één student gekoppeld kan zijn aan veel cijfers.

1.1.3 Databases produceren informatie

Je weet wellicht wat het verschil is tussen gegevens en informatie. Gegevens zijn opgeslagen cijfers of feiten. Informatie is kennis die is afgeleid van gegevens. Anders gezegd: informatie is weergave van gegevens in een zinvolle context.

Databases slaan cijfers en feiten – dus gegevens – op. Dat wordt op een manier gedaan die het mogelijk maakt informatie te produceren. De gegevens in Figuur 1-3 kunnen we gebruiken om het gemiddelde cijfer van een student te produceren, het gemiddelde cijfer van een vak, het gemiddelde aantal studenten per vak, enzovoort. In hoofdstuk 2 leer je de taal Structured Query Language (SQL) kennen, waarmee je op basis van databasegegevens informatie kunt produceren.

Samenvattend: Databases slaan gegevens op in tabellen en representeren de relaties tussen de rijen van die tabellen. Dat gebeurt op een manier die het mogelijk maakt informatie te produceren.

1.2 Voorbeelden van databases

Databases maken tegenwoordig deel uit van vrijwel elk informatiesysteem. Dat is ook niet verrassend als we bedenken dat elk informatiesysteem gegevens en relaties daartussen moet opslaan.

1.2.1 Voorbeelden van databasetoepassingen

In Figuur 1-5 zijn enkele toepassingen van databases genoemd.

Toepassing	Gebruikt door bijvoorbeeld	Aantal gebruikers	Gebruikelijke grootte
Beheer van verkoopcontacten	Verkoper	1	2000 rijen
Afspraken voor patiënten (arts, tandarts)	Medisch kantoorpersoneel	15 tot 50	100.000 rijen
Customer Relationship Management (CRM)	Verkoop-, marketing- of klantenserviceafdelingen	500	10 miljoen rijen
Enterprise Resource Processing (ERP)	Een hele organisatie	5000	Meer dan 10 miljoen rijen
Site voor e-commerce	Internetgebruikers	Kan miljoenen zijn	Meer dan 1 miljard rijen
Digitaal dashboard	Leidinggevend personeel	500	100.000 rijen
Datamining	Zakelijke analyse	25	100.000 tot meerdere miljoenen

Figuur 1-5 Voorbeelden van het gebruik van databases

1.2.2 Single-user-databasetoepassingen

De eerste applicatie in Figuur 1-5 wordt door een enkele verkoper gebruikt. Ze houdt ermee bij welke klanten ze belde en wat voor contact ze met hen had.

1.2.3 Multi-user-databasetoepassingen

De volgende toepassingen in Figuur 1-5 zijn allemaal bedoeld voor meer dan één gebruiker. De applicatie voor het plannen van afspraken voor patiënten kan bijvoorbeeld vijftien tot vijftig gebruikers hebben. Dat zullen secretaresses zijn, kantoorbeheerders, dokters, tandartsen, verpleegsters, enzovoort. Een dergelijke database zal misschien zo'n 100.000 rijen gegevens in vijf tot tien verschillende tabellen omvatten.

Wordt een databaseapplicatie door meer dan één werknemer gebruikt, dan bestaat altijd de kans dat het werk van de ene gebruiker in conflict komt met dat van een andere gebruiker. Twee secretaresses kunnen bijvoorbeeld een afspraak op hetzelfde tijdstip inplannen voor twee verschillende patiënten. Er zijn speciale mechanismen voor het coördineren van activiteiten in de database en het voorkomen van dergelijke conflicten. Je leert deze mechanismen kennen in hoofdstuk 9.

De derde rij in Figuur 1-5 toont een nog grotere databaseapplicatie. Customer Resource Management (CRM) is een informatiesysteem dat contacten met klanten beheert van de eerste aanvraag tot de acceptatie, aankoop, volgende aankopen, ondersteuning, enzovoort. CRM-systemen worden gebruikt door verkopers,

bedrijfsleiders, klantenservice en ander personeel. Een CRM-database van een wat groter bedrijf zou vijfhonderd gebruikers kunnen hebben en 10 miljoen of meer rijen in vijftig of meer tabellen.

