

# Systematisches Datenbankdesign

## Praxisfall: Erstellung einer Datenbank Kundenverwaltung

### 1 Einführung

Eine Kundenverwaltung soll zusätzlich zu den Stammdaten des Kunden (Firma, Name, Adresse usw.) jeweils den Kundenbetreuer aufnehmen, den Vertriebsweg (Großhändler, Einzelhändler, Fachhändler usw.), sowie den monatlichen Umsatz des Kunden.

Folgende Prämissen gelten: Jeder Kunde hat nur einen Kundenbetreuer, ein Kundenbetreuer kann jedoch mehrere Kunden betreuen. Jedem Kunden soll ein Vertriebsweg zugeordnet werden, mehrere Kunden können demselben Vertriebsweg angehören.

Einsteiger versuchen normalerweise das Problem mit Hilfe einer Gesamttabelle zu lösen, wobei jede Zeile einem Kunden entspricht:

Firma	Name	Kunden-Adresse	Betreuer	Betreuer-Adresse	Vertriebsweg	Umsatz
Müller AG	Hr. Peters	München, Auweg 5	Hr. Aukamp	Dortmund, Alleestr. 12	EH	1/98: 12.021,- 2/98: 13.232,-
Schneider	Hr. Wolff	Köln, Brückstr. 1	Hr. Aukamp	Dortmund, Alleestr. 122	FH	1/98: 11.890,- 2/98: 10.898,-
Büro Peters	Hr. Peters	Köln, Am Ring 23	Fr. Maisel	Dortmund, Am Teich 5	EH	1/98: 12.800,- 2/98: 8.556,-

**Tabelle 1-1: Gesamttabelle**

Folgende Fehler und Probleme werden hierbei jedoch auftreten:

- **Redundanz:** Mehrfache Datenerfassung führt zu **Datenduplikaten** (z.B. Adresse des Betreuers Hr. Aukamp), Fehler bei der Dateneingabe führen dann sehr schnell zu Mehrdeutigkeiten im Datenbestand (siehe Fehler in der Adresse). Außerdem führt Redundanz zu Speicherplatzverschwendung: In kleinen Datenbanken wäre die Speicherplatzverschwendung zwar zu verkraften, denn ob die Datenbank 96 KB oder 2 MB groß ist, ist relativ bedeutungslos auf modernen Computern. Aber in großen Datenbanken spielt es schon eine Rolle, ob sie 10 MB oder 200 MB groß sind!
- **Einfügeanomalie:** Wird ein neuer Betreuer eingestellt und in die Datenbank eingegeben, der noch keinem Kunden zugeordnet ist, müßte eine fiktive Kundennummer vergeben werden (da jede Zeile einem Kunden entspricht!)
- **Änderungsanomalie:** Die Änderungsanomalie ist in der Praxis die schwerwiegendste. **Inkonsistenzen** im Datenbestand werden durch Tippfehler sehr leicht entstehen (Wenn z.B. der Betreuer Hr. Aukamp durch einen anderen Mitarbeiter ersetzt wird, müßte in jeder Zelle der Name und die Anschrift geändert werden). Würden Zellen dabei vergessen werden, so wäre der Datenbestand nicht mehr eindeutig. Man kann zwar durch *Suchen & Ersetzen* den Namen "Aukamp" durch "Breuer" ersetzen, aber nur, wenn der Name "Aukamp" auch immer richtig geschrieben wurde. Ein Datensatz mit dem falsch geschriebenen Namen "Aukamb" würde also nicht automatisch geändert werden, die Folge wäre eine Inkonsistenz im Datenbestand.
- **Löschanomalie:** Die Kundenbetreuerin Fr. Maisel ist momentan nur für einen Kunden zuständig. Würde dieser Kunde (also die komplette Zeile) gelöscht werden, wären auch die Daten der Kundenbetreuerin Fr. Maisel gelöscht.

Aus diesem Grunde wird eine Datenbank in Access erstellt, die die o.g. Probleme vermeiden soll.

### 2 Konzeptionalisierung

Damit Dateninkonsistenzen und Anomalien nicht auftreten, wird die Datenbank im Sinne des relationalen Datenbankmodells in mehrere Tabellen aufgeteilt. Hierzu wird der Prozeß der schrittweisen Normalisierung angewendet<sup>1</sup>.



### Erste Normalform

Eine Tabelle hat dann die Erste Normalform, wenn sie nur einfache (atomisierte) Merkmalswerte enthält.

Die Erste Normalform ist erreicht, wenn eine Tabelle atomisiert ist, sich in jeder Zelle also nur ein Datenelement befindet.

Tabelle 1-1: Gesamttabelle widerspricht der Ersten Normalform, da die Zellen für den Umsatz mehrere Datenelemente enthalten. Da die Datenmenge in der Zelle Umsatz jeden Monat größer wird, läßt sich die Datenfeldgröße nicht beschränken. Dies ist jedoch nötig, da Access im Tabellenentwurf eine Datenfeldgröße benötigt<sup>2</sup>.

