

H W P

Hochschule für Wirtschaft und Politik
5. Februar 1988

Entwurf und Anwendung relationaler Datenbanken an praxisrelevanten Beispielen

- 1. EDV, Betriebswirtschaftslehre und Sozialökonomie:
einige einleitende Bemerkungen**

- 2. Datenbanken: Zweck und Struktur**
 - 2.1 Von Dateninseln zu Datenbanken
 - 2.2 Aufbau von Datenbanksystemen

- 3. RDBMS – Entwicklung: vom Problem zum logischen Datenmodell**
 - 3.1 Das Entity – Relationship – Modell (ERM)
 - 3.2 Didaktische Randbemerkungen zur Systemanalyse
 - 3.3 Das relationale Datenmodell
 - 3.4 Die Normalisierung

Manfred Sommer
Universität Bielefeld

1. EDV, Betriebswirtschaftslehre und Sozialökonomie: einige einleitende Bemerkungen

Lassen Sie mich einleitend einige Grundsatzfragen zum Verhältnis von Elektronischer Datenverarbeitung (EDV) und Betriebswirtschaftslehre (BWL) aufwerfen.

Erstens wäre methodologisch zu fragen, ob der EDV für die BWL eher eine *instrumentell-hilfswissenschaftliche* oder eine *paradigmatisch-strukturierende Funktion* beizumessen ist. Der von SCHEER (1987) gewählte Terminus der "EDV-orientierten Betriebswirtschaftslehre" deutet bereits auf die zweite Option hin: Die EDV-Orientierung soll die produktionsorientierte, die entscheidungsorientierte, die systemorientierte und die empirisch orientierte Betriebswirtschaftslehre ergänzen. Unmißverständlicher wird der paradigmatische Anspruch mit der Befürchtung artikuliert, die BWL könne sich der EDV-Orientierung dadurch entziehen, daß sie diese an die Spezialdisziplin "Wirtschaftsinformatik" abtrete – frei nach dem wissenschaftssoziologischen Ausdifferenzierungsmotto "divide et ignora". Dies hätte zur Folge, daß die Stoffinhalte der Allgemeinen und der speziellen Betriebswirtschaftslehre(n) unverändert blieben, was aber – so SCHEER – nicht der Fall sein dürfe. Die hier angedeutete Kontroverse um die "Informatisierung der Betriebswirtschaftslehre" (vgl. auch MÜLLER-MEERBACH 1981) erinnert an die langjährige Diskussion um die "Mathematisierung der Wirtschaftswissenschaften".

Zweitens wäre zu fragen nach *Gemeinsamkeiten und Unterschieden* zwischen EDV-orientierter BWL, informationsorientierter BWL, Wirtschaftsinformatik, Betriebsinformatik und – wie zu ergänzen wäre – der Verwaltungsinformatik. Da sich die unter diesen Labeln firmierenden wissenschaftlichen Bemühungen, die um den Kern der Gestaltung computergestützter Informationssysteme kreisen, in einem Prozeß ständiger Neuerungen und Akzentverschiebungen befinden, ist mit einer auch nur mittelfristig gültigen Taxonomie gar nicht zu rechnen. Um nur ein Beispiel zu nennen: in dem Grundlagenlehrbuch über "Datenverarbeitung in der Unternehmung" von GROCHLA und MELLER aus dem Jahr 1974 sind weder das heute nicht mehr wegzudenkende Teilgebiet des "Personal Computing" noch die heute erst am Anfang ihrer Bedeutungsentfaltung stehenden "Expertensysteme" zu finden. Auch wenn man davon absieht, daß gerade auch in der EDV nicht jede terminologische Neuschöpfung Ausdruck einer theoretischen oder anwendungskonzeptionellen Innovation ist, so sind doch die nachhaltigen Veränderungen durch das Vordringen der Mikrocomputer nicht zu bestreiten.

Die beiden erstgenannten Fragenkomplexe stellen sich m.E. für die EDV-Ausbildung an jedem wirtschaftswissenschaftlichen Fachbereich. Nur als Fußnote kann ich hier anmerken, daß sie sehr ähnlich auch in geistes- und sozialwissenschaftlichen Fakultäten diskutiert werden, wenn es darum geht, die "richtige" Integrationsform informationswissenschaftlicher Inhalte in die Fachdisziplin zu finden. Die sogenannte "Computerlinguistik" als wechselseitige Durchdringung von linguistischer und KI-Forschung ist hierfür nur ein Beispiel. Meine dritte Frage zielt aber auf etwas anderes, HWP-Spezifisches: Was heißt *EDV-Orientierung in einem sozialökonomischen Studiengang*, der die BWL, VWL, Soziologie und Rechtswissenschaft interdisziplinär einzubinden gedenkt? Stichwortartig lassen sich vielfältige Querverbindungen benennen:

1. Die Verwendung ökonometrischer Programmpakete (TROLL, IAS, EPS etc.) in der EDV-gestützten empirischen Wirtschaftsforschung.
2. Analoges gilt für SPSS als EDV-Werkzeug der empirischen Sozialforschung, die sich ihrerseits durchaus, z.B. bei der Marktforschung, in die BWL verlängern läßt.

Nicht nur methodische, sondern auch inhaltliche Interdependenzen sind zu beachten. Auch hier müssen Kürzel genügen, um die weitgespannten Themenfelder anzudeuten. So ist

3. gesamtwirtschaftlich nach den Beschäftigungseffekten "neuer Technologien" im allgemeinen und der Informations- und Kommunikationstechnologien im besonderen zu fragen.
4. Mikroökonomisch und mikrosoziologisch sind arbeitsorganisatorische und qualifikatorische Implikationen dieser Technologien zu thematisieren. Betriebswirtschaftliche Überlegungen, die Datenintegration als ersten Schritt auf dem Weg zu einer die Arbeitsteilung partiell wieder aufhebenden Funktionsintegration zu nehmen (vom Verrichtungsprinzip zur objektbezogenen Aufbauorganisation) können mit neueren arbeitssoziologischen Studien zum Zusammenhang von Technik- und Organisationsentwicklung konfrontiert werden ("Neue Produktionskonzepte", "Verwaltungsautomation" etc.).
5. Rechtsinformatische Bezüge bestehen u.a. beim durch das "Recht auf informationelle Selbstbestimmung" grundrechtlich untermauerten Datenschutz, aber auch in zivilrechtlichen Fragen wie dem Gewährleistungsrecht bei fehlerhafter Software oder falschen Datenbankauskünften.

Diese Liste kann nicht und muß nicht "vollständig" sein, da an dieser Hochschule keine sozial- und wirtschaftswissenschaftlich rundumerneuterten Informatiker ausgebildet werden sollen. EDV-Orientierung in einem sozialökonomischen Studiengang muß sich m.E. aber die Aufgabe stellen, im Sinne exemplarischer Interdisziplinarität auch informatisches Zusammenhangswissen zu vermitteln.

2. Datenbanken: Zweck und Struktur

"Die gegenwärtig sichtbaren wesentlichen Entwicklungs- und Anwendungstrends der EDV betreffen den verstärkten Einsatz von Datenbanksystemen und Dialogverarbeitung, die Entwicklung benutzernaher Sprachen, graphische Datenverarbeitung, die Vernetzung von unterschiedlichen Hardwaresystemen, die Entwicklung leistungsfähiger Mikrocomputer und EDV-gestützter Steuerungssysteme für Fertigungsanlagen. Von den sog. "Neuen Medien", insb. Bildschirmtext (BTX) und der Bürotechnologie gehen Impulse aus zur Einrichtung lokaler Netzwerke sowie zur Kommunikation über Electronic Mail und Electronic Conferencing" (SCHEER 1987).

"Datenbanksysteme, in den frühen siebziger Jahren noch als Exoten bestaunt oder als Beweis für fortschrittliche Unternehmensführung vorgewiesen, sind heute ein selbstverständliches Hilfsmittel der betrieblichen Organisation und der Verwaltung geworden. Finanzwesen, Fertigungsplanung, Lagerführung, Vertrieb, Handel, Banken, Versicherungen, sie alle sind auf das reibungslose Funktionieren der Verwaltung sehr großer Datenbestände angewiesen. Die Dienstleistungen der öffentlichen Hand bei Altersversorgung, Krankheitsfürsorge, Meldewesen, Verkehrsplanung oder Finanz- und Steuerverwaltung schärfen auch beim einzelnen Bürger das Bewußtsein für die Bedeutung, aber auch für die Gefahren großer Datenbanken. Neue Formen der Kommunikation wie Bildschirmtext bringen den Bürger sogar erstmals unmittelbar mit Datenbanken in Berührung. Kurz, Datenbanken sind genauso wie die Kommunikationsmedien inzwischen ein fester Bestandteil der technischen Umwelt unserer Gesellschaft geworden" (LOCKEMANN/SCHMIDT 1987).

2.1 Von Dateninseln zu Datenbanken

In der Anfangsphase der EDV standen bei der Lösung von Anwendungsproblemen die Anwendungsprogramme im Vordergrund, für die die Daten jeweils individuell in der jeweils benötigten Form bereitgestellt wurden. Diese starke *Einbindung der Datenorganisation in die Anwendungsprogramme* bewirkte

- a) eine mangelnde Flexibilität der Anwendungsprogramme (*Datenabhängigkeit*)
- b) eine Vielzahl von sich inhaltlich überschneidenden Dateien (*Datenredundanz*), die ihrerseits
- c) *erhöhte Kosten* für mehrfache Erfassung, Speicherung und Pflege der Daten mit sich brachte, ohne daß
- d) die *Integrität* und *Konsistenz* zwischen den "Dateninseln" hätte sichergestellt werden können (SCHEER 1987, GRAWE 1987, BLASER 1986).

