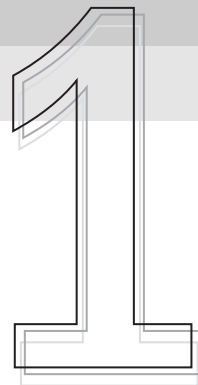


HOOFDSTUK 1

INLEIDING



Dit boek gaat over de *Structured Query Language*, of kortweg SQL, zoals er meestal over wordt gesproken. SQL, dat wordt uitgesproken als het Engelse Sequel of eenvoudigweg S-Q-L, is een computerprogrammeertaal die gebruikt wordt voor het doorzoeken (*querying*) van relationele databases op een *non-procedurele* manier. Wanneer je met SQL informatie uit een database haalt, wordt dat *querying* van de database genoemd. De term *non-procedureel* betekent dat je voor het tevoorschijn halen van de informatie kunt volstaan met het systeem te vertellen welke informatie je zoekt, zonder dat je daarvoor ook het systeem instructies hoeft te geven hoe het systeem dat zou moeten doen. Mogelijk weet je op dit moment nog niet precies wat een relationele database is. In dit hoofdstuk zullen we daarom enkele basisconcepten van relationele databases behandelen. Ons belangrijkste doel is om je de basisconcepten van SQL, een van de meestgebruikte talen voor het werken met databases, te leren. Hoewel SQL een gemakkelijk te leren taal is en het eenvoudig is om query's te schrijven, is de taal ook complex en is het mogelijk om met een overzichtelijke hoeveelheid opdrachten uiterst krachtige selectieopdrachten te realiseren. In elk hoofdstuk zul je te maken krijgen met nieuwe basisconcepten van SQL.

Doelen

In dit hoofdstuk maak je kennis met de basisconcepten van SQL, waarbij de nadruk ligt op hoe SQL wordt gebruikt in het relationele databasemanagementsysteem van Oracle. Dit hoofdstuk heeft de volgende leerdoelen:

- ontwikkelen van een goed begrip van het concept relationele database;
- aanleren van de algemene mogelijkheden van een relationele databasemanagementsysteem;
- vertrouwd raken met de kenmerken van het Oracle relationele databasemanagementsysteem;
- leren hoe SQL*Plus kan worden gebruikt;
- de basisfuncties zoals selecteren, projecteren en samenvoegen leren;
- vertrouwd raken met de basissyntax van het SELECT-statement;
- de SQL-naamconventies leren.

Gegevens en informatie

Bedrijven en organisaties, of ze nu groot of klein zijn, genereren en beheren grote hoeveelheden informatie. Informatie wordt gegenereerd op basis van feiten, die we *gegevens* noemen. Met gegevens op zich kunnen managers nauwelijks uit de voeten. Pas wanneer gegevens op de een of andere logisch manier gegroepeerd zijn krijgen de gegevens betekenis. Gegevens kunnen erg efficiënt georganiseerd en beheerd worden wanneer ze in een databasemanagementsysteem (DBMS) worden ondergebracht. Een DBMS is een uiterst complex softwarepakket. Er zijn verschillende DBMS-producten, die worden aangeboden door onder andere Oracle, IBM en Microsoft. Voorbeelden van op grote schaal toegepaste DBMS-producten zijn RDBMS van Oracle, DB2 van IBM, SQL Server van Microsoft en het voor de desktopgebruiker bedoelde Microsoft Access.

Een DBMS biedt zowel systeemontwikkelaars als de gebruikers van informatiesystemen een gemakkelijk te gebruiken interface met de databases van de organisatie. In een database kunnen twee soorten gegevens worden opgeslagen:

- *Gegevens van gebruikers.* Bedrijfsmatige gegevens die worden opgeslagen door een organisatie. Tot de gegevens van gebruikers hoort ook alle informatie die essentieel is voor de softwaretoepassingen voor de bedrijfsvoering.
- *Systeemgegevens.* Gegevens die het databasesysteem nodig heeft om gegevens van gebruikers en van zichzelf te beheren. Deze worden ook wel *metadata* of gegevens over gegevens genoemd. Onder systeemgegevens verstaan we onder andere informatie zoals het maximaal toegestane aantal karakters dat kan worden ingevoerd voor bijvoorbeeld de naam van een medewerker.