Nach der Atomisierung der Gesamttabelle (sie befindet sich nun in der Ersten Normalform) könnte sie folgendermaßen aussehen:

<u>Datensatz Nr</u>	<u>Firma</u>	<u>Name</u>	<u>Kunden-Adresse</u>	<u>Betreuer</u>	<u>Betreuer-Adresse</u>	<u>Vertriebsweg</u>	<u>Umsatz</u>
100	Müller AG	Hr. Peters	München, Auweg 5	Hr. Aukamp	Dortmund, Alleestr. 12	EH	1/98: 12.021,-
101	Müller AG	Hr. Peters	München, Auweg 5	Hr. Aukamp	Dortmund, Alleestr. 12	EH	2/98: 13.232,-
102	Schneider	Hr. Wolff	Köln, Brückstr. 1	Hr. Aukamp	Dortmund, Alleestr. 12	FH	1/98: 11.890,-
103	Schneider	Hr. Wolff	Köln, Brückstr. 1	Hr. Aukamp	Dortmund, Alleestr. 12	FH	2/98: 10.898,-
104	Büro Peters	Hr. Peters	Köln, Am Ring 23	Fr. Maisel	Dortmund, Am Teich 5	EH	1/98: 12.800,-
105	Büro Peters	Hr. Peters	Köln, Am Ring 23	Fr. Maisel	Dortmund, Am Teich 5	EH	2/98: 8.556,-

**Tabelle 2-1: atomisierte Gesamttabelle**

Das Problem der mangelnden Atomisierung ist nun zwar gelöst, dafür haben wir nun vermehrt Redundanzen. Der Kunde "Müller AG" ist beispielsweise mit 2 Datensätzen vertreten. Um die Datensätze nun aber voneinander unterscheiden zu können, wird ein künstlicher Primärschlüssel eingefügt (Datensatznummer).

Die Gesamttabelle muß in einzelne Tabellen aufgeteilt werden, um die bereits angesprochenen Probleme zu lösen. Ganz allgemein werden in den Einzeltabellen inhaltlich zusammengehörende Datenelemente zusammengefaßt. Diese Aufteilung der Einzeltabellen wird regelkonform nach den Regeln der Normalisierung vorgenommen.



### Zweite Normalform

Eine Tabelle liegt in der Zweiten Normalform vor, wenn sie sich bereits in der [Ersten Normalform](#) befindet und wenn jedes Merkmal (außer dem Schlüssel) unmittelbar vom Schlüssel abhängt.

Warum befindet sich die Tabelle 2-1: atomisierte Gesamttabelle noch nicht in der Zweiten Normalform? Hängt jedes Merkmal (außer dem Schlüssel) unmittelbar vom Schlüssel **KuNr** ab? Anders ausgedrückt darf es also keine Werte mehr geben, die noch von der Kundennummer unabhängig sind.

- **Hängt der Nachname von der Kundennummer ab?** Ja, denn jeder Kunde (und damit sein Nachname) soll durch die Kundennummer eindeutig identifizierbar sein. Der Nachname ist also, anders ausgedrückt, nicht unabhängig von der Kundennummer.
- **Hängt der Betreuer von der Kundennummer ab?** Nein, denn nichts spräche dagegen, wenn dem Kunden "Müller AG" mit der Kundennummer 100 die Betreuerin Fr. Maisel zugewiesen wäre. Der Betreuer Hr. Aukamp ist also nicht zwangsweise mit der Kundennummer 100 verbunden. Das zeigt sich auch darin, daß Hr. Aukamp noch einen anderen Kunden mit einer anderen Kundennummer betreut. Der Betreuer ist also, anders ausgedrückt, unabhängig von der Kundennummer.
- **Hängt der Vertriebsweg von der Kundennummer ab?** Nein, denn der Vertriebsweg EH ist auch dem Kunden "Büro Peters" zugewiesen. Der Vertriebsweg ist also nicht zwangsweise mit der Kundennummer 100 verbunden, er ist also von ihr unabhängig.
- **Hängt der Umsatz von der Kundennummer ab?** Nein, denn ein anderer Kunde könnte zufällig den gleichen Umsatz erzielt haben, er ist also von der Kundennummer unabhängig.

Um nun die atomisierte Gesamttabelle in die Zweite Normalform zu überführen, werden inhaltlich zusammengehörende Datenelemente in Teiltabellen zusammengefaßt und anschließend die Prüfung in den Teiltabellen auf ein Vorliegen der [Zweiten Normalform](#) wiederholt.

Bei der Aufteilung einer Gesamttabelle muß das Prinzip der Verlustfreiheit gewährleistet sein, d.h. daß durch die Aufteilung keine Informationen verlorengehen dürfen.



#### **Verlustfreiheit**

*Eine Zerlegung ist dann verlustfrei, wenn der Verbund der Zerlegungen wieder die ursprünglich zerlegte Tabelle ergibt.*

Um die Verlustfreiheit zu gewährleisten ist bei der Aufteilung einer Gesamttabelle in Einzeltabellen die Vergabe von Schlüsselfeldern (Primärschlüsselfelder) nötig, durch die der Zusammenhalt der Tabellen hergestellt wird<sup>3</sup>.

Die Tabellen in der Zweiten Normalform könnten folgendermaßen aussehen:

<b>KuNr</b>	<b>Firma</b>	<b>Name</b>	<b>PLZ</b>	<b>Ort</b>	<b>Strasse</b>	<b>Telefon</b>	<b>PersNr</b>
100	Müller AG	Peters	80976	München	Auweg 5	089/765465	1
101	Schneider	Wolff	43526	Köln	Brückstr. 1	0201/65465	1
102	Büro Peters	Schmitt	43526	Köln	Am Ring 23	0221/87433	2

**Tabelle 2-2: Teiltabelle KUNDEN**

<b>PersNr</b>	<b>Betreuername</b>	<b>PLZ</b>	<b>Ort</b>	<b>Strasse</b>
1	Aukamp	44136	Dortmund	Alleestr. 12
2	Maisel	44126	Dortmund	Am Teich 5

**Tabelle 2-3: Teiltabelle BETREUER**

<b>VwNr</b>	<b>Vertriebsweg</b>
100	Großhandel
101	Fachhandel
102	Einzelhandel

**Tabelle 2-4: Teiltabelle VERTRIEBSWEG**

<b>UNr</b>	<b>Datum</b>	<b>Umsatz</b>	<b>KuNr</b>
1	1/98	12.021,-	100
2	2/98	13.232,-	100
3	1/98	11.890,-	101
4	2/98	10.898,-	101

**Tabelle 2-5: Teiltabelle UMSATZ**

Warum befindet sich jetzt beispielsweise die Tabelle 2-3: Teiltabelle BETREUER in der Zweiten Normalform?

