

Grundlagen

Access 97/2000



Max Woodtli

Inhaltsverzeichnis

1.	Informationen und Daten	4
1.1.	Datenverwaltung	4
2.	Betrachtungsebenen	4
2.1.	Reale Ebene	6
2.2.	Logische Ebene	6
2.3.	Physische Ebene	6
3.	Was ist eine Datenbank?	8
3.1.	Datenbanksystem	8
4.	Entwerfen einer Datenbank	10
4.1.	Entwurfsziele	12
5.	Erstellen einer ACCESS-Datenbank	18
5.1.	Erstellen einer Tabelle	20
5.2.	Hinzufügen und Speichern von Datensätzen	24
5.3.	Formulare erstellen	26
5.3.1.	Formulare verwenden	28
5.4.	Erstellen von Berichten	30
5.5	Zusätzliche Felder in Formulare und Berichte einfügen	34
5.6.	Erstellen von Abfragen	36
5.6.1.	Einbinden von Abfragen in einen Bericht	38
5.6.2.	Selektieren/Auswahlabfragen	40
5.6.3.	Auswahlabfragen-Aufgaben	42
5.6.4.	Aktionsabfragen	48
5.6.4.1.	Aktualisierungs-Abfrage	48
5.6.4.2.	Tabellenerstellungs-Abfrage	50
5.6.4.3.	Anfüge-Abfrage	52
5.6.4.4.	Lösch-Abfrage	54
5.7.	Fortgeschrittene Funktionen in Tabellen, Formularen und Berichten	56
5.7.1.	Gültigkeitsregeln definieren	56
5.7.2.	Steuerelemente in Formulare einfügen	58
5.7.3.	Mit Objekten gestalten	62
5.7.4.	Individuelle Berichte	66
6.	Projekt: FAKTURA	70
6.1.	Schrittweises Vorgehen	70
6.1.1.	Formularanalyse	70
6.1.2.	Erstellen des ER-Diagramms	72
6.1.3.	Erweiterung des ER-Diagramms	76
6.1.4.	Das Problem der M:M-Beziehung	76
6.1.5.	Beziehungsentität	78

6.2.	Normalisierung	82
6.2.1.	Die Normalisierungsregeln	84
6.2.2.	Aufgaben zur Normalisierung	88
6.3.	Realisierung mit ACCESS	90
6.3.1.	Tabellen erstellen	90
6.3.2.	Tabellen verknüpfen	92
6.3.3.	Abfragen (Views) erstellen als "Gerüst" für das Rechnungsformular	94
6.3.4.	Berechnete Felder in Abfragen	98
6.3.5.	Rechnungsformular erstellen	100
6.3.6.	Rechnungen bearbeiten	102
6.3.7.	Berechnung der Summen	104
6.3.8.	Rechnungen drucken	106
6.3.9.	Ergänzungsaufgaben zum Projekt: FAKTURA	110
7.0	Literaturverzeichnis	111

4. Entwerfen einer Datenbank

Grundsätzliches

Bevor der erste Griff zur Maus erfolgt und Sie sich auf das Anlegen einer Datenbank stürzen, sollten Sie sie skizzieren: Was soll rein-, was soll rauskommen, wie viele Datensätze könnten es werden, und welche Daten sind verknüpft? Ein sorgfältig ausgearbeitetes Konzept dient hierbei als Bauplan für die Struktur Ihrer Datenbank. Nur allzuleicht übersieht man beim Entwurf von Tabellen selbstgebaute Fallen und Engpässe, die später nur schwer zu beheben sind.

Ein Beispiel: Eine Mitgliederverwaltung für mehrere Vereine muss berücksichtigen, dass eine Person vielleicht in mehr als einem Verein vertreten ist, ohne dass man die Stammdaten des Mitglieds zweimal erfassen muss. **Daher ist die sorgfältige Planung der allererste und wichtigste Schritt beim Design einer Datenbank.**

Designphasen

Um nun eine solche Datenbank effizient planen zu können, teilt man das ganze Entwurfs-Prozedere in verschiedene **Designphasen** ein. Die erste Designphase ist die **Analyse des Informationsbedarfs** (Was soll in meiner Datenbank gespeichert werden? Welche Tabellen, Felder, Felddatentypen gibt es?).

In der zweiten Designphase folgt die Erstellung des **konzeptionellen Schemas** der Datenbank.

Datenmodelle

Um Daten und Datensysteme programmunabhängig beschreiben zu können, werden geeignete **Beschreibungswerkzeuge** benötigt. Man nennt dies ein **Datenmodell**. Es handelt sich hierbei um eine **Struktursprache**, welche sich zur Beschreibung von Datenbeständen eignet. Drei Datenmodelle haben sich eingebürgert, wobei wir auf die ersten beiden nicht näher eingehen werden. Es handelt sich um das **hierarchische**, das **netzwerkartige** und das **tabellarische Modell**, auch **Relationen-Modell** genannt.