Erschwert wird die klassische Dateiverwaltung noch dadurch, daß ihre Speicherungs- und Zugriffstechniken (direkte Adressierung, sequentielle Speicherung, indexsequentielle Speicherung) komplexe Datenstrukturen gar nicht redundanzfrei speichern können. So werden bei konventioneller Dateiverarbeitung z.B. die bei der Auftragsbearbeitung verwendeten Stammdaten aus den Kunden- und Artikeldateien sowie die erzeugte Auftragsdatei in einem statistischen Anwendungsprogramm zumindest teilweise in anderer Sortierung erneut gehalten (Anhang 1). Werden die Stammdaten zudem in der Auftragsbearbeitung einerseits und der Tourenplanung andererseits getrennt erfaßt (Anhang 2), können Inkompabilitäten z.B. hinsichtlich des Formats auftreten, die durch ein Bridge-Modul überbrückt werden müssen.

2.2 Aufbau von Datenbanksystemen

Konsequent lassen sich die genannten Schwierigkeiten nur angehen, wenn die Daten nicht mehr als Anhängsel der Programme, sondern unabhängig von einzelnen Anwendungen nach einem a priori erstellten Gesamtkonzept integriert verarbeitet werden (*Datenintegration*). Die Zusammenfassung inhaltlich verschiedener, aber zusammengehöriger Dateien (= Datenbestände) bezeichnet man als *Datenbank (DB)*, deren Aufbau, Kontrolle und Manipulation von einem betriebssystemnahen Programmsystem, dem sog. *Datenbankverwaltungssystem (DBMS)*, geleistet wird (Anhang 3 und 4). Hierzu zählen im einzelnen

- Methoden und Werkzeuge zur Erstellung des konzeptionellen Modells¹⁾, der externen und internen Schemata,
- die Datendefinitionssprachen (DDL) zur Festlegung des konzeptionellen Modells
- die Datenmanipulationssprache (DML) zum Einfügen, Ändern, Löschen und Abfragen der Daten

1) Konzeptionelles Modell = Konzeptionelles Schema.

- die Hilfsmittel zur Gewährleistung der Datensicherheit (Vollständigkeit und Korrektheit der Daten)
- die Hilfsmittel zur Gewährleistung des Datenschutzes (Schutz der Daten vor dem Zugriff unberechtigter Benutzer)
- das "Hochfahren" und "Herunterfahren" der Datenbank
- das job accounting
- das Importieren externer Datenbestände sowie das Exportieren u.ä.m. (HANSEN 1986).

Neben der in der Datenbank materialisierten physischen oder *internen Datensicht*²⁾ gehört zu einem Datenbanksystem die *externe Benutzersicht*³⁾, die den parametrischen und den gelegentlichen Endbenutzern eine benutzerfreundliche, auf das für sie wesentliche beschränkte Sicht des konzeptionellen Schemas gestattet. Der *parametrische Endbenutzer*, der klassischerweise auf der Sachbearbeiterebene angesiedelt wird (Lohnbüro, Finanzbuchhaltung, Bankschalter, Versicherungen, Meldeämter, Reisebüros etc.), kommuniziert mit der Datenbank über vorgefertigte Anwendungsprogramme, die er mit einem relativ beschränkten Kommandovorrat zur Ausführung bringt. Es handelt sich also stets um wohldefinierte, formalisierte und routinisierte Aufgaben, für die sich die Programmierung von Anwendungsmodulen lohnt⁴⁾. Historisch rangierte diese Rationalisierung von Massenvorgängen an erster Stelle bei der EDV – Einführung (*operative Datenverarbeitung*; MATHY 1987).

Der *gelegentliche Endbenutzer* oder "problem solver" kommuniziert mit der Datenbank über eine flexible Abfragesprache, die ihm die für die Lösung seiner wechselnden, schlecht strukturierten Problemstellungen erforderlichen Informationen ad hoc zur Verfügung stellt. Die interaktive Kommunikation ist hier typisch, da erste Antworten des DBMS zu Verschiebungen oder Erweiterungen der Problemstellung führen und weitere Nachfragen an die Datenbank auslösen können. Diese Nutzungsart dient also der strategischen Entscheidungsfindung und findet sich in den Stellen einer Organisation, die Entscheidungen vorbereiten bzw. selbst treffen (*dispositive Datenverarbeitung*; MATHY 1987).

2) Synonyme: physische Datenorganisation, interne Datenrepräsentation, internes Schema.

3) Synonyme: externes Schema, externe Datensicht.

4) Hierbei kann es sich sowohl um Massendatenverarbeitung im Batchbetrieb wie auch um dialogorientierte Einzelfallbearbeitung handeln. Relevantes Abgrenzungskriterium ist der administrative Standardcharakter der fest umrissenen Aufgabe.

Während die gelegentlichen Endbenutzer also ihre ad-hoc-Anfragen an das System selbst formulieren – dies zu beherrschen ist z.B. auch Gegenstand der Führungskräfteweiterbildung in PC-Seminaren – werden die Anwendungsprogramme der parametrischen Endbenutzer von der dritten Benutzergruppe, den *Anwendungsprogrammierern*, erstellt. Der *Datenbankadministrator* schließlich ist für alle mit der Handhabung des Datenbankverwaltungssystems verbundenen Funktionen verantwortlich.

Dieses klassische Modell des Datenbanksystems mit seiner idealtypischen Klassifizierung von vier Benutzergruppen trägt deutlich sichtbar den Stempel der *Mainframe-EDV*. Schon der Datenbankbetrieb auf Minirechnern unter UNIX lockert die Grenzen auf. Hier kann schon aus personalkapazitiven Gründen der Datenbankadministrator zugleich auch Anwendungsprogrammierer sein. Beim *Personal Computing* sind die organisatorischen Freiheitsgrade für die Aufgabenschneidung noch größer. Sie liegen zwischen den Extremen eines parametrischen Endbenutzers einerseits, der seinen portablen PC vorwiegend zur Datenaufnahme im Außendienst einsetzt, während ein PC-information-center des Betriebs die Informationsverdichtung und -auswertung samt der dafür erforderlichen Daten- und Programmpflege erledigt, und eines "Hybridbenutzers" andererseits, der alle vier Benutzergruppen in sich vereint und mit seinem PC wirklich allein steht ("stand alone"), wobei ihn die Integration seines Datenbanksystems in eine sog. *Integrierte Arbeitsplatz-Software* mit zusätzlicher Tabellenkalkulation, einer zumindest rudimentären Methodenbank, Businessgraphik und Textverarbeitung unterstützt (Anhang 5).

Es deutet sich also an, daß das klassische Modell der vier Benutzergruppen nicht nur durch neuere Hardwareentwicklungen – vor allem PC's und kleinere Mehrplatzsysteme – sondern auch durch die sowohl von dieser "neuen technologischen Basis" wie auch von der "Humanisierungsdiskussion" ausgelösten Veränderungstendenzen der Arbeitsorganisation aufgeweicht wird. Mit der *Funktionsintegration* und dem damit verbundenen "job enlargement" und "job enrichment" wird der ausschließlich parametrische Benutzer obsolet. Vielmehr wird er seine Anwendungsprogramme selbst programmieren und/oder die Informationen von Fall zu Fall so abrufen, wie es komplexere Aufgabenstellungen erfordern.

3. RDBMS – Entwicklung: vom Problem zum logischen Datenmodell

Dreh- und Angelpunkt eines Datenbanksystems ist das *Konzeptionelle Datenmodell*, das erstens ein Realweltssystem durch den Vorgang der Modellierung via

- Abstraktion
- Deskription
- Klassifikation

in ein bestimmtes *Informationsstrukturmodell* abbildet, das seinerseits mit Blick auf das *logische Datenmodell* eines spezifischen DBMS formalisiert wird⁵⁾.

3.1 Das Entity – Relationship – Modell (ERM)

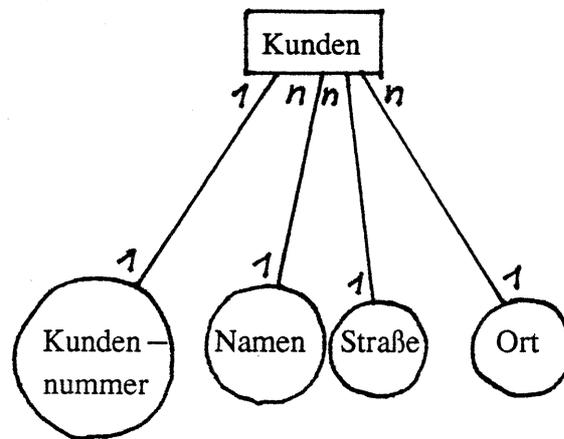
Bei dem mit jeder Modellbildung verbundenen Abstraktionsprozeß werden vier methodologische Aspekte m.E. gerade in der EDV – Literatur nicht immer hinreichend betont. Ein mentales Modell eines Realitätsausschnitts kann nicht auf "Totalität" zielen, sondern muß selektiv sein (Anhang 6). Wovon abstrahiert wird, ist

1. *zweckgebunden*
2. *erfahrensbedingt* (Personen – und Umweltbedingtheit)
3. *interessenbedingt* (Struktur – und Zielbedingtheit) und
4. bei computergestützten Modellen auch *hardware – /softwarebedingt*.