- **Hängt der Betreuername von der Personalnummer ab?** Ja, denn jeder Betreuername soll ja gerade durch eine eindeutige Nummer identifiziert werden können. Der Betreuername ist also, anders ausgedrückt, nicht unabhängig von der **Personalnummer**.
- **Hängt der Ort von der Betreuernummer ab?** Strenggenommen nein, denn es könnten noch andere Betreuer (wie z.B. Fr. Maisel) in Dortmund wohnen. Strenggenommen müßte also diese Tabelle wiederum noch weiter aufgeteilt werden, um z.B. eine Tabelle "ORTE" zu erhalten. An dieser Stelle soll jedoch aus didaktischen Gründen, um also das Beispiel nicht zu sehr zu verkomplizieren, diese Tatsache ignoriert werden und diese Teiltabelle als der Zweiten Normalform genügend angesehen werden.

Es soll eine weitere Tabelle GEHALT erstellt werden, in der die Gehälter der Kundenbetreuer gespeichert sind. Dies ist eine Erweiterung der Datenbank, die lediglich aufgenommen wird, um einen weiteren Beziehungstyp vorzustellen<sup>4</sup>, sowie die Möglichkeiten der Zugriffsbeschränkung zu demonstrieren.

<b>PersNr</b>	<b>Gehalt</b>
1	DM 6290,-
2	DM 5890,-

**Tabelle 2-6: Teiltabelle GEHALT**

<sup>3</sup> Durch eine Abfrage über alle Teiltabellen muß wieder die Ursprungstabelle erzeugt werden können.



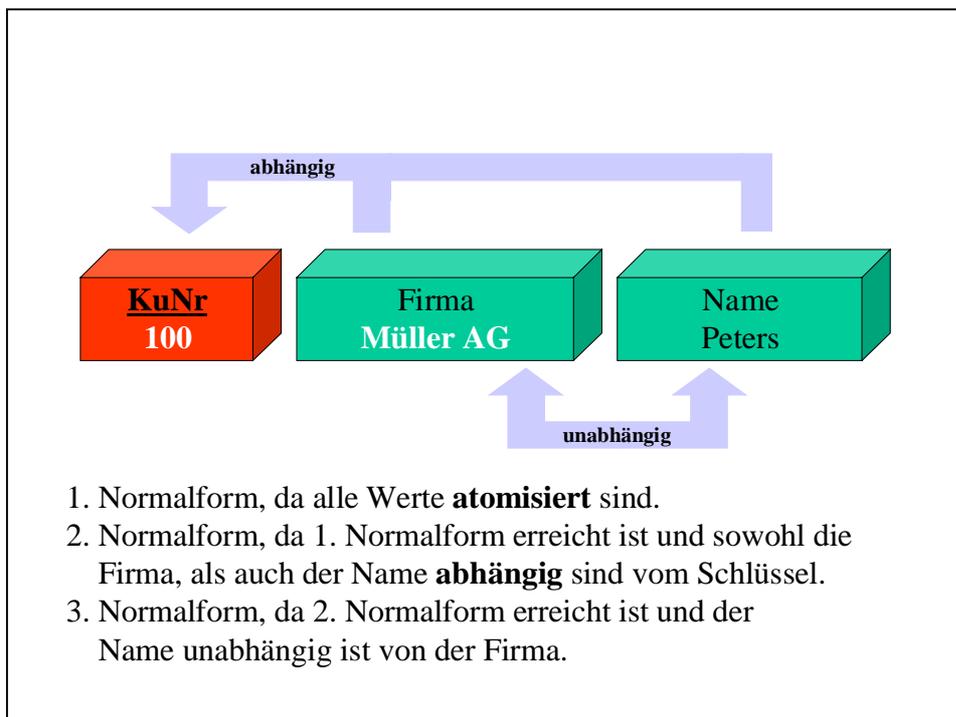
### Dritte Normalform

Eine Tabelle hat dann die Dritte Normalform erreicht, wenn sie sich in der [Zweiten Normalform](#) befindet und wenn ihre Merkmale (außer dem Schlüssel) untereinander unabhängig sind, also keine transitiven Abhängigkeiten mehr bestehen.

Liegt beispielsweise die Tabelle 2-2: Teiltabelle KUNDEN in der Dritten Normalform vor?

- **Hängt der Name (Inhaber oder Ansprechpartner) von dem Firmennamen ab?** Nein, der Inhaber oder Ansprechpartner könnte sich ändern, ohne daß sich zwangsweise der Firmenname ebenfalls ändern müßte. Oder, anders ausgedrückt, die Firma könnte umfirmieren, ohne daß sich der Ansprechpartner ändern müßte. Beide Merkmale hängen zwar vom Schlüssel Kundenummer ab, sind jedoch untereinander unabhängig. Wären untereinander noch Abhängigkeiten, so läge noch eine sogenannte transitive Abhängigkeit vor, die Dritte Normalform wäre also noch nicht erreicht, die Tabelle müßte demnach noch weiter aufgeteilt werden.

Eine komprimierte Darstellung zur Tabelle 2-2: Teiltabelle KUNDEN ist in Abbildung 3.1.1-1 zu sehen.



**Abbildung 3.1.1-1: Normalformen**

Obwohl es in der Theorie noch höhere Normalformen als die Dritte Normalform gibt, reicht in der Regel das Erreichen der Dritten Normalform aus.