Relationele database

Een database is een geïntegreerde eenheid verzamelde gegevens. Een database wordt meestal opgeslagen op een permanent opslagmedium dat altijd online is, zoals een schijfunit. De DBMS vormt de interface met het besturingssysteem van een computer en maakt het op die manier mogelijk dat gegevens uit een database kunnen worden opgehaald en erin kunnen worden opgeslagen. Voordat ze bewerkt en omgezet kunnen worden, moeten de gegevens eerst vanaf het opslagmedium worden opgehaald en in het geheugen van de computer worden opgeslagen. In een gebruikelijke client-serversituatie betekent dit dat de gegevens die opgeslagen zijn op een of meer *servers* via een computernetwerk naar de *client*computer van de medewerker van de organisatie moeten worden gekopieerd. Dit netwerk kan zowel een lokaal als een wide-area netwerk zijn.

Het meestgebruikte type DBMS-software wordt een *relationeel* DBMS of RDBMS genoemd. Een relationele database slaat gegevens op in de vorm van tabellen, zoals is weergegeven in figuur 1.1. Een *tabel* is gedefinieerd als een verzame-

ling *rijen* en *kolommen*. De tabellen worden formeel aangeduid als *relaties*; dat is dan ook de reden dat dergelijke databases relationele databases worden genoemd. In de praktijk zal het echter zelden voorkomen dat je iemand tabellen als relaties zult horen aanduiden. Nagenoeg altijd zal de term *tabellen* worden gebruikt.

Zoals je in figuur 1.1 kunt zien, vormen de rijen de records en de kolommen de velden. Het DBMS maakt het mogelijk om elke combinatie van rijen en kolommen te bewerken met behulp van een speciale manipulatietaal of queryingtaal zoals SQL.

Tabel-naam	Kolomnaam				Kolom alias naam	
	MDW_SOFI_NR	MDW_ACHTERNAAM	MDW_GEB_D	MDW_SALARIS	Geslacht	Afdeling
Rij	999666666	Bordoloi	10-NOV-67	55000	M	1
	999555555	Jochems	20-JUN-71	43000	V	3
	999444444	Zuiderweg	08-DEC-75	43000	M	7
	999887777	Muiden	19-JUL-78	25000	V	3
	999222222	Amelsvoort	29-MAR-69	25000	M	3
	999111111	Bock	01-SEP-55	30000	M	7
	999333333	Joosten	15-SEP-72	38000	M	7
	999888888	Pregers	31-JUL-72	25000	V	7
	Primaire sleutel		Kolom		Vreemde sleutel	

Figuur 1.1

Een relationele database biedt de mogelijkheid om gegevens op te slaan en weer op te halen op een manier die consistent is met een gedefinieerd gegevenstype, dat het *relationele model* wordt genoemd. Het relationele model werd in 1970 door E. F. Codd ontwikkeld. Dit model bestaat uit een aantal richtlijnen voor het opnemen van gegevens in een relationele database, plus een aantal operatoren die gebruikt kunnen worden om de informatie te bewerken. De kenmerken van een relationele database zijn:

- *De database is gestructureerd op basis van de gegevens en heeft geen statisch ontwerp.* Dit betekent dat de structuur van een database meestal gedurende lange tijd onveranderd zal blijven, omdat de soorten gegevens die een organisatie opslaat meestal ook lange tijd onveranderd blijven. Een database die op basis van de gegevens die erin opgeslagen moeten worden is ontworpen zal ook geen dubbel opgeslagen gegevens bevatten. Oudere, statisch ontworpen databases, dwongen organisaties vaak tot het bouwen van informatiesystemen die niet onderling konden communiceren, omdat de gegevensdefinities in de verschillende systemen niet consistent waren.

- *De gegevens zijn beschrijvend.* Dat betekent dat de namen van tabellen en kolommen in tabellen een duidelijke betekenis hebben.
- *De consistentie van gegevenswaarden wordt tussen alle toepassingen gehandhaafd.* Bij oudere technologieën was het mogelijk dat bijvoorbeeld het adres van een klant door een organisatie in twee verschillende gegevensbestanden opgeslagen werd. Als de klant van adres veranderde, en die wijziging slechts in het ene bestand maar niet de andere werd doorgevoerd, ontstond er een inconsistentie in de gegevens. Relationele databases minimaliseren het dubbel opslaan van gegevens.
- *Er zijn regels gedefinieerd die afdwingen hoe gegevens worden opgeslagen.* Dat betekent dat de gegevens die in de database zijn opgeslagen valide zijn (wat aangeduid wordt als *integriteit van gegevens*). Een organisatie kan bijvoorbeeld de regel hebben dat het uurloon van een medewerker nooit hoger kan zijn dan 75 euro. Een andere voorbeeld van integriteit van gegevens is het afdwingen van de beperking dat er nooit een verkooporder in de database aanwezig kan zijn zonder een corresponderend klantrecord.