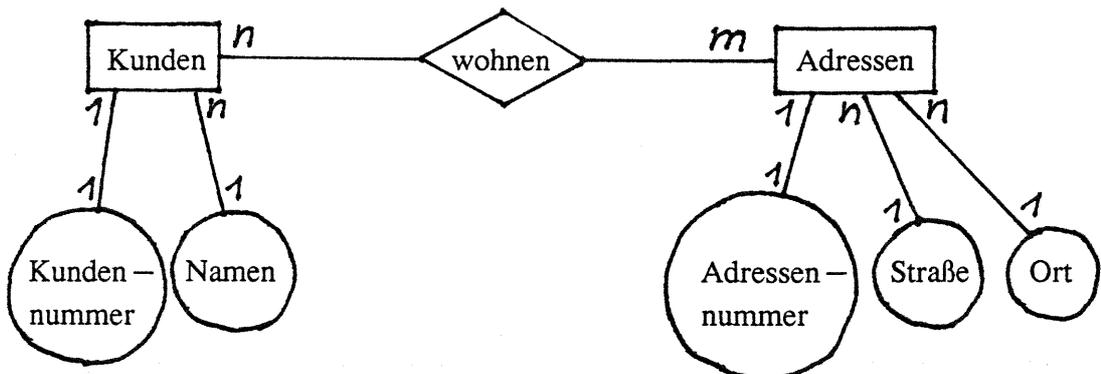
Daraus folgt unmittelbar, daß es zu jedem Realsystem nicht nur ein konzeptionelles Datenmodell, sondern unbestimmt viele gibt – eine für methodologisch geschulte Modell- und Theoriebildner nicht neue Erkenntnis. Diese möglicherweise konkurrierenden Datenmodelle lassen sich nicht in "richtige" und "falsche" einteilen – auch das würde einem "raffinierten Falsifikationisten" gar nicht erst in den Sinn kommen – sondern höchstens in "geeigneter" und "wenig geeignete", ohne daß ein eindeutiges Effizienzkriterium angegeben werden kann. Dies folgt allein schon aus der Zweckgebundenheit.

5) Bei keiner Komponente der Datenbanksystemarchitektur herrscht eine größere babylonische Sprachverwirrung als hier: logische Datenorganisation, logische Datenstruktur, logischer Datenbankentwurf (SCHWEPPE 1987), logisches Datenmodell (BLASER 1986), konzeptionelles Modell (HANSEN 1987), konzeptionelle Datensicht, konzeptionelles Schema (BORG 1987, BLASER 1986).

Eine Form der Deskription und Klassifikation und damit ein spezifisches Informationsstrukturmodell ist das *Entity-Relationship-Modell (ERM)*. Es besteht aus drei Elementen. *Entities* (Objekte) sind reale oder abstrakte Dinge, wie Kunden, Artikel, Aufträge, Beschäftigte etc., die zu Entitytypen zusammengefaßt werden, deren Eigenschaften durch *Attribute* beschrieben und deren logische Verknüpfungen als *Relationen* gefaßt werden (SCHEEER 1987; SCHWEPPE 1987). Die Unterscheidung zwischen Entitytyp und Attributen ist ebenfalls zweckgebunden. Zwischen Entities und Attributen soll höchstens eine $n:1$ -Beziehung bestehen, aber keine $n:m$ -Beziehung. Dies sei am Standardbeispiel von Adressen illustriert.



Mehrere Kunden können den gleichen Namen haben, in derselben Straße oder erst recht im selben Ort wohnen ($n:1$ -Relation). Deshalb hat keines dieser Attribute eine identifizierende Eigenschaft, weshalb eine Kundennummer als *Schlüsselattribut* eingeführt wird ($1:1$ -Relation). Wenn ein Kunde aber mehrere Adressen (Straße, Wohnort) haben kann (mehrere Wohnsitze oder Privat- und Dienstadresse), ist es sinnvoll, diese Attribute zu einem eigenen Entitätstyp zusammenzufassen. Zwischen den Entitätstypen Kunden und Adressen besteht dann eine $m:n$ -Relation.



Während zur Identifizierung eines Objekts *ein* Schlüsselattribut ausreicht, sind zur Identifizierung einer Relation *mindestens zwei* Schlüsselattribute erforderlich (SCHEER 1987), unabhängig davon ob es sich um eine 1:n-, n:1- oder n:m-Relation handelt.

In meinem Seminar über "EDV-Anwendungen in Erziehungs-, Beratungs- und Bildungseinrichtungen" in diesem Semester haben wir viel Zeit auf die Erarbeitung von Informationsstrukturen und logischen Datenmodellen für zwei Beispiele verwendet und dabei verschiedene Zugangsweisen gewählt. Am Beispiel von *Kindertagesstätten*, deren Trägervereine dem Deutschen Paritätischen Wohlfahrtsverband (DPWV) angeschlossen sind, haben wir die Informationsstruktur jenseits der Finanz- und Lohnbuchhaltung aus den dort verwendeten Formularen (z.B. Anhang 7) eruiert⁶⁾. Beim zweiten Beispiel der *Medienstelle* unserer Fakultät haben die Seminarteilnehmer die relevanten Daten, ihre Verknüpfung und ihren Verarbeitungszweck durch ein Expertengespräch mit dem für die Medienstelle Verantwortlichen erhoben. Interessant war hierbei, daß die Aufbereitungsform der Ausgangsinformationen den ersten Eindruck von der Systemkomplexität psychologisch offensichtlich beeinflusst. Das im Vergleich zum "schein-strukturierten" Formularwesen eher "fuzzy" wirkende Gesprächsergebnis führte bei der Mehrzahl der Seminarteilnehmer zu der spontanen Vermutung, daß zweite Beispiel sei wesentlich "komplexer" als das erste – ein Irrtum, wie bereits der Vergleich der respektiven Entity-Relationship-Modelle zeigt (Anhang 8).

3.2 Didaktische Randbemerkungen zur Systemanalyse

Ohne Systemanalyse bleibt der Entwurf eines Datenbanksystems – wie der eines jeden computergestützten Informationssystems (SCHEER 1987) – konzeptionslos. Nun kann man sich auf den Standpunkt stellen, die Systemanalyse sei – wie in dem oben besprochenen Modell der Mainframe-Datenbanken – vor allem Sache der IS-Spezialisten, also primär der Informatiker; Studenten der Betriebswirtschaftslehre als spätere "gelegentliche Benutzer" bräuchten sich hiermit im Studium nicht zu

6) Die Lohnbuchhaltung wird vom Diakonischen Rechenzentrum Bethel kostengünstig erledigt; die BAT-Pflege ist selbst für größere Organisationen zu aufwendig. Die EDV-gestützte Lohnbuchhaltung ist deshalb kein Dezentralisierungskandidat. Für die Finanzbuchhaltung hat der Landesverband NRW des DPWV einen Entwicklungsauftrag für ein ORACLE-Standardprogramm erteilt. Dieses wird später an die örtlichen und fachlichen Gegebenheiten (Kontenrahmen) der einzelnen Mitgliedsorganisationen angepaßt und soll in den Kreisgruppen, die ORACLE-Datenbanksysteme haben, eingesetzt werden.

beschäftigen. Ich teile diese Auffassung nicht und halte eine exemplarische Beschäftigung mit systemanalytischen Problemen für eine EDV-Hochschulausbildung nicht nur bei Betriebswirten, sondern bei allen Wirtschafts- und Sozialwissenschaftlern für ausgesprochen wichtig. Sie vermittelt die Erkenntnis,

- daß die Entwicklung von Informationssystemen längst vor dem Einschalten des Computers beginnen muß,
- daß ihr Erfolg ganz wesentlich von der Kenntnis der bisherigen Organisationsstruktur und Informationsflüsse, ihrer Stärken und Schwächen und den Verbesserungsvorstellungen der Beteiligten abhängt und
- daß die im IS abzubildende Realität sich beim ersten Eintauchen wesentlich komplexer darstellt als a priori vermutet, um mit fortschreitender Strukturierung wieder etwas an Einfachheit zu gewinnen; gerade in dieser Phase kommt der im engeren Sinne fachwissenschaftliche spill-over einer EDV-Ausbildung zum Vorschein (Anhang 9).
- last not least fördert sie die Einsicht, daß die Entwicklung computergestützter Informationssysteme mit einer Analyse des gegenwärtigen und absehbaren Informationsbedarfs beginnen und möglichst wenig von subjektiven Vorlieben für bestimmte Betriebssysteme, Programmiersprachen, Softwarepakete oder Rechnerkategorien präformiert werden darf.

3.3 Das relationale Datenmodell

Grundsätzlich kann jedes Entity-Relationship-Modell mit einem

- hierarchischen,
- netzförmigen oder
- relationalen Datenmodell

computergerecht beschrieben werden. Postrelationale Datenmodelle (z.B. das NF²-Modell) sollen eine Brücke zwischen Datenbanksystemen (Verwaltung großer Datenmengen) und Expertensystemen (Repräsentation von Regelwissen) bauen, liegen aber z.Zt. noch im Forschungs- und nicht im Anwendungsbereich (SCHWEPPE 1987; BLASER 1986).

Obwohl die Frage nach dem richtigen Datenmodell wissenschaftlich nicht entscheidbar ist, sprechen folgende Gesichtspunkte für das auf CODD (1970) zurückgehende relationale Datenmodell, das man sich als eine Menge von Entitytypen und Relationen beschreibenden Tabellen vorstellen kann, deren Attribute (einschließlich der identifizierenden Schlüsselattribute) die Spalten bilden und deren einzelne Entities zeilenweise als Datensätze eingetragen werden (vgl. Anhang 10 zur Tabellen – Terminologie):

1. *Strukturflexibilität und Datenunabhängigkeit*

Während bei netzwerkartigen Datenbanksystemen die Beziehungen zwischen den Datenelementen in der Datenbankstruktur mit *Zeigern* vorab festgelegt werden (Starrheit, erheblicher Strukturveränderungsaufwand), werden bei relationalen Datenbanksystemen die Beziehungen zwischen den Entitytypen genauso wie die Beziehungen zu den beschreibenden Nicht – Schlüssel – Attributen durch *Tabellen* festgelegt (Überflüssigkeit von *Zeigern*). Strukturveränderungen reduzieren sich damit auf Tabellenänderungen (SCHWEPPE 1987), neue Relationen zu alten oder neuen Objekttypen werden durch neue Tabellen ergänzt.

