

Skript zur Lehrveranstaltung EDV I/II

Datenbanktheorie

Ein Grundlagentext für Studenten

Version 1.0.7 f

Eitel v. Maur
1994

INHALT

VORWORT	1
1. EINLEITUNG	1
2. VON DER SEPARATEN DATEIVERWALTUNG ZUM DATENBANKVERWALTUNGSSYSTEM	2
2.1. ALLGEMEINES.....	2
2.2. DATEIVERWALTUNG	3
2.2.1. Redundanz	3
2.2.2. Inkonsistenz	3
2.2.3. Gemeinsame Datenbasis.....	4
2.2.4. Physische Datenabhängigkeit.....	4
2.2.5. Logische Datenabhängigkeit.....	4
2.3. DATEIVERWALTUNGSSYSTEM	5
2.4. DATENBANKVERWALTUNGSSYSTEM.....	6
2.5. DAS RELATIONALE DATENBANKMODELL.....	7
3. DATENBANKENTWURF.....	8
3.1. DATENMODELLIERUNG.....	8
3.2. NORMALISIERUNG.....	9
3.2.1. Datensammlung.....	9
3.2.2. Primärschlüssel	10
3.2.3. 1. Normalform	10
3.2.4. 2. Normalform	12
3.2.5. 3. Normalform	12
3.2.6. Zusammenfassung.....	16
3.2.7. Weitere Normalformen.....	16
3.3. DAS ENTITY-RELATIONSHIP-MODELL	16
3.4. INTEGRITÄTSANFORDERUNGEN AN EINE DATENBANK	21
3.4.1. Referenzintegrität	21
3.4.2. Nachschlagetabelle.....	21
3.4.3. Indizes	22
3.4.4. Gültigkeitsprüfungen	22
3.4.5. Paßwortschutz	23
LITERATUR.....	23

VORWORT

Die vorliegende Arbeit erhebt nicht den Anspruch, hochwissenschaftlich sein zu wollen. Sie ist nicht geschrieben worden, um einen akademischen Grad erlangen zu helfen. Das vorliegende Werk wurde für Studenten geschrieben. Es soll den Lehrstoff bewußt einfach vermitteln. Dies ist leider nicht immer gelungen. Doch hat der Autor mehr Wert auf eine verständliche Vermittlung des Stoffs gelegt, als auf 110% ige Korrektheit der Definitionen. Für Finden der Fehler gibt es keine Belohnung, außer der innerlichen Freude. Die Nachfolger werden jedenfalls dankbar sein. Besonders für Tips zur weiteren Verbesserung der Verständlichkeit ist der Autor dankbar. Bei einem so komplexen Gebiet wie der Datenbanktheorie kann man keine ideale sequentielle Abfolge der Einzelthemen festlegen. Das beste Mittel ist einfach - am Ende angelangt, noch mal von vorne lesen.

1. EINLEITUNG

Durch die enormen Fortschritte, die in der Informationstechnologie erzielt worden sind, und durch die rapide Ausbreitung dieses Bereiches, wurde längst erkannt, welchen Stellenwert der Produktionsfaktor Information in unserer heutigen Gesellschaft hat. So bildet heute in fast jedem bedeutenden Unternehmen der Produktion, des Handels oder des Dienstleistungsreiches eine Datenbank die Basis der Informationsflüsse. Produktions-, Planungs- und Steuerungssysteme (PPS) bzw. Warenwirtschaftssysteme (WWS) werden solche umfangreichen Softwareprogramme genannt, die mehr und mehr Funktionen des Unternehmens übernehmen. Durch diese stetige Substitution von menschlicher Arbeit hin zu computergestützter Arbeit, ist der Anteil des Rechners an der Ablauforganisation ein immer bedeutenderer. Die Grundlage eines solchen Unternehmenssystems bildet jedoch das Datenmodell des Unternehmens, das in einem Datenbankverwaltungssystem (DBVS) implementiert wird. Nur wenn Datenbankmodell und Datenbankverwaltungssystem solide sind, ist die darauf aufgesetzte Computertechnologie, damit das gesamte System und damit der langfristige Bestand des entsprechenden Unternehmens, gewährleistet. Trotzdem wird in der Praxis diese Tatsache noch all zu oft ignoriert.

Nachdem im Unterricht bisher ausführlich auf Algorithmen eingegangen wurde, folgt nun der zweite große Teil: die Datenorganisation. Oft wird in der Literatur eine Trennung von Algorithmen und Daten angenommen, obwohl dies praktisch nicht möglich ist. Sinnvollerweise werden bei einem Algorithmus Daten verarbeitet. Werden Daten verarbeitet, so geschieht dies durch Algorithmen.

Mit der Programmiersprache, die Du bisher kennengelernt hast (Pascal bzw. 4GL), konntest Du dem Rechner komfortabel Algorithmen beibringen. Die Verarbeitung von Daten anhand z. B. der sequentiellen Dateiorganisation war kaum in der Lage, Freude über gelungene Ergebnisse zu erzielen. Mit entsprechenden Datenbankverwaltungssystemen ist dies anders. Diese sind zu Leistungen in der Lage, die selbst den erfahrenen Anwender immer wieder in Entzücken versetzen können, sofern es um die Verarbeitung von Daten geht. Sind also die Datenbankverwaltungssysteme die Lösung aller Dinge? Nein! Mit diesen lassen sich nur einfache Algorithmen abbilden. Erst die Kombination von Datenbankverwaltungssystem und Programmiersprache (z. B. Pascal oder 4GL) eröffnet die gewünschten Möglichkeiten. Bei Informix/4GL ist dies schon integriert. Bei Pascal wurde von uns versucht, dies durch eine Schnittstelle mit dem Namen "psql" zu lösen. Diese Schnittstelle verbindet Pascal (Algorithmen) und Informix (Datenbankverwaltungssystem).

Doch halt halt! Vor der endgültigen Lösung kommt erst die Datenbanktheorie. Der Umgang mit Algorithmen war schwer. Der Umgang mit Datenbanken ist nicht leichter. Doch keinen Schrecken: Es macht Spaß und bietet einiges Kniffliges!

2. VON DER SEPARATEN DATEIVERWALTUNG ZUM DATENBANKVERWALTUNGSSYSTEM

2.1. ALLGEMEINES

Um auf den heutigen Stand der Möglichkeiten bei der Datenverarbeitung zu kommen, waren vor den Lösungen die Probleme notwendig. Erst wenn Bewußtsein über das Problem besteht, kann dafür eine Lösung gefunden werden. Deshalb ist es auch notwendig, die Schritte, die zum heutigen Standard geführt haben, nachvollziehen zu können. Nur dann kannst Du

verstehen, welche Vorteile Datenbankverwaltungssystem bieten und warum diese so wichtig sind.

Anfänglich wurde die Datenorganisation stiefmütterlich behandelt. Wie im Unterricht stand zuerst der Algorithmus im Vordergrund. Die Entwicklung der Datenorganisation begann bei der separaten Dateiverwaltung. Dort entwickelten sich nach und nach die vier Grundformen der Dateiorganisation.

1. sequentielle Dateiorganisation
2. gestreut-direkte Dateiorganisation
3. gestreut-indirekte Dateiorganisation
4. index-sequentielle Dateiorganisation

Diese Dateiorganisationsformen solltest Du schon kennen.¹ Wie Du Dich erinnern wirst, war für diese Entwicklung vor allem die Entwicklung von direkt adressierbaren Speichermedien verantwortlich. Nach den stetig steigenden Anforderungen an die Dateiverwaltung wurden Dateiverwaltungssysteme entwickelt. Auch deren Leistungen konnten den Ansprüchen der Benutzer nicht lange genügen. Die ersten Datenbankverwaltungssysteme entstanden. Erst das hierarchische, dann das netzartige und schließlich das relationale Datenbankverwaltungssystem.² Von dem letzteren wollen wir Dir einiges erzählen, doch nun erst die Anfänge.

2.2. DATEIVERWALTUNG

Dateiverwaltung ist inzwischen zur Genüge geübt worden! Im Programmcode (Pascal / 4GL) werden RECORDs und Dateien vereinbart. Die RECORDs können dann satzweise in die entsprechenden Dateien ein- oder ausgelesen werden. Soll in die Datei geschrieben werden, so muß diese vorher zum Schreiben geöffnet worden sein. Willst Du einen Satz aus der

¹siehe Jancke / Jeiter 89

²Inzwischen gibt es auch eine erneute Entwicklung, die objektorientierten Datenbanken. Dies würde hier jedoch viel zu weit führen. In ein paar Jahren wird diese Entwicklung mehr reden von sich machen.

Datei lesen, so mußt Du diese vorher zum Lesen öffnen. Mehr kann man mit Dateien nicht machen.

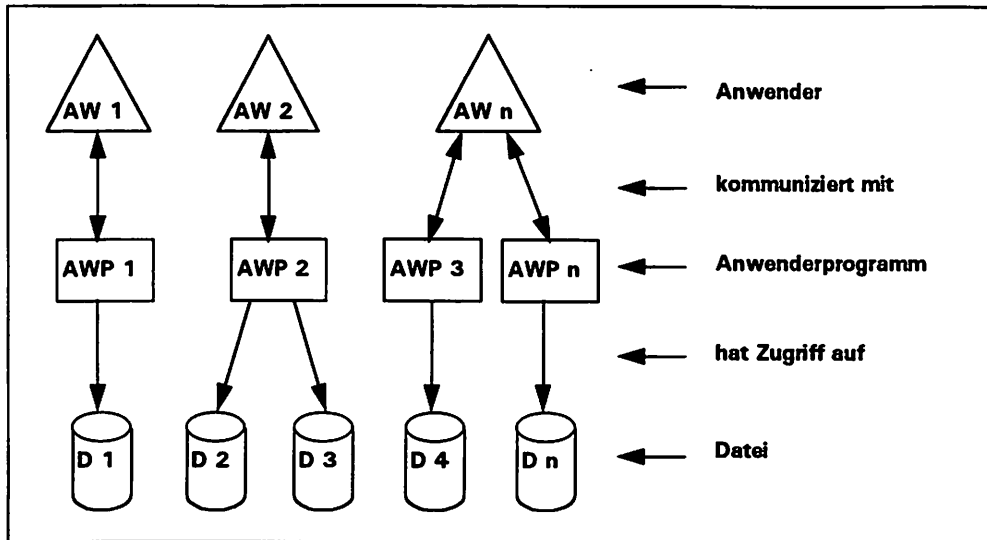


Abbildung 1: separate Dateiverwaltung

Immerhin, die Vorteile von Dateien sind schon enorm:

1. dauerhafte Speicherung von Daten (auf externen Speicher)
 - ⇒ Datensicherheit
 - ⇒ Übertragbarkeit der Daten
2. unbegrenzte Speicherkapazität (nur durch das Speichermedium)
3. Kostenersparnis durch Nutzung externer Speichermedien

2.2.1. REDUNDANZ

Für die Anforderungen von komplexen Anwendungen reicht dies allerdings nicht aus. Eine Datei ist immer nur für eine ganz bestimmte Anwendung (Programm) geschrieben. Dadurch werden Daten, die in verschiedenen Anwendungen benötigt werden, immer wieder erneut gespeichert. Beispielsweise mußt Du Deine Daten einmal beim Immatrikulationsamt abgeben. Diese werden dann in das Studentenverwaltungsprogramm eingegeben. Doch auch das Prüfungsamt will Deine Daten haben, noch mal. Die

geben Deine Daten in das Studentennotenprogramm ein. Natürlich sind in beiden Dateien die Daten nicht identisch, da beide Ämter verschiedene Interessen an Dir haben. Sie bilden, wie unten ausführlicher erklärt, verschiedene Modelle von Dir. Doch die vielen Daten, die doppelt gespeichert werden müssen, sind das Problem. Diese Daten sind überflüssig, sie sind redundant!

Definition Redundanz

Redundanz ergibt sich durch mehrfache Speicherung derselben Daten.

2.2.2. INKONSISTENZ

Diese Daten sind aber nicht nur überflüssig. Sie sind schädlich! Was passiert, wenn Du Deine Adressänderung bei der Rückmeldung angibst? Die Änderung wird vom Immatrikulationsamt erfaßt. Das Prüfungsamt? Das bekommt davon nichts mit! Schade!³ Somit entstehen widersprüchliche Daten. Diese Daten sind dann inkonsistent! Niemand weiß, welche Adresse die richtige ist.

