

Datenbanksysteme

**LESEPROBE
zum Kurs 817
Datenbanksysteme**

Autor:

Prof. Dr. Hermann Gehring

unter Mitarbeit von:

Dipl.-Inform. Jörg Homberger

I. Inhaltsübersicht

Kurseinheit 1: Entwicklung von Datenbanksystemen Architektur von Datenbanksystemen

1 Entwicklung von Datenbanksystemen.....	7
1.1 Dateiverwaltung durch Anwendungsprogramme.....	8
1.2 Dateiverwaltungssysteme.....	12
1.3 Datenbankkonzept.....	15
2 Architektur von Datenbanksystemen	21
2.1 Datenbankebenen	21
2.2 Datenbankverwaltungssysteme	33

Kurseinheit 2: Logische Datenorganisation

3 Logische Datenorganisation.....	6
3.1 Datenmodellierung.....	7
3.1.1 Grundelemente von Datenmodellen.....	7
3.1.2 Entity Relationship-Modell	18
3.2 Hierarchisches Datenmodell	21
3.3 Netzwerkartiges Datenmodell.....	31
3.4 Relationenmodelle	39
3.4.1 Klassisches Relationenmodell.....	39
3.4.2 Unternehmensdatenmodellierung mit einem erweiterten Relationenmodell	54
3.5 Logischer Datenbankentwurf.....	74
3.5.1 Schritte des logischen Datenbankentwurfs.....	75
3.5.2 Datenbankentwurf und Softwareentwicklungsprozess	86

Kurseinheit 3: Datenbankbenutzung

4 Datenbankbenutzung	5
4.1 Datenbanksprachen	5
4.2 Benutzungsformen und Benutzerklassen	9
4.3 Relationenalgebra.....	23
4.4 Structured Query Language (SQL)	33
4.4.1 SQL und das System R.....	34
4.4.2 Datenbeschreibung mit SQL.....	38
4.4.3 Datenmanipulation mit SQL.....	46
4.4.4 Externe Ebene des Systems R.....	72
4.4.5 Datenlexikon des Systems R	74

Kurseinheit 4: Physische Datenorganisation Datenintegrität

5 Physische Datenorganisation	5
5.1 Entwurf interner Sätze.....	6
5.2 Primärorganisation von Dateien.....	9
5.3 Sekundärorganisation von Dateien.....	11
5.4 Verbindung von Dateien	14
6 Datenintegrität	21
6.1 Datenkonsistenz	23
6.2 Datensicherheit.....	35
6.2.1 Synchronisation paralleler Datenbankzugriffe	35
6.2.2 Datenrekonstruktion	52
6.3 Datenschutz	59

II. Einleitung

Integrierte Informations-
verarbeitung durch
Datenbanksysteme

Im Gesamtsystem der betrieblichen Informationsverarbeitung spielen Datenbanksysteme eine zentrale Rolle. Einerseits dienen sie der Verwaltung der Stamm-, Bestands- und Bewegungsdaten, die bei dem Einsatz der verschiedenen betrieblichen Informationssysteme – insbesondere auch den die Leistungserstellung unterstützenden operativen Systemen bzw. ERP-Systemen – benötigt und verarbeitet werden. Zudem beinhalten Datenbanksysteme Funktionalitäten, die der Sicherstellung der Konsistenz und der Integrität der verwalteten und verarbeiteten Daten dienen. Schließlich ermöglichen Datenbanksysteme einen Zugriff unterschiedlicher Informationssysteme auf einen (umfassenden) Datenbestand. Insofern sind sie ein zentrales Mittel zur Realisierung des Grundgedankens der integrierten betrieblichen Informationsverarbeitung, d.h. eines Verarbeitungskonzeptes, das z.B. für die umfassende informationstechnologische Unterstützung der betrieblichen Leistungserstellung unabdingbar ist.

Datenbankkonzept

Das der Realisierung von Datenbanksystemen zugrunde liegenden Datenbankkonzept umfasst zwei Kernelemente:

- eine Datenbank als zentrale Datensammlung einschließlich eines die Struktur der Datensammlung bestimmenden Datenmodells und
- ein Datenbankverwaltungssystem zum Aufbau, zur Pflege und zur Erschließung der Datenbank für die Gesamtheit der Datenbankbenutzer.