2. *Benutzerspezifische Datensichten*

Da sich neben den in der Datenbank eingetragenen Tabellen auch virtuelle Tabellen (Views) definieren lassen, die dem jeweiligen Benutzer eine für ihn zweckadäquate Sicht auf die Datenbasis ermöglicht, werden gleichzeitig die Benutzerfreundlichkeit und der Datenschutz verbessert.

3. *Solide mathematische Basis*

Mit der Relationenalgebra werden neben den Mengenoperationen Vereinigung, Durchschnitt, Differenz und Kreuzprodukt auch die relationalen Operationen Auswahl, Projektion und Verbindung zur Datenmanipulation zur Verfügung gestellt.

4. *Mengenorientierte Datenmanipulationssprachen*

Während traditionelle Anwendungsprogramme "Satz für Satz" vorgehen, arbeiten relationale Sprachen mengenorientiert mit der "multiple – record – at – a – time" – Logik. Ihre Nonprozeduralität orientiert sich am "Was ist zu tun" statt am "Wie ist es zu tun" und trägt wesentlich zur Benutzerfreundlichkeit bei. Die zum Standard avancierte *Structured Query Language (SQL)* ist zudem so natürlich – sprachlich – wenn auch in englisch – gehalten, daß sie Nichtmathematikern leichter zugänglich ist.

5. *Intuitive Anschaulichkeit*

der tabellarischen Darstellung – gerade für Ökonomen

6. ERM – Bezug

Das relationale Modell und insbesondere die Normalformenlehre knüpft nahtlos an die Entwurfsprinzipien des Entity – Relationship – Modells an (SCHEER 1987).

3.4 Die Normalisierung

Um die wesentlichen funktionalen Vorteile des Relationenmodells zur Geltung zu bringen, muß das relationale Datenmodell normalisiert werden. Was ist nun unter Normalisierung zu verstehen? Mit den Worten von Chris DATE: "Basically, normalization helps you structure your thinking ... In essence, normalization tells you how to derive better tables out of tables you have already. It doesn't tell you where those original tables come from. You need some other methodology for that" (DATE 1986). Eine solche andere Methodologie habe ich mit dem Entity – Relationship – Modell bereits beschrieben.

Es bleibt zu fragen, was die "besseren Tabellen" von den "schlechteren Ausgangstabellen" unterscheidet. Ein Bestellformular beispielsweise (Anhang 11) läßt sich leicht in eine einzige Tabelle überführen (Anhang 12), die eine *unnormalisierte Relation* darstellt (Anhang 13). Jede Bestellung bildet einen Datensatz mit variabler Satzlänge, da sich eine Bestellung auf unterschiedlich viele Produkte beziehen kann.

Um Datensätze mit fester Satzlänge zu erhalten, müssen die sich auf die gesamte Bestellung beziehenden Daten für jedes Produkt erneut mit abgespeichert werden. Dadurch entsteht eine erhebliche Redundanz hinsichtlich der produktunabhängigen Daten, vor allem soweit sie den Entitytyp Lieferant betreffen, was den Änderungsdienst bei diesen Daten aufwendig und fehleranfällig macht (UPDATE – Problem). Das 1. Gebot der Normalisierung lautet deshalb, solche Wiederholungsgruppen zu entfernen, indem die eingeschachtelte Relation – hier die Produktangaben – selbstständig werden. Im Anhang 14 wird die so entstandene *erste Normalform (1NF)* für die Beispiele Bestellwesen und Medienstelle parallel dargestellt. Wie unschwer zu erkennen ist, erleichtert eine Systemanalyse mit Hilfe der ERM – Heuristik den Einstieg in die erste Normalform, da an den Kanten ihrer Relationen bereits die Wiederholungsgruppeneigenschaft zwischen den Objekttypen vermerkt ist.

Jede Zeile einer Relation, also jeder Datensatz, muß durch ein Attribut bzw. eine Attributkombination eindeutig identifizierbar sein, die man als *Schlüssel* bezeichnet und durch Unterstreichung hervorhebt. Jeder Schlüsselwert darf deshalb in einer Tabelle nur einmal vorkommen. Dies ist mit ganzzahligen, einfachen Schlüsseln wie

Bestellnummern, Produktnummern, Cassetten-Nummern, Benutzernummern – sog. Primärschlüsseln –, einfacher zu realisieren als mit aus Nicht-Schlüsselattributen zusammengesetzten Schlüsseln (unübersichtlich, unvollständig bei fehlenden Attributwerten, nicht mehr identifizierend bei späterem Wegfall ganzer Attribute).

Durch die Auslagerung von Wiederholungsgruppen entstehen neue Tabellen mit notwendigerweise zusammengesetzten Schlüsseln. In der Tabelle **PRODUKT-ANGABEN** muß neben der Produkt-Nummer auch die "entity-fremde" Bestell-Nummer mit herübergenommen werden, um den logischen Zusammenhang zwischen den beiden Objekttypen zu erhalten. Analoges gilt für die zusammengesetzten Schlüssel Cassetten-Nummer, Sendungs-Nummer und Benutzer-Nummer, Cassetten-Nummer im Medienstellen-Beispiel.

Nach der Beseitigung von Entity-Redundanzen durch die erste Normalform werden in weiteren Normalisierungsschritten Attribut-Redundanzen ausgeschaltet. In der *zweiten Normalform (2NF)* werden zunächst alle Attribute entfernt, die nicht vom Gesamtschlüssel einer Tabelle voll funktional abhängen (Anhang 15). Naheliegenderweise ist dies gerade bei allen durch den ersten Normalisierungsschritt entstandenen Tabellen zu prüfen, weil deren Datensätze ja gerade durch solche Gesamtschlüssel identifiziert werden. Während die Produktmenge und damit auch der Produkt-Gesamtpreis auftragsspezifisch sind, hängen die Produkt-Bezeichnung und der Produkt-Preis nur von der Produkt-Nummer und damit nicht vom Gesamtschlüssel ab. Beim Produktpreis könnte man freilich auch eine Abhängigkeit vom Gesamtschlüssel bejahen, wenn dieser von auftragsspezifischen Konditionen wie z.B. Mengenrabatten abhängt. Sauberer wäre es aber auch dann, eine entsprechende Rabattstaffel durch ein gesondertes Attribut auszudrücken⁷⁾. Gerade hier wird sehr deutlich, daß funktionale Abhängigkeit eine Eigenschaft des Gegenstandsbereichs und seiner Perzeption ist, die Normalisierung also ein formal- und sachlogisches Verfahren ist.

Die Ausgliederung einer **PRODUKT**-Tabelle in der 2. Normalform überwindet drei grundlegende Datenmanipulationsprobleme der 1. Normalform

- a) Informationen über ein Produkt könnten nicht eingefügt werden, bevor es hierfür eine Bestellung gibt (INSERT-Problem),

7) Anders wäre es, wenn es statt um die Beschaffungs- um die Absatzseite des Betriebes ginge. Hier wäre eine funktionale Abhängigkeit des Absatzpreises von der Kunden-Nummer und der Produkt-Nummer Reflex einer Preisdifferenzierung.

- b) wenn ein Bestellvorgang gelöscht wird, könnte die Produktinformation mit verloren gehen, sofern sie nur in dieser Bestellung auftrat (DELETE – Problem),
- c) jede Änderung z.B. der redundanten Produktbezeichnung müsste wiederholt durchgeführt werden (UPDATE – Problem).

In dem Beispiel der Medienstelle wird die 2. Normalform wegen der Vielzahl reinesendungsspezifischer Attribute noch anschaulicher.

Ähnliche Probleme wie die soeben diskutierten treten auch dann auf, wenn einzelne Attribute von anderen Nicht–Schlüsseldaten voll funktional abhängen. Auf diese Möglichkeit hin sind vor allem die Tabellen ohne zusammengesetzte Schlüssel zu untersuchen, die nach dem ersten Normalisierungsschritt zugleich schon in der 2. Normalform waren (1NF = 2NF). Die *dritte Normalform* fordert also zwecks weiterer Redundanzminimierung die Eliminierung von Attributen, die von anderen Nicht–Schlüsselattributen voll funktional abhängen, m.a.W. von transitiven Abhängigkeiten zwischen Nicht–Schlüsselattributen. Im Beschaffungsbeispiel ist es die Abhängigkeit des Lieferantennamens und seiner Adresse von der Lieferer–Nummer, am Beispiel der Medienstelle etwas gekünstelt die Abhängigkeit der Sendewoche vom Sendedatum (Anhang 16).

Fassen wir zusammen: die Systemanalyse führt in einer ersten Modellierungsphase zu einer Klassifikation von Objekttypen, ihrer Beschreibung durch zweckrelevante Attribute und einer Identifikation der zwischen den Objekttypen bestehenden Abhängigkeiten. Diese Abhängigkeiten können im Entity–Relationship–Modell hinsichtlich der vier möglichen Beziehungstypen (1:1, n:1, 1:n, m:n) exakter definiert werden, womit zugleich der Einstieg in die Normalisierung geschafft wird, die ihrerseits der Redundanzminimierung durch Strukturzerlegung dient.

In der dritten Normalform beschränkt sich die Redundanz dann auf die zwei– oder mehrfache Speicherung von Schlüsselattributen in den Tabellen, die Beziehungen zwischen den Entitytypen festhalten. Man kann sie als *relationale Minimalredundanz* bezeichnen. Am Ende des Normalisierungsprozesses steht eine Menge von Tabellen, die – bei zusätzlicher Angabe der Datentypen und –länge – direkt als leere Tabellen in die Datenbank eingetragen werden. Der CREATE TABLE–Befehl von SQL übernimmt diese Aufgabe einer Datendefinitionssprache (DDL).