Het voordeel van relationele databases is dat ze over het algemeen gemakkelijker te gebruiken zijn en een sterkere mate van gegevensafhankelijkheid hebben dan oudere databasetechnologieën. Gegevensafhankelijkheid is de mogelijkheid om veranderingen in de structuur van een database aan te brengen, zonder dat daarvoor ook de computertoepassingen moeten worden aangepast waarmee de database wordt benaderd. Voorbeelden hiervan zijn onder andere een programma waarmee een gebruiker nieuwe informatie over een verkooporder of informatie over een nieuwe klant kan invoeren.

Databasemanagementsystemen (DBMS)

Een databasemanagementsysteem (DBMS) beheert de gegevens in een database. Het fungeert als een laag tussen de gebruiker en de database, waardoor gebruikers met de database kunnen werken. Een DBMS is een verzameling toepassingen waarmee gebruikers een database kunnen maken en beheren. Een DBMS is een veelzijdig softwaresysteem waarin databases voor allerlei toepassingen gedefinieerd, gemaakt en beheerd kunnen worden. Zonder een DBMS is het onmogelijk om gegevens in een database te selecteren, bij te werken of te verwijderen. Alleen het DBMS weet hoe en waar de gegevens op een van de permanente opslagmedia van de organisatie zijn opgeslagen. Een DBMS maakt het ook mogelijk dat gegevens gedeeld kunnen worden; gebruikers en managers van een informatiesysteem kunnen meer *informatiewaarde* halen uit dezelfde hoeveelheid gegevens wanneer de gegevens gedeeld worden.

Een DBMS is een compleet softwarepakket voor het beheren van een database. Het levert de volgende diensten:

- gegevensdefinities voor het definiëren en opslaan van alle objecten waaruit een database bestaat, zoals tabellen en indexen;
- gegevensonderhoud voor het actueel houden van rijen (records) voor elke tabel in een database;
- gegevensmanipulatie voor het invoegen, bijwerken en sorteren van gegevens in een database;
- gegevensweergave waarmee de gegevens voor de gebruiker zichtbaar gemaakt kunnen worden;
- integriteitsbewaking van de gegevens waardoor de nauwkeurigheid van de gegevens gegarandeerd is;
- gegevensbeveiliging om te garanderen dat alleen gebruikers met voldoende toegangsrechten bepaalde gegevens kunnen benaderen;
- back-up- en herstelfunctionaliteit waarmee de belangrijkste gegevens van de organisatie automatisch worden veilig gesteld en herstel na een systeemfout mogelijk is.

Het relationele DBMS van Oracle

Een relationele database wordt geïmplementeerd met behulp van een relationeel databasemanagementsysteem (RDBMS). Een RDBMS voert alle hierboven genoemde basisfuncties van de DBMS-software uit, plus nog een groot aantal andere functies waarmee het relationele model begrijpelijker en gemakkelijker te implementeren wordt. Een van de meest belangrijke kenmerken van een RDBMS is dat het diensten levert waarmee systeembeheerders de structuur van een database gemakkelijk kunnen aanpassen.

Het RDBMS van Oracle behoort tot de meestgebruikte RDBMS-pakketten. Het wordt op grote schaal gebruikt omdat het een krachtig gegevensbeheerprogramma is en omdat Oracle versies ervan levert voor nagenoeg elk type computer, voor pc's en Apple Macintosh-computers, maar ook voor minicomputers en enorme mainframes. We zullen het Oracle RDBMS in het vervolg kortweg *Oracle* noemen. De software van Oracle werkt op alle computerplatformen, of ze nu groot of klein zijn, nagenoeg op dezelfde wijze. Een informaticus of gebruiker die kennis van Oracle heeft op een bepaald systeem kan daardoor deze vaardigheden ook gemakkelijk gebruiken op een andere computer. Dit maakt dat deskundige Oracle-gebruikers en ontwikkelaars veelgevraagd zijn en kennis van en vaardigheden in Oracle veelzijdig toepasbaar zijn. De onderscheidende kenmerken van Oracle zijn:

- *Beveiligingsmechanismen.* De geavanceerde beveiligingsmechanismen van Oracle beperken de toegang tot gevoelige gegevens met behulp van diverse typen rech-

ten, bijvoorbeeld het recht om bepaalde informatie in een database te mogen lezen of te mogen invoeren.