Eine Gesamtübersicht über den derzeitigen, konzeptionellen Stand unserer Datenbank gibt Abbildung 3.1.1-2 und Abbildung 3.1.1-3.



**Vorsicht:** Verändern Sie nach der Konzeptionsphase Ihre Tabellen nicht mehr! Insbesondere das Hinzufügen von Merkmalen (Datenfeldern), weil man feststellt, daß unbedingt noch dieses oder jenes Merkmal aufgenommen werden müßte, ist kritisch! Wenn Sie Veränderungen durchführen, überprüfen Sie Ihr Konzept nochmals und überprüfen Sie vor allem die Einhaltung der Regeln der Normalformen. Oftmals stellt sich nach der Implementierung der Datenbank heraus, daß gegen wichtige Regeln verstoßen wurde. Dies kann dazu führen, daß Ihr Datenbankdesign für größere Datenmengen oder den von Ihnen gewünschten Zweck ungeeignet ist. Eine komplette Neuimplementierung ist dann oftmals notwendig.

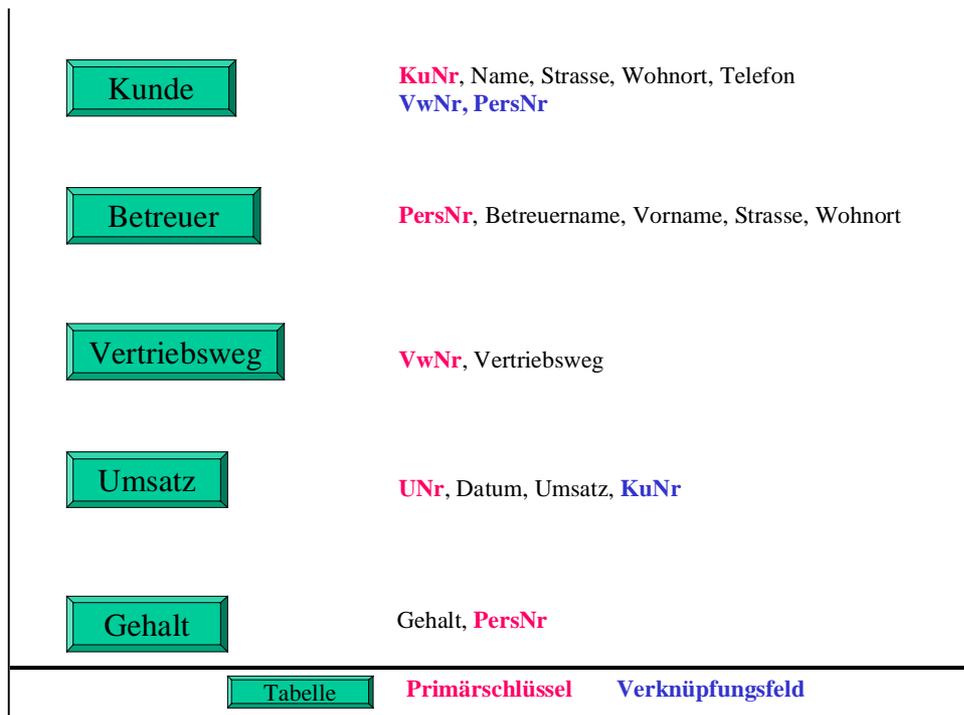


Abbildung 3.1.1-2: Tabellen

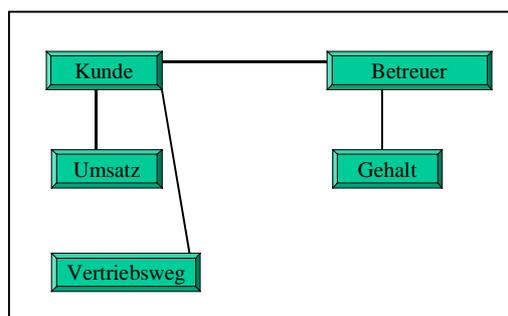


Abbildung 3.1.1-3: Verknüpfungen

### 3 Implementierung

#### 3.1 Tabellenerstellung

In der Tabellenentwurfsansicht sollen exemplarisch die Tabelle 2-2: Teiltabelle KUNDEN und die Tabelle 2-5: Teiltabelle UMSATZ definiert werden.

##### 3.1.1 Teiltabelle Kunden

Nach der Vergabe des Feldnamens **KuNr** für das erste Feld muß entschieden werden, welchen Datentyp der Feldname aufweisen soll. Sinnvolle Alternativen wären die Datentypen AUTOWERT oder ZAHL:

Beim Datentyp AUTOWERT werden automatisch für jeden neuen Datensatz fortlaufende, eindeutige Zahlen vergeben.

Ist das in diesem Falle sinnvoll? Wenn bereits Kundennummern existieren und sie daher nicht neu vergeben werden sollen, oder wenn sie eine bestimmte Syntax aufweisen, sollte der Datentyp AUTOWERT nicht benutzt werden. In unserem Beispiel wollen wir den Datentyp ZAHL verwenden.

Die nächste Frage, die beantwortet werden muß, bezieht sich auf die Feldgröße des Datentyps ZAHL: Sie bestimmt u.a. die Anzahl der (Nachkomma-)Stellen, den damit darstellbaren Wertebereich, sowie die weiteren Verarbeitungsmöglichkeiten<sup>5</sup>.