Definition Inkonsistenz

Inkonsistenz ist eine mögliche Folge von Redundanz. Inkonsistenz besteht, wenn die redundant gespeicherten Daten verschiedene Werte enthalten.

2.2.3. GEMEINSAME DATENBASIS

Doch nicht genug davon! Nicht nur, daß eine Datei immer nur für eine bestimmte Anwendung angelegt werden muß. Jeder Anwender, der dieses Programm benutzt, produziert seine eigenen Dateien. Sachbearbeiter Kolb sitzt an einem anderen Computer als Sachbearbeiter Heifrau. Beide

³Die in diesem Skript geäußerten Mängel der universitären Verwaltung sind frei erfunden und entsprechen in keinsten Weise der Realität. Es handelt sich lediglich um Beispiele zur didaktischen Veranschaulichung des Lehrstoffes.

haben das Studentennotenprogramm auf ihrem Rechner. Und so erzeugt jeder seine eigene Datei. Redundanz und Inkonsistenz sind dann vorprogrammiert. Keine gute Lösung! Im Zeitalter der Vernetzung sollte so etwas nicht vorkommen.

2.2.4. PHYSISCHE DATENABHÄNGIGKEIT

Was passiert, wenn sich die Modernisierung auch in der Verwaltung durchsetzt? Durch die Einführung von Festplatten ergibt sich die Möglichkeit, von einer sequentiellen Dateiorganisation auf eine gestreut-direkte Dateiorganisation umzusteigen. Alle Algorithmen der Anwendungsprogramme sind dann falsch. Sie müssen alle angepaßt oder sogar neu geschrieben werden. Denn die physische Dateiorganisation ist ganz entscheidend für den Algorithmus der Software. Dies nennt man **physische Datenabhängigkeit**.

Gewünscht ist also eine Datenorganisation, die Redundanzfreiheit und Konsistenz gewährleistet. Jedes Datum soll nur einmal gespeichert sein. Ist dies so, können widersprüchliche Daten nicht entstehen. Diese Redundanzfreiheit soll auch über verschiedene Computer hinweg gewährleistet sein. Dies setzt natürlich einen Netzverbund der Computer voraus. Denn nur dann können die zentral gelagerten Daten zu den verschiedenen Rechnern (Clients) geschickt werden. Außerdem sollen natürlich nicht nur auf den verschiedenen Computern die Daten nur einmal vorhanden sein. Nur wenn alle Anwendungsprogramme auf dieselbe Datenbasis zugreifen, gibt es wirklich Redundanzfreiheit.

2.2.5. LOGISCHE DATENABHÄNGIGKEIT

Noch mehr Schwächen! Stell Dir vor, Kobolt und Heifrau haben es geschafft. Sie arbeiten mit zwei Rechnern, doch alle ihre Anwendungen arbeiten mit nur einer gemeinsamen Datei. Außerdem haben sie sich mit dem Immatrikulationsamt zusammengetan. Die Datenbestände sind nicht mehr doppelt. Das Notenprogramm und das Verwaltungsprogramm greifen auf die selben Daten zu. Stell Dir nun noch vor, die Prüfungsordnung ändert sich. Es gibt statt zehn plötzlich zwanzig Noten, um die sich das Prüfungsamt zu kümmern hat. Das Datenmodell Deiner Person ändert sich.

Die Datenstruktur muß angepaßt werden. Aber nicht nur die Dateien sind zu ändern. Alle Programme, die mit den Dateien in Verbindung stehen, müssen geändert werden. Alle RECORDs müssen angepaßt werden. Also nicht nur das Notenprogramm des Prüfungsamtes. Auch beim Verwaltungsprogramm des Immatrikulationsamtes müssen alle RECORDs geändert werden. Dies nennt man **logische Datenabhängigkeit**.

Nicht zu vergessen ist auch, daß es sich bei den Daten im Prüfungsamt um sehr sensible Daten handelt. Zu leicht könnte jemand auf die Idee kommen, für ein wenig Sponsoring ein paar Daten zu verändern. Nach oben oder unten. Kommt eben auf den Sponsor an. Es ergeben sich also auch im Datenschutz umfangreiche Gefahren. Und die können bei der separaten Dateiverwaltung nur schwer behoben werden.

Jetzt noch mal zusammengefaßt. Die größten Probleme der Dateiverwaltung sind:

1. physische Datenabhängigkeit
2. logische Datenabhängigkeit
3. Zugriffspfadunabhängigkeit
4. Redundanz
5. Inkonsistenz
6. Datenschutz

Dafür gibt es inzwischen Lösungen!

Fragen:

Welche Nachteile hat die separate Dateiverwaltung? Welche Konsequenzen hat dies in der Praxis? Ist das wirklich wichtig?

Antworten:

Die Redundanz verursacht einen enormen Zusatzeufwand, der mit dem heutigen Stand der Technik unvereinbar ist. Bei einer unternehmensweit eingesetzten separaten Dateiverwaltung für ein mittleres oder gar großes Unternehmen ist die Überlegung angebracht, ob man besser keine EDV einsetzt als eine solche Lösung zu wählen.

Die Redundanz führt zwangsläufig zu einer Inkonsistenz, die ein solches Chaos in einem Unternehmen auslösen kann, so daß dem Unternehmen dadurch die Existenzgrundlage entzogen werden kann.

2.3. DATEIVERWALTUNGSSYSTEM

Die Entwicklung von Dateiverwaltungssystemen führte zu Zugriffsfunktionen auf die verschiedenen Dateien. Diese Zugriffsfunktionen sind nicht in den jeweiligen Anwendungsprogrammen implementiert, sondern eigene Programme. Sie werden von den jeweiligen Anwendungen benutzt. Benutzt Du ein Dateiverwaltungssystem, so greift die Anwendung nicht mehr direkt auf die benutzte Datei zu. Stattdessen benutzt die Anwendung die von dem Dateiverwaltungssystem zur Verfügung gestellten Funktionen. Die Funktionen greifen auf die Dateien zu. Die Daten werden von den Funktionen an die Anwendungen geliefert. {Noch mal lesen!}

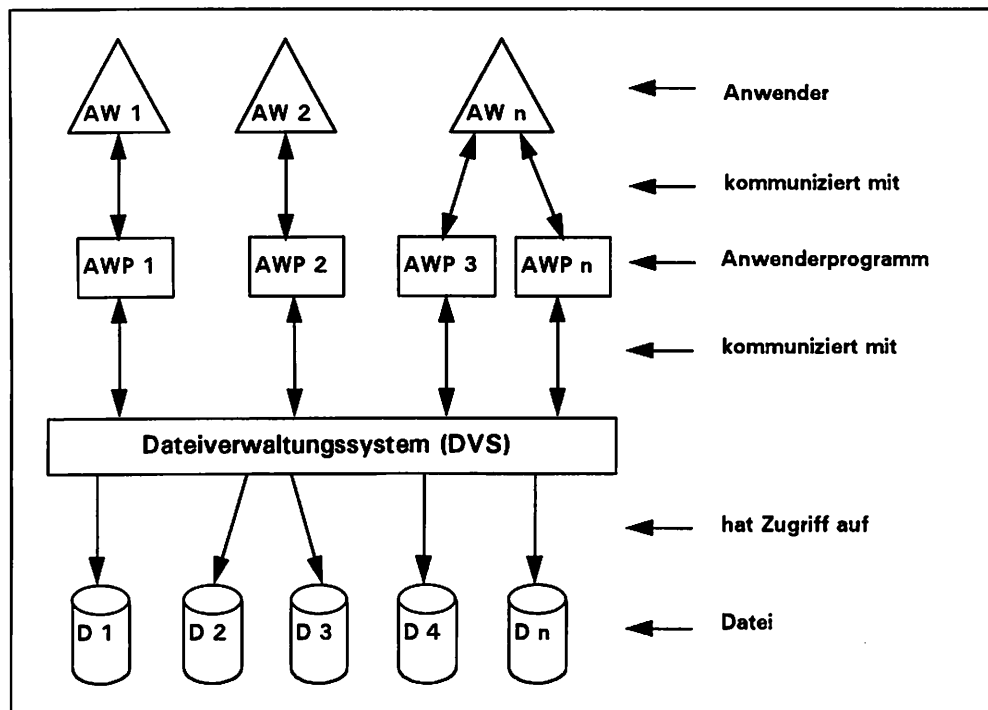


Abbildung 2: Dateiverwaltungssystem

Beispiele für Zugriffsfunktionen sind:

- FUNCTION DatenSatzLesen (Parameter)
- FUNCTION DatenSatzSchreiben (Parameter)
- FUNCTION DatenSatzSuchen (Parameter)
- FUNCTION DatenSatzÄndern (Parameter)
- FUNCTION DateiÖffnen (Parameter)
- FUNCTION DateiSchließen (Parameter)

Wo ist jetzt der Vorteil gegenüber der separaten Dateiverwaltung? Für die Anwendungen ist es jetzt egal, welche Dateiorganisationsform gewählt wurde. Wird die Dateiorganisationsform geändert, muß nur noch ein Programm angepaßt werden, das Dateiverwaltungssystem. Alle Anwendungsprogramme bleiben unverändert.

Physikalische Änderungen der Datenorganisation können nun durchgeführt werden. Die Anwendungsprogramme sind davon nicht betroffen. Jedoch kontrollieren die Programme immer noch die Struktur der Daten. Eine Änderung der Datenstruktur verlangt eine Änderung aller auf die Datei zugreifenden Anwenderprogramme.

Vom Dateiverwaltungssystem wird der Zugriff auf die Daten übernommen. Auf den Aufbau der Daten hat dieses jedoch keinen Einfluß. Deshalb können verschiedene Anwenderprogramme immer noch nicht wirklich mit den selben Dateien arbeiten. Nur wenn die RECORDs identisch sind, ist dies möglich. Und dies führt zur oben erläuterten Problematik. Doch was gehen das Immatikulationsamt Deine Noten an? Und was sollten die auch jedes mal mit all den Daten? Ganz zu schweigen vom Datenschutz!

Viele Fortschritte bringt ein Dateiverwaltungssystem mithin nicht! Redundanz und Inkonsistenz wird dadurch nicht beseitigt. Der praktische Sinn eines Dateiverwaltungssystems ist damit fraglich. Ein verbreitetes, etwas besseres Dateiverwaltungssystem ist dBase, im Moment im Eigentum der Firma Borland.⁴

⁴Fairerweise muß man zugeben, daß auch dieses Produkt Fortschritte gemacht hat.

2.4. DATENBANKVERWALTUNGSSYSTEM

Die restlich anstehenden Probleme verspricht das Datenbankverwaltungssystem⁵ (DBVS) zu lösen. Dieses entstand in den 60'er Jahren. Im wesentlichen gibt es dabei drei verschiedene Datenbankmodelle.