Ziele des
Datenbankkonzepts

Mit dem Datenbankkonzept verfolgt man u.a. folgende, die Gesamtheit der betrieblichen Datenverarbeitungsprozesse betreffenden Ziele:

- Beherrschung des Massendatenaufkommens in einem Unternehmen.
- Strukturierung der gesamten Datenwelt eines Unternehmens in Form eines sogenannten konzeptionellen Schemas.
- Gewährleistung der Auswertbarkeit der Daten für beliebige Anwendungszwecke.
- Sicherstellung der Korrektheit und der Konsistenz der Daten.

Die Entwicklung und die Ausgestaltung des Datenbankkonzepts verliefen in mehreren Stufen. Zu Beginn entwickelte man hierarchische Systeme, die lediglich die Verwendung restriktiver, hierarchischer Datenstrukturen gestatteten. Mehr Freiheitsgrade bei der Datenmodellierung eröffneten die dann folgenden netzwerkartigen Systeme. Doch eine grundlegende Auseinandersetzung mit dem Problem der redundanzarmen Datenmodellierung fand erst bei der Konzipierung der heute in der Regel eingesetzten relationalen Systeme statt.

Heute ist die betriebliche Datenverarbeitung ohne den Einsatz von Datenbanken nicht mehr denkbar. Datenbankgestützte Anwendungssysteme dominieren im gesamten betrieblichen Anwendungsspektrum, von laufzeitintensiven, interaktionsarmen (Standard-) Anwendungen bis hin zu dialogintensiven Abfragesystemen.

Ziel des Kurses

Ziel des vorliegenden Kurses ist es nun nicht, möglichst viele Datenbanksysteme und Datenbanksprachen zu behandeln, und auch nicht, Details über die Funktionsweise von Datenbanksystemen zu vermitteln. Neben der Darstellung der prinzipiellen Funktionsweise und der Architektur von Datenbanksystemen stehen

vielmehr die Datenmodellierung, die Datenmanipulation und die Datenabfrage im betrieblichen Anwendungsbezug im Vordergrund. Zugrunde gelegt werden hierbei ein erweitertes relationales Datenmodell und die Abfragesprache SQL, die mittlerweile als Standardabfragesprache gelten kann.

Sämtliche theoretischen Ausführungen zur relationalen Datenmodellierung und zur Datenmanipulation mit SQL werden an einem praxisorientierten Beispiel demonstriert. Mit dem Beispiel verknüpfen sich zwei Ziele. Einerseits soll es das Erlernen der behandelten Vorgehensweisen und Sprachelemente erleichtern. Andererseits soll es eine Hilfestellung bei der selbständigen Anwendung der vermittelten Kenntnisse - beispielsweise bei der Anfertigung von Abschlussarbeiten im Bereich Wirtschaftsinformatik - geben.

Der Kurs ist in vier Kurseinheiten mit insgesamt sechs Kapiteln gegliedert.

Gliederung des Kurses

In der ersten Kurseinheit wird im einleitenden Kapitel 1 die Entwicklung von Datenbanksystemen in den Stufen Datenverwaltung durch Anwendungsprogramme, Einsatz von Dateiverwaltungssystemen und Datenbankkonzept beleuchtet. Auf die Architektur von Datenbanksystemen geht das Kapitel 2 ein. Vorgestellt wird insbesondere das sogenannte ANSI-Architekturmodell, das mit seiner 3-Ebenen-Architektur sowie mit seinen ebenenbezogenen Datenbanksprachen und personellen Instanzen einen Meilenstein in der Entwicklung der Datenbank-Technologie darstellt.

erste Kurseinheit

Gegenstand der aus dem Kapitel 3 bestehenden zweiten Kurseinheit ist die logische Datenorganisation. Behandelt werden einige Grundzusammenhänge der Datenmodellierung sowie das hierarchische, das netzwerkartige und das relationale Datenmodell. Besonderes Gewicht wird hierbei auf das klassische Relationenmodell und auf ein erweitertes Relationenmodell gelegt.

