

131

€ 4,-

KnowWare
SQL leicht gemacht

72 Seiten

Start mit
Datenbanken und
SQL



Petra Bilke

www.KnowWare.de

2. Ausgabe

Deutschland: 4,- EUR Österreich: 4,60 EUR
Schweiz: 8 SFR Luxemburg: 4,70 EUR Italien: 5,50 EUR

Acrobat Reader: Wie ...

F5/F6 öffnet/schließt die Ansicht **Lesezeichen**

Strg+F sucht

Im Menü Ansicht stellst du ein, wie die Datei gezeigt wird

STRG+0 = Ganze Seite **STRG+1** = Originalgrösse **STRG+2** = Fensterbreite

Im selben Menü kannst du folgendes einstellen:: **Einzelne Seite**, **Fortlaufend** oder **Fortlaufend - Doppelseiten** .. Probiere es aus, um die Unterschiede zu sehen.

Navigation

Pfeil Links/Rechts: eine Seite vor/zurück

Alt+ Pfeil Links/Rechts: Wie im Browser: Vorwärts/Zurück

Strg++ vergrößert und **Strg+-** verkleinert

Bestellung und Vertrieb für den Buchhandel

Bonner Pressevertrieb, Postfach 3920, D-49029 Osnabrück

Tel.: +49 (0)541 33145-20

Fax: +49 (0)541 33145-33

bestellung@knowware.de

www.knowware.de/bestellen

Autoren gesucht

Der KnowWare-Verlag sucht ständig neue Autoren. Hast du ein Thema, daß dir unter den Fingern brennt? - ein Thema, das du anderen Leuten leicht verständlich erklären kannst?

Schicke uns einfach ein paar Beispielseiten und ein vorläufiges Inhaltsverzeichnis an folgende Adresse:

lektorat@knowware.de

Wir werden uns deinen Vorschlag ansehen und dir so schnell wie möglich eine Antwort senden.

Vorwort	4	Der Operator IN	40
Grundlagen	5	Der Bereichsoperator BETWEEN	41
Welche Aufgaben hat eine Datenbank?	5	Der Operator LIKE	42
Die Hauptbestandteile einer Datenbank.....	5	Ordnen von Daten (ORDER BY)	42
Das relationale Datenbankmodell	6	Gruppenbildung mit GROUP BY und Gruppenselektion mit HAVING	43
Beziehungen	7	Verknüpfen von mehreren Tabellen	45
Der Normalisierungsprozeß	12	Tabellen vereinigen.....	54
Erste Normalform.....	12	Unterabfragen	55
Zweite Normalform.....	13	SQL-Anweisungen speichern und ausführen.....	57
Dritte Normalform.....	14	Benutzersichten (Views).....	57
Der Entwurfsprozess	15	... in MS-Access-SQL	57
Die Datenbanksprache SQL in verschieden Systemen.....	16	... in MS-SQL-Server	57
MS-Access	17	Anweisungen zur Datenmanipulation	60
Datenbank anlegen	17	UPDATE	60
SQL nutzen.....	17	Datensätze löschen	61
MS-SQL-Server	18	Datenbankdesign	62
Datenbank anlegen	18	Tabellen ändern (ALTER TABLE)	62
Grundlagen der Datenbanksprache SQL (Structured Query Language)	20	Löschen von Views, Indizes und Tabellen	62
Regeln zur Eingabe von SQL- Anweisungen.....	21	Zugriffsschutz und Erteilung von Rechten mit SQL	63
Erstellen von Datenbanken.....	22	Benutzer und Gruppen.....	63
Tabellen erstellen	22	Systemberechtigungen.....	64
Möchtest du verhindern, dass mehrere Datensätze mit identischer Personalnummer erfasst werden, ist ein Identifikationsschlüssel in Form des Primärschlüssels zu bilden.	24	Objektberechtigungen.....	64
Tabellen indizieren	25	Entziehen der Zugriffsrechte mit REVOKE	65
Datensätze (Tupel) einfügen	26	Systemberechtigungen entziehen.....	65
Datenabfrage mit SELECT	28	Objektberechtigungen	65
Inhaltsanzeige einer Tabelle	29	Anhang	66
Auswahl von Datensätzen mit WHERE	31		
Mathematische Aufbereitung von Ergebnissen	33		
Mathematische Ausdrücke	33		
Einsatz von Funktionen in mathematischen Ausdrücken.....	34		
Zusammengesetzte Vergleiche	38		
Die Operatoren NOT, AND und OR.....	38		

Vorwort

5 Jahre ist das SQL-Heft auf dem Markt. Die Nachfrage ist ungebrochen. Nun möchte ich das Angebot von Michael Maardt nutzen, in der neuen Auflage mehr Seiten zu füllen und dir somit umfangreichere Informationen zum Thema Datenbanken und SQL zur Verfügung zu stellen.

Datenbanken sind in der heutigen Zeit von immenser Wichtigkeit. Praktisch in jedem Geschäftsbereich und auch im privaten Leben fallen Daten und Informationen an, die verwaltet werden wollen.

Für die Datensammlungen eignen sich Datenbanken. Diese müssen gepflegt werden, wenn sie nicht an Wert verlieren sollen. Alte, ungültige oder falsche Daten bereiten Frust und haben keinen Nutzen. Eine Datenbank bringt nur dann einen hohen Wert, wenn mit Hilfe einer ausgefeilten Abfragesprache auch Daten der Sammlung hinzugefügt oder aktualisiert werden können. Anliegen dieses Heftes ist es, dir die Abfragesprache SQL näher zu bringen.

Das Kürzel SQL bedeutet **Structured Query Language** oder Strukturierte Abfragesprache. Das Haupteinsatzgebiet von SQL sind Datenbank-Abfragen. SQL ist eine Teilsprache, das heißt sie kann nicht allein verwendet werden, sondern nur in Verbindung mit einer anderen Programmiersprache oder einem speziellen Datenbankprogramm.

In meinem Heft möchte ich mich zwei Schwerpunkten widmen.

Zunächst werden wir uns mit einigen Grundlagen zu den Datenbanken beschäftigen, da es schwer ist, ohne Datenbankgrundkenntnisse, effektive Dateistrukturen zu erstellen.