- *Back-up- en herstelmechanismen.* De geavanceerde back-up- en hersteltoepassingen van Oracle minimaliseren gegevensverlies en downtime in het geval van problemen.
- *Opslagcapaciteitsbeheer.* De flexibele opslagcapaciteitsbeheermechanismen van Oracle maken het mogelijk om opslagcapaciteit naar behoefte toe te wijzen. Deze mechanismen zorgen daarna ook voor de toewijzing van opslagcapaciteit.
- *Open connectivity.* De open connectivity-functionaliteit van Oracle zorgt ervoor dat de database continu toegankelijk is. Ook kan het systeem zonder meer samenwerken met systemen van andere softwarefabrikanten.
- *Tools en toepassingen.* Oracle ondersteunt een groot aantal ontwikkelgereedschappen, gereedschappen voor eindgebruikers en in de handel verkrijgbare standaardtoepassingen waarmee zakelijke processen en gegevens worden gemodelleerd en programmacode automatisch kan worden gegenereerd.

SQL en SQL*Plus van Oracle

Gebruikers van een RDBMS manipuleren gegevens met behulp van een speciale *gegevensmanipulatietaal*. De structuur van databases wordt gedefinieerd met behulp van een *gegevensdefinitietaal*. De opdrachten die gebruikers willen laten uitvoeren om gegevens op te slaan en te selecteren kunnen op een terminal met RDBMS-interface of met behulp van een grafische interface worden ingevoerd. Het DBMS verwerkt vervolgens de opdrachten.

SQL is de meest populaire databasetaal en heeft opdrachten waarmee het kan worden gebruikt voor zowel het manipuleren als het definiëren van relationele databases, omdat de syntax gebaseerd is op natuurlijk Engels. Oracle was een van de eerste bedrijven die RDBMS-software op de markt bracht die werkte met een gestructureerde querytaal die gebaseerd was op de Engelse taal. Alle gegevenstoegang- en manipulatiegereedschappen van Oracle zijn gebaseerd op de huidige SQL-versie van het American National Standards Institute (ANSI).

SQL is een uitgebreide databasetaal. Voor SQL er was, waren er geen standaard gegevensbenaderingstalen. IBM ontwikkelde aan het eind van de jaren zeventig van de vorige eeuw een SQL-interface voor relationele databases. De ANSI en de International Standards Organisation (ISO) hebben beide SQL als standaardtaal voor relationele databasemanagementtoegang vastgesteld. De populariteit van SQL is steeds verder toegenomen, omdat gebruikers geen programmeerervaring hoeven te hebben.

Oracle gebruikt SQL voor alle interactie met de database. De SQL-expressies kunnen verdeeld worden in de volgende twee belangrijke categorieën:

- *De gegevensdefinitietaal (Engels: Data Definition Language of kortweg DDL)*: een reeks SQL-opdrachten die objecten in een database creëren en definiëren en deze definities in een gegevensbibliotheek opslaan. Een voorbeeld van een DDL-opdracht is de CREATE TABLE-opdracht. Met DDL kunnen de gebruikers een databaseobject maken, verwijderen of wijzigen en rechten aan een databaseobject toekennen of juist verwijderen.
- *De gegevensmanipulatietaal (Engels: Data Manipulation Language of kortweg DML)*: een reeks SQL-opdrachten waarmee gebruikers de gegevens in een database kunnen manipuleren. Een voorbeeld van een DML-opdracht is de INSERT-opdracht. Met DML kunnen de gebruiker gegevens in een database invoegen, bewerken, verwijderen of selecteren.

SQL is een taal zonder specifieke opmaakeisen. Dat betekent dat er geen specifieke regels zijn voor de manier waarop een SQL-opdracht wordt ingevoerd. Daarnaast is de vocabulaire van SQL erg beknopt. Daarom is het relatief gemakkelijk om te leren werken met SQL.