zweite Kurseinheit

Auf die Datenbankbenutzung geht die aus dem Kapitel 4 bestehende dritte Kurseinheit ein. Neben allgemeineren Aspekten der Datenbankbenutzung, die vor allem die Arten der Datenbankbenutzer und die Formen der Datenbankbenutzung betreffen, werden eingehend zwei Datenbanksprachen behandelt. Zum einen die speziell auf Relationensysteme zugeschnittene Relationenalgebra und zum anderen die ebenfalls für relationale Systeme entwickelte Sprache SQL.

dritte Kurseinheit

Die vierte Kurseinheit umfasst zwei Kapitel. Das Kapitel 5 thematisiert die Abbildung konzeptionell-logischer Datenstrukturen auf physische Strukturen unter Verwendung bekannter Methoden der physischen Datenorganisation. Der Problemkreis der Integrität von Datenbanken mit den drei Integritätsebenen "Datenkonsistenz", "Datensicherheit" und "Datenschutz" wird im Kapitel 6 aufgegriffen. Im Vordergrund stehen hierbei Fragen der Datenkonsistenz sowie die Formulierung von Konsistenzbedingungen mit der Sprache SQL.

vierte Kurseinheit

Der Kurs ist so abgefasst, dass sich ein ergänzendes Literaturstudium erübrigt. Die Literaturhinweise am Ende der einzelnen Kurseinheiten sind in zwei Gruppen untergliedert. Die Hinweise der ersten Gruppe beziehen sich auf die im jeweiligen Lehrtext zitierte Literatur und die Hinweise der zweiten Gruppe auf weiterführende Literatur.

III. Leseprobe

⋮

3.1 Datenmodellierung

3.1.1 Grundelemente von Datenmodellen

In nahezu allen Datenmodellen treten bestimmte Elemente auf, die hier als Grundelemente bezeichnet seien. Zu den Grundelementen gehören Entitäten, Beziehungen zwischen Entitäten, Attribute zur Charakterisierung von Entitäten und Schlüssel. Nachfolgend werden diese Grundelemente behandelt.

a) Entitäten

Entitäten benötigt man für die datenmäßige Darstellung von Phänomenen eines Realitätsausschnitts. Sie sind gleichsam Abbilder dieser Phänomene in der Datenwelt. Die abgebildeten Phänomene können realer Natur, z.B. Kunden, Lieferanten, Artikel, Maschinen usw., oder gedanklicher Natur, z.B. Kosten, Preise, Termine usw., sein. Für den Entitätsbegriff folgt damit:

Begriff der Entität

Eine Entität (engl. entity) ist ein Element der Datenwelt, welches ein reales oder ein gedankliches Einzelphänomen in einem betrachteten Realitätsausschnitt repräsentiert.

Beispiele für Entitäten sind der Kunde "ALFRED HUPFER KG", der Fertigungsauftrag "H 240543", das Auslieferungsfahrzeug mit dem Kennzeichen "OHZ-ME 72" usw. Jede dieser Entitäten stellt ein individuelles Exemplar aus einer Klasse von Exemplaren dar.

Objektklassen

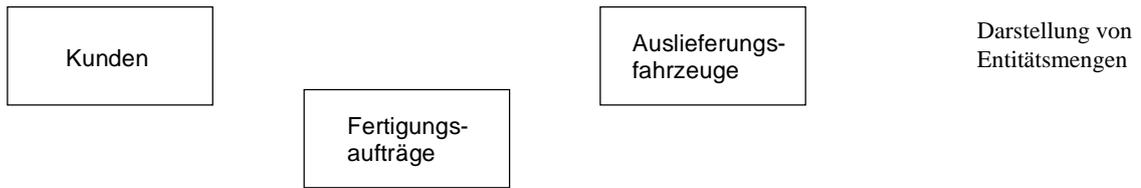
Ein zentrales Modellierungskonzept besteht in der Bildung von Klassen von Objekten. Man denke etwa an die Klasse der ganzen Zahlen, die bekanntlich durch den Datentyp INTEGER beschrieben wird. In vollkommen analoger Weise lassen sich Entitäten zu Klassen von Entitäten zusammenfassen. Klassen von Entitäten werden in der Literatur meist als Entitätsmengen bezeichnet. Zum Begriff der Entitätsmenge:

Eine Entitätsmenge (engl. entity set) fasst alle Entitäten zusammen, die durch gleiche Merkmale, nicht notwendigerweise aber durch gleiche Merkmalsausprägungen, charakterisiert werden.