Feldgröße	Beschreibung	Dezimale Genauigkeit	Speichergröße
Byte	Speichert Zahlen von 0 bis 255	Keine	1 Byte
Integer	Speichert Zahlen von -32.768 bis 32.767	Keine	2 Bytes
Long Integer	(Voreinstellung) Speichert Zahlen von -2.147.483.648 bis 2.147.483.647	Keine	4 Bytes
Single	Speichert Zahlen von -3.402823E38 bis -1.401298E-45 für negative Werte, und von 1.401298E-45 bis 3.402823E38 für positive Werte.	7	4 Bytes
Double	Speichert Zahlen von -1.79769313486231E308 bis -4.94065645841247E-324 für negative Werte, und von 1.79769313486231E308 bis 4.94065645841247E-324 für positive Werte.	15	8 Bytes
Replikations- ID	Globally unique identifier (GUID)	N/A	16 Bytes

**Tabelle 3-1: Überblick über Feldgrößen von Zahl**

Da die Kundennummer keine Nachkommastellen aufweist, sondern mit Ganzzahlen dargestellt werden kann, beschränkt sich die Auswahl auf die Feldgrößen Byte, Integer und Long Integer. Wäre sichergestellt, daß es niemals mehr als 32.767 Kunden geben wird, würde Integer ausreichen, ansonsten müßte Long Integer gewählt werden<sup>6</sup>. Wir wollen die Feldgröße Long Integer nehmen.

Als nächstes überlegen wir, ob die Eingabe einer Kundennummer notwendig ist. Da die Kundennummer einen Kunden eindeutig identifiziert, sollte auch die Eingabe einer Kundennummer erzwungen werden. Dies erreicht man, indem die Eigenschaft *Eingabe erforderlich* auf „Ja“ gesetzt wird (der Standardwert „0“ sollte dann gelöscht werden).

Im nächsten Schritt wird die Kundennummer **KuNr** als Primärschlüsselfeld definiert.

Der Primärschlüssel ist ein eindeutiges Kennzeichen eines Datensatzes und die Eindeutigkeit wird vom System überwacht. Ist es in diesem Falle überhaupt nötig, die Eigenschaft *Eingabe erforderlich* auf „Ja“ zu setzen?

Strenggenommen nein, denn (laut Vorgabe) weist die Eigenschaft *Standardwert* den Wert „0“ auf. Würde also beim erstenmal ein Datensatz ohne Kundennummer eingegeben, würde sie automatisch „0“ lauten! Beim zweitenmal hingegen käme die Fehlermeldung *„Die von Ihnen vorgenommenen Änderungen an der Tabelle konnten nicht vorgenommen werden, da der Index, Primärschlüssel oder die Beziehung mehrfach vorkommende Werte enthalten würde“*. Wenn hingegen die Eigenschaft *Eingabe erforderlich* auf „Ja“ gesetzt wird, würde bei sonst gleichen Bedingungen lediglich eine andere Fehlermeldung erscheinen: *„Feld <Name> kann keinen Null-Wert enthalten, da die Required-Eigenschaft für dieses Feld den Wert True hat. Geben Sie in das Feld einen Wert ein“*.

Als letztes sollte im Feld Beschreibung ein erklärender Text zum Feldnamen aufgenommen werden. Einerseits dient es der Dokumentation und andererseits wird der Beschreibungstext beim Einsatz von Formularen als Feldeingabehilfe in der Statuszeile angezeigt.

Der Feldname **Firma** bekommt den Datentyp TEXT zugewiesen. Die Feldgröße wird mit 50 Zeichen vorgeschlagen und kann auch in aller Regel übernommen werden. Unnötigen Speicherplatzbedarf brauchen Sie an dieser Stelle nicht zu befürchten, da die Feldgröße für den Datentyp Text dynamisch vergeben wird und Sie hier lediglich die obere Grenze vorgeben. Damit bei der Dateneingabe der Firmenname nicht ausgelassen werden kann, sollte die Eigenschaft *Eingabe erforderlich* auf „Ja“ gesetzt werden.

Der Feldname **Name** bekommt ebenfalls den Datentyp TEXT zugewiesen, wobei die Eigenschaft *Eingabe erforderlich* „Nein“ bleiben kann.

Für den Feldnamen **PLZ** muß überlegt werden, welcher Datentyp zugewiesen werden sollte. Die Postleitzahl besteht aus 5 Zahlen, ohne Nachkommastellen, es läge also nahe, den Datentyp ZAHL zu benutzen. Hierbei sollten jedoch folgende Überlegungen vorgenommen werden: Die Feldgröße Integer ist zwar fünfstellig, geht aber nur bis zur Zahl 32.767. Es müßte also die Feldgröße Long Integer benutzt werden. Aber was würde mit ostdeutschen Postleitzahlen, die mit einer 0 beginnen, passieren? Diese könnten nicht erfaßt werden (Vornull-

<sup>6</sup> Allerdings kann bei Bedarf auch nachträglich der Wertebereich durch Auswahl einer anderen Feldgröße

Unterdrückung). Ist es denn überhaupt nötig, den Datentyp ZAHL zu benutzen? Nötig wäre es, wenn mit den Postleitzahlen mathematische Berechnungen durchgeführt werden sollen, dies ist jedoch nicht der Fall. Der Datentyp TEXT ist hier besser geeignet. Er löst einerseits das Problem mit der führenden Null und andererseits könnte man noch die Staatskürzel integrieren<sup>7</sup>. Die Feldgröße könnte auf 8 Zeichen begrenzt werden, ist jedoch nicht zwingend nötig. Die Eingabe sollte nicht erzwungen werden, da es durchaus denkbar wäre, daß zum Zeitpunkt der Datenerfassung diese Daten noch nicht bekannt sind und später nachgetragen werden können.

Die Felder **Wohnort** und **Strasse** sind ebenfalls vom Datentyp TEXT, wobei die Feldgröße 50 Zeichen betragen und die Eigenschaft *Eingabe erforderlich* auf „Nein“ gesetzt bleiben sollte.

Das Feld **Telefon** sollte ebenfalls vom Datentyp TEXT und nicht ZAHL sein, da Berechnungen nicht nötig sind, führende Nullen die Regel sind und meistens ein Zeichen zum Trennen von Vorwahl und Rufnummer benutzt wird. Die Feldgröße könnte auf 20 Zeichen reduziert werden.