We stelden eerder dat SQL een *non-procedurele taal* is. Bij procedurele talen, zoals COBOL of C++, moet een computerprogrammeur tot in detail de stappen programmeren die nodig zijn om een bepaalde taak uit te voeren. Een gewone procedurele toepassing kan bestaan uit honderden regels programmacode. Bij een non-procedurele taal hoeft je alleen maar in te voeren welke taak je door het DBMS wilt laten uitvoeren en is het niet nodig om gedetailleerde programmacode te schrijven. Non-procedurele toepassingen zijn daarom meestal erg kort. Om SQL te kunnen gebruiken hoeft je ook niet te weten hoe gegevens fysiek of logisch opgeslagen zijn. Het DBMS converteert of *ontleedt* (Engels: *parse*) de SQL-opdrachten die je invoert en voert de taak vervolgens eenvoudigweg uit. Het is echter wel belangrijk dat je de logische databasestructuur begrijpt. Je moet bijvoorbeeld de namen van de tabellen en van de kolommen daarin kennen. In hoofdstuk 2 leer je hoe je tabellen kunt maken en hoe je de namen van de tabellen en de kolommen daarin kunt definiëren.

Databasbeheerders, programmeurs, analisten en andere informatici die willen werken met het Oracle DBMS gebruiken SQL*Plus als primaire interface. SQL*Plus verzorgt een volledige implementatie van standaard ANSI-SQL. Daarnaast bevat het ook nog talloze uitbreidingen die je kunt gebruiken voor speciale taken, zoals het beschrijven van de logisch structuur van een tabel. Je kunt SQL*Plus ook gebruiken om een query in een tabel uit te voeren, de structuur van een tabel en de kolommen daarin te definiëren of gegevens te manipuleren door rijen gegevens in te voegen, bij te werken of te verwijderen. Ook heeft SQL*Plus een uitgebreide online helpfunctie. SQL*Plus bevat daarnaast een aantal Oracle-specifieke functies waarmee je rapporten op het scherm of op papier kunt maken.

Met deze functies kun je de manier waarop de uitvoer wordt weergegeven op een monitor of afdruk vormgeven. Deze functies werken bovenop standaard SQL. Met SQL*Plus kunnen gebruikers:

- SQL-opdrachten en PL/SQL-blokken invoeren, bewerken, opslaan, selecteren en uitvoeren PL/SQL-blokken zijn kleine toepassingen die geschreven zijn in de procedurele taalversie van SQL. PL/SQL is een geavanceerde programmeertaal waarmee je toepassingen kunt maken die gegevens procedureel verwerken op een vergelijkbare manier als talen als C++ en Visual Basic.NET;
- wiskundige bewerkingen uitvoeren op queryresultaten, deze opmaken, opslaan en afdrukken in de vorm van rapporten;
- kolomdefinities voor elke willekeurige tabel weergeven;
- gegevens tussen SQL-databases benaderen en kopiëren;
- berichten verzenden naar en antwoorden ontvangen van een gebruiker van een informatiesysteem.

Relationele bewerkingen

Met SQL-bewerkingen kun je nieuwe tabellen creëren, gegevensrijen in tabellen invoegen, rijen in een tabel bijwerken, rijen in een tabel verwijderen en query's uitvoeren in tabellen om de daarin opgeslagen gegevens weer te geven. In het resterende deel van dit hoofdstuk zullen we stilstaan bij het concept van het uitvoeren van een query in tabellen en kennismaken met het gebruik van SQL*Plus om eenvoudige SQL-query's te schrijven.

Het SELECT-statement wordt voornamelijk gebruikt om query's te schrijven die informatie uit databasetabellen halen. Eerder hebben we gesteld dat een database een verzameling aan elkaar gerelateerde tabellen is, waarbij elke tabel bestaat uit rijen en kolommen. De kracht van het SELECT-statement zit in de mogelijkheid ervan om gegevens uit meerdere tabellen met elkaar te combineren en zo uitvoer te genereren in de vorm van een resultatentabel. Bekijk het SELECT-statement in voorbeeld 1.1 en de bijbehorende resultatentabel. Hier voert het SELECT-statement een query uit in de tabel medewerker en selecteert waarden uit twee kolommen met de namen mdw_sofi_nr en mdw_achternaam.

```
/* Voorbeeld 1.1 */  
SELECT mdw_sofi_nr, mdw_achternaam  
FROM medewerker;
```

MDW_SOFI_	MDW_ACHTERNAAM
-----	-----
999666666	Bordoloi


```
999555555    Jochems
999444444    Zuiderweg
overige rijen zijn hier weggelaten...
```