1. Hierarchisches Datenmodell
2. Netzartiges Datenmodell
3. Relationales Datenmodell

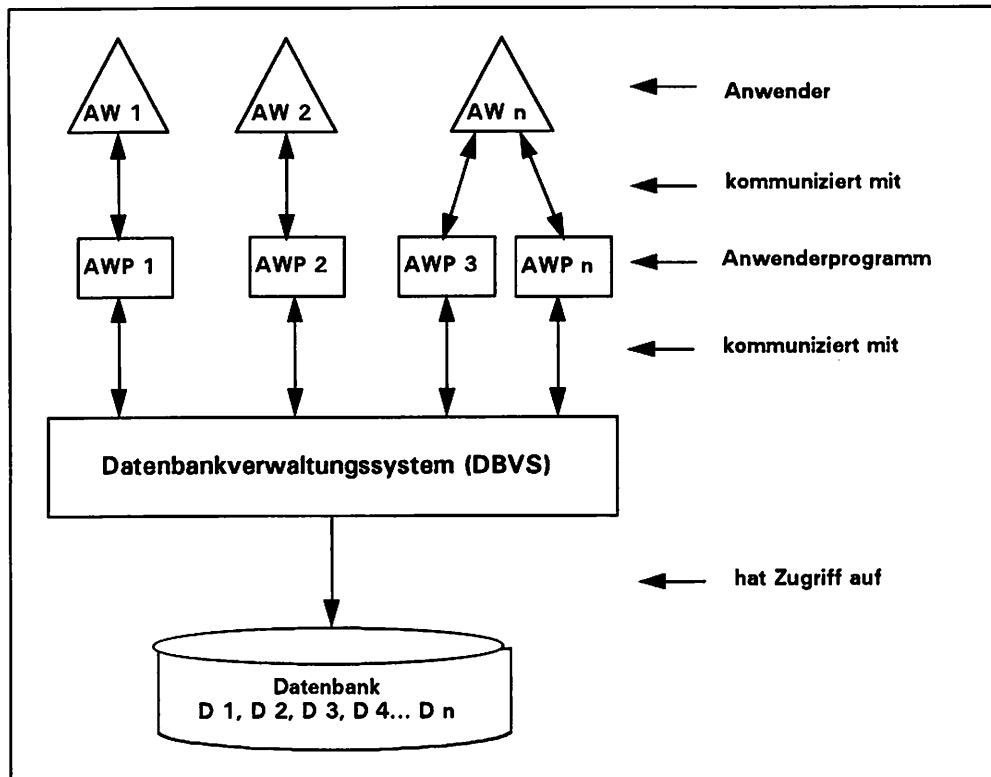


Abbildung 3: Datenbankverwaltungssystem

Die ersten beiden spielen heute kaum noch eine Rolle. Zu erwähnen ist, daß Daimler Benz und die Deutsche Bank eine Datenbank betreiben, dem ein hierarchisches Datenmodell zugrunde liegt. Beide Firmen besitzen das erste kommerzielle DBVS. Es stammt von IBM, wurde Ende der 60'er Jahre entwickelt und heißt IMS (Information Management System). Ein Umstieg scheint aus Kostengründen unmöglich.⁶ Doch auch die Tatsache, daß jeder Fehler dieser Datenbank bekannt und dokumentiert ist, trägt dazu bei. Dies gewährleistet eine enorme Betriebssicherheit. Die Deutsche Bank ist sich der Bedeutung ihrer Datenbank bewußt. Ein Ausfall dieser Datenbank würde die Deutsche Bank schon nach wenigen Tagen zur Anmeldung des Konkurses zwingen!!!

Das relationale Datenmodell stammt von E. F. Codd. Er hat es Anfang der 70'er Jahre entwickelt. Durch seine Überlegenheit den anderen Datenmodellen gegenüber hat es sich schnell durchgesetzt. Dieses Datenmodell bildet die Grundlage der meisten kommerziell genutzten Datenbanken. Doch was ist ein Datenmodell?

Definition Datenmodell

Zur formalen Beschreibung aller in der Datenbank enthaltenen Daten und ihrer Beziehungen untereinander verwendet man Datenmodelle.

2.5. DAS RELATIONALE DATENBANKMODELL

Das relationale Datenmodell verwendet zur Darstellung von Daten Tabellen. Objekte und Beziehungen werden jeweils durch Tabellen dargestellt.

⁵Oft auch Datenbankmanagementsystem (DBMS) genannt.

⁶Selbst für diese beiden!

Relation $\approx$ Tabelle $\approx$ Datei $\approx$ modellierter Objekttyp $\approx$ Entitytyp

Kunde	KundenNr	Nachname	Vorname	Telefon	PLZ
	1234	Maier	Karl	741 25 15	12431
	1238	Moustache	Ludwig	478 12 45	25478
	1345	Knecht	Hans	478 12 45	25478
	1458	Koller	Susanne	321 54 78	84710

Datensatz $\approx$ Tupel $\approx$ Tabellenzeile $\approx$ Element $\approx$ Entity

1234	Maier	Karl	741 25 15	12431
------	-------	------	-----------	-------

Anzahl der Attribute = Grad

	1	2	3	4	5
$\Rightarrow 5$	1234	Maier	Karl	741 25 15	12431

Anzahl der Datensätze = Mächtigkeit

1	1234	Maier	Karl	741 25 15	12431
2	1238	Moustache	Ludwig	478 12 45	25478
3	1345	Knecht	Hans	478 12 45	25478
4 $\Rightarrow 4$	1458	Koller	Susanne	321 54 78	84710

Attribut $\approx$ Spaltentitel $\approx$ Feldname $\approx$ Merkmal

Nachname

Jedes Attribut hat einen Datentyp = Attributtyp

KundenNr	INTEGER
Nachname	STRING
Vorname	STRING
Telefon	STRING
PLZ	INTEGER

Domäne =

Zusammenfassung aller möglicher Attributwerte eines Attributs

PLZ
12431
25478
25478
84710

$\Rightarrow$

Domäne PLZ {INTEGER}

$(PLZ > 0) \wedge (PLZ \leq 99999)$

3. DATENBANKENTWURF

Der erste Schritt beim Anlegen einer Datenbank ist immer gründliche Planungsarbeit. Auch dies wird allzuoft vergessen. Als erstes müssen die genauen Anforderungen bekannt sein. Für die jeweilige Anwendung muß ermittelt werden, welche Daten benötigt werden. Es entsteht ein Datenmodell. Dieses Datenmodell wird in eine der verwendeten Datenbank entsprechende Form gebracht. Meistens entsteht mithin ein relationales Datenmodell. Dies bedeutet, daß die für die Anwendung gesammelten Daten in eine spezielle Form gebracht werden. Beim relationalen Datenmodell entstehen lauter Tabellen⁷. Ist die Redundanz der Tabellen beseitigt, wird die Referenzintegrität hergestellt, Nachschlagetabellen angelegt und Sekundärindizes festgelegt. Es werden Gültigkeitsprüfungen für die einzelnen Daten erzeugt und verschiedene Ebenen des Paßwortschutzes erstellt. Erst wenn alle diese Arbeiten auf Papier entstanden sind, kann daran gedacht werden, die Schritte in die Praxis umzusetzen. Hältst Du diese Reihenfolge nicht ein, wirst Du dafür "teuer bezahlen" müssen. Ist der gesamte Entwurf der Datenbank in das Datenbankverwaltungssystem implementiert worden, können die eigentlichen Daten eingegeben werden. Anschließend⁸ entstehen Masken, Listen (Reports), Abfragen und auf den Datenbestand zugreifende komplexere Algorithmen.

3.1. DATENMODELLIERUNG

Von Modellen und Modellierung wirst Du noch genug hören. Die Systemanalyse wird es bringen! Deshalb soll an dieser Stelle nur ein kurzer Einstieg erfolgen.

Fragst Du zwei Freunde, sie mögen Dich beschreiben. Der eine findet Dich dick, Deinen Schatz echt schnuckelig und meint, Du rauchst zu viel. Den anderen interessiert nur, daß Du was von Joop! und Versace trägst. Dem

⁷Wobei anzumerken ist, daß diese Tabellen nicht nur willkürlich zusammengestellte Tabellen seien, sondern sich in der 3. Normalform befinden müssen. Normalisierung wird weiter unten ausführlich behandelt.

⁸Dieser Vorgang kann u. U. schon parallel beginnen.

nicht genug interessiert sich Dein zukünftiger Arbeitgeber nur für die EDV- und anderen Noten. Deine Mutter möchte diese auch erfahren, bekundet Neugierde gegenüber Deines Partners, doch das reicht Ihr an Informationen noch lange nicht. Keiner von ihnen ist in der Lage, Dich in Deiner Ganzheit zu erfassen. Alle können sie nur Merkmale erkennen, an denen sie meinen, Dich zu erkennen. Anhand dieser Merkmale bilden sie vereinfachte Abbildungen von Dir. Diese Abbildungen nennt man Modelle. Deine Freunde haben Dich also unterschiedlich modelliert. Jeder nach seinen Bedürfnissen. Während das Datenmodell von Deinen Freunden kurz und einfach ist, das von Deiner Mutter ist schon um einiges komplexer.

Bei der Datenmodellierung kommt es mithin darauf an, welche Daten für die jeweilige Anwendung interessant sind. Bei einer Datenbank müssen alle Daten aller Anwendungen zusammengetragen werden. So muß sich das Immatrikulationsamt, genauso wie das Prüfungsamt, überlegen, welche Daten es von Dir benötigt. Beide Datenmodelle werden dann für eine gemeinsame Datenbank zusammengefaßt, so daß ein gemeinsames Datenmodell entsteht.

Im folgenden zeigen wir Dir die Vorgehensweise an einem einfachen Beispiel. Dies reicht natürlich mit nichts für die auf Dich zukommende Praxis aus, liefert aber die Möglichkeit, schnell und bequem nachvollzogen werden zu können.

Wir zeigen Dir, wie man eine einfache Adressverwaltung für zu Hause entwerfen könnte. Als erstes haben wir uns die Daten überlegt, die uns von unseren Freunden, Verwandten und Geschäftspartnern interessieren. Sie sind in der folgenden Abbildung 4 der Reihe nach aufgeführt.

Attribut	Beispiel	Abkürzung
Anrede	Herr	Anr
Titel	cand. rer. pol.	Titel
Vorname	Karl	VName
Nachname	Maier	Name
Namenszusatz	von	Zus
Priorität	7	Pri
Kategorie	Freund	Kateg

Straße / Hausnummer	Lärchenstr. 17	Straße
PLZ	15427	PLZ
Ort	Berlin	Ort
Land	Deutschland	Land
Notizen	Matr.: 114.541, Kto.: 147821, Liebt gegrillten Fisch!	Notizen
Geburtsdatum	19. 5. 1923	Geburtsdat
Spitzname	Specht	SName
Telefon	030-314-7 34 82	Tel_Nr
Fax	0711-28 00-181	s. u.

Abbildung 4: Attribute für die Adressverwaltung mit Beispielausprägungen

3.2. NORMALISIERUNG

3.2.1. DATENSAMMLUNG

Nach der Attributesammlung haben wir uns dafür ein paar Beispieldaten überlegt, sie sind im folgenden aufgeführt.

1000	Herr Prof. Karl Kahle "Glatze" 19.05.1923 71 Jahre gesch. 1
	Büro: TU Berlin Str. d. 17 Juni 11452 Berlin Deutschland
	T 314-0
	T 314-2 24 54
	T 314-7 34 82
	F 314-7 34 95
	privat: Familie Kahle Plattstr. 21 14525 Berlin Deutschland
	T 321 45 74
	F 478 41 41
	Geliebte: Gerda Hui Lustweg 6 24513 Hannover Deutschland
	T 321 21 21
	T 412 41 78

1001	Frau Dr. Susi zu Stulle "Mausi" 18.03.1974 20 Jahre Freund 7 "guter Typ"
	privat: Susi zu Stulle Am Baguette 2 10212 Berlin Deutschland
	T 321 54 78
	T 874 54 21
	Freund: Heiko Hensel Tomi-Weg 52 14235 Berlin Deutschland
	T 784 12 12
	Eltern: Familie zu Stulle Brotgasse 1 15423 Berlin Deutschland
	F 447 78 78
	T 447 78 79
	T 447 78 80
1003	Frau cand. ing. Moni Mauer 08.10.1970 24 Jahre verw. 10 "klasse"
	Freund: Axel Rocker Kohlenweg 11 12457 Berlin Deutschland
	T 357 15 98
	Büro: IKEA Ivarstiege 21 a 22455 Hamburg Deutschland
	T 386-25 45
	T 386-78 36
	Eltern: Familie Nabel Am Loch 3 44125 Mölln Deutschland
	T 247 87 32
	Freund: Arndt Ampel Kreuzach Str. 4 84125 St. Joseph Deutschland
	T 774 75 11
1004	Herr Peter Panther "Quicki" 21.07.1955 39 Jahre gesch. 5 "reich!!!"
	Bruder: Axel Rocker Kohlenweg 11 12457 Berlin Deutschland
	T 357 15 98
	Büro: IKEA Ivarstiege 21 a 22455 Hamburg Deutschland
	T 386-25 45
	T 386-78 36
1005	Frau Dipl.-Kff. Vicki von Vierhuß 15.02.1968 26 Jahre verw. 2 "schrecklich"
	Büro: Das Buch Seitenstr. 2 12457 Berlin Deutschland
	T 654 73 21
	F 654 73 55
	privat: WG am Brunnen Am Loch 17 44125 Berlin Deutschland

Abbildung 5: Beispieldaten für die Adressverwaltung

Wie Du siehst, gibt es verschiedene Leute, die mehrere Adressen und Telefon- bzw. Faxnummern haben.