Begriff der Entitätsmenge

Beispiele für Entitätsmengen sind Kunden, Fertigungsaufträge, *Auslieferungsfahrzeuge* usw. Zur Charakterisierung der Entitätsmenge *Auslieferungsfahrzeuge* mögen die Merkmale *Kennzeichen*, *Motorleistung*, *Ladegewicht* und *Höchstleistung* dienen. Auf die Entitäten dieser Mengen treffen zwar durchweg die genannten Merkmale zu, jedoch können die Merkmalsausprägungen bzw. -werte von Entität zu Entität variieren.

Zur grafischen Darstellung von Entitätsmengen werden Rechtecke verwendet. Abb. 3.2 zeigt einige Beispiele.



Darstellung von Entitätsmengen

Abb. 3.1. Grafische Darstellung von Entitätsmengen.

Sinnvoll ist die Bildung von Entitätsmengen deshalb, weil alle individuellen Exemplare einer Entitätsmenge in prinzipiell gleicher Weise behandelt oder manipuliert werden. Statt viele individuelle Entitäten zu betrachten, genügt es daher, sich auf der "Typebene" mit Entitätsmengen zu beschäftigen. Es gelten hier die gleichen Überlegungen, die im Kurs "Algorithmen und Datenstrukturen" bei der Einführung des Begriffs des Datentyps angestellt wurden.

Betrachtung auf Typebene

Alle in Abb. 3.2 angegebenen Entitätsmengen weisen eine Gemeinsamkeit auf. Sie enthalten ausschließlich Entitäten, die nicht gleichzeitig in anderen Entitätsmengen auftreten. Entitätsmengen mit dieser Eigenschaft heißen disjunkt.

Nicht disjunkte Entitätsmengen überlappen sich. Und für überlappende Entitätsmengen kann man immer eine Entitätsmenge angeben, welche die überlappenden Mengen umfasst. Eine umfassende Entitätsmenge kann sich allerdings auch aus nicht überlappenden Entitätsmengen zusammensetzen.

Die genannten Arten von Entitätsmengen seien begrifflich wie folgt präzisiert:

Zwei Entitätsmengen A und B heißen disjunkt, wenn keine Entität existiert, die sowohl zu A als auch zu B gehört.	disjunkte
Zwei Entitätsmengen A und B überlappen sich, wenn zumindest eine Entität existiert, die sowohl zu A als auch zu B gehört.	überlappende
Eine Entitätsmenge A umschließt eine Entitätsmenge B genau dann, wenn sämtliche in B auftretenden Entitäten auch zu A gehören.	umfassende Entitätsmengen

Abbildung 3.3 veranschaulicht diese Definitionen.

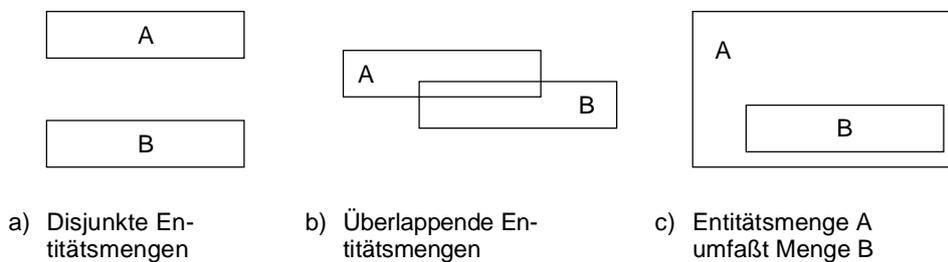


Abb. 3.2. Disjunkte, überlappende und umfassende Entitätsmengen.