Literatur

BLASER, A. (1986)

Datenbanksysteme aus Benutzersicht – Stand und Entwicklungstendenzen. In: Schulz, A. (Hrsg.): Die Zukunft der Informationssysteme. Lehren der 80er Jahre. Berlin–Heidelberg–New York–London–Paris–Tokyo, 1986, S. 2–37.

BORG, B. (1987)

Didaktisch–methodische Aspekte des Einsatzes von Datenbanksystemen. In: LOG IN, 1987, Jg. 7, H. 5/6, S. 30–35.

CODD, E.F. (1970)

A Relational Model of Data for Large Shared Data Banks. In: Communications of the ACM, 1970, pp. 377–387.

DATE, C.J. (1986)

Relational Database. Selected Writings. Reading (Mass.) 1986.

DIEMER, W. (1987)

Relationale Datenbanken kurz und bündig. Sicherheit, Integrität und Unabhängigkeit in der Datenverwaltung. Würzburg 1987 (2. Aufl.), Vogel–Buchverlag.

DWORATSCHEK, S. (1986)

Grundlagen der Datenverarbeitung. Berlin–New York 1986, 7. Aufl. (Walter de Gruyter).

EILERS, H./JANSEN, W./DE VOLDER, H. (1986)

SQL in der Praxis. Bonn 1986.

GRAWE, H. (1987)

Effizientes Marketing mit Personal Computing. Marketing–Management, Softwarebewertung, PC–Datenbanken. Wiesbaden 1987 (Gabler).

GROCHLA, E./MELLER, F. (1974)

Datenverarbeitung in der Unternehmung, Bd. 1: Grundlagen (rororo studium 60). Reinbek 1974.

HANSEN, H.R. (1986)

Wirtschaftsinformatik I. Einführung in die betriebliche Datenverarbeitung (UTB 802). Stuttgart 1986, 5. Aufl. (Gustav Fischer Verlag).

LOCKEMANN, P.C./SCHMIDT, J.W. (HRSG.) (1987)

Datenbankhandbuch. Berlin–Heidelberg–New York–London–Paris. Tokyo 1987.

MATHY, G. (1987)

Informatik – Strategie und Relationale Datenbanken. In: Information Management, 1987, H. 2, S. 6–16.

MÜLLER – MERBACH, H. (1981)

Betriebsinformatik am Ende? In: Zeitschrift für Betriebswirtschaft 1981, H. 3, S. 274–282.

SCHEER, A.W. (1987)

EDV – orientierte Betriebswirtschaftslehre. Berlin – Heidelberg – New York – London – Paris – Tokyo, 1987 (3. Aufl.).

SCHNEIDER, H.J. (1986)

Formale Gestaltungsaspekte in der Systementwicklung. Eine einheitliche Beschreibungs- und Entwicklungsmethode für DV – Problemlösungen ist nicht in Sichtweite. In: Handbuch der Modernen Datenverarbeitung, 1986, H. 130, 23. Jg., S. 3–17.

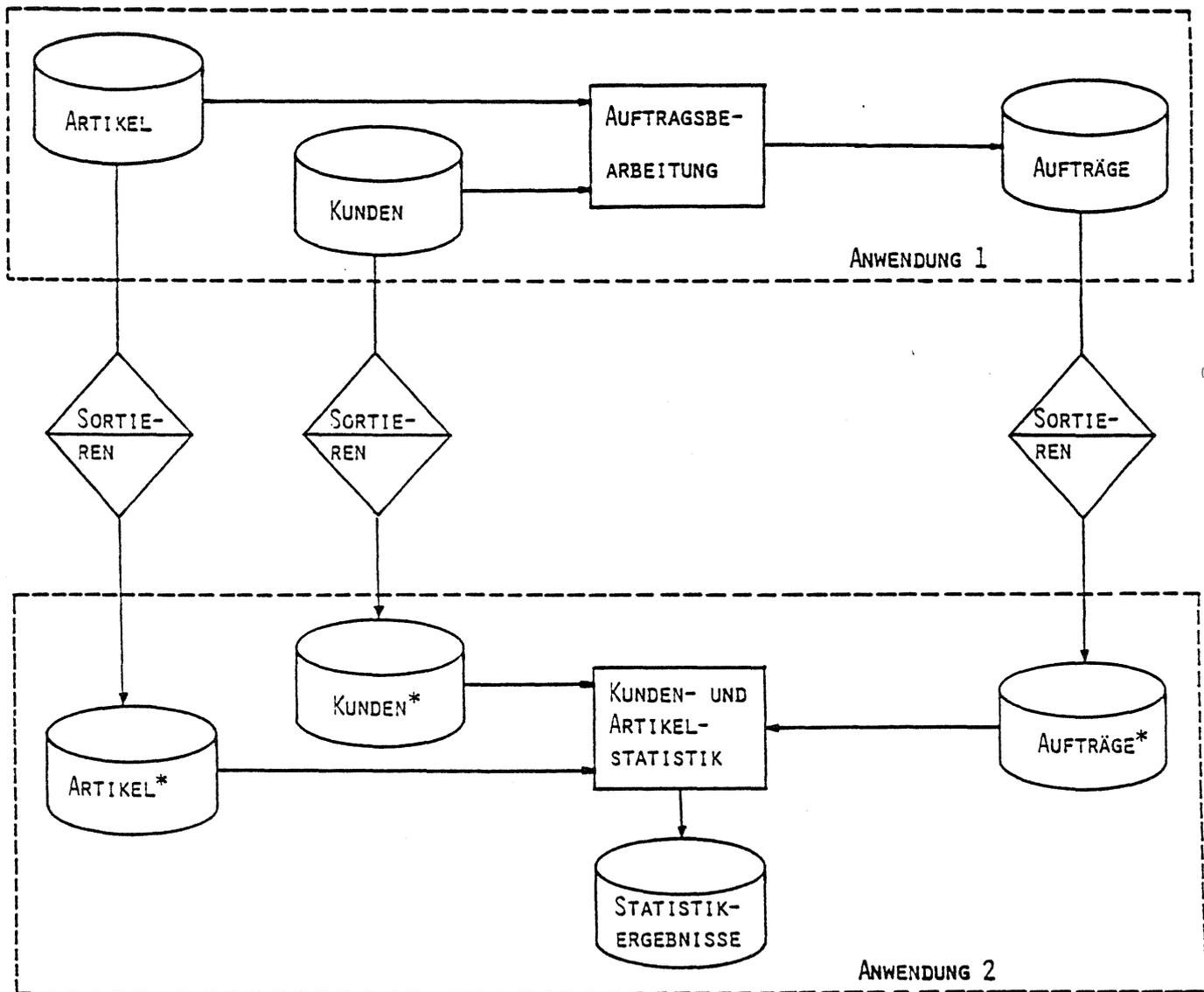


Abb. 2.1.01: Konventionelle Dateiverarbeitung

Quelle: A.-W. Scherer, EDV - orientierte Betriebswirtschaftslehre (1987:11)

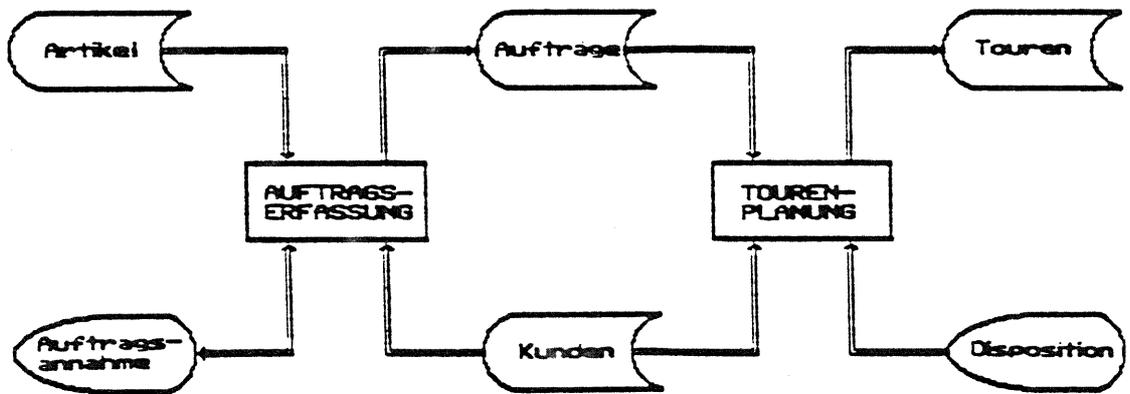
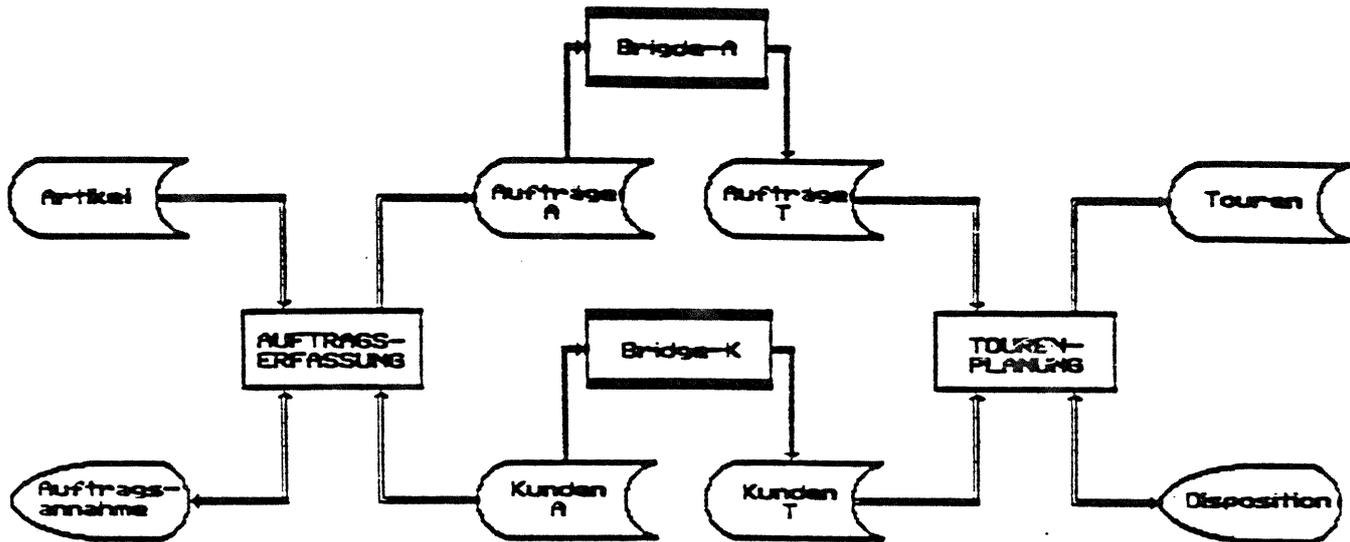