Das Relationen-Modell

Das **Relationen-Modell** (Tabellen) geht auf **E.F. Codd** zurück. Es hat seine Bedeutung wegen der Systematik der Datendarstellung und der Entkoppelung der Datenbereitstellung von ihrer zukünftigen Verwendung. Dieses Modell gewinnt ständig an Bedeutung, insbesondere seit der Verfügbarkeit von entsprechenden Softwareprogrammen wie z.B. ACCESS, die sich auf dieses Modell abstützen.

Tabellenform

Man kann sich eine relationale Datenbank als **Menge von Tabellen** vorstellen, auf denen gewisse Operationen ausgeführt werden. Die **Spaltenüberschriften** sind **Attribute (Eigenschaften)**, eine **Zeile** ist ein **Datensatz**.

Naheliegende Operationen wären beispielsweise:

- eine Liste nach Buchtiteln zu bilden
- Bücher nach bestimmten Autoren zu selektieren

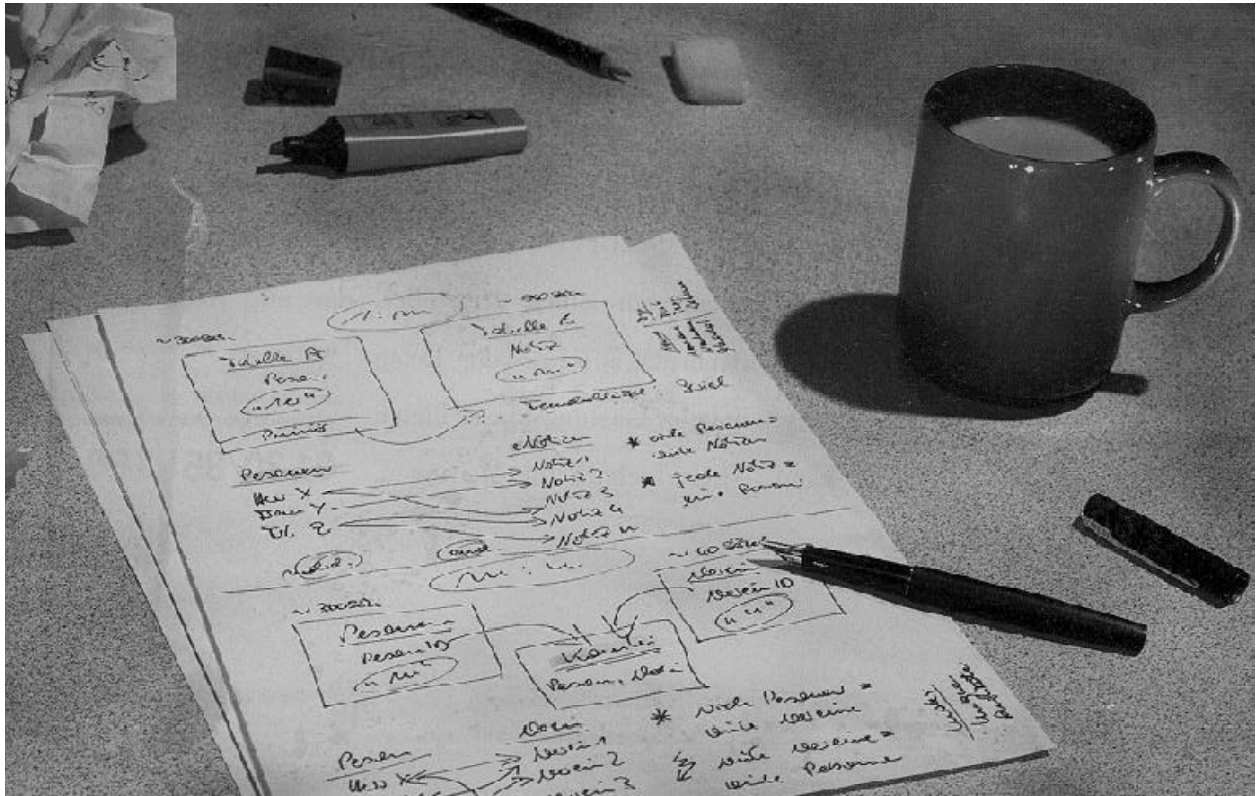


Abb. 5
Nichts geht über eine sorgfältige Planung

Quelle: WIN 94

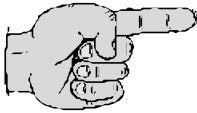
BUCH					
Buchtitel	Autor	Verlag	Datum	ISBN-Nr	Anzahl Seiten
Sichtweisen der Informatik	Wolfgang Coy	vieweg	1992	3-528-05263-5	109
Client/Server Technologie	Albert Kare	Springer	1994	3-540-57330-5	425
Probleme lösen mit Methode	J. Beetz	Oldenbourg	1994	3-486-21790-9	177
Wer erfindet die Computermysterien?	J. Weizenbaum	Herder	1993	3-451-04192-8	153
Informatik und Philosophie	P. Schiefe	Wissensch.	1993	3-411-16521-9	326
Aufbruch zu einer neuen Lernkultur	Bruno Kropf	Haupt	1993	3-258-04791-X	261