Im zweiten Teil des Heftes wirst du die Abfragesprache SQL (Structured Query Language) kennenlernen, die von vielen Datenbanksystemen genutzt wird.

Beherrscht du die Grundlagen von SQL, kannst du in verschiedensten Datenbanksystemen und unterschiedlichsten Rechnerplattformen Daten auf einfache Art und Weise abrufen und aktualisieren. Möchtest du meine Beispiele am eigenen PC nachvollziehen, kannst du z.B. Microsoft-Access, ORACLE oder Microsoft-Query nutzen.

Die besprochenen Beispiele stehen zum Download auf der Seite www.bilke.de/sql bereit.

Als begleitendes Beispiel wird das Problem eines Systemhauses geschildert, was vor kurzen einen Bereich Schulung und Beratung eröffnet hat. Er hat die Aufgabe Schulungs- und Beratungsleistungen im Bereich der Weiterbildung bereitzustellen.

Grundlagen

Welche Aufgaben hat eine Datenbank?

- Eine **Datenbank** ist eine Sammlung von Daten, die sich auf ein bestimmtes Thema oder einen bestimmten Zweck beziehen.
Das Beispiel in diesem Heft beruht darauf, dass ein Mitarbeiter für Weiterbildung eines Unternehmens in einer Datenbank die besuchten Schulungen der Betriebsangehörigen erfassen möchte.
- Eine **Datenbank** ermöglicht dem Benutzer den Zugriff auf gespeicherte Daten, ohne dass man wissen muss, wie die Daten im Datenbanksystem organisiert sind.
- Eine **Datenbank** gewährleistet, dass kein Benutzer ohne Zugriffsberechtigung Daten sichten oder manipulieren kann.
- Eine **Datenbank** stellt sicher, dass es nicht wegen Fehlmanipulationen des Benutzers passiert, dass Daten zerstört werden oder der ganze Datenbestand unbrauchbar wird.

Die Hauptbestandteile einer Datenbank

Das Datenbankverwaltungssystem

bildet den Kern der Datenbank und beinhaltet alle für die gesamte Datenverwaltung notwendigen Routinen.

Die Datenbanksprache

bildet die Schnittstelle zwischen dem Benutzer und dem Datenbankverwaltungssystem. Bei vielen Datenbanksystemen wird als Datenbanksprache die Sprache SQL (Structured Query Language) verwendet.

SQL besitzt Elemente für folgende Aufgabenbereiche:

- **Die Datendefinition (Data Definition Language; DDL)** wird benötigt, um die Datenstruktur aufzubauen (Tabellen einrichten, Felder definieren etc.).
- **Die Datenmanipulation (Data Manipulation Language; DML)** wird genutzt um Daten einzugeben, zu löschen und zu ändern. Auch die Anweisung zur **Datenabfrage** wird zur DML gezählt.
- **Die Datensteuerung (Data Control Language; DCL)** dient zur Wahrung der Datenintegrität. Du kannst Datenbankbenutzer einrichten und so die gespeicherten Daten vor dem Zugriff unberechtigter Personen schützen.

Der Formulareditor

erlaubt das Erstellen von Formularen zur Dateneingabe und Datenänderung für den Benutzer.

Der Berichtseditor

gibt die Möglichkeit, die gespeicherten Daten in einer übersichtlichen Form darzustellen.

Der Menüeditor

erlaubt es dem Benutzer per Knopfdruck entsprechende Programme, Formulare oder Berichte zu aktivieren.

Weiterhin gibt es je nach Softwareprodukt weitere Komponenten, wie z.B.

Netzwerkprogramme für Client/Server-Architekturen,

Programmierspracheninterfaces, grafik- und mausunterstützte Abfragewerkzeuge für eine vereinfachte Datenbankabfrage.

Das relationale Datenbankmodell

Ein Datenbankmodell dient der exakten Beschreibung von Datenbankschemata. Bekannte Datenbankmodelle sind das hierarchische Modell, das Codasyl-Netzwerkmodell und das hier beschriebene relationale Modell.

Bei relationalen Datenbanken werden die Informationen geordnet nach Themenkreisen in Form von Tabellen abgelegt. Dieses Datenmodell ist einfach zu verstehen und zeichnet sich durch Flexibilität bezüglich Strukturänderungen und -ergänzungen aus.

Relationale Datenbanken bezwecken, dass **Redundanzen** (mehrfaches Speichern von gleichen Informationen) verhindert werden und damit die **Datenkonsistenz** (Eindeutigkeit der Daten) gewährleistet werden kann.

Im weiteren werden Begriffe verwendet, die kurz erklärt werden. Die in Klammern stehenden Bezeichnungen stammen aus der Datenbanktheorie, die davor sind gebräuchliche Informatikbezeichnungen.

Tabelle (Relation):

Die Tabelle umfasst den Tabellennamen, die Feldnamen und die Menge aller Datensätze.

Die Tabelle hat einen eindeutigen Tabellennamen und besitzt folgende Merkmale:

- Jede Spalte der Tabelle besitzt einen eindeutigen Namen (Spaltenname) und einen zugehörigen Wertebereich.
- Die Reihenfolge der Spalten und Zeilen ist unerheblich.
- Es existieren keine zwei Zeilen, die identische Werte besitzen.

Tabellename (Entität)

Der Tabellename erfasst einen Themenkreis, welcher Elemente mit gleichen Merkmalen umfasst. Im Heftbeispiel wurden die Tabellen **TEILNEHMER** oder **SCHULUNGEN** genannt.

Feldname / Spaltenname (Attribut):

Der Feldname ist ein Element einer Tabelle, das eine bestimmte Kategorie von Informationen enthält, wie z.B. Vornamen oder Nachnamen von Personen.

Datensatz (Tupel):

Ein Datensatz ist eine Zusammenstellung von Daten über ein Objekt. Im unserem Beispiel sind die Teilnehmer oder die Schulungen Objekte. Ein Datensatz wird in einer Tabelle als Zeile dargestellt.

Feldinhalt (Attributwert):

Der Feldinhalt ist ein Datenwert.

Wertebereich (Domäne):

Ein Wertebereich ist eine Gruppe von Datensätzen, die durch Einschränkung eines Feldes auf bestimmte Feldinhalte definiert wird.