Abb. 4.1.1.A: Insellösung (oben) und Integrierte Lösung (unten)

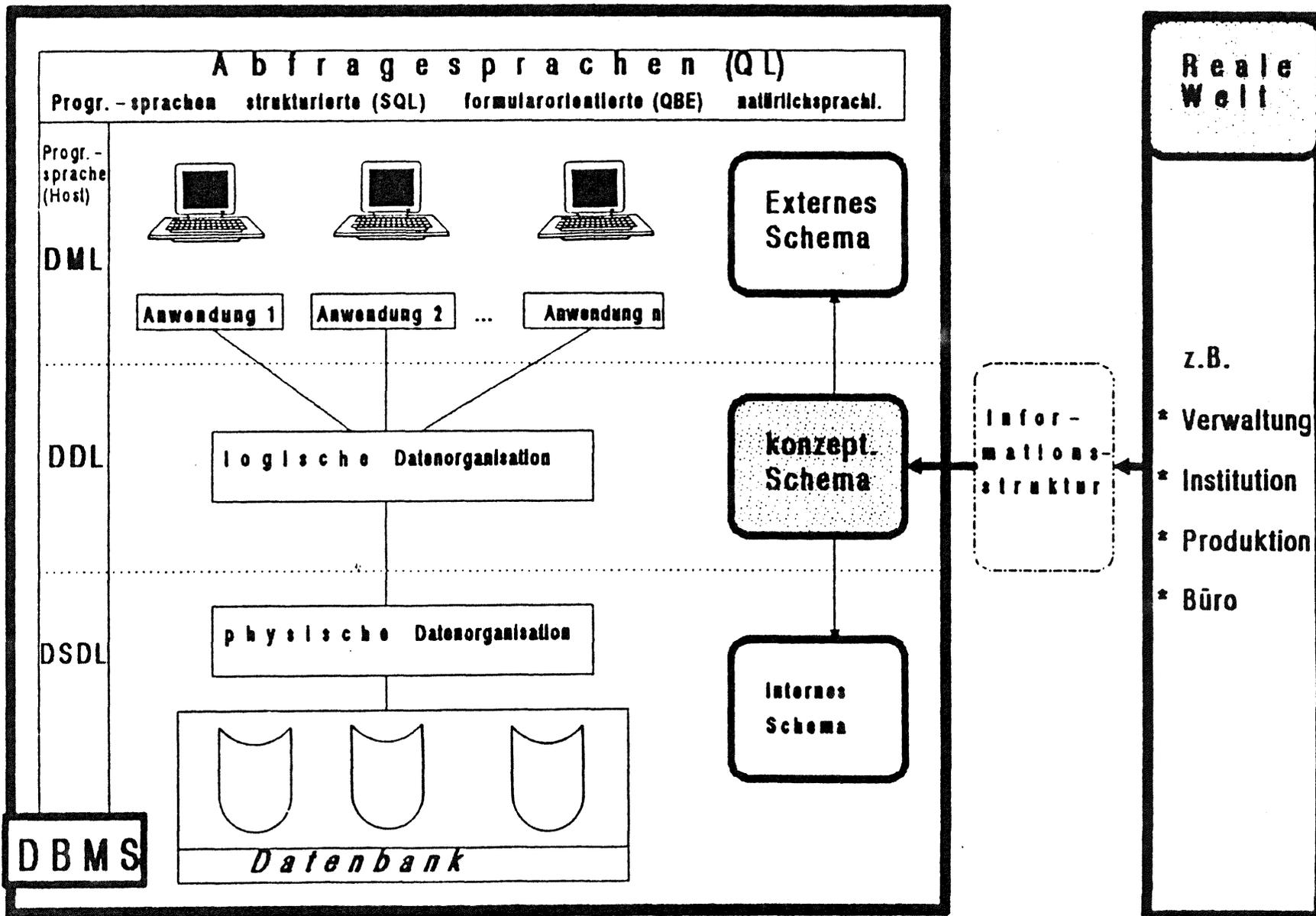
Quelle: H. Grawe, Effizientes Marketing mit Personal Computing (1987:89)

Anwendungs-
programmierer

gelegentliche
Benutzer

parametrische#
Benutzer

Datenbank-
verwalter



Quelle: B. Borg, Didaktisch - methodische Aspekte des Einsatzes von
Datenbanksystemen (1987:32)

Bild 5: Struktur eines Datenbanksystems.

