

DBI Zusammenfassung

Leon Schlömmner

April 3, 2020

Contents

1	Modellierung	3
1.1	Datenmodellierung	3
1.2	Datenmodelle	4
1.2.1	Hierarchisches Datenmodell	4
1.2.2	Netzwerk Datenmodell	5
1.2.3	Relationales Datenmodell	6
2	Normalisierung	9
3	SQL - DDL	10
4	SQL DCL	11
5	SQL - DML	12
6	PL/SQL	13
7	Gleichzeitigkeit	14
8	Indizes	15
9	Portieren von Projekten	16
10	Datenbank Tuning	17
11	Data Warehousing	18
12	Struktur und Aufbau einer Datenbank anhand von Oracle	19
13	Datenintegrität	20
14	Trigger	21
15	JPA	22

16 Apex	23
17 Verteilte Datenbanken	24
18 OracleNet	25
19 XML	26
20 Objektorientierte und Objektrelationale Datenbanken	27
21 Datenbank-Entwurfstools	28
22 Data Analytics	29
23 Docker	30
24 Bäume	31
25 NoSQL	32

Chapter 1

Modellierung

1.1 Datenmodellierung

Modellieren ist der Weg von der realen Welt bis zum Datenmodell. Ein Datenmodell verdeutlicht die Beziehungen zwischen Daten. Ergebnis des Modellierungsprozesses ist ein sogenanntes Datenschema, das zumeist grafisch visualisiert wird. Im Hinblick auf die Datenmodellierung eignen sich folgende Ansätze:

Konzeptuelles Datenmodell Beschreibt die globale logische Struktur aller Daten Implementierungs-unabhängig und stellt diese in einer systematischen Form dar.

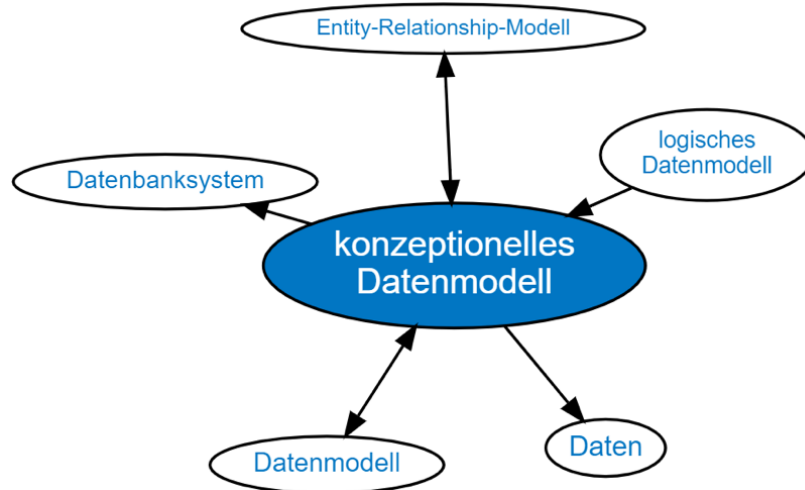


Figure 1.1: Konzeptuelles Datenmodell

Logisches Datenmodell Auf die spätere Implementierung ausgerichtetes Datenmodell, das die Daten für den späteren Einsatz bereits vorstrukturiert.