Die Felder Personalnummer **PersNr** und Vertriebswegnummer **VwNr** sind Fremdschlüssel<sup>8</sup>, mit deren Hilfe die Anbindung an die Tabellen Betreuer und Kunden vorgenommen werden. Die Datentypen sollten identisch, zumindest kompatibel, mit den jeweiligen Feldern aus den Teiltabellen sein. Wir wollen den Datentyp ZAHL, Feldgröße Long Integer nehmen<sup>9</sup>.

Die Eigenschaft *Eingabe erforderlich* sollte auf „Nein“ gesetzt bleiben, da es möglich sein soll, den Betreuer erst später einzutragen.

tblKunden : Tabelle			
	Feldname	Felddatentyp	Beschreibung
🔑	KuNr	Zahl	Primärschlüssel Kundennummer
	Firma	Text	Firmenname
	Name	Text	Name des Ansprechpartners in der Firma
	Strasse	Text	Strassenname des Firmensitzes
	PLZ	Text	Postleitzahl des Firmensitzes
	Wohnort	Text	Wohnort des Firmensitzes
	Telefon	Text	Telefonnummer der Firma bzw. des Ansprechpartners
	PersNr	Zahl	Fremdschlüssel zur Verknüpfung mit der Tabelle Betreuer
▶	VwNr	Zahl	Fremdschlüssel zur Verknüpfung mit der Tabelle Vertriebsweg

Abbildung 3.1.1-1: Tabelle tblKunden

Die Tabelle sollte schließlich unter einem selbsterklärenden Namen gespeichert werden: Der Tabellenname KUNDEN bietet sich an, allerdings hat es sich vielfach eingebürgert, den Objekttyp als Kürzel voranzustellen. Bei Tabellen also das Kürzel „tbl“<sup>10</sup>. Der Tabellenname sollte also „tblKunden“ lauten.

### 3.1.2 Teiltabelle Umsatz

Zuerst betrachten wir den Grund, warum überhaupt eine eigene Teiltabelle für den Umsatz genommen werden soll: Würden wir in der Tabelle Kunden Felder für den Umsatz eintragen oder reservieren, z.B. in der Form Jan98, Feb98, Mrz98 usw., müßten demzufolge auch jeden Monat die Eingabemasken auf die geänderten Felder abgestimmt werden. Außerdem würden wir recht schnell an eine Grenze bezüglich der Tabellenbreite (also Anzahl der Felder) stoßen. Darüberhinaus hätten wir enorme Probleme, Berechnungen durchzuführen, wie beispielsweise den Jahresumsatz eines Kunden.

Daher wird der Umsatz in eine Teiltabelle ausgelagert und so aufgebaut, daß sich die Tabelle nicht in der Breite vergrößert, sondern in der Länge (also eine Vergrößerung der Anzahl der Datensätze).

Nach der Vergabe des Feldnamens **UNr** für das erste Feld muß entschieden werden, welchen Datentyp der Feldname aufweisen soll. Die Umsatznummer **UNr** ist ein eindeutiger Identifikator (Primärschlüssel) für jeden Umsatzdatensatz. Er braucht nicht von Hand eingegeben zu werden, sondern die Vergabe soll automatisch durch

<sup>7</sup> Man könnte beispielsweise die Unterscheidung in „D-44135“ und „CH-4088“ treffen.

<sup>8</sup> In der Tabelle BETREUER wäre **PersNr** der Primärschlüssel, in der Tabelle KUNDEN ist **PersNr** jedoch ein Fremdschlüssel

<sup>9</sup> Die Feldtypen Zahl und Autowert sind beispielsweise identisch. Für den Fremdschlüssel können wir nicht den Datentyp Autowert benutzen, da wir die Kontrolle über den Fremdschlüssel behalten müssen, um eine Verknüpfung zu ermöglichen.

ein Autofeld erfolgen. Das Feld **Datum** soll vom Datentyp DATUM/UHRZEIT sein, damit später Datumsberechnungen leichter vorgenommen werden können. Das Feld **Umsatz** ist vom Datentyp WÄHRUNG. Bei den Feldern **Datum**, **Umsatz** und **KuNr** sollte die Eingabe erforderlich sein, da sonst die Anlage eines Datensatzes sinnlos wäre. Außerdem sollte die Eigenschaft Standardwert nicht auf „0“ stehen, damit die erforderliche Eingabe nicht übergangen werden kann.

Das letzte Feld, die Kundennummer **KuNr**, ist wieder ein Fremdschlüssel, mit dessen Hilfe die Anbindung an die Tabelle der Kunden vorgenommen wird. Der Datentyp sollte wieder identisch, zumindest kompatibel, zu dem Feld **KuNr** in der Tabelle Kunden sein. Wir wollen den Datentyp ZAHL, Feldgröße Long Integer nehmen. Gespeichert wird die Tabelle unter dem Namen „tblKunden“.

tblUmsatz : Tabelle			
	Feldname	Felddatentyp	Beschreibung
PK	UNr	AutoWert	Primärschlüssel Umsatznummer
	Datum	Datum/Uhrzeit	Umsatzdatum
	Umsatz	Währung	Umsatz der Firma
	KuNr	Zahl	Fremdschlüssel zur Verknüpfung mit der Tabelle Kunden

Abbildung 3.1.2-1: Tabelle tblUmsatz

Die restlichen Tabellen werden analog erstellt.

### 3.2 Beziehungen zwischen den Tabellen

Nachdem die Tabellen erzeugt wurden, werden die Beziehungen zwischen den Tabellen eingerichtet. Im Fenster Beziehungen wird definiert, in welchen Beziehungen die Tabellen zueinander stehen, und über welche Schlüsselfelder sie miteinander verknüpft sind.