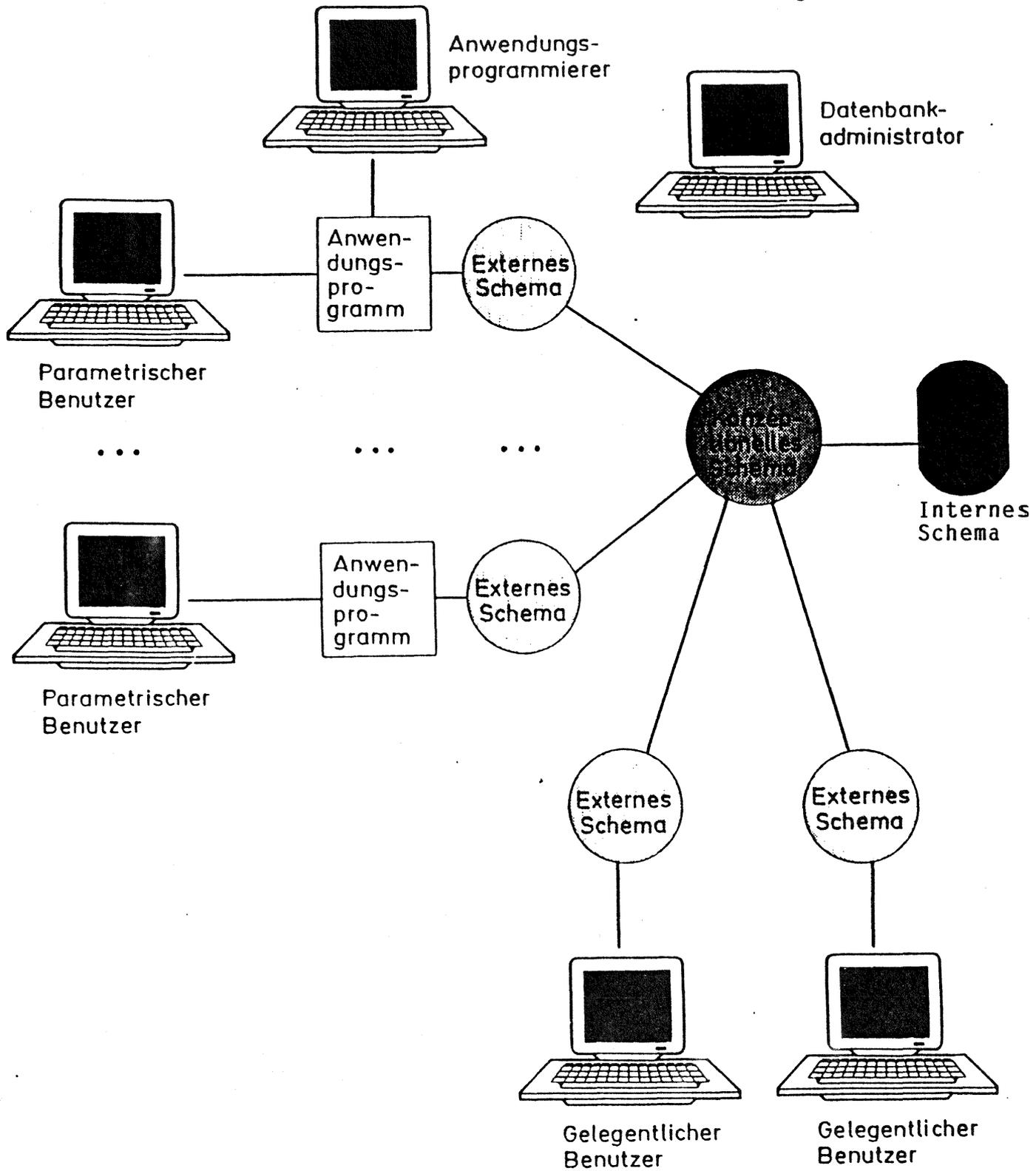


Abb. 3.2.2.1.4/1: Architektur eines Datenbanksystems

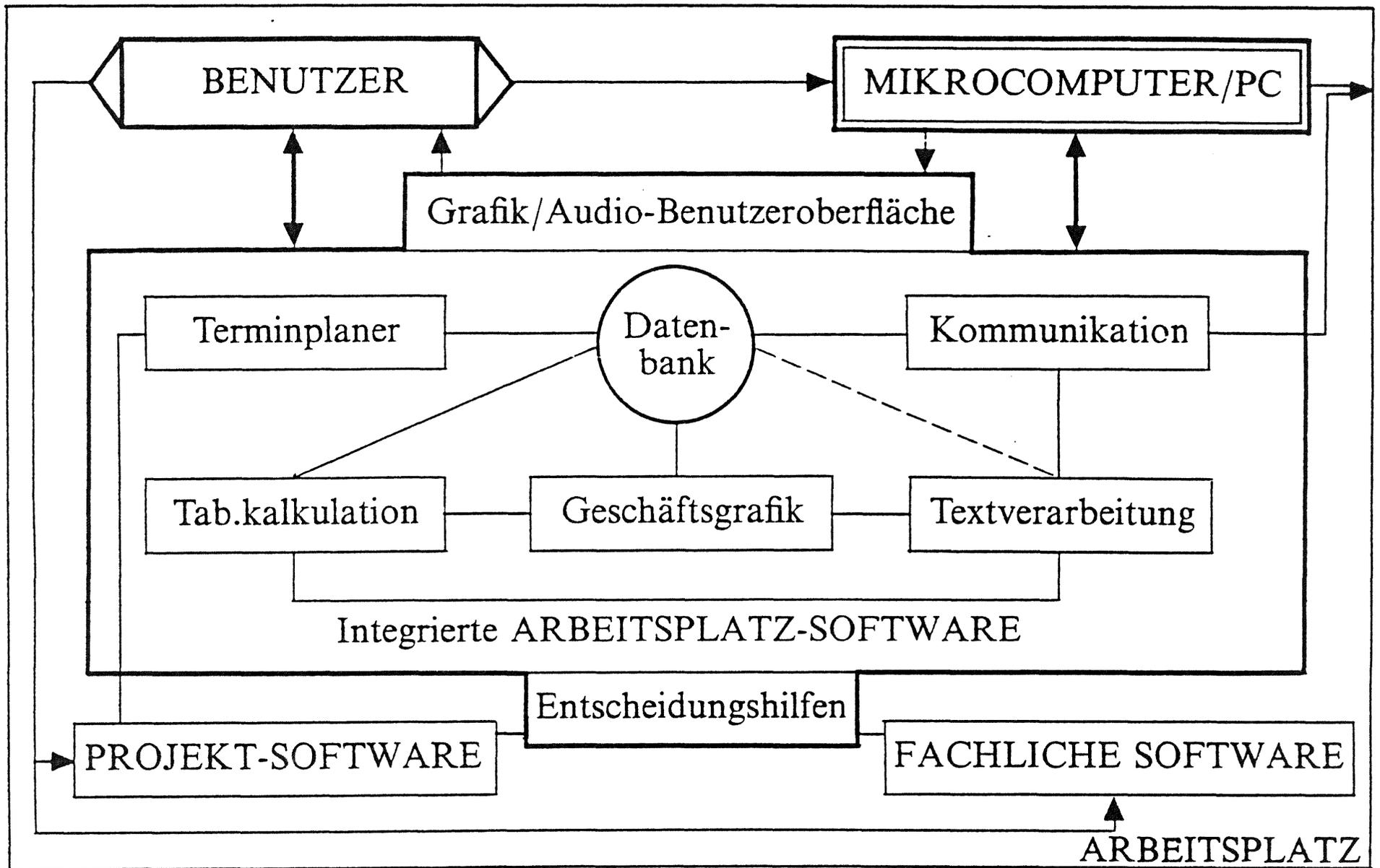


Abb. VI 10: (integrierte) Arbeitsplatz-Software und ihre Nutzungsformen

Quelle: S.Dworatschek, Grundlagen der Datenverarbeitung (1986:476)

1. Name(n) der (des) Erziehungsberechtigten

a) _____

b) _____

2. Kinder

Vor- und Zuname	Geburtsdatum	Kindergarten	Aufnahmedatum
-----------------	--------------	--------------	---------------

a) _____

b) _____

c) _____

d) _____

3. Selbsteinschätzung

Mein/Unser Einkommen betrug im Jahre 19.....

bis zu 50.000,--

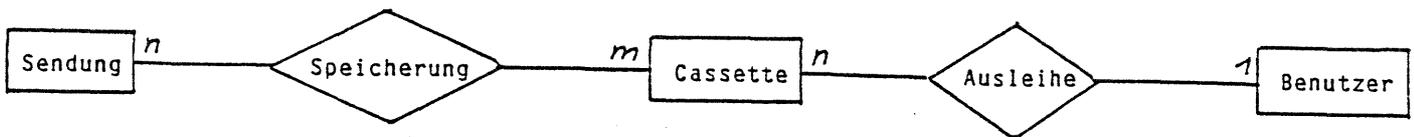
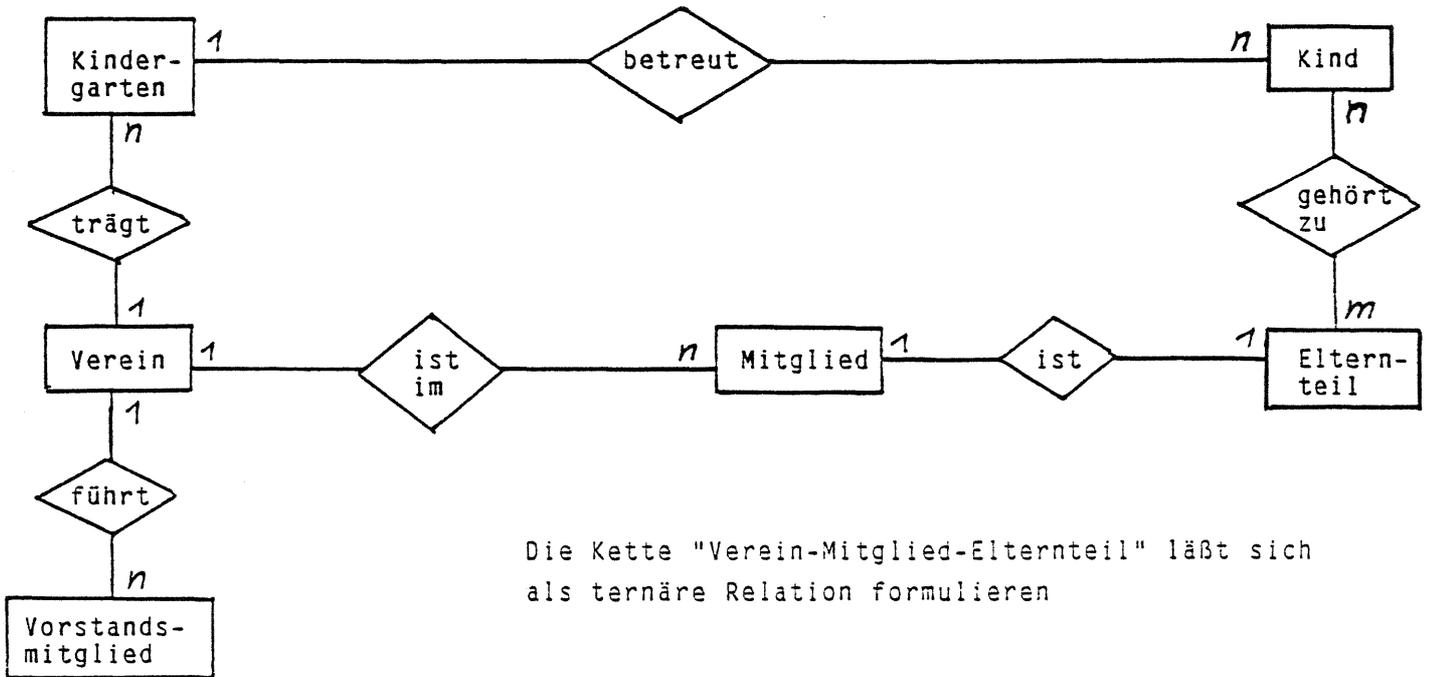
über 50.000,--