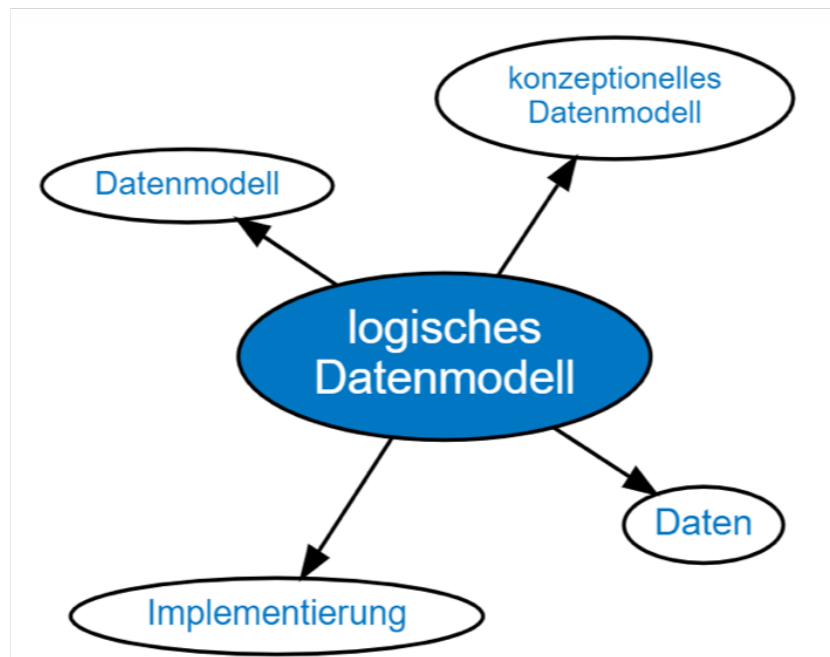


Figure 1.2: Logisches Datenmodell

Physisches Datenmodell Basiert auf der Grundlage des logischen Datenmodells. Im physischen Datenmodell wird besonders auf Effizienz der SQL-Abfragen geachtet, beispielsweise durch den Entwurf von Indexstrukturen.

Aufgabe der Datenmodellierung ist es, bei der Konzeption eines Informationssystems dessen *Objekte mittels Attribute und Beziehungen gemäß den Anforderungen zu strukturieren* und diese Struktur formal zu dokumentieren.

Ziele: Redundanzfreie Datenspeicherung und hohe Datenkonsistenz. Eine redundanzfreie Datenspeicherung liegt dann vor, wenn jede Information in einer Datenbank genau einmal vorkommt. Des Weiteren muss eine hohe Datenkonsistenz verfolgt werden, so dass Daten eindeutige Informationen darstellen.

1.2 Datenmodelle

Ein Modell ist ein *vereinfachtes Abbild der Wirklichkeit*. Zur Beschreibung der Art und Weise, wie Daten in einer Datenbank gespeichert werden, gibt es verschiedene Datenmodelle. Einige dieser Modelle findet man auch in der betriebswirtschaftlichen Organisationslehre wieder.

1.2.1 Hierarchisches Datenmodell

Das hierarchische Datenmodell kennen wir vom Dateisystem unserer Festplatte (Laufwerksbuchstabe, Ordner, Dateien). Dieses Modell wird auch „Baumstruktur“ genannt.

Ganz links (oder oben) befindet sich die Wurzel (Root). Von ihr sind alle Objekte abhängig, die weitere abhängige Objekte haben können. In der betriebswirtschaftlichen Lehre ist es mit einer Stablinienorganisation vergleichbar. Das hierarchische Datenmodell war früher ein sehr gebräuchliches Modell, deshalb bildet es die Grundlage älterer Datenbanksysteme.