3.2.2. PRIMÄRSCHLÜSSEL

Eine elementar wichtiges Thema für den Datenbankentwurf ist der Primärschlüssel. Den Primärschlüssel kennst Du schon aus der Dateiorganisation. Dort erfüllt er die selbe Aufgabe - er identifiziert einen Datensatz eindeutig.

Definition Primärschlüssel

Der Primärschlüssel identifiziert einen Datensatz eindeutig. In einer Relation darf jeder Primärschlüssel höchstens einmal vorkommen. Ein Primärschlüssel kann auch aus mehr als einem Attribut zusammengesetzt sein, es handelt sich dann um einen kombinierten Primärschlüssel.

Willst Du Redundanzfreiheit in der Datenbank erreichen, so ist entscheidend, daß keine Daten mehrfach gespeichert werden. (Eben nicht redundant!) Alle Datensätze einer Datei bzw. Tabelle müssen sich also unterscheiden. Dem trägt der Primärschlüssel Rechnung. In einer Datenbank kann die selbe Ausprägung des Schlüssels gar nicht zweimal eingegeben werden. Das läßt ein Datenbankverwaltungssystem nicht zu.

Der Schlüssel kann aus bereits vorhandenen Attributen bestehen (z. B.: Name, PLZ) oder aus künstlichen, die der Tabelle angehängt werden. Dabei gibt es sprechende Schlüssel, wie z. B. eine Artikelnummer, die sich aus Kürzeln für das Produkt, den Hersteller und der Packungsgröße zusammensetzt. Und es gibt Schlüsse!, die einfach eine fortlaufende Nummer darstellen. Für den letzten Fall gibt es in einigen Datenbankverwaltungssystemen bereits vordefinierte Typen, die für eine automatische Numerierung sorgen. (Sehr praktisch!) Diese Variante wird von Praktikern auch als die in der Regel best geeignete empfohlen.⁹ Einen Schlüssel wie den Personennamen kann man nicht sinnvoll verwenden. Wieviele Müllers und Schmidts gibt es doch in Deutschland. Die PLZ oder die Telefonnummer dagegen kann durchaus mal als sinnvoller Schlüsselteil eingesetzt werden.

⁹siehe Finkenzeller 93, S. 25 ff

Vorsicht ist bei Datenbankverwaltungssystemen auch bei kombinierten Schlüsseln geboten. Mehr als zwei Schlüsselattribute pro Schlüssel sind nur in sehr seltenen Fällen anzuraten. Wie weiter unten deutlich werden wird, erschwert dies die Arbeit beim täglichen Umgang mit der Datenbank erheblich! Desweiteren besteht dadurch die Gefahr der Verletzung höherer Normalformen als der 3. Normalform¹⁰.

3.2.3. 1. NORMALFORM

Der nächste Schritt ist nun, die Daten in die 1. Normalform zu bringen. Dazu wird einfach eine Tabelle genommen, werden die Spalten mit den Attributen beschriftet und die Beispieldaten eingetragen.

Definition 1. Normalform

Eine Relation befindet sich in der 1. Normalform, wenn sie nur einfache Wertebereiche hat. Die 1. Normalform stellt lediglich sicher, daß es sich überhaupt um eine Tabelle handelt. Sie verlangt, daß Attributwerte Einzelwerte und keine Wertemengen sind.

¹⁰Diese Normalformen sind in einer Grundlagenveranstaltung überflüssig. Doch es gibt sie wirklich!

Nr	P_Nr	Anr	Titel	VName	Zus	Name	SName	Geburtsdat	Alt	Kateg	Pri	Notizen	Typ	Bewohner	Straße	PLZ	Ort	Land	K	Tel_Nr
1	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		Büro	TU Berlin	Str. d. 17 Juni	11452	Berlin	Deutschland	T	314-0
2	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		Büro	TU Berlin	Str. d. 17 Juni	11452	Berlin	Deutschland	T	314-2 24 54
3	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		Büro	TU Berlin	Str. d. 17 Juni	11452	Berlin	Deutschland	T	314-7 34 82
4	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		Büro	TU Berlin	Str. d. 17 Juni	11452	Berlin	Deutschland	F	314-7 34 95
5	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		privat	Familie Kahle	Plattstr. 21	14525	Berlin	Deutschland	T	321 45 74
6	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		privat	Familie Kahle	Plattstr. 21	14525	Berlin	Deutschland	F	478 41 41
7	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		Geliebte	Gerda Hui	Lustweg 6	24513	Hannover	Deutschland	T	321 21 21
8	1000	Herr	Prof.	Karl		Kahle	Glatze	19.05.1923	71	gesch.	1		Geliebte	Gerda Hui	Lustweg 6	24513	Hannover	Deutschland	T	412 41 78
9	1001	Frau	Dr.	Susi	zu	Stulle	Mausi	18.03.1974	20	Freund	7	guter Typ	privat	Susi zu Stulle	Am Baguette 2	10212	Berlin	Deutschland	T	321 54 78
10	1001	Frau	Dr.	Susi	zu	Stulle	Mausi	18.03.1974	20	Freund	7	guter Typ	privat	Susi zu Stulle	Am Baguette 2	10212	Berlin	Deutschland	T	874 54 21
11	1001	Frau	Dr.	Susi	zu	Stulle	Mausi	18.03.1974	20	Freund	7	guter Typ	Freund	Heiko Hensel	Tomi-Weg 52	14235	Berlin	Deutschland	T	784 12 12
12	1001	Frau	Dr.	Susi	zu	Stulle	Mausi	18.03.1974	20	Freund	7	guter Typ	Eltern	Familie zu Stulle	Brotgasse 1	15423	Berlin	Deutschland	F	447 78 78
13	1001	Frau	Dr.	Susi	zu	Stulle	Mausi	18.03.1974	20	Freund	7	guter Typ	Eltern	Familie zu Stulle	Brotgasse 1	15423	Berlin	Deutschland	T	447 78 79
14	1001	Frau	Dr.	Susi	zu	Stulle	Mausi	18.03.1974	20	Freund	7	guter Typ	Eltern	Familie zu Stulle	Brotgasse 1	15423	Berlin	Deutschland	T	447 78 80
15	1003	Frau	cand. ing.	Moni		Mauer		08.10.1970	24	verw.	10	klasse	Freund	Axel Rocker	Kohlenweg 11	12457	Berlin	Deutschland	T	357 15 98
16	1003	Frau	cand. ing.	Moni		Mauer		08.10.1970	24	verw.	10	klasse	Büro	IKEA	Ivarstieg 21 a	22455	Hamburg	Deutschland	T	386-25 45
17	1003	Frau	cand. ing.	Moni		Mauer		08.10.1970	24	verw.	10	klasse	Büro	IKEA	Ivarstieg 21 a	22455	Hamburg	Deutschland	T	386-78 36
18	1003	Frau	cand. ing.	Moni		Mauer		08.10.1970	24	verw.	10	klasse	Eltern	Familie Nabel	Am Loch 3	44125	Mölln	Deutschland	T	247 87 32
19	1003	Frau	cand. ing.	Moni		Mauer		08.10.1970	24	verw.	10	klasse	Freund	Arndt Ampel	Kreuzach Str. 4	84125	St. Joseph	Deutschland	T	774 75 11
20	1004	Herr		Peter		Panther	Quicki	21.07.1955	39	gesch.	5	reich!!!	Bruder	Axel Rocker	Kohlenweg 11	12457	Berlin	Deutschland	T	357 15 98
21	1004	Herr		Peter		Panther	Quicki	21.07.1955	39	gesch.	5	reich!!!	Büro	IKEA	Ivarstieg 21 a	22455	Hamburg	Deutschland	T	386-25 45
22	1004	Herr		Peter		Panther	Quicki	21.07.1955	39	gesch.	5	reich!!!	Büro	IKEA	Ivarstieg 21 a	22455	Hamburg	Deutschland	T	386-78 36
23	1005	Frau	Dipl.-Kff.	Vicki	von	Vierhuß		15.02.1968	26	verw.	2	schrecklich	Büro	Das Buch	Seitenstr. 2	12457	Berlin	Deutschland	T	654 73 21
24	1005	Frau	Dipl.-Kff.	Vicki	von	Vierhuß		15.02.1968	26	verw.	2	schrecklich	Büro	Das Buch	Seitenstr. 2	12457	Berlin	Deutschland	F	654 73 55
25	1005	Frau	Dipl.-Kff.	Vicki	von	Vierhuß		15.02.1968	26	verw.	2	schrecklich	privat	WG am Brunnen	Am Loch 17	44125	Berlin	Deutschland		

Abbildung 6: Relation PERSON in der 1. Normalform

Die Lücken, die dadurch entstehen, müssen mit den entsprechenden Attributwerten aufgefüllt werden. Nur dann handelt es sich schließlich um eine echte Tabelle! Dadurch entstehen etliche Wiederholungen. Diese gilt es im nächsten Schritt zu eliminieren.

Die Telefon- bzw. Faxnummer haben wir gleich zusammengefaßt, da sich dies als sinnvoll erweist. Es gibt reine Telefon- und Faxanschlüsse und Nummern, an denen Kombigeräte angeschlossen sind, so daß eine Nummer zum Telefonieren und Faxen genutzt werden kann. Um dieser Unterscheidung Rechnung zu tragen, wird der jeweiligen Nummer ein Kennzeichen angefügt. T für Telefon. F für Faxanschluß. B für beides, Fax- und Telefonanschluß in einem.

3.2.4. 2. NORMALFORM

Um diese Redundanzen wegzubekommen, wird die Tabelle in die zweite Normalform gebracht.

Definition 2. Normalform

Die 2. Normalform verlangt, daß alle nicht dem Primärschlüssel angehörigen Attribute von dem gesamten Primärschlüssel abhängig sein müssen, nicht jedoch von Schlüsselteilen. Verletzungen dieser Forderungen sind nur dann möglich, wenn der Primärschlüssel aus mehreren Attributen zusammengesetzt ist.

Besondere Beachtung findet der letzte Satz der Definition. Handelt es sich um einen Primärschlüssel, der aus nur einem Attribut besteht, was meistens der Fall ist, ist die 2. Normalform identisch mit der 1. Normalform. Es stellt sich hier also die Frage, welches ist der Primärschlüssel?

Wie oben schon erläutert, ist es die Personenummer (P_Nr)! Ist es die Personenummer? Wie in der Definition des Primärschlüssels besonders hervorgehoben wird, darf dieser nur einmal pro Tabelle vorkommen. Deshalb ist es ja gerade der Primärschlüssel! In der obigen Tabelle siehst Du deutlich, daß durch die Bildung der 1. Normalform die Personenummer ihre "Schlüsselrolle" verloren hat. Die Personenummer kann mithin nicht der Primärschlüssel sein. Was dann?

Du hast nun zwei Möglichkeiten. Die eine Möglichkeit besteht darin, daß Du einfach einen neuen Schlüssel einführst, indem Du jede Zeile durchnummerierst. In der Tabelle oben haben wir dies bereits getan (Attribut Nr: 1 bis 25). Die zweite Möglichkeit besteht darin, aus den vorhandenen Attributen einen Primärschlüssel zu "bauen". Einen kombinierten Primärschlüs-

sel. Die zweite Möglichkeit scheint in unserem Fall nicht sinnvoll zu sein, da es einem Anfänger schwer fallen wird zu erkennen, welche Kombination von welchen Attributen sinnvoll ist. So wählen wir hier die erste Möglichkeit. Doch gräme Dich nicht, es ist alles zu Deinem Besten.

3.2.5. 3. NORMALFORM

Daß die 3. Normalform nicht so einfach zu erreichen ist, scheint wahrscheinlich. Schließlich versprochen wir Kniffliges. Später!

Definition 3. Normalform

Die dritte Normalform ist verletzt, wenn es in der Relation Abhängigkeiten zwischen nicht dem Primärschlüssel angehörigen Attributen gibt, so daß ein solches Attribut eindeutig auf ein anderes verweist.

Schlüsselattribut ist immer noch die von uns eingeführte Nr. Doch um die geht es bei der 3. Normalform nicht. Es geht um den Rest. Du mußt prüfen, ob es Abhängigkeiten zwischen diesen restlichen Attributen gibt.