Voor het moment zullen we de syntax of de opmaak van het SELECT-statement laten voor wat het is. Merk op dat de *resultatentabel* tabelvormig is en kolommen en rijen heeft. Elke kolom heeft de kolomnaam als kop. Oracle zal nauwgezet elke rij in de tabel weergeven. Later zullen we zien hoe je het aantal weergegeven rijen kunt beperken tot een subset van de rijen in een tabel. Je zult ook leren om SELECT-statements te schrijven waarmee je kolommen uit meer dan één tabel kunt selecteren. De mogelijkheid om bepaalde rijen en kolommen uit een of meer tabellen te selecteren vormen de fundamentele relationele bewerkingen, en er zijn drie van dergelijke bewerkingen: **select**, **project** en **join**.

De bewerking Select

Een selectiebewerking selecteert een subset rijen (records) in een tabel (relatie) die voldoen aan een selectiecriterium. De subset kan variëren van nul rijen, als geen van de rijen voldoet aan het selectiecriterium, tot alle rijen van een tabel. Het SELECT-statement in voorbeeld 1.2 selecteert een subset rijen door gebruik te maken van een WHERE-clausule. De resultatentabel bevat alleen de rijen die voldoen aan de voorwaarde van de WHERE-clausule. Hoofdstuk 3 behandelt de WHERE-clausule in detail.

```
/* Voorbeeld 1.2 */
SELECT mdw_sofi_nr, mdw_voornaam
FROM medewerker
WHERE mdw_sofi_nr = '999111111';

MDW_SOFI_    MDW_VOORNAAM
-----
999111111    Douglas
```

Omdat elke medewerker een uniek sofi-nummer heeft (*mdw_sofi_nr*), selecteert voorbeeld 1.2 een subset die bestaat uit exact één rij. Als echter geen van de medewerkers een sofi-nummer heeft dat overeenkomt met het sofi-nummer dat in de WHERE-clausule werd gespecificeerd, zal de resultatentabel geen enkele rij bevatten.

De bewerking Project

Een projectbewerking selecteert een beperkt aantal uitgeschreven kolommen (velden) in een tabel. De resultaatentabel bevat een subset van de beschikbare kolommen en kan maximaal alle beschikbare kolommen bevatten. Voorbeeld 1.3 selecteert een subset kolommen uit de tabel *medewerker*, om precies te zijn het sofi-nummer, de voornaam en de achternaam van elke medewerker. In hoofdstuk 3 komen we uitgebreid terug op projectbewerkingen.

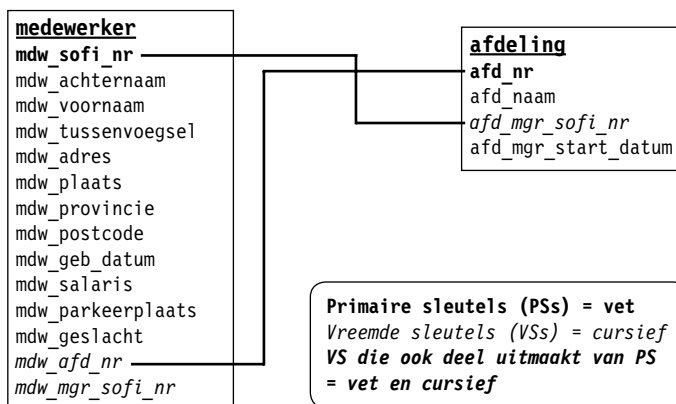
```
/* Voorbeeld 1.3 */
```

```
SELECT mdw_sofi_nr, mdw_voornaam, mdw_achternaam
FROM medewerker;
```

MDW_SOFI_	MDW_VOORNAAM	MDW_ACHTERNAAM
-----	-----	-----
999666666	Bijoy	Bordeloi
999555555	Suzan	Jochems
999444444	Willem	Zuiderweg
<i>overige rijen zijn hier weggelaten...</i>		

De bewerking Join

Een join-bewerking combineert gegevens afkomstig uit twee of meer tabellen op basis van een of meer *gezamenlijke kolomwaarden*. Bekijk de tabellen *medewerker* en *afdeling* in figuur 1.2. We weten dat ondernemingen vaak opgedeeld zijn in afdelingen. De medewerkers worden ingedeeld bij een afdeling, en elke afdeling kan meer dan één medewerker hebben.



Figuur 1.2