Enterprise Resource Planning (ERP) is een informatiesysteem waar alle afdelingen van een fabricagebedrijf gebruik van maken, bijvoorbeeld verkoop, inventarisbeheer, productplanning, inkoop en andere bedrijfsfuncties. SAP is de grootste leverancier van ERP-applicaties en heeft als een van de belangrijkste productonderdelen een database die gegevens van deze diverse bedrijfsfuncties integreert. Een ERP-systeem kan vijfduizend of meer gebruikers hebben en meer dan 100 miljoen rijen in honderden tabellen.

1.2.4 E-commerce-databasetoepassingen

E-commerce is ook een belangrijke toepassing van databases. Databases worden gebruikt voor het invoeren van bestellingen, factureren, verzenden en klantenondersteuning. De grootste databases bij een e-commercesite zijn verrassend genoeg niet de databases voor het verwerken van bestellingen, maar die voor het bijhouden van het browsergedrag van klanten. De meeste prominente e-commercebedrijven, zoals Amazon.com (*www.amazon.com*), houden bij welke webpagina's en onderdelen van webpagina's hun klanten bekijken. Ze houden ook bij waarop de klanten klikken, wat ze aan hun boodschappenkarretjes toevoegen, wat ze bestellen, welke boodschappenkarretjes ze annuleren, enzovoort. E-commercebedrijven gebruiken databases met webactiviteiten om te bepalen welke webpagina-items populair en succesvol zijn en welke niet. Ze voeren ook experimenten uit om te bepalen of een paarse achtergrond meer bestellingen oplevert dan een blauwe achtergrond, enzovoort. Dergelijke databases voor gebruik van het web zijn enorm.

1.2.5 Rapportage en datamining

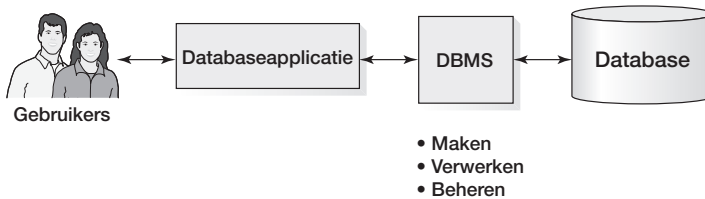
De laatste twee voorbeelden uit Figuur 1-5 zijn digitale dashboards en applicaties voor datamining. Deze applicaties gebruiken de bestaande gegevens en vatten deze samen om de gegevens inzichtelijker te maken of nieuwe informatie te creëren als hulp voor de bedrijfsleiding. Digitale dashboards en andere rapportsystemen maken overzichten van de voormalige en huidige bedrijfsprestaties. Applicaties voor datamining voorspellen de toekomstige prestaties. In hoofdstuk 15 kijken we naar dergelijke applicaties.

Opmerking

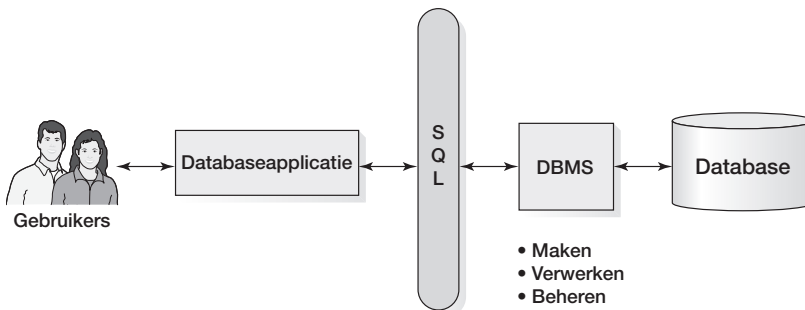
Dat een database klein is, betekent niet dat deze ook een eenvoudige structuur heeft. Neem bijvoorbeeld een bedrijf dat per jaar voor € 1 miljoen aan onderdelen verkoopt en een bedrijf dat per jaar voor € 100 miljoen aan onderdelen verkoopt. Deze bedrijven hebben soortgelijke databases, ongeacht het verschil in omzet. Ze hebben beide dezelfde soorten gegevens, ongeveer hetzelfde aantal tabellen met gegevens en hetzelfde niveau aan ingewikkeldheid in de gegevensrelaties. Het enige verschil tussen deze twee zit in de hoeveelheid gegevens. Een database voor een klein bedrijf mag dan misschien wel klein zijn, het betekent niet noodzakelijk dat deze ook eenvoudig is.