So hängen Anr, Titel, VName, Zus, Name, SName, Geburtsdat, Alter, Kateg, Prio, Notizen nur vom Nichtschlüsselattribut P_Nr ab! Diese Verletzung der 3. Normalform wird dadurch behoben, daß eine neue Relation gebildet wird. Diese neue Relation bekommt den Namen PERSON und hat die Personenummer als Primärschlüssel. Das Alter hängt vom Nichtschlüsselattribut Geburtsdatum ab. Schließlich kann man das Alter einer Person ohne weiteres ausrechnen, wenn man das Geburtsdatum und das aktuelle Datum kennt. Diese Rechenleistung wird auch von einem Datenbankverwaltungssystem erwartet. Das Alter ist somit vom Geburtsdatum abhängig (nicht umgekehrt!) und wird deshalb ersatzlos gestrichen!

<u>P_Nr</u>	AnNr	Titel	VName	Zus	Name	Pri	SName	Geburtsdat	Kateg	Notizen
1000	1	Prof.	Karl		Kahle	1	Glatze	19.05.1923	gesch.	
1001	2	Dr.	Susi	zu	Stulle	7	Mausi	18.03.1974	Freund	guter Typ
1003	2	cand. ing.	Moni		Mauer	10		08.10.1970	verw.	klasse
1004	1		Peter		Panther	5	Quicki	21.07.1955	gesch.	reich!!!
1005	2	Dipl.-Kff.	Vicki	von	Vierhuß	2		15.02.1968	verw.	schrecklich

Abbildung 7: Relation PERSON in der 3. Normalform

Straße, PLZ, Ort, Land hängen nur vom Nichtschlüsselattribut Bewohner ab. Daraus ergibt sich die Tabelle ADRESSE. Als Primärschlüssel könnte Bewohner dienen. Aus den oben erläuterten Gründen ist dies kein sinnvoller Schlüssel. Familie Kahle könnte noch einen Sommersitz haben, wodurch es zweimal als Ausprägung den Primärschlüssel "Familie Kahle" gäbe. Dies darf nicht sein! Es empfiehlt sich also eine fortlaufende Nummer als Schlüssel einzuführen, die Adressnummer (Adr_Nr).

<u>Adr_Nr</u>	Bewohner	Straße	PLZ	Ort	Land
1500	TU Berlin	Str. d. 17 Juni	11452	Berlin	Deutschland
1501	Familie Kahle	Plattstr. 21	14525	Berlin	Deutschland
1502	Gerda Hui	Lustweg 6	24513	Hannover	Deutschland
1503	Familie zu Stulle	Brotgasse 1	15423	Berlin	Deutschland
1504	Heiko Hensel	Tomi-Weg 52	14235	Berlin	Deutschland
1505	Susi zu Stulle	Am Baguette 2	10212	Berlin	Deutschland
1506	IKEA	Ivarstieg 21 a	22455	Hamburg	Deutschland
1507	Familie Nabel	Am Loch 3	44125	Möln	Deutschland
1508	Arndt Ampel	Kreuzach Str. 4	84125	St. Joseph	Deutschland
1509	Axel Rocker	Kohlenweg 11	12457	Berlin	Deutschland
1510	Das Buch	Seitenstr. 2	12457	Berlin	Deutschland
1511	WG am Brunnen	Am Loch 17	44125	Berlin	Deutschland

Abbildung 8: Relation ADRESSE in der 3. Normalform

Das Kennzeichen (K) hängt nur von der Telefonnummer (Tel_Nr) ab. Daraus ergibt sich die Tabelle TEL_FAX mit dem Primärschlüssel Tel_Nr.

Tel_Nr	K
247 87 32	T
314-0	T
314-2 24 54	T
314-7 34 82	T
314-7 34 95	F
321 21 21	T
321 45 74	T
321 54 78	T
357 15 98	T
386-25 45	T
386-78 36	T
412 41 78	T
447 78 78	F
447 78 79	T
447 78 80	T
478 41 41	F
654 73 21	T
654 73 55	F
774 75 11	T
784 12 12	T
874 54 21	T

Abbildung 9: Relation TEL_FAX in der 3. Normalform

Auch hier hat die Normalisierung immerhin vier Zeilen erspart. Drei Nummern waren redundant.

Die Telefonnummer wiederum hängt nur vom Nichtschlüsselattribut Bewohner ab. Dadurch ergibt sich die Relation ADRESSE_TEL_FAX. Da eine Adresse N Telefone haben kann, ein Telefon aber immer zu genau einer Adresse gehört, liegt eine 1:N-Relation vor. Deshalb ist als Schlüssel die Telefonnummer zu wählen - nicht der Bewohner. Statt des Bewohners wurde bei der Relation Adresse ein neuer Primärschlüssel eingeführt, die Adr_Nr. Deshalb wird in die Relation ADRESSE_TEL_FAX nicht das Attribut Bewohner sondern Adr_Nr aufgenommen.

Tel_Nr	Adr_Nr
247 87 32	1507
314-0	1500
314-2 24 54	1500
314-7 34 82	1500
314-7 34 95	1500
321 21 21	1502
321 45 74	1501
321 54 78	1505
357 15 98	1509
386-25 45	1506
386-78 36	1506
412 41 78	1502
447 78 78	1503
447 78 79	1503
447 78 80	1503
478 41 41	1501
654 73 21	1510
654 73 55	1510
774 75 11	1508
784 12 12	1504
874 54 21	1505

Abbildung 10: Relation ADRESSE_TEL_FAX in der 3. Normalform

Die Tabelle TEL_FAX und die Tabelle ADRESSE_TEL_FAX sind fast identisch. Es liegt zwar keine Redundanz im eigentlichen Sinne vor, jedoch könnten die beiden Relationen zu einer zusammengefaßt werden. Dies erleichtert die spätere Arbeit mit der Datenbank enorm. Es ist leicht einsichtig, daß weniger Tabellen übersichtlicher sind als mehr. Im obigen Fall sind sich die Experten jedoch nicht einig!!! Also mach es wie Du willst. Hier trotzdem die zusammengefaßte Relation TEL_FAX* zur Verdeutlichung.

Tel_Nr	K	Adr_Nr
247 87 32	T	1507
314-0	T	1500
314-2 24 54	T	1500
314-7 34 82	T	1500
314-7 34 95	F	1500
321 21 21	T	1502
321 45 74	T	1501
321 54 78	T	1505
357 15 98	T	1509
386-25 45	T	1506
386-78 36	T	1506
412 41 78	T	1502
447 78 78	F	1503
447 78 79	T	1503
447 78 80	T	1503
478 41 41	F	1501
654 73 21	T	1510
654 73 55	F	1510
774 75 11	T	1508
784 12 12	T	1504
874 54 21	T	1505

Abbildung 11: Relation TEL_FAX* in der 3. Normalform

Nachdem schon soviel aus der ursprünglichen Tabelle verschwunden ist, stellt sich die Frage, was noch da ist.

<u>Nr</u>	P_Nr	Bewohner	Typ
-----------	------	----------	-----

Bewohner wird sinnvollerweise wie oben durch Adr_Nr ersetzt.

<u>Nr</u>	P_Nr	Adr_Nr	Typ
-----------	------	--------	-----

Typ stellt die Beziehung der Adresse zur Person her. Handelt es sich bei der Adresse für die entsprechende Person um das Büro, ist es die Privatadresse oder die geliebte Mutter? Die Kennzeichnung wäre also bei keiner der beiden Attribute P_Nr oder Adr_Nr alleine aussagekräftig. Es bedarf

beider Attribute, um den Typ in sinnvollen Zusammenhang zu bringen. Trotzdem hängt der Typ von diesen beiden Nichtschlüsselattributen direkt ab. Es entsteht die Tabelle PERSON_ADRESSE. Da eine Person mehrere Adressen haben kann, eine Adresse aber auch von mehreren Personen bewohnt werden kann, handelt es sich in unserem Fall um eine N:M-Beziehung. Beispielsweise wohnen Moni Mauer und Peter Panther beide bei Axel Rocker. Für Moni ist Axel der Freund. Für Peter ist Axel der Bruder.

P_Nr	Adr_Nr	Typ
1000	1500	Büro
1000	1501	privat
1000	1502	Geliebte
1001	1503	Eltern
1001	1504	Freund
1001	1505	privat
1003	1506	Büro
1003	1507	Eltern
1003	1508	Freund
1003	1509	Freund
1004	1506	Büro
1004	1509	Bruder
1005	1510	Büro
1005	1511	privat

Abbildung 12: Relation PERSON_ADRESSE in der 3. Normalform

Die folgende Grafik faßt nochmals alle Relationen und deren Primärschlüssel zusammen.

Relation	Primärschlüssel
Person	P_Nr
Adresse	Adr_Nr
Tel Fax*	Tel_Nr
Person_Adresse	P_Nr + Adr_Nr

Abbildung 13: Zusammenfassung der Relationen in der 3. Normalform

Nun sind alle Tabellen redundanzfrei. Die Schwierigkeit bestand letztlich darin, herauszufinden, welche einzelnen Relationen gebildet werden müssen. Dazu ist die Normalisierung eine formale Möglichkeit. Ist jedoch nach einiger Übung das Prinzip des Relationenbildens etwas in die Fingerspitzen gewandert, so wirst Du nicht mehr verstehen, wie man so umständlich vorgehen konnte - wie wir nur auf die Idee kommen konnten, alles so kompliziert zu machen. Das ist dann später.

Oben hatten wir von einer 2. Möglichkeit gesprochen, die Schlüssel zu wählen. Wir hätten statt eines neuen Schlüssels (Nr), der am Ende sowieso wieder weggefallen ist, den kombinierten Primärschlüssel aus P_Nr + Adr_Nr¹¹ + Tel_Nr verwenden können. Dies wäre sinnvoll gewesen, da genau dieser Schlüssel für die obige Tabelle der 1. Normalform der richtige Schlüssel ist. Auf der jetzigen Stufe der Erkenntnis ist dies leicht nachvollziehbar, da es sich ja genau um die Schlüssel der einzelnen Relationen handelt. Wenn Du genug geübt hast, wirst Du mit einem oder zwei Blicken erkennen, daß sich alle anderen Attribute nur wiederholt haben. Es sind sogenannte Wiederholungsgruppen aufgetreten.

Wie aber, ist die Frage, wäre die Normalisierung weiter von statten gegangen? Durch den kombinierten Schlüssel ist es schließlich nicht mehr möglich, die 2. Normalform zu übergehen. Sobald es einen kombinierten Primärschlüssel gibt ist es möglich, daß die 2. Normalform nicht erfüllt ist. Wir hätten in diesem Fall nicht die Abhängigkeiten zwischen den Nichtschlüsselattributen aufspüren müssen, sondern die Abhängigkeiten der Nichtschlüsselattribute von Teilen des kombinierten Primärschlüssels. Für jeden Teil des Schlüssels (z. B.: P_Nr) hätten wir entsprechende Nichtschlüsselattribute (z. B.: Anr, Name, Titel) gefunden, so daß am Ende genau dieselben vier Tabellen entstanden wären. Unterschied? Eine Frage des Geschmacks!

¹¹Auf dem obigen Stand natürlich das Attribut Bewohner statt der fortlaufenden Nummer Adr_Nr.

3.2.6. ZUSAMMENFASSUNG

Nochmal in Kürze! Durch die Normalisierung wird ein **redundanzfreies** relationales Datenmodell erzeugt. In der 1. Normalform werden die Daten in eine Form gebracht, so daß dabei eine Tabelle entsteht. Die entstehenden Lücken werden mit den zugehörigen Daten gefüllt, so daß dadurch Redundanzen entstehen. Bei der 2. Normalform versucht man die Nichtschlüsselattribute herauszusuchen, die nicht vom gesamten Primärschlüssel abhängen. Besteht der Primärschlüssel aus nur einem Attribut (z. B.: PersonenNr) ist die 1. Normalform immer identisch mit der 2. Normalform. Bei der 3. Normalform werden die gegenseitigen Abhängigkeiten zwischen den Nichtprimärschlüsselattributen festgestellt. Attribute, die sich durch andere berechnen lassen werden ersatzlos gestrichen. Attribute die von anderen Attributen abhängen werden ausgegliedert und bilden eine neue Tabelle.