Dazu müssen wir erst einmal untersuchen, welche Beziehungstypen zwischen den Tabellen bestehen. Anhand der [Prämissen](#) können wir dies feststellen: „Jeder Kunde hat nur einen Kundenbetreuer, ein Kundenbetreuer kann jedoch mehrere Kunden betreuen.“

Hier besteht also eine 1:N-Beziehung zwischen Kunde (N) und Kundenbetreuer (1). Im Beziehungsfenster wird das Schlüsselfeld **PersNr** aus der Tabelle tblKunden auf das Schlüsselfeld **PersNr** der Tabelle tblBetreuer gezogen. Die Ziehrichtung ist hierbei unwichtig, da automatisch die Tabelle zur „1“-Tabelle wird, die den Primärschlüssel besitzt.

Die zweite Prämisse lautete: „Jedem Kunden soll ein Vertriebsweg zugeordnet werden, mehrere Kunden können demselben Vertriebsweg angehören.“. Hier besteht also eine 1:N-Beziehung zwischen Kunde (N) und Vertriebsweg (1).

Welche Beziehung besteht zwischen den Tabellen Umsatz und Kunden? Ein Kunde kann mehrere Datensätze aus der Tabelle Umsatz zugeordnet haben (z.B. Jan98, Feb98 usw.). Ein Datensatz aus der Tabelle Umsatz kann jedoch immer nur einem Kunden zugeordnet sein. Auch hier haben wir also wieder eine 1:N-Beziehung.

Die 1:N-Beziehung ist die Häufigste, daneben gibt es aber noch die 1:1-Beziehung. Damit wir eine solche 1:1-Beziehung hier vorstellen können, haben wir die Tabelle Gehalt aufgenommen: Jedem Betreuer ist genau ein Gehalt zugeordnet. Strenggenommen hätten wir die Tabelle Gehalt nicht ausgliedern müssen. 1:1-Beziehungen lassen sich normalerweise problemlos in die verknüpfte Tabelle integrieren, es würde nicht gegen die Regeln des Normalisierungsprozesses verstoßen. Gelegentlich macht es trotzdem Sinn, die Tabellen zu trennen, wenn man beispielsweise unterschiedliche Zugriffsrechte vergeben möchte, oder eine umfangreiche Tabelle teilen möchte.

Das Ergebnis der Verknüpfungen sehen wir in Abbildung 3.1.2-1: Beziehungsfenster mit verknüpften Tabellen.

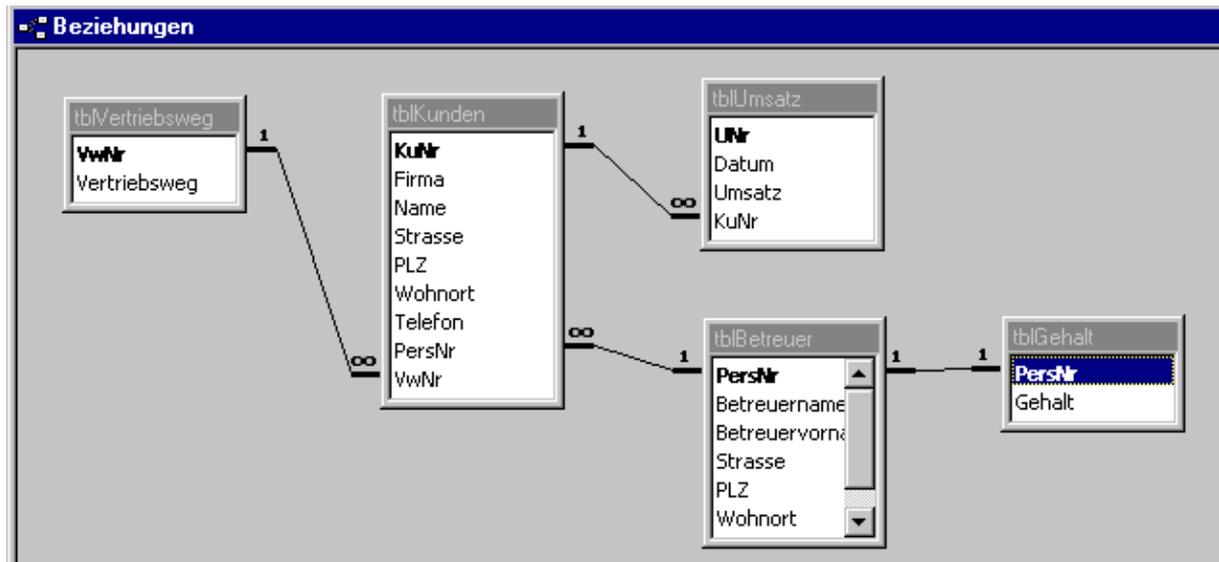


Abbildung 3.1.2-1: Beziehungsfenster mit verknüpften Tabellen

Die Symbole 1 und ∞ weisen daraufhin, daß die referentielle Integrität eingeschaltet wurde, also die Beziehungen zwischen den Tabellen automatisch überwacht werden.

Was bedeutet die referentielle Integrität?

Die Einhaltung der referentielle Integrität bedeutet, daß zwischen der sogenannten Mastertabelle („1“) und der Detailtabelle („N“) übereinstimmende Schlüssel vorhanden sein müssen. So soll z.B. einem Kunden (Detailtabelle, „N“) ein existierender Kundenbetreuer (Mastertabelle, „1“) zugewiesen werden.