Nullwerte:

Wenn ein Feld einen Nullwert enthält, besitzt dieses Feld keinen Feldinhalt und beinhaltet somit keine Information. Der Nullwert darf nicht mit der Zahl Null verwechselt werden. Die Zahl Null stellt eine Information dar, der Nullwert jedoch nicht.

Die erklärten Begriffe haben ich einem Schema zusammengefasst. Bei dem Beispiel handelt es sich um einen Ausschnitt aus einer Personaltabelle.

Tabellenname: PERSONAL

Personalnummer	Vorname	Name	Ort	
1002	Kai	Müller	Eilenburg	← Feldnamen
1101	Guenter	Maus	Halle/Saale	
1015	Erik	Wicki	Hamburg	
1016	Ulrich	Müller	Hohenpriessnitz	← Datensatz
1005	Ede	Pfau	Leipzig	
1100	Kerstin	Gans	Moertitz	
1008	Eike	Taro	Musterdorf	← Wertebereich
1010	Jürgen	Marx	Musterdorf	
1112	Hannes	Heik	Musterdorf	
1113	Karl	Seppa	Musterdorf	
1114	Karsten	Karn	Musterdorf	
1017	Klaus	Ecke	Wurzen	

↑
Feldinhalt

Beziehungen

In der Regel besteht eine relationale Datenbank nicht nur aus einer Tabelle, sondern aus mehreren. Die einzelnen Tabellen dürfen nicht isoliert betrachtet werden. Zwischen den Tabellen können Beziehungen bestehen. Die Anzahl der möglichen Beziehungen ist begrenzt

und ergibt sich aus der Kombination der möglichen Assoziationstypen.

Assoziation:

Eine Assoziation bestimmt, wie viel Datensätze einer Tabelle 2 zu einem Datensatz der Tabelle 1 gehören können. Es gibt vier verschiedene Assoziationstypen:

Assoziationstyp	Abkürzung	Datensatzanzahl in Tabelle 2
einfache Assoziation	1	genau ein Datensatz (1)
konditionelle Assoziation	c	kein oder genau ein Datensatz (0/1)
multiple Assoziation	m	mindesten ein Datensatz (>=1)
multipl-konditionelle Assoziation	mc	beliebig viele Datensätze (>=0)

Ein Beispiel für eine einfache Assoziation

Es sind zwei Tabellen vorhanden. In der ersten Tabelle sind allgemeine und in der zweiten spezielle Personaldaten gespeichert. In beiden Tabellen muss genau ein Datensatz pro Person vorhanden sein.

Personal

PNR	Vorname	Name
1002	Kai	Müller
1005	Ede	Pfau
1008	Eike	Taro

Gehalt

PNR	Gehalt
1002	4500
1005	4900
1008	5000

Ein Beispiel für eine konditionelle Assoziation

Es sind zwei Tabellen vorhanden. In der ersten Tabelle sind allgemeine Personaldaten und in der zweiten spezielle Personaldaten gespeichert. In der zweiten Tabelle gibt es in Abhängigkeit davon, ob die Person einen Pass besitzt, keinen oder genau einen Datensatz.

Personal

PNR	Vorname	Name
1002	Kai	Müller
1005	Ede	Pfau
1008	Eike	Taro

Pass

PNR	Passnummer
1002	4500
1008	5000

Ein Beispiel für eine multiple Assoziation

Es sind zwei Tabellen vorhanden. In der ersten Tabelle sind allgemeine Personaldaten und in der zweiten Tabelle die Adressen der Haupt- und Nebenwohnsitze der Personen gespeichert. Pro Person muss in der Tabelle Adressen mindestens ein Datensatz vorhanden sein. Herr Müller und Herr Pfau sind mit jeweils 2 Datensätzen vorhanden, da beide über eine Nebenwohnung verfügen.

Personal

PNR	Vorname	Name
1002	Kai	Müller
1005	Ede	Pfau
1008	Eike	Taro

Adressen

PNR	PLZ	Ort	Straße
1002	04838	Eilenburg	Hallesche Str. 7
1002	22898	Hamburg	Musterstr. 2
1005	04232	Leipzig	Nicoleiplatz 2
1005	09999	Muster	Allee 3
1008	09999	Musterdorf	Musterstr. 6

Ein Beispiel für multipel-konditionelle Assoziation

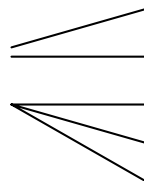
Es sind zwei Tabellen vorhanden. In der ersten Tabelle sind allgemeine Personaldaten und in der zweiten Tabelle die Vornamen der Kinder gespeichert. Da Personen beliebig viele Kinder besitzen können, gibt es pro Person kein, ein oder mehrere Datensätze.

Personal

PNR	Vorname	Name
1002	Kai	Müller
1005	Ede	Pfau
1008	Eike	Taro

Kinder

PNR	Vorname
1005	Eva
1005	Susi
1008	Max
1008	Tim
1008	Tom



Jede Assoziation (Tabelle 1 zu Tabelle 2) besitzt auch eine Gegenassoziation (Tabelle 2 zu Tabelle 1). Kombiniert man diese zwei Assoziationen miteinander, so erhält man eine Beziehung. Ausgehend von den vier verschiedenen Assoziationstypen gibt es max. 16 verschiedene Beziehungstypen, welche in folgender Tabelle aufgeführt sind:

	1	c	m	mc		
1	1-1	c-1	m-1	mc-1	hierarchische Beziehungen	
c	1-c	c-c	m-c	mc-c		konditionelle Beziehungen
m	1-m	c-m	m-m	mc-m		netzwerkförmige Beziehungen
mc	1-mc	c-mc	m-mc	mc-mc		

Für die obigen Beispiele lassen sich die Beziehungen wie folgt darstellen:

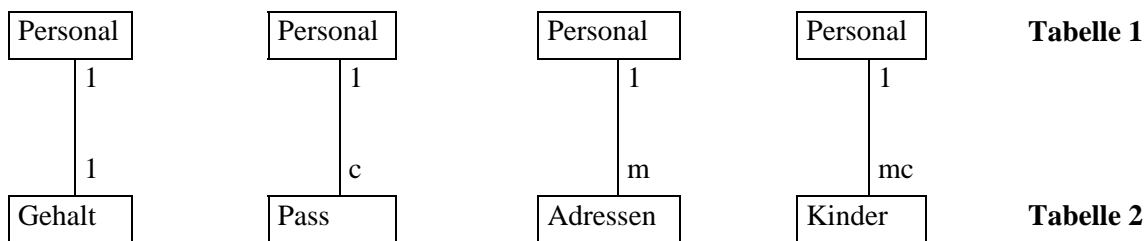


Tabelle 1

Tabelle 2

Um Beziehungen zwischen den Tabellen herstellen zu können, müssen die Begriffe **Identifikationsschlüssel** und **Fremdschlüssel** bekannt sein.