3.2.7. WEITERE NORMALFORMEN

Wir haben bisher von drei Normalformen gesprochen. Das reicht eigentlich! Doch es gibt noch mehr davon. Es gibt tatsächlich noch die 4. und 5. Normalform. Diese können, wie oben bereits erwähnt, relevant werden, wenn kombinierte Primärschlüssel aus mehr als zwei Attributen zusammengesetzt sind. Theoretisch gibt es sogar noch mehr Normalformen. Es soll jedoch mit dieser Erkenntnis unsere Genugtuung im Bereich der Normalisierung erreicht sein. Drei sind genug!

3.3. DAS ENTITY-RELATIONSHIP-MODELL

Bis hierher war es schon sehr schön. Jetzt macht es richtig Spaß! Natürlich ist die obige Vorgehensweise des Datenbankentwurfs ganz nett - doch ziemlich unbefriedigend. Es gibt Abhilfe! Für den Datenbankentwurf gibt es Methoden.

Bei Algorithmen gab es als Hilfe das Nassi-Shneiderman-Diagramm (Struktogramm). Auch bei der Modellierung von Datenmodellen ist man nicht alleine gelassen. Einfach und enorm hilfreich ist das Entity-Relationship-

Modell (ERM) von CHEN¹². Ein paar Symbole verdeutlichen schnell, worum es geht.

Wie einfach das Ganze ist und wie schwer es werden kann, verdeutlichen wir an unserem schönen Adressverwaltungsbeispiel.

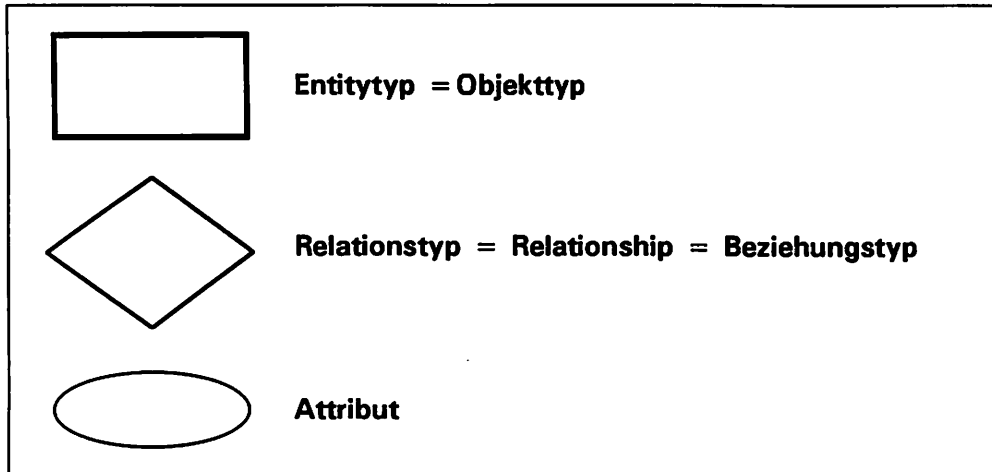


Abbildung 14: Symbole im Entity-Relationship-Diagramm (ERD)

Beim Entity-Relationship-Modell ist die Herangehensweise eine andere als oben gezeigt. Sie wird von Verfechtern oft als "natürliche Herangehensweise" bezeichnet. Statt sich alle Attribute zu überlegen, diese zu sammeln und dann künstlich zusammenzuwerfen, wird anders vorgegangen. Man betrachtet Objekte, wie Rechnung, Kunde oder Telefon. Und man ermittelt die Beziehungen zwischen diesen Objekten, wie Kunde *hat* Telefon oder Kunde *bekommt* Rechnung. Die Objekte werden Entitytypen, deren Ausprägungen Entitäten genannt. Die Beziehungstypen nennt man Relationstypen, deren Ausprägungen Relation¹³. (Viele Worte!¹⁴)

¹²siehe Balzert 90, S. 43 ff

¹³Dieser Ausdruck ist verwirrend, da er nicht mit dem identischen Ausdruck aus dem relationalen Datenmodell verwechselt werden darf. Dort werden Relationen durch Tabellen dargestellt.

¹⁴Der Autor fragt sich, ob es Zweck dieser mannigfachen Begriffe sein könnte, einfache Sachen kompliziert erscheinen zu lassen. Doch wer sollte derartige Absichten hegen? Wozu?

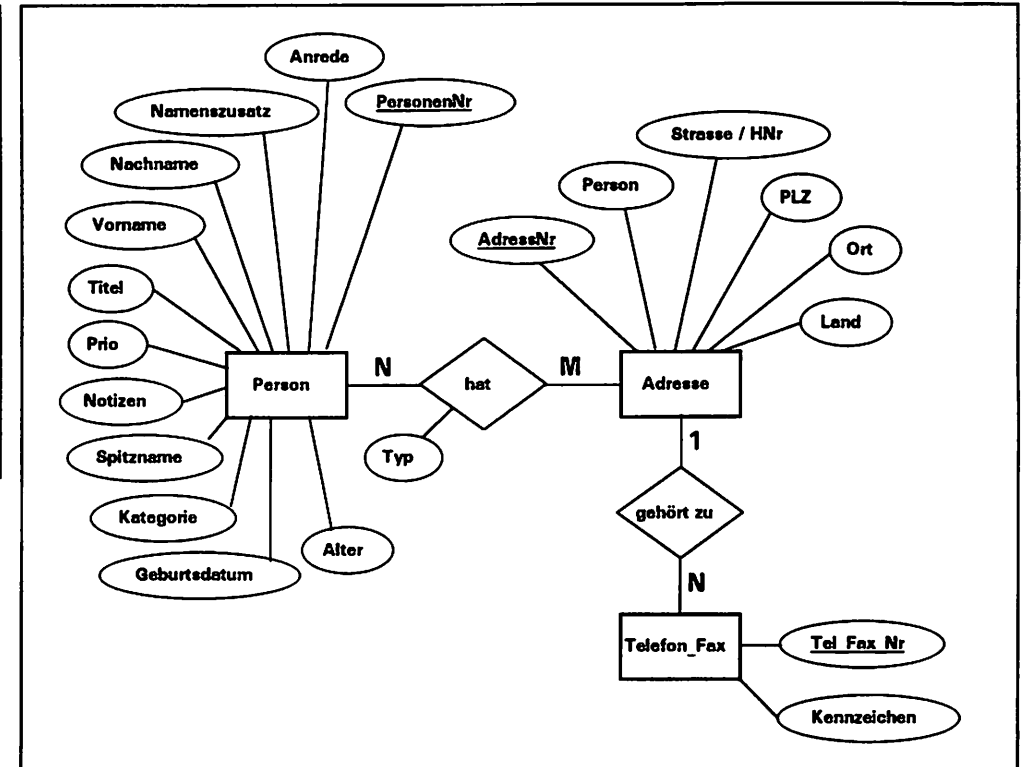


Abbildung 15: Entity-Relationship-Diagramm der Adressverwaltung

In der Abbildung 15 siehst Du die drei bekannten Entitäten Person, Adresse und Telefon. Das sind die Rechtecke. Zwischen zwei Entitäten muß immer eine Beziehung vorhanden sein. Das sind die Rauten. Eine Person *hat* eine Adresse. Eine Adresse *hat* eine Person. Zugegebenermaßen ist "hat" keine glückliche Beziehungsbezeichnung. Es ist jedoch manchmal schwierig, passendes zu finden. Beachten muß Du, daß die Linien in beide Richtungen deuten. Pfeile werden nicht verwendet. Die Elipsen stehen für die Attribute. Der Primärschlüssel ist unterstrichen. Wie Du siehst, haben nicht nur die Entitäten Attribute, sondern auch Beziehungen können Attribute besitzen. Wie oben festgestellt, hängt beispielsweise der Typ, der die

Beziehung zwischen der Person und der Adresse angibt (privat, Büro, Mutter etc.), von der Person und der Adresse ab. Logisch! Oder?

Diese einfache Darstellung ist schnell gemacht. Der Dienst, der durch sie geleistet wird, kann kaum hoch genug bewertet werden. Das Entity-Relationship-Diagramm hilft nicht nur beim Entwurf der Datenbank, sondern ist auch beim täglichen Betrieb für Abfragen eine Notwendigkeit.

Wie beim Nassi-Shneiderman-Diagramm kann auch das Entity-Relationship-Diagramm schrittweise verfeinert werden. So könnte in einem ersten Schritt nur die Person als Entität erscheinen. Als Attribute könnte die Person Name, Adresse und Telefon haben. Daraus wird im zweiten Schritt Abbildung 15. Attribute können also zu eigenen Entitäten werden. Diese Top-Down-Verfeinerung ist wichtig, da mit dieser Methode Datenmodelle ganzer Unternehmen modelliert werden müssen. Hierfür setzte sich in den letzten Jahren allerdings mehr und mehr eine Weiterentwicklung des Entity-Relationship-Modells, nämlich das Structured-Entity-Relationship-Modell (SERM) durch.¹⁵

Zurück! Das Entity-Relationship-Diagramm kann einem helfen, so daß man es auf Anhieb schafft, Relationen in die 3. Normalform zu bringen. Dies ist unwahrscheinlich, doch einen großen Schritt in Richtung 3. Normalform nimmt es einem ab! In unserem Fall ist lediglich das Attribut Alter der Relation PERSON zu streichen, da es den Anforderungen der 3. Normalform nicht genügt.¹⁶ Es läßt sich aus dem Geburtsdatum ableiten.

Nun die Frage, wie der Übergang von der Zeichnung hin zu den Tabellen geschieht. Ganz einfach! Alle Objekte, Entitäten und Beziehungen werden jeweils eine Tabelle. Bei den Entitäten ist das ganz unproblematisch. Als Primärschlüssel nimmt man den Primärschlüssel. Dieser sollte sinnvollerweise aus nur einem Attribut bestehen. Bei den Beziehungen wird zuerst geschaut, von welchem Typ sie sind. N:M, N:1 oder 1:1. Bei N:M Beziehungen werden beide Primärschlüssel der angrenzenden Entitäten über-

nommen. Sie dienen als kombinierter Primärschlüssel in der Beziehungstabelle. Dies ist bei der Beziehung zwischen Person und Adresse der Fall. Eine Person kann mehrere Adressen bewohnen, eine Adresse kann von mehreren Personen bewohnt werden. Es handelt sich somit um eine N:M Beziehung. Die Tabelle PERSON_ADRESSE¹⁷ bekommt deshalb als Primärschlüssel die PersonenNr + die AdressNr. Keiner der beiden Schlüsselteile für sich allein könnte die Primärschlüssel-Anforderungen erfüllen.

Die PersonenNr und die AdressNr fungieren allerdings nicht nur als Primärschlüssel. Beide sind jeweils Fremdschlüssel. Über den Fremdschlüssel AdressNr wird auf den Primärschlüssel AdressNr in der Relation Adresse zugegriffen. Nur durch diesen Verweis des Fremd- auf den Primärschlüssel ist die Verbindung im Datenmodell möglich. Du möchtest alle Adressen zu einer Person herausuchen. Finde die Person in der Relation PERSON. Lese den Primärschlüssel (PersonenNr). Suche in der Relation PERSON_ADRESSE¹⁸ alle Sätze mit der identischen PersonenNr (Fremdschlüssel) heraus. Lese von all diesen gefundenen Datensätzen die AdressNr (Fremdschlüssel der Relation ADRESSE). Suche in der Relation ADRESSE alle Datensätze mit identischer AdressNr. {Noch mal lesen! Und dabei mit dem Finger auf dem Entity-Relationship-Diagramm Abbildung 16 entlangfahren!}

Definition Fremdschlüssel

Ein Fremdschlüssel ist ein Attribut, welches in einer anderen Relation als Primärschlüssel vorhanden ist. Durch die Verknüpfung von Fremdschlüssel und Primärschlüssel entsteht die Verbindung von Relationen.

Bei einer Beziehung vom Typ 1:N gibt es wieder zwei Möglichkeiten, diese in Tabellen umzusetzen. Die erste Alternative besteht darin, eine Tabelle zu erzeugen. Als Primärschlüssel dient der Primärschlüssel der Relation, die dem N zugeordnet werden kann. Eine Adresse hat N Telefone. Ein Telefon

¹⁵Es sei auf Sekundärliteratur verwiesen!