Abb. 6
Der Aufbau einer Tabelle

6.3.2. Tabellen verknüpfen

Grundsätzliches

Die Verknüpfung der einzelnen Tabellen erfolgt über die **Primär-** bzw. **Fremdschlüssel** der zu verknüpfenden Tabellen. Handelt es sich hierbei um eine 1 : M-Beziehung (z.B. KUNDE : RECHNUNGSKOPF), so spricht man auch von einer hierarchischen Verknüpfung, bei der z.B. die Tabelle KUNDE die **Master-** und die Tabelle: RECHNUNGSKOPF die **Detailtabelle** bildet. (Siehe Abb. 100)



Damit Sie Tabellen über Primär- bzw. Fremdschlüssel verknüpfen können, müssen diese sog. **Referenzfelder denselben Felddatentyp** und die **gleiche Feldlänge** haben. Dabei ist zu beachten, dass ACCESS den Felddatentyp **AutoWert** (geeignet für Primärschlüssel) als **Long Integer** speichert.

Referentielle Integrität

Bei der Bearbeitung von relational verknüpften Tabellen sind einige Sonderfälle zu berücksichtigen. Angenommen, Sie legen eine Rechnung für die Kundennummer 4711 an.

Wenn Sie anschliessend die Kundennummer 4711 in einem getrennten Arbeitsschritt löschen würden, entstünde eine Rechnung mit einer nicht mehr gültigen Kundennummer und die **referentielle Integrität** Ihrer Datenbank ginge verloren.

Es muss daher gewährleistet werden, dass **Referenzfelder** (Primär- und Fremdschlüssel), auf die sich ein anderer Datensatz bezieht, nicht gelöscht werden können.

Da ACCESS diese Einstellung in der Datenbank speichert, erkennt es die einmal definierte Verknüpfung von Datenbanktabellen automatisch.

Aufgabe

Verknüpfen Sie mit ACCESS die Tabellen gemäss **ER-Diagramm** (Abb. 86).

Vorgehen

1. Aktivieren Sie den Menüpunkt **Extras Beziehungen...**
2. In der darauffolgenden Dialogbox **Tabelle anzeigen** klicken Sie nacheinander alle 4 Tabellen an und fügen Sie über die Befehlsschaltfläche **Hinzufügen** hinzu.
3. Nachdem Sie alle 4 Tabellen hinzugefügt haben, schliessen Sie die Dialogbox **Tabelle anzeigen** über die Befehlsschaltfläche **Schliessen**.
4. Ziehen Sie im Fenster **Beziehungen** das **Primärschlüsselfeld** der Tabelle **KUNDE** zum entsprechenden **Fremdschlüsselfeld** der Tabelle **RECHNUNGSKOPF**. (Abb. 99)

./..