Bei der Definition von Entitätsmengen bestehen gewisse Freiheitsgrade, die von den jeweiligen Gegebenheiten abhängen. Zur Verdeutlichung der Freiheitsgrade im Modellierungsprozess sei das in Abb. 3.4 angegebene Beispiel betrachtet.

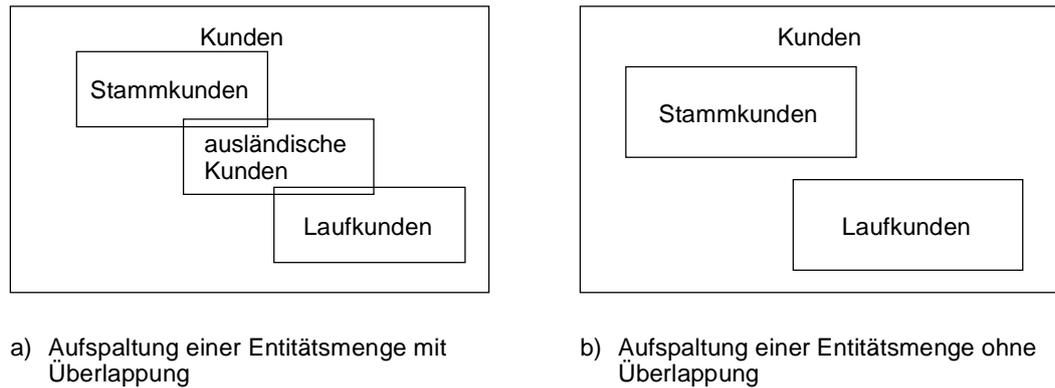


Abb. 3.3. Freiheitsgrade im Modellierungsprozess.

Aufspaltung von Entitätsmengen

Beide in Abb. 3.4 gezeigten Fälle entstehen durch die Detaillierung der Entitätsmenge *Kunden* in speziellere Entitätsmengen. Im Fall a) überlappen sie sich und im Fall b) nicht. Ob eine Überlappung vorzusehen ist, hängt von der Bedeutung bestimmter Merkmalseigenschaften ab. Während bei der Aufspaltung ohne Überlappung lediglich die Zugehörigkeit zur Stamm- oder Laufkundschaft interessiert, findet im anderen Fall auch die Auslandszugehörigkeit Berücksichtigung.

Man beachte, dass die Entitätsmenge *Kunden* noch auf viele andere Arten aufgespalten werden könnte. Beispielsweise unter Berücksichtigung von Umsatzklassen, Absatzregionen, Liefer- und Zahlungsmodalitäten usw. Die Frage, welche der möglichen Spezialisierungen angezeigt ist und ob überhaupt eine Spezialisierung vorzunehmen ist, lässt sich anhand der die Entitätsmenge *Kunden* manipulierenden Verarbeitungsprozesse entscheiden. Existieren beispielsweise Prozesse, die ausschließlich auf Stammkunden zugreifen, so könnte die Spezialisierung in eine Entitätsmenge Stammkunden sinnvoll sein.

b) Beziehungen zwischen Entitäten

⋮

4.4 Structured Query Language (SQL)

4.4.3 Datenmanipulation mit SQL

⋮

a) Auswahl von Daten (SELECT-Anweisung)

Zur Auswahl von Daten steht in SQL die SELECT-Anweisung zur Verfügung. Die Grundform der SELECT-Anweisung lautet:

Grundform der SELECT-Anweisung

SELECT	daten
FROM	basistabellen
WHERE	bedingung

Nach dem Schlüsselwort **SELECT** sind die auszugebenden Daten zu spezifizieren und nach **FROM** die Basistabellen, auf die sich die Abfrage bezieht. Welche Bedingungen die abzufragenden Daten erfüllen müssen, ist nach dem Schlüsselwort **WHERE** anzugeben. Die Grundform der **SELECT**-Anweisung kann noch einige Erweiterungen erfahren, beispielsweise durch das Anhängen einer Sortieranforderung oder durch das Einbinden einer weiteren **SELECT**-Anweisung im Bedingungsteil.