¹⁶Abhängigkeiten zwischen Nichtschlüsselattributen.

¹⁷Im Entity-Relationship-Diagramm noch als "hat" bezeichnet.

¹⁸dito

gehört zu genau einer Adresse. Die Relation ADRESSE_TELEFON¹⁹ erhält den Primärschlüssel Tel_Fax_Nr.

Frage:

Warum wird im obigen Beispiel bei der 1:N-Beziehung die Tel_Fax_Nr als Schlüssel gewählt? Warum nicht die AdressNr?

Antwort:

Wäre die AdressNr der Primärschlüssel, so könnte es mehrere Datensätze mit dem selben Primärschlüssel geben. Das geht nicht!

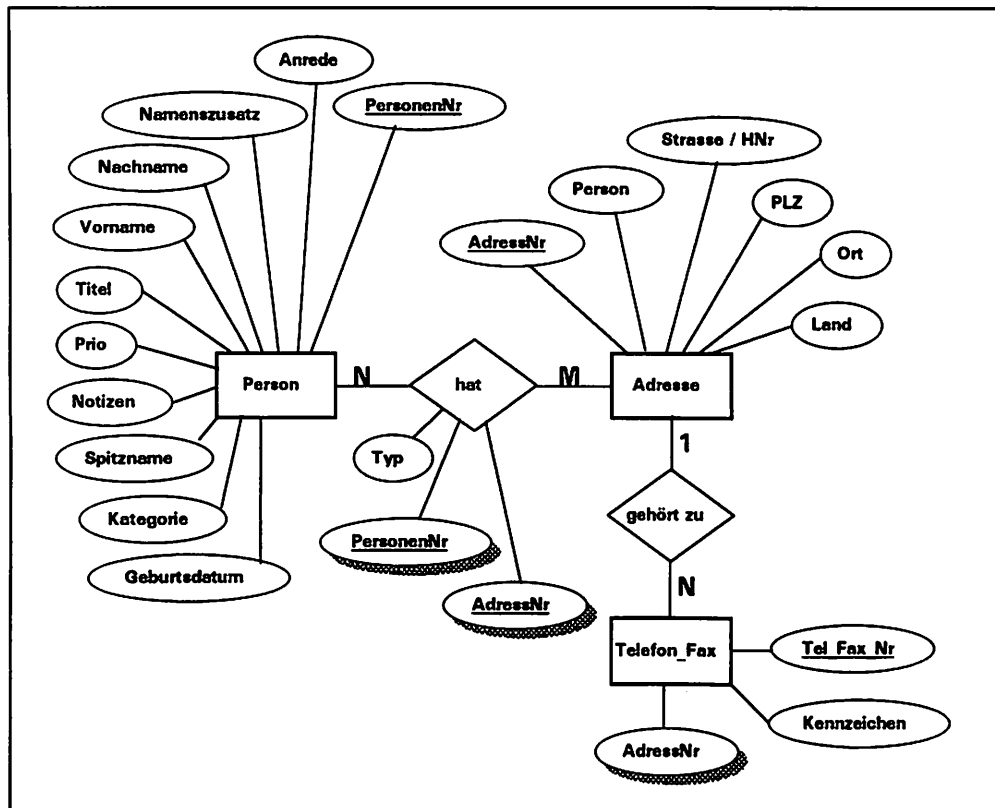


Abbildung 16: Entity-Relationship-Diagramm der Adressverwaltung in 3. Normalform

¹⁹Im Entity-Relationship-Diagramm noch als "gehört zu" bezeichnet.

Die 2. Alternative kann darin bestehen, die Beziehung nicht in eine Tabelle umzusetzen. Sie wird einfach weggelassen. Statt dessen wird nur der Primärschlüssel der Relation, der der 1 zugeordnet ist als Fremdschlüssel in die Tabelle aufgenommen, die dem N zugeordnet ist. Die Tabelle Telefon erhält das zusätzliche Attribut AdressNr.

Bei einer 1:1-Beziehung zweier Entitäten stellt sich die Frage, ob überhaupt zwei Tabellen zweckmäßig sind. Sie werden höchstwahrscheinlich den selben Primärschlüssel besitzen. Eine Aufspaltung in zwei Tabellen kann aber tatsächlich Sinn machen.²⁰ Argumente für die Zweckmäßigkeit einer weiteren Tabelle, in der die Beziehung hinterlegt ist, konnten wir bisher nicht finden.

Zur nochmaligen Verdeutlichung ist in der Abbildung 16 unser Endergebnis in 3. Normalform zu bewundern. Fremdschlüssel tauchen im Entity-Relationship-Diagramm normalerweise nicht auf! Diese Konvention haben wir hier im Namen der Didaktik ausnahmsweise fallen gelassen. Die Fremdschlüssel sind durch Schatten gekennzeichnet.

Es sind dann alle Entitäten und Beziehungen in Tabellen verwandelt. Und dann? Dann werden alle Tabellen überprüft, ob sie in der 1. Normalform sind, ob sie in der 2. Normalform sind und ob sie sich last but not least in der 3. Normalform befinden. Und dann? Dann sind die Relationen bereit, implementiert zu werden.

Alle Vorgänge, die nun am Entity-Relationship-Modell erläutert wurden, sind im Grunde nur Wiederholung gewesen. Das selbe ist auch durch die reine Normalisierung entstanden. Die beste Methode ist zweifelsfrei eine Kombination aus Entity-Relationship-Diagramm und Normalisierung. Um zu veranschaulichen, wie kompliziert es werden kann, die folgende Überlegung.²¹

²⁰Dies hat sich der Autor von Spezialisten versichern lassen.

²¹Es sei dem Leser, welcher sich beim Lesen des folgenden dabei ertappt, durch ein sich verstärkendes Unwohlsein befallen zu werden, dadurch geholfen, in dem wir ihm schon vorher mitteilen, daß diese Überlegungen lediglich zur Veranschaulichung der Problematik dienen und in keinsten Weise zum hoch

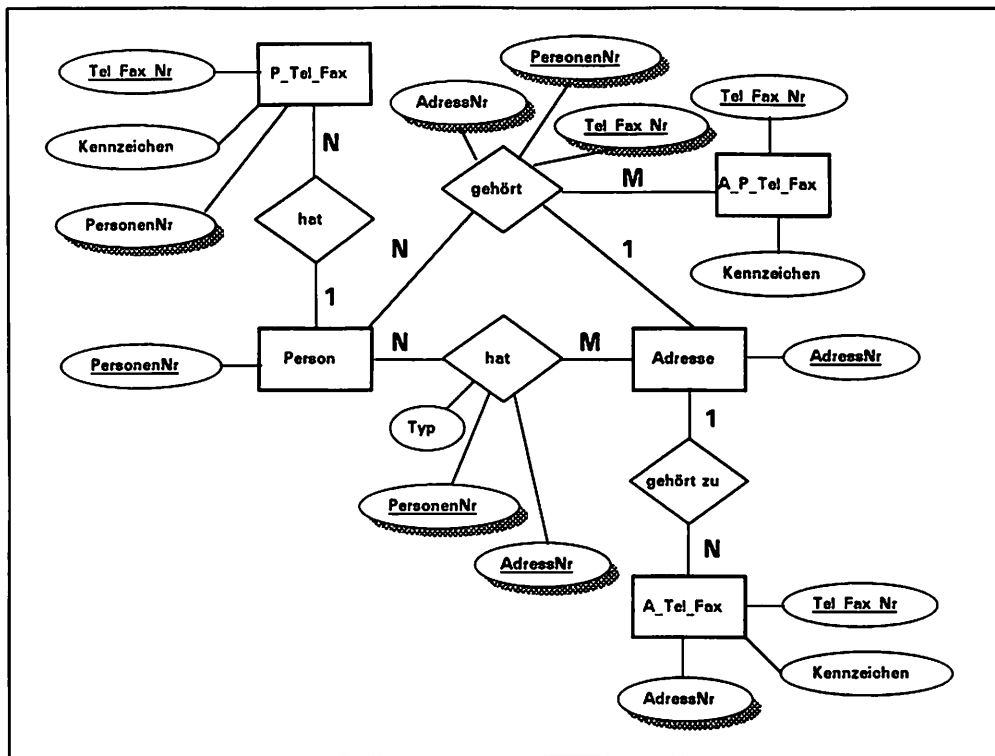


Abbildung 17: chaotisches Entity-Relationship-Diagramm zur Verdeutlichung der Telefonproblematik

Bisher hatten wir Dir erzählt, die Beziehung zwischen ADRESSE und TELEFON sei eine 1:N-Beziehung. Dies ist der sinnvolle Normalfall. Dieser Fall ist auch in Abbildung 17 rechts unten dargestellt. Die Zeichnung ist um einige Nichtschlüsselattribute beraubt. Es soll den Überblick erleichtern.

Was ist aber mit den Handys los? Alle Mobiltelefone zeichnen sich durch ihre Mobilität aus. Und dadurch sind sie gerade nicht einer Adresse zu zu-

ordnen. Sie gehören zur PERSON! Jede PERSON kann mehrere TELEFONE besitzen. Jedes TELEFON gehört zu einer PERSON. Nehmen wir also an, es handele sich um eine 1:N-Beziehung²². Dieser Fall ist in Abbildung 17 rechts oben eingezeichnet.

Der dritte Fall behandelt die Problematik, daß eine Person bei einer Adresse mehrere Telefone haben kann, diese Telefone aber nicht einfach der Adresse zurechenbar sind. Es handelt sich also um eine Beziehung zwischen Person, Adresse und Telefon. Am Beispiel: Prof. Kahle hat ein Telefon in der Universität. Diese Nummer einfach nur der Adresse zu zuordnen wäre schlecht. Im grünen Buch²³ der TU-Berlin erschienen dann alle Telefonnummern z. B. des Hauptgebäudes mit Anschrift. Keiner könnte jedoch sagen, zu welcher Person die jeweilige Nummer gehört. Würde man das Telefon der Person zuordnen, so könnte nicht mehr unterschieden werden, ob es sich bei der Nummer von Prof. Kahle um die Privat- oder Uniadresse handelt. Oder evtl. um das Auto. Es kann also nur die Kombination der drei Relationen Person, Adresse und Telefon das notwendige Ergebnis liefern.

Dadurch ergeben sich drei Tabellen für das Telefon und bis zu drei Tabellen für die Beziehungen. Nicht nur, daß das Ganze höchst kompliziert ist, es ist auch völlig unpraktikabel. Gesucht ist eine bessere Lösung. Und dafür ist das Entity-Relationship-Modell ausgezeichnet geeignet! Du fängst noch mal von vorne an. Du zeichnest die drei Entitäten. Und nun beginnst Du Dir die Beziehungen zu überlegen. Wie in der Abbildung 18 zu sehen, ergibt sich eine wunderbare Lösung. Redundanzfrei!

akuten, prüfungsrelevanten Stoff gehören. Wer an derlei Modellierung Freude bekommt, was der Autor mit viel Verständnis nachempfinden kann, sei an Prof. Weber "Datenbanken und Informationssysteme" im Hauptstudium Informatik verwiesen.

²²Ungünstigerweise für unser Beispiel wäre es jedoch auch realistisch von einer N:M-Beziehung auszugehen, da Herr Müller und Frau Müller das *selbe* Handy benutzen.

²³Das ist das Telefonverzeichnis der TU-Berlin.