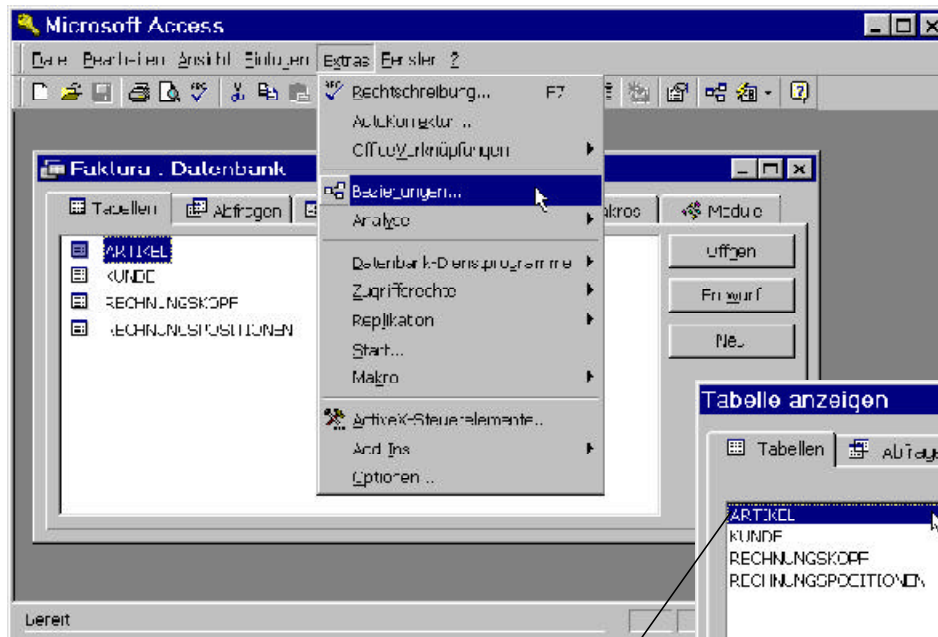


Abb. 97
Der ACCESS-Bildschirm
beim Verknüpfen von
Tabellen

Hier doppelklicken, um die
Tabelle hinzuzufügen

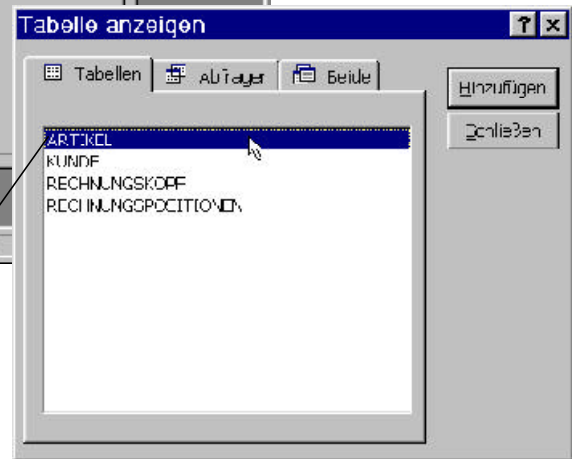
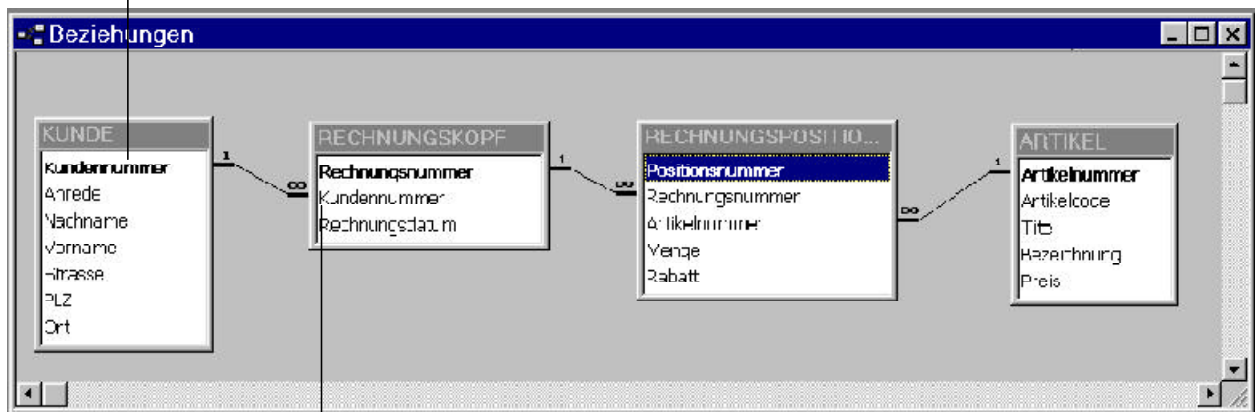


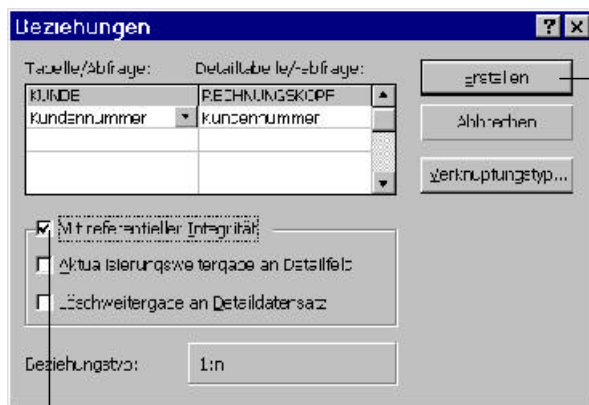
Abb. 98
Die Dialogbox "Tabelle anzeigen"

Primärschlüssel
von hier ...



... nach hier ziehen.

Abb. 99
Das Fenster "Beziehungen" mit sämtlichen für das
Projekt FAKTURA benötigten Tabellen



hier klicken, um die Beziehung
zu erstellen

Hier Referentielle Integrität aktivieren

Abb. 100
Die Dialogbox "Beziehungen"