Nachfolgend werden die verschiedenen Formen der **SELECT**-Anweisung vorgestellt, und zwar gegliedert in elf Gruppen. Jede Anweisung erhält eine Nummer, um die Übersicht zu erleichtern. Außerdem wird jede Anweisung auf die in Beispiel 4.4 beschriebenen Basistabellen angewandt.

(1) Einfache Auswahl

Bei der einfachsten Form der Auswahl erübrigt sich die Angabe eines Bedingungsteils. Der Teil "WHERE bedingung" fällt daher weg.

Projektion mit Duplikaten

```
(1.1) SELECT ArtikelNr, DispoMenge
      FROM ArtikelBew
```

Ergebnis:

ArtikelNr	DispoMenge
810010	0
820050	0
910030	0
910030	0
910030	0

Die Anweisung (1.1) ist nichts anderes als eine Projektion der Relation *ArtikelBew* auf die Attributkombination (ArtikelNr, DispoMenge), wobei das Ergebnis Duplikate enthalten kann.

```
(1.2) SELECT DISTINCT ArtikelNr,
                    DispoMenge
      FROM ArtikelBew
```

Ergebnis:

ArtikelNr	DispoMenge
810010	0
820050	0
910030	0

Projektion ohne Duplikate mittels **DISTINCT**-Spezifikation

Sollen Duplikate im Ergebnis nicht auftreten, so ist dies durch den Zusatz **DISTINCT** explizit anzugeben. Die Anweisung (1.2) schließt Duplikate aus.

```
(1.3) SELECT *
      FROM ArtikelBew
```

Ergebnis: Kopie von *ArtikelBew*

Kopieren einer Basistabelle

Eine Kopie der Tabelle *ArtikelBew* erzeugt die Anweisung (1.3). Das Symbol * (Stern) steht hierbei als Abkürzung für sämtliche Attribute bzw. Attributnamen der Relation *ArtikelBew*.

(2) Qualifizierte Auswahl

Bei der qualifizierten Auswahl müssen die Ergebnisdaten einer Bedingung genügen, die im Teil "WHERE bedingung" zu spezifizieren ist. Betrachtet werde zunächst eine einfache Bedingung.

Auswahl mit Einzelbedingung

```
(2.1) SELECT ArtikelNr
        FROM ArtikelBew
        WHERE BewDatum = 920505
        AND ProjektNr = 140100
```

Ergebnis:

ArtikelNr
820050

Ermittelt werden die Nummern der Artikel, die am 5.5.92 eine Bewegung erfuhren und dem Projekt 140100 zuzuordnen sind.

Im Falle der Anweisung (2.1) besteht der Qualifikationsteil aus zwei Einzelbedingungen, die mit dem logischen Operator AND verknüpft werden. Außerdem richten sich beide Einzelbedingungen an die gleiche Tabelle, nämlich ArtikelBew.

Die Auswertung der Anweisung vollzieht sich wie folgt:

Für jeden Tupel der Relation ArtikelBew wird geprüft, ob

- der Wert von BewDatum gleich 920505 ist und ob
- der Wert von ProjektNr gleich 140100 ist.

Sind beide Bedingungen für einen Tupel erfüllt, so liegt ein zutreffender Tupel vor. Das auszugebende Ergebnis wird nun aus den Attributwerten der zutreffenden Tupel gebildet.

Eine Anweisung zur qualifizierten Auswahl kann sich wesentlich komplexer darstellen. Einerseits kann sie mehrere Tabellen betreffen und andererseits kann sie aus mehr als zwei verknüpften Einzelbedingungen bestehen. Für die Verknüpfung von Einzelbedingungen im Qualifikationsteil kommen die logischen Operatoren

Verknüpfung von Einzelbedingungen mit

logischen Operatoren

AND, OR und NOT

und

in Frage. In den Einzelbedingungen selbst sind die Vergleichsoperatoren

Vergleichsoperatoren

=, <>, <, <=, > und >=

zulässig. Zudem kann durch das Setzen von Klammern im Qualifikationsteil die Auswertungsfolge vorgegeben werden.

⋮