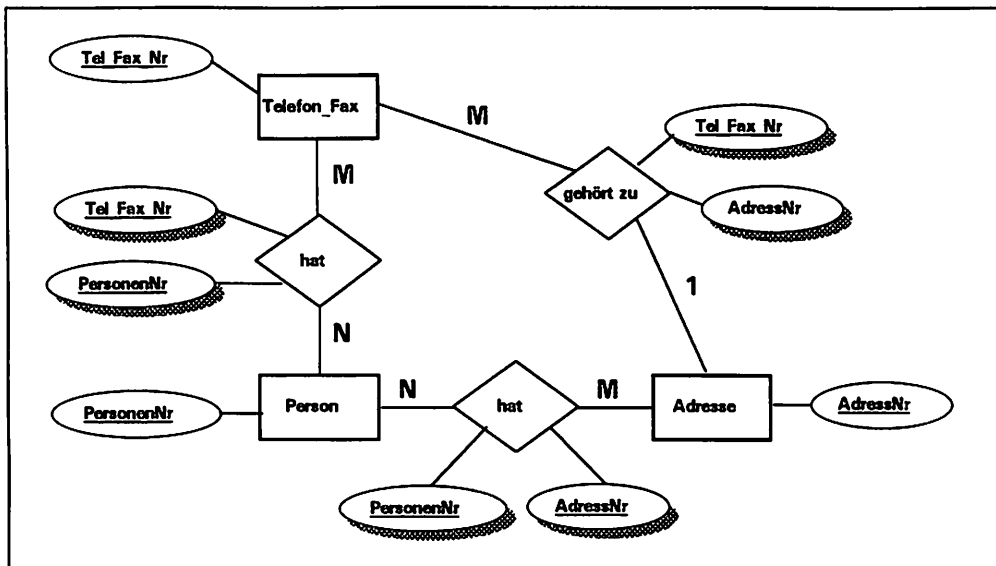


Abbildung 18: wunderschönes Entity-Relationship-Diagramm mit gelöster Telefonproblematik

3.4. INTEGRITÄTSANFORDERUNGEN AN EINE DATENBANK

Auf dem jetzigen Stand bist Du in der Lage, eine redundanzfreie, konsistente Datenbank zu entwerfen. Für den Betrieb sind damit aber noch nicht alle der wichtigsten Anforderungen an ein Datenbankverwaltungssystem erfüllt.

3.4.1. REFERENZINTEGRITÄT

Durch das Implementieren der Tabellen ist noch keine echte Datenbank entstanden. Was noch fehlt, ist die referentielle Integrität. Durch die referentielle Integrität wird ein wichtiger Schritt zur langfristigen Bewahrung der Datenkonsistenz getan. Dabei wird dem Datenbankverwaltungssystem mitgeteilt, welche Tabellen wie verknüpft werden. Die Verbindung erfolgt über die Fremdschlüssel-Primärschlüssel-Verknüpfung. In unserem Beispiel

(siehe Abbildung 16) wird die Relation **PERSON_ADRESSE**²⁴ mit den beiden Relationen **PERSON** und **ADRESSE** verknüpft. Dazu wird dem Datenbankverwaltungssystem mitgeteilt, daß der Fremdschlüssel **PersonenNr** in der Relation **PERSON_ADRESSE** auf die Relation **PERSON** verweist. Das DBVS verbindet dann **PersonenNr** (**PERSON_ADRESSE**) mit **PersonenNr** (**PERSON**). Mit den anderen Fremdschlüsseln wird ebenso verfahren. Die Abbildung 19 zeigt die Verknüpfungen.

Tabelle	Fremdschlüssel ²⁵	Primärschlüssel	Bezugstabelle
Person Adresse	PersonenNr	PersonenNr	Person
Person Adresse	AdressNr	AdressNr	Adresse
Telefon	AdressNr	AdressNr	Adresse

Abbildung 19: Referenzintegrität der Adressverwaltung

Ist die referentielle Integrität im Datenbankverwaltungssystem hinterlegt, so kontrolliert das DBVS bei allen Operationen, ob die Integrität verletzt werden könnte. Beispielsweise kann keine Adresse gelöscht werden, solange noch eine Telefonnummer zu dieser Adresse existiert.

Frage:

Worum darf eine Adresse nicht gelöscht werden, solange dazu noch eine Telefonnummer gespeichert ist?

Antwort:

Weiß die Datenintegrität des Datenbestandes sonst nicht mehr gegeben wäre.

3.4.2. NACHSCHLAGETABELLE

Eine weitere sehr praktische Sache sind Nachschlagetabellen. Sie dienen dazu, für einzelne Attribute mögliche Ausprägungen bereit zu halten. So könnte es beispielsweise eine Nachschlagetabelle für das Attribut **Titel** in der Relation **PERSON** geben. In dieser Nachschlagetabelle stehen dann Ausprägungen wie **Prof.**, **Dipl.-Kff.**, **Dipl.-Hdl.**, **Dr. med.**

²⁴"hat"

²⁵Fremdschlüssel und Primärschlüssel könnten verschiedene Namen haben, dies ist jedoch nicht sinnvoll.

Damit können Nachschlagetabellen ähnlich fungieren wie Tabellen, zu denen eine referentielle Integrität hergestellt wurde. Der Unterschied liegt aber mindestens darin, daß bei einer Nachschlagetabelle nicht die Notwendigkeit besteht, daß die verwendete Ausprägung in der Nachschlagetabelle vorhanden ist. So kann ein Titel durchaus Prof. Dr. Dr. Dr. Dr. lauten, ohne daß dieser in der Nachschlagetabelle vorhanden sein muß.

3.4.3. INDIZES

Die Primärschlüssel sind die Grundlage jeder Datenbank. Nach diesen Primärschlüsseln sind die Tabellen auch meist geordnet. Dann ist der Primärschlüssel gleichzeitig der Primärindex.

Definition Primärindex

Primärindex ist das Attribut, nach dem die Tabelle sortiert wird. Dies ist üblicherweise der Primärschlüssel.

Nach dem Primärindex kann folglich auch am schnellsten gesucht werden. Dies hat zur Konsequenz, daß man als Kunde bei sämtlichen Firmen immer als erstes nach seiner Kundennummer gefragt wird. Wer erhält schon bei der Bank eine Auskunft, kennt er nicht seine Kontonummer? Viel einfacher ist es da, wird man nach seinem Namen gefragt. Unseren Namen kennen wir alle. Das Suchen nach diesem dauert allerdings länger, da die Tabellen sequentiell durchgelesen werden müssen, bis alle Datensätze mit dem entsprechenden Namen gefunden sind. Das bedeutet, daß jedesmal alle Datensätze gelesen werden müßten. Hierfür gibt es Abhilfe.

Definition Sekundärindex

Wird nach einem bestimmten Attribut häufig gesucht, so empfiehlt es sich, dieses Attribut als Sekundärindex anzugeben. Sortiert nach diesem Attribut wird dann eine Extra-Tabelle angelegt, die für jedes dieser Attribute jeweils auf den Primärschlüssel der Tabelle verweist. Dies empfiehlt sich zumindest für alle Fremdschlüssel.

Durch den Sekundärindex kann wesentlich schneller nach Attributen gesucht werden. Das Prinzip ist schon aus der Dateiorganisation bekannt.²⁶ Am besten wird das Ganze wieder durch unser Beispiel klar. Hier ein Sekundärindex auf den Nachnamen der Personentabelle.

<u>Name</u> (Sekundärindex)	<u>P_Nr</u> (Primärschlüssel)
Kahle	1000
Mauer	1003
Panther	1004
Stulle	1001
Vierhuß	1005

Abbildung 20: Tabelle des Sekundärindex für den Nachnamen der Relation PERSON.

3.4.4. GÜLTIGKEITSPRÜFUNGEN

Elementar wichtig ist nicht nur, daß die Daten nicht doppelt vorkommen. Dadurch ist zwar Inkonsistenz vermieden. Die Integrität fordert allerdings noch mehr. Auch die nur einmal gespeicherten Daten müssen richtig sein. Das Geburtsdatum eines lebenden Menschen kann kaum 1492 lauten. Die PLZ darf keine Buchstaben enthalten. Dazu bieten Datenbankverwaltungssysteme Gültigkeitsprüfungen. So kann also verlangt werden, daß die PLZ eine ganze Zahl zwischen 0 und 99999 ist. Man kann auch gewisse Muster für Eingaben festlegen. Beispielsweise könnte ein Attribut immer erst einen großen Buchstaben, dann einen Bindestrich, drei Ziffern, einen Backslash und dann noch eine Zahl zwischen 0 und 3000. Das sieht etwa so aus:

A-245\2541

Gerade diese Funktion erspart im Alltagsgefecht dem Sachbearbeiter so manchen Fehler, der teuer werden kann.

Eine andere Alternative, die möglichen Werte der Attribute einzuschränken, sind natürlich auch Tabellen, in die die einzelnen Ausprägungen

²⁶siehe Jancke / Jeiter 89, S. 37 ff

aufgenommen werden. Man könnte so festlegen, daß in das Feld Anrede der Personentabelle nur Herr, Frau oder Firma eingegeben werden kann. Die Tabelle hat dann nur eine Spalte.²⁷

An dieser Stelle muß außerdem noch erwähnt werden, daß bei einzelnen Attributen eine Pflichteingabe festgelegt werden kann. Dann muß bei der Dateneingabe auf jeden Fall etwas in dieses Feld eingegeben werden. Dies ist notwendig bei allen Schlüsselfeldern: Primär- und Fremdschlüsseln. Es ist angeraten, bei allen elementaren Attributen. Doch das ist wieder eine Frage der Sichtweise. Das Marketing kann die Telefonnummer für unverzichtbar halten. Für den Vertrieb ist diese u. U. nebensächlich. Zu beachten gilt es, daß Datensätze nicht gespeichert werden können, sollte eine Pflichteingabe fehlen. Dies umgeht der Sachbearbeiter dann evtl. dadurch, daß er falsche Werte eingibt (PLZ = 00000).

3.4.5. PAßWORTSCHUTZ

Gerade der Paßwortschutz von Datenbankverwaltungssystemen ist einer der wichtigsten Vorteile. Bei Dateien konnte jeder alles lesen, verändern oder löschen. Bei einem DBVS bekommt jeder nur die Rechte für die Daten, die er braucht. Datenbankverwaltungssysteme liefern nicht nur umfangreiche Daten über Personen und Sachen, sie bieten auch die Möglichkeit mannigfache Auswertungen zu produzieren. Die Gefahren, die dadurch für die Datensicherheit und besonders für den Datenschutz²⁸ entstehen, sind enorm.

Bei Datenbankverwaltungssystemen können nicht nur Tabellen einzelne Paßwörter erhalten. Jedes einzelne Attribut kann ein oder mehrere Paßwörter erhalten. Es ist teilweise sogar möglich, einzelne Felder mit Paßwörtern zu belegen. Dabei können allen diesen Paßwörtern unterschiedliche

Rechte zugewiesen werden. Nur lesen, nur zusätzliche Datensätze einfügen, nur vorhandene Datensätze ändern oder alles: lesen, ändern, hinzufügen und löschen.

LITERATUR

- | | |
|--------------------|--|
| Balzert 90 | Balzert, Helmut; CASE Systeme und Werkzeuge, 2. Auflage, BI-Wissenschaftsverlag, Mannheim 1990 |
| Duden Inf. 93 | Duden Informatik; 2. Auflage, Dudenverlag, Mannheim 1993 |
| Finkenzeller 93 | Finkenzeller, Hermann; Paradox für Windows, 1. Auflage, Markt- und Technik-Verlag, München 1993 |
| Hansen 87 | Hansen, Hans Robert; Wirtschaftsinformatik I, 5. Auflage, Gustav Fischer Verlag, Stuttgart 1987 |
| Krallmann 94 | Krallmann, Hermann; Systemanalyse im Unternehmen, 1. Auflage, Oldenburg Verlag, München 1994 |
| Stahlknecht 93 | Stahlknecht, Peter; Einführung in die Wirtschaftsinformatik, 6. Auflage, Springer-Verlag, Berlin 1993 |
| Wortmann 92 | Wortmann, Jan; Datenbanksysteme, Skript zur Lehrveranstaltung EDV I/II, 1. Auflage, Berlin 1992 |
| Jancke / Jeiter 89 | Jancke, Michael; Jeiter, Vera; Datenorganisation, Skript zur Lehrveranstaltung EDV I/II, 3. Auflage, Berlin 1989 |
| Hagenbach 93 | Hagenbach, Dirk; Informix, Skript zur Lehrveranstaltung EDV I/II, 1. Auflage, Berlin 1993 |

²⁷Siehe auch unter Nachschlagetabellen.

²⁸Hierfür sei auf umfangreich verfügbare Sekundärliteratur verwiesen. Bei größerem Interesse kannst Du Dich an Prof. Lutterbeck vom Fachgebiet Informatik und Gesellschaft wenden. Datenschutz kann auch im Rahmen der wirtschaftswissenschaftlichen Studiengänge als Wahlfach besucht werden.