1.3 De onderdelen van een databasesysteem

Zoals je in Figuur 1-6 ziet, bestaat een databasesysteem gewoonlijk uit vier verschillende onderdelen: gebruikers, de databaseapplicatie, het databasemanagementsysteem (DBMS) en de database zelf. Omdat de **Structured Query Language (SQL)** een belangrijke internationale standaardtaal is die alle commerciële DBMS-producten ondersteunen, kunnen we het plaatje van het databasesysteem verfijnen tot dat van Figuur 1-7.



Figuur 1-6 De onderdelen van een databasesysteem



Figuur 1-7 De onderdelen van een databasesysteem met SQL

In Figuur 1-7 zie je aan de rechterkant de **database**, een verzameling van onderling gerelateerde tabellen en andere structuren. Het **databasemanagementsysteem (DBMS)** is een groot en ingewikkeld computerprogramma om de database te maken, te gebruiken en te beheren. Het DBMS ontvangt opdrachten die gecodeerd zijn in SQL en vertaalt deze opdrachten in acties op de database.

Een **databaseapplicatie** is een verzameling van een of meer computerprogramma's die als intermediair fungeert tussen de gebruiker en het DBMS. De applicatie leest of verandert de gegevens in een database door SQL-opdrachten naar het DBMS te sturen. Een applicatie kan gegevens aan de gebruiker tonen in de vorm van invoerformulieren en rapporten.

De **gebruikers** zijn het vierde onderdeel van het databasesysteem. Zij gebruiken de databaseapplicatie om allerlei gegevens bij te houden.

1.3.1 Databaseapplicaties en SQL

De basisfuncties van databaseapplicaties zijn:

- maken en verwerken van invoerformulieren;
- verwerken van query's;
- maken en verwerken van rapporten;
- uitvoeren van de applicatielogica;
- beheer voeren over de hele applicatie.

In het vervolg van dit boek behandelen we bovengenoemde punten uitvoerig, maar een paar ervan lichten we hier kort toe.

VAK

Vaknummer: 40

Vaknaam: Accounting 1

Trimester: E04

Sectie: 1

STUDENT

Studentnum	Achternaam	Voornaam	Emailadres
100	Kok	Sam	Kok@OnzeU.edu
400	Groen	Gerda	Groen@OnzeU.edu
*			

Record: 1 van 2 | Zoeken

Figuur 1-8 Invoerformulier

In Figuur 1-8 zie je een typisch invoerformulier voor de student-vak-cijferdatabase uit de vorige paragrafen. Merk op dat dit formulier de structuur van de onderliggende tabellen voor de gebruiker verbergt. Het doel van dit formulier is dan ook, net als dat van alle invoerformulieren, de gebruikers de gegevens in een vorm te presenteren die voor hen bruikbaar en zinvol is, ongeacht de onderliggende tabelstructuur.

Een belangrijke functie van de databaseapplicatie is het verwerken van query's. De applicatie stuurt de query in de vorm van een SQL-opdracht aan het DBMS. Hieronder zie je een voorbeeld van een SQL-opdracht voor het verwerken van de gegevens uit de STUDENT-tabel van Figuur 1-1:


```

SELECT    Achternaam, Voornaam, E-mailadres
FROM      STUDENT
WHERE     Studentnummer > 200;

```

Dit SQL-statement haalt de namen en e-mailadressen op van alle studenten die een Studentnummer hebben dat groter is dan 200. Uitgaande van de gegevens in Figuur 1-1 levert dit statement de Achternaam, de Voornaam en het E-mailadres van de studenten Holman en Groen.

1.3.2 Het DBMS

Het DBMS, of databasemanagementsysteem, verwerkt SQL-opdrachten en maakt, verwerkt en beheert de database. Een DBMS is een groot en ingewikkeld computerprogramma. Er zijn maar weinig organisaties die hun eigen DBMS-programma's schrijven. In plaats daarvan schaffen ze licenties aan voor DBMS-producten bij grote databaseleveranciers. Een van de DBMS-producten is Microsoft Access. Andere commerciële DBMS-producten zijn Oracle Database van Oracle, DB2 van IBM, SQL Server van Microsoft en MySQL van Sun Microsystems/Oracle. Er bestaan tientallen andere DBMS-producten maar deze vijf hebben het leeuwendeel van de markt.