über 100.000,--

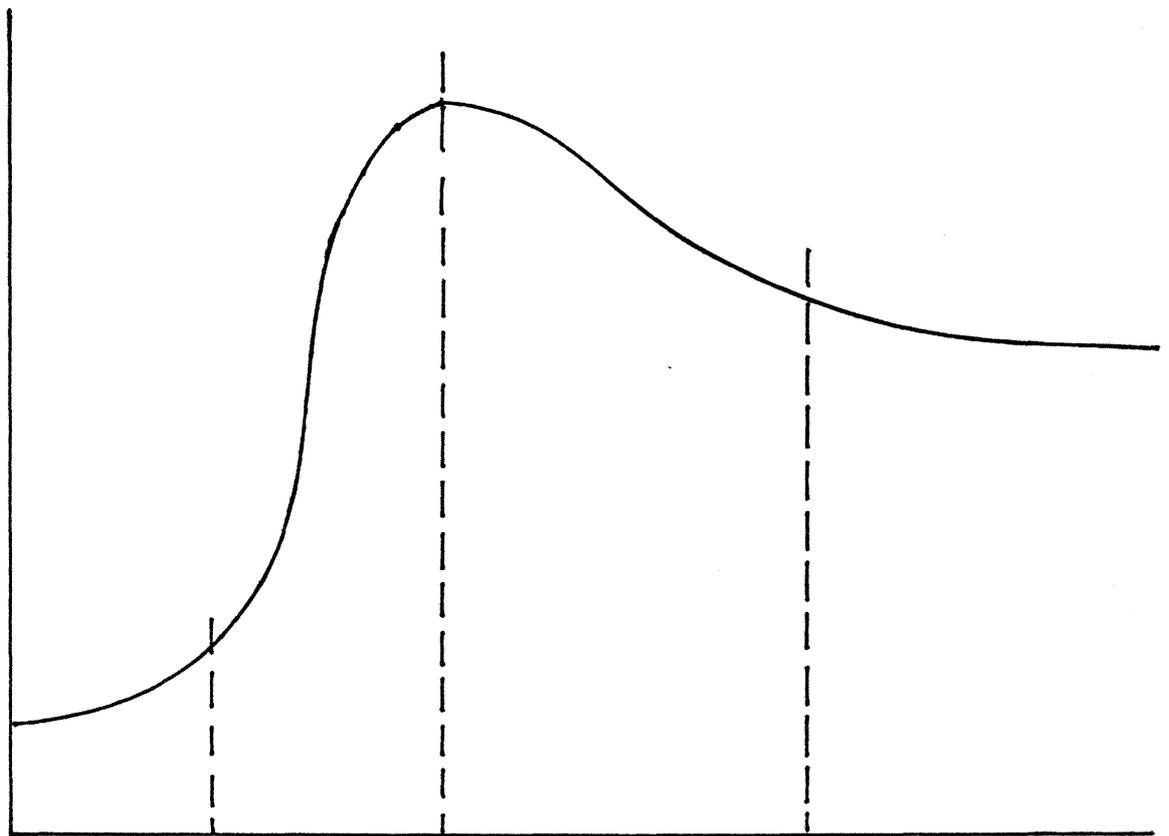
4. Ich/Wir erkläre(n), daß die obigen Angaben richtig sind. Mir/Uns ist bekannt, daß das Jugendamt bzw. Landesjugendamt meine/unsere Angaben nachprüfen kann. (§ 14 Abs. 5 KgG)

.....

(Datum/Unterschrift aller Erziehungsberechtigten)



Komplexitäts-



Systemanalyse

-ignoranz

-zuwachs

-reduktion

-stagnation

← Anwendungs-
programmierung →

LIEFERANTEN				
	LFR_CODE	NAME	ADRESSE	WOHNORT
(Zeile) →	035	FLORA GMBH	UFERSTR. 76	AACHEN
	004	HOVEN G.	SANDWEG 50	LINZ
	011	STRAUCH GMBH	BEERENWEG 1	LINZ

Beispiel 2.1. Tabelle LIEFERANTEN

Erläuterung:

- Tabellenname = LIEFERANTEN
- Spaltennamen, Kennzeichen = LFR_CODE, NAME, ADRESSE, WOHNORT
- Schlüssel = LFR_CODE (Primärschlüssel!)

Beachte:

Wäre das Kennzeichen LFR_CODE nicht vorhanden, könnte es einen aus den Kennzeichen NAME und ADRESSE zusammengesetzten Schlüssel geben.

Vorkommende Begriffe	Alternativen (genaugenommen ist die Bedeutung leicht abweichend)
Daten	Sammelname (unbestimmt)
Tabelle 1)	Relation, Entität, Datensatz, Bestand, File, Gegenstand, Begriff, Objekt, Themengruppe, Dateityp
Zeile 1)	Reihe, Gruppe, Entität, Tuple, Segment, Vorkommen, Occurrence, Record
Spalte 1)	Rubrik, Feld, Kennzeichen
Kennzeichen	Attribut, Datum, Item
Schlüssel	(Record-)Key
View	virtuelle Tabelle, externes Schema

Schema 2.2. Vorkommende Begriffe und ihre Alternativen.

Datum: 16. Juni 1984		Bestell-Nummer: 123456	Lieferer-Nummer: 75621	Lieferer-Name: Becker & Co.	Lieferer-Adresse: Bahnhofstraße 17 6200 Wiesbaden
Bitte liefern Sie:					
Produkt-Nr.	Bezeichnung	Stückpreis	Menge	Gesamtpreis	
98765123	Farbfernseher	2200,-	1	2200,-	
43567812	Stecker	3,70	10	37,-	
43567814	Steckdosen	6,50	10	65,-	
Gesamtwert der Bestellung: DM 2302,-					

Bild 4.6 Bestellformular

Tabelle Bestellung

Bestell- Nummer	Datum	Lieferer- Nummer	Lieferer- Name	Lieferer- Straße	Lieferer- PLZ	Lieferer- Ort	Bestell- wert	Produkt- Nummer	Produkt- Bezeichnung	Produkt- Preis	Produkt- Menge	Produkt- Gesamtpreis
123456	16.6.1984	75621	Becker & Co.	Bahnhof- str. 17	6200	Wiesbaden	2.302,-	98765123	Farbfernseher	2.200,-	1	2.200,-
								43567812	Stecker	3,70	10	37,-
								43567814	Steckdosen	6,50	10	65,-
...
...
...
...

UNNORMALISIERTE RELATIONEN

Beispiel: Beschaffungswesen

Bestell – Nummer
 Datum
 Lieferer – Nummer
 Lieferer – Name
 Lieferer – Straße
 Lieferer – PLZ
 Lieferer – Ort
 Bestellwert
 Produkt – Nummer
 Produkt – Bezeichnung
 Produkt – Preis
 Produkt – Menge
 Produkt – Gesamtpreis

Beispiel: Medienstelle

Cassetten – Nummer
 Sendungstitel
 Themenfelder
 Schlagworte
 Beschreibung
 Sender
 Sendedatum
 Sendewoche
 Sendezeit
 Sendelänge
 Archivierungsdatum
 Autor
 Sendereihe
 Ausleihfrist
 Aufnahmekapazität
 Laufwerkzählerstand

Personalausweis – Nr.

Vorname
 Nachname
 Status
 Ausleihkontingent
 Cassetten – Nummer
 Entleihdatum
 Rückgabedatum
 Entleihzweck
 Vormerkungsdatum
 Gewünschtes Entleihdatum
 Gewünschtes Rückgabedatum

ERSTE NORMALFORM (1NF)

Beispiel: Beschaffungswesen

Bestell – Nummer

Datum

Lieferer – Nummer

Lieferer – Name

Lieferer – Straße

Lieferer – PLZ

Lieferer – Ort

Bestellwert

Bestell – NummerProdukt – Nummer

Produkt – Bezeichnung

Produkt – Preis

Produkt – Menge

Produkt – Gesamtpreis

Beispiel: Medienstelle

Cassetten – Nummer

Aufnahmekapazität

Cassetten – NummerSendungs – Nummer

Sendungstitel

Themenfelder

Schlagworte

Beschreibung

Sender

Sendereihe

Sendedatum

Sendewoche

Sendezeit

Sendelänge

Aufnahmegeschwindigkeit

Laufwerkzählerstand

Archivierungsdatum

Ausleihfrist

Benutzer – Nummer

Vorname

Nachname

Personalausweis – Nr.

Status

Ausleihkontingent

Benutzer – NummerCassetten – Nummer

Entleihdatum

Rückgabedatum

Entleihzweck

Vormerkungsdatum

Gewünschtes Entleihdatum

Gewünschtes Rückgabedatum

ZWEITE NORMALFORM (2NF)

Beispiel: Beschaffungswesen

Bestell – Nummer

Datum

Lieferer – Nummer

Lieferer – Name

Lieferer – Straße

Lieferer – PLZ

Lieferer – Ort

Bestellwert

Bestell – NummerProdukt – Nummer

Produkt – Menge

Produkt – Gesamtpreis

Produkt – Nummer

Produkt – Bezeichnung

Produkt – Preis

Beispiel: Medienstelle

Cassetten – Nummer

Aufnahmekapazität

Cassetten – NummerSendungsnummer

Aufnahmegeschwindigkeit

Laufwerkzählerstand

Archivierungsdatum

Ausleihfrist

Sendungs – Nummer

Sendungstitel

Autor

Themenfeld

Schlagworte

Beschreibung

Sender

Sendereihe

Sendedatum

Sendewoche

Sendezeit

Sendelänge

Benutzer – Nummer

Vorname

Nachname

Personalausweis – Nr.

Status

Ausleihkontingent

Benutzer – NummerCassetten – Nummer

Entleihdatum

Rückgabedatum

Entleihzweck

Vormerkungsdatum

Gewünschtes Entleihdatum

Gewünschtes Rückgabedatum

DRITTE NORMALFORM (3NF)

Beispiel: Beschaffungswesen

Bestell – Nummer

Datum

Lieferer – Nummer

Bestellwert

Lieferer – Nummer

Lieferer – Name

Lieferer – Straße

Lieferer – PLZ

Lieferer – Ort

Bestell – NummerProdukt – Nummer

Produktmenge

Produkt – Gesamtpreis

Produkt – Nummer

Produkt – Bezeichnung

Produkt – Preis

Beispiel: Medienstelle

Cassetten – Nummer

Aufnahmekapazität

Cassetten – NummerSendungsnummer

Aufnahmegeschwindigkeit

Laufwerkzählerstand

Archivierungsdatum

Ausleihfrist

Sendungs – Nummer

Sendungstitel

Autor

Themenfeld

Schlagworte

Beschreibung

Sender

Sendereihe

Sendedatum

Sendezeit

Sendelänge

Sendedatum

Sendewoche

Benutzer – Nummer

Vorname

Nachname

Personalausweis – Nr.

Status

Ausleihkontingent

Benutzer – NummerCassetten – Nummer

Entleihdatum

Rückgabedatum

Entleihzweck

Vormerkungsdatum

Gewünschtes Entleihdatum

Gewünschtes Rückgabedatum