Identifikationsschlüssel (Primärschlüssel):

Jeder Datensatz einer Tabelle muss eindeutig identifizierbar sein. Dies kann durch ein Feld oder eine Kombination von Feldern gewährleistet werden. Bei der Tabelle Personal bietet sich zum Beispiel die Personalnummer als eindeutiger Identifikationsschlüssel an.

Der Identifikationsschlüssel muss folgende Kriterien erfüllen:

- Der Identifikationsschlüssel muss eindeutig sein.
- Jeder neue Datensatz erhält sofort den entsprechenden Wert des Identifikationsschlüssels.

Fremdschlüssel:

Ein Fremdschlüssel in einer Tabelle 2 ist ein Feld oder eine Feldkombination, welche in einer Tabelle 1 den Identifikationsschlüssel bildet.

Der Fremdschlüssel muss folgende Kriterien erfüllen:

- Ein Fremdschlüssel in Tabelle 2 kann nur diejenigen Feldinhalte annehmen, welche bereits im Identifikationsschlüssel der Tabelle 1 existieren.
- Er darf somit nicht leer sein.
- Der Feldname von Identifikationsschlüssel und Fremdschlüssel muß nicht identisch sein.

In Tabelle 1 (Personal) wurde als Identifikationsschlüssel die Personalnummer vergeben. Um eine Beziehung herstellen zu können, braucht man in Tabelle 2 einen Fremdschlüssel mit entsprechenden Feldinhalten. In den obigen Beispielen sind es wiederum die Personalnummern.

Betrachten wir noch ein letztes Beispiel:

Es sind 2 Tabellen vorhanden. In der ersten Tabelle sind allgemeine Personaldaten und die Nummer des Dienstwagens gespeichert. Die zweite Tabelle beinhaltet Autodaten und die Personalnummer des jeweiligen Fahrzeugnutzers.

Personal

PNR	Vorname	Name	ANR
1002	Kai	Müller	
1005	Ede	Pfau	L-AD 1234
1008	Eike	Taro	L-XX 1887

Autos

ANR	Marke	PNR
L-YY 1111	Skoda	
L-AD 1234	VW	1005
L-YY 1112	Audi	
L-XX 1887	BMW	1008
L-YY 1113	Ford	

Jede Person hat kein oder genau einen Dienstwagen. Zu jedem Auto gehört kein oder genau ein Fahrzeugnutzer.

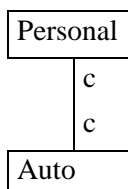


Tabelle 1

Es handelt sich hier um eine c-c Beziehung.

Tabelle 2

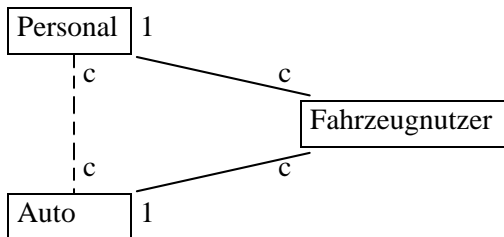
Identifikationsschlüssel und Fremdschlüssel werden wie folgt vergeben:

Tabelle	Identifikationsschlüssel	Fremdschlüssel
Personal	PNR (Personalnummer)	ANR (Autonummer)
Auto	ANR (Autonummer)	PNR (Personalnummer)

Sieh dir die Inhalte der Fremdschlüssel etwas genauer an, dann wirst du feststellen, dass Feldwerte nicht belegt sind. Sie haben einen Nullwert. Das widerspricht der Definition des Fremdschlüssels, wo es heißt:

- Ein Fremdschlüssel in Tabelle 2 kann nur diejenigen Feldinhalte annehmen, welche bereits im Identifikationsschlüssel der Tabelle 1 existieren.
- Er darf somit nicht leer sein.

Konditionelle und netzwerkförmige Beziehungen sind im vorgestellten Datenmodell somit nicht zulässig und müssen deshalb in hierarchische Beziehungen umgewandelt werden.



Es wird eine neue Tabelle Fahrzeugnutzer erstellt und somit die c-c Beziehung in zwei 1-c Bedingungen umgewandelt.

Die Tabellen haben nun nachfolgende Feldinhalte:

Personal

PNR	Vorname	Name
1002	Kai	Müller
1005	Ede	Pfau
1008	Eike	Taro

Autos

ANR	Marke
L-YY 1111	Skoda
L-AD 1234	VW
L-YY 1112	Audi
L-XX 1887	BMW
L-YY 1113	Ford

Fahrzeugnutzer

PNR	ANR
1005	L-AD 1234
1008	L-XX 1887

Der Normalisierungsprozeß

Die Normalisierung bezweckt die redundanzfreie Speicherung von Informationen innerhalb der Tabellen. Unter **redundanzfreier Datenspeicherung** versteht man, dass gleiche Informationen nicht mehrfach gespeichert werden.

Um eine optimale Datenstruktur zu erreichen, bedarf es verschiedener Techniken. Sie werden im folgenden anhand eines einfachen Beispiels, der Erfassung von Daten von Schulungsteilnehmern eines Unternehmens, beschrieben.

Zunächst werden alle notwendigen Informationen zu der Problematik unsystematisch und unstrukturiert gesammelt. Mir erscheint es sinnvoll, folgende Daten zu erfassen:

- **Personendaten der Kursteilnehmer**
- **Kursort**
- **Kursbezeichnung**
- **Kursthema**
- **Datum der Kursbesuche**

Man hat so eine Grobstrukturierung der Daten erreicht, die bereits zu einer Tabelle zusammengefasst werden könnte.

Der Informationsgehalt der Tabelle ist noch sehr allgemein und ungenau. Um eine bessere Datenstruktur zu erreichen, kann man auf die Methoden des Normalisierungsverfahrens zurückgreifen.