1.3.3 De database

Het laatste onderdeel dat je in Figuur 1-7 ziet, is de database. Een database is een zichzelf beschrijvende verzameling van geïntegreerde tabellen. **Geïntegreerde tabellen** zijn tabellen die zowel gegevens als de relaties tussen deze gegevens bevatten. De tabellen in Figuur 1-3 zijn *geïntegreerd*, want ze slaan niet alleen gegevens op over studenten, vakken en cijfers, maar ook gegevens over de relaties tussen de rijen van verschillende tabellen.

Een database is *zichzelf beschrijvend*, want deze bevat een beschrijving van zichzelf. Dat wil zeggen dat databases niet alleen tabellen bevatten met gegevens van gebruikers, maar ook tabellen die deze gegevens beschrijven. Dergelijke gegevens noemen we **metagegevens**, omdat dit gegevens over gegevens zijn. De vorm en de indeling van de metagegevens variëren van DBMS tot DBMS.

Figuur 1-9 toont algemene metagegevenstabellen, die de tabellen en de kolommen beschrijven van de database uit Figuur 1-4.

Tabelnaam	AantalKolommen	PrimaireSleutel
STUDENT	3	Studentnummer
VAK	4	Vaknummer
CIJFER	3	(Studentnummer, Vaknummer)

Kolomnaam	Tabelnaam	Datatype	Lengte (bytes)
Studentnummer	STUDENT	Integer	4
Achternaam	STUDENT	Text	25
Voornaam	STUDENT	Text	25
E-mailadres	STUDENT	Text	100

Kolomnaam	Tabelnaam	Datatype	Lengte (bytes)
Vaknummer	VAK	Integer	4
Naam	VAK	Text	25
Trimester	VAK	Text	12
Sectie	VAK	Integer	4
Studentnummer	CIJFER	Integer	4
Vaknummer	CIJFER	Integer	4
Cijfer	CIJFER	Decimal	(2,1)

Figuur 1-9 Een voorbeeld van metagegevenstabellen

Je kunt de metagegevens bekijken om te bepalen of bepaalde tabellen, kolommen, indexen of andere structuren in een database bestaan. Het volgende statement vraagt SQL Server bijvoorbeeld in de metagegevenstabel SYSOBJECTS na te kijken of er een gebruikerstabel (Type = 'U') met de naam VAK bestaat in de database. Is dat het geval, dan wordt deze tabel uit de database verwijderd.

```

If Exists
    (SELECT *
    FROM SYSOBJECTS
    WHERE [Name] = 'VAK'
    AND Type = 'U')
DROP TABLE VAK;

```

Maak je niet druk over de syntaxis van dit statement. Je leert verderop in dit boek wat dit statement precies betekent en hoe je zelf dergelijke statements schrijft. Je hoeft hier alleen te weten dat dit een van de manieren is waarop databasebeheerders gebruikmaken van metagegevens.

1.4 Persoonlijke versus bedrijfs-databasesystemen

We kunnen databasesystemen en DBMS-producten in twee groepen verdelen: persoonlijke databasesystemen en bedrijfssystemen.

1.4.1 Wat is Microsoft Access?

Voordat we verdergaan, moeten we een veelvoorkomende misvatting uit de weg ruimen: Microsoft Access is *niet* alleen een DB. Het is een **persoonlijk databasesysteem**: een DBMS *plus* een applicatiegenerator. Microsoft Access omvat een DBMS, dat de database maakt, verwerkt en beheert. Het omvat ook onderdelen voor formuleren, rapporten en query's, die samen de applicatiegenerator van Microsoft Access vormen.

De onderdelen van Microsoft Access zie je in Figuur 1-10. De applicaties van Microsoft Access maken intern gebruik van SQL voor het communiceren met het DBMS. De DBMS-engine (de 'motor') van Access heet Jet. Je zult maar zelden iets over Jet horen, want Microsoft verkoopt Jet niet als een afzonderlijk product.