- Sie können in das Fremdschlüsselfeld der Detailtabelle keinen Wert eingeben, der nicht im Primärschlüsselfeld der Mastertabelle enthalten ist<sup>11</sup>.
- Sie können keinen Datensatz aus der Mastertabelle **löschen**, wenn übereinstimmende Datensätze in einer Detailtabelle enthalten sind<sup>12</sup>. Beispielsweise können Sie einen Datensatz eines Betreuers aus der Tabelle Betreuer nicht löschen, wenn diesem Betreuer in der Tabelle Kunden Kunden zugeordnet sind. (Es sei denn, Sie haben die Option „Löschweitergabe an Detaildatensatz“ eingeschaltet.)
- Sie können keinen Primärschlüsselwert in der Mastertabelle **ändern**, wenn es zu diesem Datensatz Detaildatensätze gibt<sup>13</sup>. Beispielsweise können Sie die Personalnummer eines Betreuers in der Tabelle Betreuer nicht ändern, wenn diesem Betreuer in der Tabelle Kunden Kunden zugeordnet sind. (Es sei denn, Sie haben die Option „Aktualisierungweitergabe an Detailfeld“ eingeschaltet.)

Betrachten wir uns die Beziehung zwischen der Tabelle Kunden und der Tabelle Betreuer in Abbildung 3.1.2-2.: referentielle Integrität zwischen Kunden und Betreuer. Die Option „Aktualisierungweitergabe an Detailfeld“ ist eingeschaltet, damit bei einer Änderung der Personalnummer eines Betreuers (aus welchem Grund auch immer) die Personalnummer im Fremdschlüssel der Tabelle Kunden ebenfalls geändert wird und dadurch auch die Verknüpfung erhalten bleibt.

In Abbildung 3.1.2-3: referentielle Integrität zwischen Betreuer und Gehalt ist zusätzlich die Option „Löschweitergabe an Detaildatensatz“ eingeschaltet<sup>14</sup>. Dadurch wird bewirkt, daß z.B. das Ausscheiden eines Betreuers aus dem Unternehmen gleichzeitig ein Löschen seines Datensatzes aus der Tabelle Gehalt erfolgt.

<sup>11</sup> Sie können jedoch in das Fremdschlüsselfeld einen Nullwert eingeben und damit angeben, daß die Datensätze nicht miteinander verknüpft sind. So ist es z.B. nicht möglich, einen Kunden zu haben, der einem nichtvorhandenen Betreuer zugeordnet ist. Sie können jedoch einen Kunden haben, der niemandem zugeordnet ist, indem Sie in das Feld **PersNr** einen Nullwert eingeben.

<sup>12</sup> Detaildatensätze können Sie jederzeit löschen.

<sup>13</sup> Einen Fremdschlüsselwert in der Detailtabelle können Sie jederzeit ändern, um den Detaildatensatz z.B. einem anderen Datensatz in der Mastertabelle zuzuordnen.

<sup>14</sup> Die „Ziehrichtung“ von der Tabelle Betreuer zur Tabelle Gehalt beim Verknüpfen ist in diesem Fall

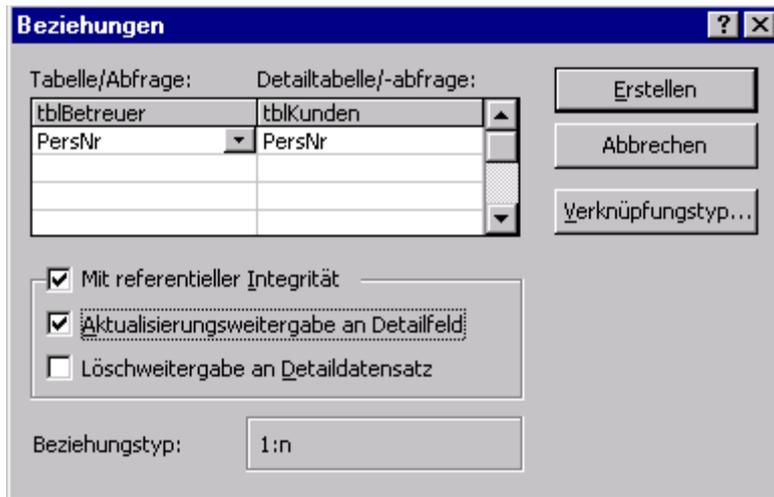


Abbildung 3.1.2-2.: referentielle Integrität zwischen Kunden und Betreuer

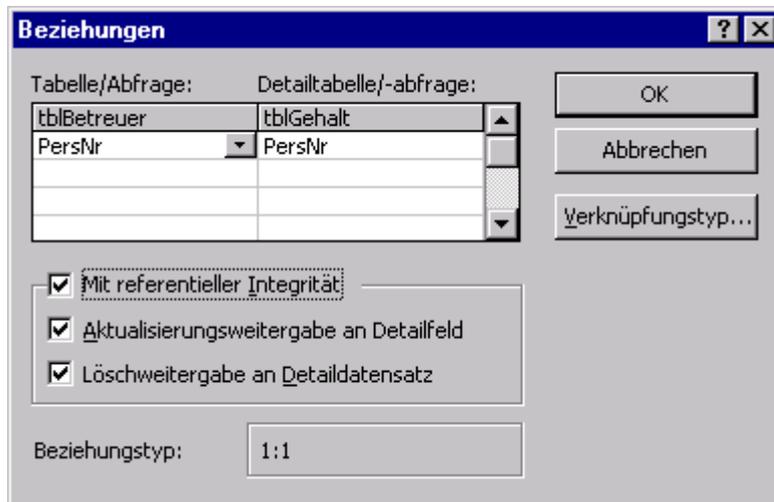


Abbildung 3.1.2-3: referentielle Integrität zwischen Betreuer und Gehalt

Das Ergebnis der bisherigen Schritte finden Sie in der Datenbank Kundenverwaltung 1.mdb.