Erste Normalform

Eine Tabelle ist nach der ersten Normalform ausgerichtet, wenn alle elementaren Informationen in einzelne Felder aufgeteilt werden.

Die Arbeit mit einem Datenbanksystem wird wesentlich erleichtert, wenn die Feldnamen nach folgenden Regeln vergeben werden:

- Feldnamen sollen kurz und prägnant sein.
- Sie sollen Hinweis auf den Feldinhalt gewähren.

Unter Berücksichtigung dieser Regeln ergeben sich die Feldnamen.

Feldliste	Feldnamen
Teilnehmervorname	Vorname
Teilnehmername	Name
Straße	Straße
Postleitzahl	PLZ
Ort	Ort
Geburt	Geburt
Geschlecht	Geschlecht
Gehalt	Gehalt
Schulungsbezeichnung	Schulungsbezeichnung
Kursbeginn	Kursbeginn
Kursende	Kursende
Veranstaltungsort	Veranstaltungsort
Veranstalteradresse	Adresse
Veranstaltername	Veranstalter
Kurskosten	Kurskosten
Veranstaltung besucht	Besucht

Einen Tabellenausschnitt der gefüllten Datei könnte folgendem gleichen:

Vorname	Name	...	Schulungsbezeichnung	Kurskosten	Kursbeginn	...
Kai	Müller	...	ABA Schulung	800	14.03.02	...
Kai	Müller	...	Grundlagen PC / Windows XP	500	13.12.02	...
Klaus	Ecke	...	ABA Schulung	800	14.03.02	...
Klaus	Ecke	...	Teamtraining	1400	07.04.02	...
Manfred	Stach	...	Projektmanagement	750	07.11.01	...
Elfi	Kaufmann	...	Laborschulung 2	1000	14.04.02	...
Elfi	Kaufmann	...	Grundlagen PC / Windows XP	500	13.12.02	...
Eike	Taro	...	Telefonmarketing-Training	1210	26.04.02	...
...

Man erkennt, dass in dieser Form Redundanzen auftreten. Für jeden neuen Lehrgang, der besucht wird, müssen die Personaldaten noch einmal aufgeführt werden. Auch die Schulungs- informationen sind für jeden Teilnehmer mehrmals vorhanden.

unterscheidbar macht; und zwar ohne Ausnahme. Es dürfen keine Datenin- haltswiederholungen vorkommen.

Zur Erfüllung dieser Anforderung ist das Verständnis sogenannter Identifikations- schlüsselfelder (vgl. Seite 10) von großer Bedeutung.

Zweite Normalform

Die zweite Normalform basiert auf der ersten Normalform und fordert eine eindeutige Identifikationsmöglichkeit der einzelnen Datensätze. Es muss also ein Kriterium gefunden werden, dass die Datensätze eindeutig

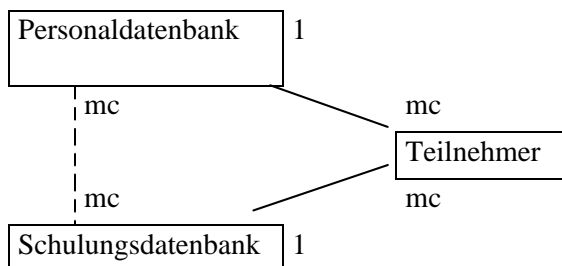
Die Personaldaten und die Schulungs- informationen werden in zwei Tabellen untergebracht. Es werden neue Felder - die Personalnummer und die laufende_NR - zur eindeutigen Identifizierung eingeführt.

Personal

Personal- nummer	Vorname	Name	Strasse	PLz	Ort	Geburt	Geschlecht	Gehalt
1002	Kai	Müller	Hallesche Str. 16	04838	Eilenburg	28.12.51	M	1800
1005	Ede	Pfau	Nicoleiplatz 2	04232	Leipzig	02.08.52	M	3000
1008	Eike	Taro	Musterstr. 6	09999	Musterdorf	10.09.47	M	2700
1010	Jürgen	Marx	Musterstr. 12	09999	Musterdorf	22.03.56	M	1900
1015	Erik	Wicki	Bolten-Weg 3	22587	Hamburg	09.07.50	M	2300
1016	Ulrich	Müller	Weide Str. 2 a	04838	Hohenpriessnitz	02.01.51	M	4900
1017	Klaus	Ecke	Schulze-Str. 11	04808	Wurzen	13.04.55	M	1300
1100	Kerstin	Gans	Hauptstr.10	04838	Moertitz	19.03.61	W	3000
...

Schulung

laufen- de NR	Schulungsbezeichnung	Kurs- kosten	Kurs- beginn	Kursende	Veranstalt- ingsort	Adresse	Veranstalter
1	ABA Schulung	800	14.03.02	15.03.02	Mannheim	...	S & B
2	Laborschulung 2	1000	14.04.02	16.04.02	Leipzig	...	S & B Leipzig
3	Teamtraining	1400	07.04.02	08.04.02	Berlin	...	S & B Berlin
4	Grundkurs Excel	600	25.11.02	26.11.02	München	...	S & B
5	Grundlagen PC / Windows XP	900	13.12.02	13.12.02	Chemnitz	...	S & B
6	Grundkurs Word	600	14.12.02	15.12.02	Leipzig	...	S & B Leipzig
7	Grundkurs Powerpoint	600	04.01.02	04.01.02	Leipzig	...	S & B Leipzig
8	Projektmanagement	750	07.11.01	07.11.01	Berlin	...	S & B Berlin
9	Telefonmarketing-Training	1210	26.04.02	27.04.02	Rostock	...	S & B Rostock
10	Personalplanung in der Praxis	1780	03.09.02	04.09.02	Essen	...	S & B Essen
11	Arbeitszeugnisse richtig formulieren & analysieren	980	18.09.02	19.09.02	Essen	...	S & B Essen



Die beiden Tabellen besitzen eine mc-mc Beziehung.

Diese ist nicht zulässig und muss aufgelöst werden.

Durch Einführung einer neuen Tabelle Teilnehmer wird die eine mc-mc Beziehung in zwei 1-mc Beziehungen umgewandelt.

Diese neue Tabelle könnte folgende Inhalte haben:

Teilnehmer

laufende NR	Personalnummer	Besucht
1	1002	Ja
1	1017	Ja
1	1101	Ja
1	1429	Ja
2	1114	Ja
2	1439	Ja
3	1010	Ja
3	1015	Ja
...

Dritte Normalform

Die dritte Normalform basiert auf der zweiten Normalform und stellt eine weitere Verfeinerung dar. Es dürfen innerhalb einer Tabelle die Feldwerte nur vom Identifikationsschlüssel abhängig sein und untereinander keine Abhängigkeiten haben.

Wenn du dir z.B. die Personaltabelle ansiehst, stellst du fest, dass sich diese Tabelle bereits in der 3. Normalform befindet. Ein Name kann z.B. mehrmals auftreten und dient nicht zur

eindeutigen Identifikation des Vornamens oder der Anschrift.

Auch die Tabelle **TEILNEHMER** befindet sich bereits in der dritten Normalform.

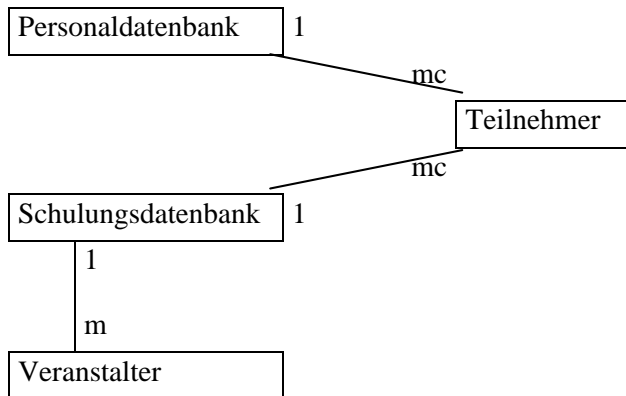
Anders sieht das bei der Schulungstabelle aus. Hier könnte man anhand der Veranstalternummer eindeutig auf Veranstalter, Veranstalteradresse und Veranstaltungsort schließen. Diese Daten sind zu separieren. Die Tabelle **SCHULUNG** muss also weiter unterteilt werden. Die beiden aufgeteilten Tabellen siehst du hier:

Schulung

laufende_NR	Schulungsbezeichnung	Kurskosten	Kursbeginn	Kursende	Veranstalter_NR
1	ABA Schulung	800	14.03.02	15.03.02	1
2	Laborschulung 2	1000	14.04.02	16.04.02	4
3	Teamtraining	1400	07.04.02	08.04.02	5
4	Grundkurs Excel	600	25.11.02	26.11.02	2
5	Grundlagen PC / Windows XP	500	13.12.02	13.12.02	3
...

Veranstalter

Veranstalter_NR	Veranstaltungsort	Adresse	Veranstalter
1	Mannheim	...	S & B Mannheim
4	Leipzig	...	S & B Leipzig
5	Berlin	...	S & B Berlin
2	München	...	S & B München
3	Chemnitz	...	S & B Chemnitz
6	Rostock	...	S & B Rostock
7	Essen	...	S & B Essen



Zusammenfassend sind hier die Beziehungen aller vier Tabellen dargestellt.

Der Entwurfsprozess

Der Entwurfsprozess kann kein geradliniges "Kochrezept" darstellen, bei dem aus der vage formulierten Aufgabenstellung automatisch die perfekte Datenbankapplikation resultiert. Vielmehr handelt es sich um einen Prozess, bei dem ein schrittweises Entwurfsverfahren immer wieder angewendet wird, bis zuletzt ein Daten-system resultiert, welches alle gestellten Anforderungen erfüllt. Dieser Prozess kann folgenderweise dargestellt werden:

Bestimme den Zweck der Datenbank.
Sammele alle Informationen, die du erfassen möchtest.
Lege fest, welche Tabellen notwendig sind. Ordne die relevanten Informationen verschiedenen Themen zu, z.B. Teilnehmer oder Lehrgänge. Für jedes dieser Themen wird in der Datenbank eine Tabelle erstellt. Bestimme, welche Felder zur Aufnahme der Informationen benötigt werden
Definiere Identifikationsschlüssel und Fremdschlüssel.
Vereinbare die Beziehungen zwischen den einzelnen Informationen. Sieh dir jede Tabelle an und entscheide, in welcher Beziehung die Daten dieser Tabelle zu den Daten anderer Tabellen in der Datenbank stehen. Füge eventuell in die Tabellen weitere Felder hinzu oder erstelle neue Tabellen.
Erstelle nun die Tabellen und einige Datensätze mit Beispieldaten. Prüfe, ob damit die gewünschten Ergebnisse erzielt werden und nimm bei Bedarf Änderungen vor.

Ein gut durchdachter Datenbankentwurf kann somit folgende **Probleme vermeiden** helfen:

- Nachträgliche Datenstrukturänderungen**
 Sie haben meist Auswirkungen auf andere Bereiche der Anwendung. So müssten Formulare oder Berichte ebenfalls verändert werden.
- Langsame Zugriffszeiten**
- Datenredundanzen**
 Sie sind problematisch, wenn sich der redundante Teil einer oder der redundante Teil in einer anderen Tabelle ändert. Es muss immer dafür gesorgt werden, dass die zu ändernde Information auch in allen Datenbereichen geändert wird. Das gleiche Problem tritt auch bei Löschvorgängen auf. Werden redundante Datenteile vergessen zu löschen, entstehen sog. "Datenleichen".
- Dateninkonsistenzen**
 Sie treten auf, wenn Datenteilbereiche sich ändern, diese Änderung aber an betroffene andere Datenteilbereiche nicht weitergegeben werden.

Die Datenbanksprache SQL in verschiedenen Systemen

Das relationale Datenbankmodell wurde 1970 von Dr. E. F. Codd eingeführt. Auf dieser Basis wurde 1974 von IBM die Datenbanksprache SEQUEL (Structured English QUery Language) entwickelt, die im Jahr 1976 zu SEQUEL2 weiterentwickelt wurde. In den ersten kommerziell verfügbaren relationalen Datenbanksystemen wurde eine Untermenge dieser Sprache integriert, die SQL genannt wurde. Diese hat im Laufe der Zeit verschiedene Normungsprozesse durchlaufen, die als SQL-86, SQL-89, SQL-92 oder SQL-99 bekannt geworden sind.

Die Umsetzung des firmenunabhängigen Standards in die Datenbankprodukte nimmt gewisse Zeit in Anspruch. Die meisten im Einsatz befindlichen Datenbanken unterstützen daher nur SQL92.

Dank der beschriebenen Normung verfügen wir heute über zahlreiche Implementierungen von SQL, wie z.B. Oracle, MS-Access, MS-SQL-Server, MySQL. Welches System du bei der Erstellung deiner Übungen nutzt, bleibt dir selbst überlassen.

Verschiedene Systeme kannst du als Test- oder Vollversion erhalten.

- kostenloser **ORACLE9i** Download
otn.oracle.com/software/content.html
- **SQL SERVER** als Evaluierungspaket
www.microsoft.com/germany/sql/
- kostenloser Download von **MYSQL**
www.mysql.de/downloads/index.html

Im vorliegenden Heft nutze ich zur Demonstration MS-Access und den MS-SQL-Server. Bist du speziell an der Problematik MySQL interessiert, empfehle ich das KnowWare EXTRA Heft 7 „PHP & MySQL - dynamische Webseiten“.

Nachdem du dich für ein Datenbanksystem entschieden hast, besteht der nächste Schritt darin, eine Datenbank anzulegen. Sie soll den Namen **SCHULUNG** erhalten.

Je nach Produkt gibt es verschiedene Verfahren die Datenbank **SCHULUNG** anzulegen. Bei einigen Datenbanksystemen steht dir der Befehl **CREATE DATABASE schulung**

zur Verfügung. Schau am besten in der jeweiligen Datenbankdokumentation nach, ob dir dieser Befehl zur Verfügung steht und welche Parameter noch anzugeben sind.

Der SQL-92-Standard definiert keinen Mechanismus zum expliziten Erzeugen einer Datenbank. Das ist Aufgabe der jeweiligen Datenbankprodukte.

MS-Access

MS-Access ist ein relationales Datenbanksystem, in dem die Vorzüge von WINDOWS - Übersichtlichkeit und leichte Bedienbarkeit - mit der Leistungsfähigkeit einer professionellen und bedienerfreundlichen Datenbank kombiniert sind. Als vor Jahren MS-Access 1.0 vorgestellt wurde, konnte sich kaum einer vorstellen, dass eine Datenbank für den Personalcomputer, die nicht xBase-kompatibel (xBase - Programmiergrundlage von z.B. dBase, Clipper u.a.) war, bedeutende Marktanteile erzielen kann. Heute jedoch hat Microsoft auf dem Datenbankmarkt zahlreiche Mitbewerber aus dem Rennen geworfen. Mittlerweile steht uns die Version MS Access 2002 zur Verfügung. Um SQL-Befehle zu testen, kannst du jede verfügbare MS-Accessversion nutzen.

Datenbank anlegen

Nachdem du MS-Access gestartet hast, wirst du über einen Assistenten aufgefordert, eine Datenbank anzulegen oder eine vorhandene Datenbank zu öffnen. Alternativ kannst du aber auch zum Anlegen einer Datenbank den Menüpunkt **DATEI | NEU** nutzen.

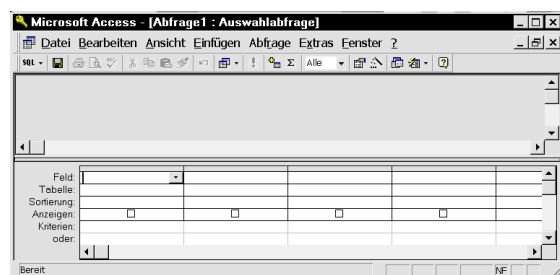


SQL nutzen

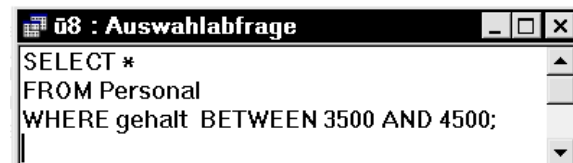
Wähle nun das Register **ABFRAGE**. Um SQL-Anweisungen einzugeben, ist immer eine neue Abfrage zu erstellen. Drücke also auf die Schaltfläche **NEU**.

Das Dialogfenster **TABELLE ANZEIGEN** schließt du.


Im Abfragefenster klickst du in der Symbolleiste auf die Schaltfläche **SQL**. Eine zweite Möglichkeit zum Start von SQL ist, dass du aus dem Menü **ANSICHT** den Befehl **SQL** wählst.



Es öffnet sich ein Abfragefenster, in das die jeweilige SQL-Anweisung eingetragen wird.



Möchtest du eine Anweisung abspeichern, so klicke auf das Diskettensymbol oder wähle **ABFRAGE SPEICHERN UNTER...** im Menü **DATEI**. Es ist ein Abfragenname einzugeben, der dann im Datenbankfenster erscheint. Jederzeit kann von dort auf die Taste Entwurf geklickt und so die SQL-Anweisung geändert werden.

Soll die Anweisung ausgeführt werden, klicke in der Symbolleiste auf  oder nutze im Menü **Ansicht** das Untermenü **Ausführen**.

MS-SQL-Server

Die Antwort von Microsoft auf die Oracle-Datenbank heißt MS-SQL Server.

Der MS-SQL-Server ist ein RDBMS-System (Relational Database Management System) für Client / Server.

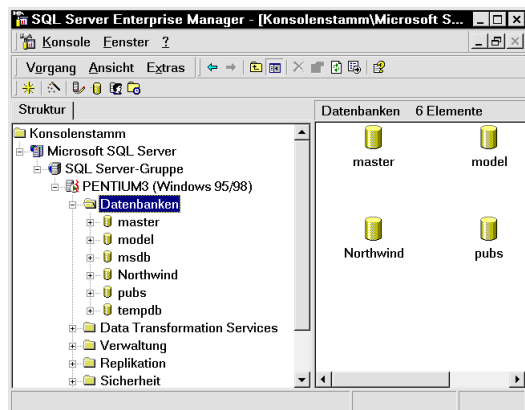
Ein Client ist in diesem Zusammenhang eine Systemkomponente, die Dienste oder Ressourcen von anderen Komponenten des Systems anfordert.

Ein Server ist eine Systemkomponente des Systems, die Dienste oder Ressourcen anderen Komponenten des Systems bereitstellt.

Anders als bei dateibasierten Systemen, wie z.B. bei MS-Access werden bei Client / Server Systemen die Datenbankdienste vom Client getrennt. Die Datenbankdienste werden auf dem leistungsstarken Server durchgeführt. Auf dem Clientrechner läuft die Anwendung, die den Datenzugriff gestaltet.

Datenbank anlegen

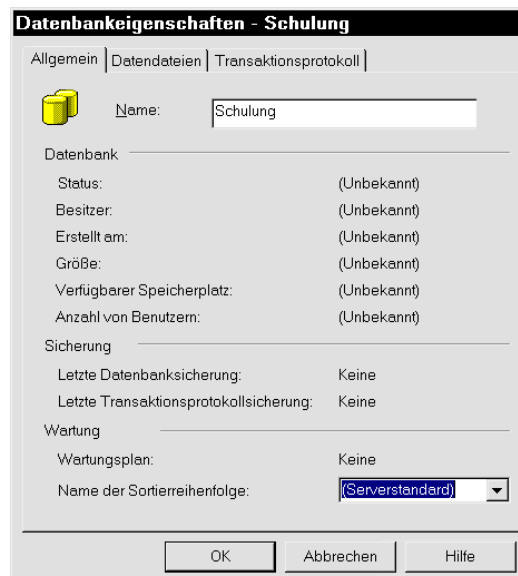
Für die administrativen Aufgaben steht dir der Enterprise Manager der SQL Servers zur Verfügung.



Um die Datenbank anzulegen, starte den Enterprise Manager und wähle den Datenbankserver aus.

Im Ordner Datenbanken befinden sich alle auf dem System befindlichen Datenbanken. Die Datenbanken **MASTER**, **MODEL**, **MSDB** und **TEMPDB** sind Systemdatenbanken und sollten nicht verändert werden. Bei **NORTHWIND** und **PUBS** handelt es sich um Testdatenbanken.

Um eine Datenbank anzulegen, markiere den Ordner Datenbank und wähle den Menüpunkt **VORGANG / NEUE DATENBANK**.



Gib den Dateinamen **SCHULUNG** ein. In der Registerkarte Datendateien kannst du den Speicherort und die Anfangsgröße festlegen. Für unsere Übungen solltest du die Standardeinstellungen übernehmen. Für das Transaktionsprotokoll übernimmst du auch die Standardeinstellungen. Im Transaktionsprotokoll werden alle Änderungen für den Fall der Wiederherstellung der Daten aufgezeichnet.

Für die Eingabe der SQL-Befehle hast du mehrere Möglichkeiten.

Auf der Ebene der Kommandozeile stehen dir die Programme **ISQL** bzw. **OSQL** zur Verfügung. Mit **ISQL** erfolgt die Kommunikation mit dem Datenbankserver über die DB-Library. Das Programm **OSQL** kommuniziert mittels ODBC mit dem Datenbankserver.

Um z.B. **OSQL** zu nutzen, gib in der Eingabeaufforderung zunächst

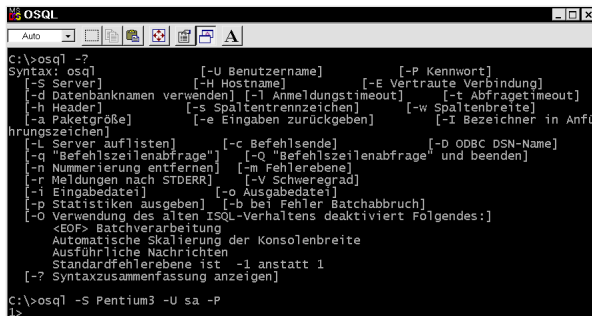
```
osql -?
```

ein. Als Ergebnis siehst du eine Übersicht aller Parameter.

Die Anmeldung an den Datenbankserver erfolgt über den Befehl

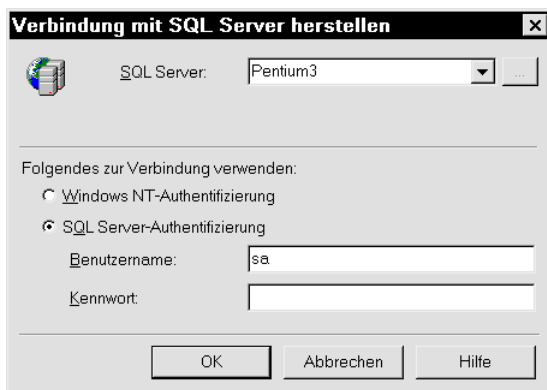
```
osql -S Server -U Benutzername -P Kennwort
```

Anschließend kannst du deine SQL-Befehle eingeben.



```
C:\>osql -?
Syntax: osql [-U Benutzername] [-P Kennwort]
[-S Server] [-H Hostname] [-E Vertraute Verbindung]
[-d Datenbanknamen verwenden] [-I Anmeldezeitlimit] [-t Abfragezeitlimit]
[-h Header] [-s Spaltenanzahl] [-w Spaltenbreite]
[-a Paketgröße] [-e Eingaben zurückgeben] [-I Bezeichner in Anfr
hungszeichen]
[-L Server auflisten] [-c Befehlsende] [-D ODBC DSN-Name]
[-q "Befehlszeilenabfrage" [-Q "Befehlszeilenabfrage" und beenden]
[-n Nummerierung entfernen] [-m Fehlerebene]
[-r Meldungen nach STDERR] [-V Schweregrad]
[-i Eingabedatei] [-o Ausgabedatei]
[-p Statistiken ausgeben] [-b bei Fehler Batchabbruch]
[-O Verwendung des ältesten ISQL-Verhaltens deaktiviert Folgendes:]
<EOF> Batchverarbeitung
Automatische Skalierung der Konsolenbreite
Ausführliche Nachrichten
Standardfehlerebene ist -1 anstatt 1
[-? Syntaxzusammenfassung anzeigen]
C:\>osql -S Pentium3 -U sa -P
1>
```

Komfortabler jedoch ist die Befehlseingabe über den **QUERY ANALYSER**. Nach dem Start ist wieder der Benutzername und ein Kennwort zur Anmeldung am Datenbankserver einzugeben.



In einem Datenbankfenster können nun die Befehle eingegeben werden.