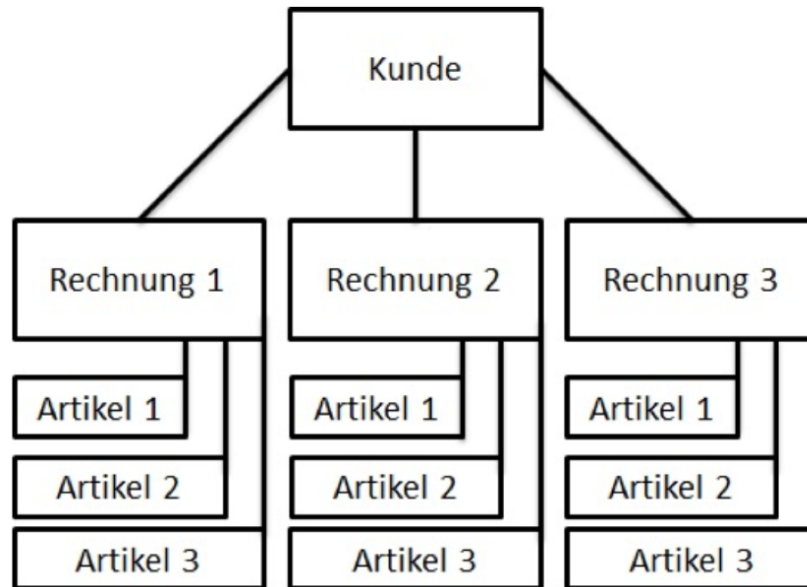


Figure 1.3: Stablinienorganisation

Durch die *hierarchische Baumstruktur* ist der *lesende Zugriff extrem schnell*. Der Nachteil der baumstrukturierten Verweise liegt bei der Speicherung der Daten und deren Verknüpfungen, da die Verweise untereinander vorab ermittelt werden müssen. Die Verknüpfungen werden über Eltern-Kind-Beziehungen realisiert und in der Baumstruktur abgebildet. Der große Nachteil bei diesem Modell ist es, dass man nur eine Baumstruktur verwenden kann: Es ist also nicht möglich, zwei Baumstrukturen miteinander zu verknüpfen. *Das hat zur Folge, dass dieses Modell sehr starr ist und wenig Freiheit für den Entwickler bietet.*

1.2.2 Netzwerk Datenmodell

Definition: Datenmodell, mit dem Netzwerkstrukturen zwischen Datensätzen beschrieben werden können. Es dient als Grundlage vieler Datenbanksysteme. Die drei Datenbanksprachen DML, DDL und DCL werden angewandt. Durch das netzwerkartige Modell existieren meist unterschiedliche Suchwege, um einen bestimmten Datensatz zu ermitteln. Es ähnelt dem hierarchischen Datenbankmodell und kann einer Matrixorganisation gegenübergestellt werden. Das Netzwerk Datenbankmodell besitzt keine strenge Hierarchie. Ein Datenfeld besteht aus einem Namen und einem Wert. Es sind m:n Beziehungen möglich, d.h. ein Datensatz kann mehrere Vorgänger haben. Des Weiteren

können auch mehrere Datensätze an erster Stelle stehen. Der Vorteil dieses Modells ist, das es unterschiedliche Suchwege als Lösungsweg angibt, was natürlich auch zu Problemen führen kann, wenn der Entwickler genau einen Lösungsweg benutzen will. Auch die Übersichtlichkeit verringert sich, wenn das Modell ständig weiter wächst.

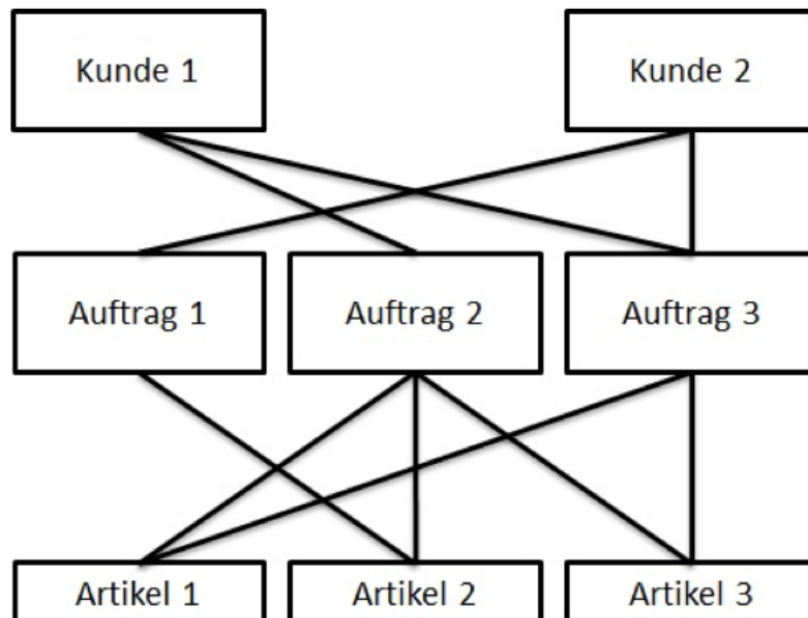


Figure 1.4: Netzwerk Datenmodell

Heute wird das Netzwerkdatenbankmodell als eine Art Verallgemeinerung des hierarchischen Datenbankmodells gesehen.

1.2.3 Relationales Datenmodell

= Auf den Arbeiten von Edgar F. Codd von 1970 basierendes Datenmodell, mit dem Beziehungen zwischen Daten in Form von Relationen beschrieben werden.

In einfachen Worten: Ein relationales Datenmodell ist eine Ansammlung von Tabellen, die miteinander verknüpft sind. Das relationale Datenmodell ist das am weitesten verbreitete Datenmodell, welches in der Datenbankentwicklung als Standard genutzt wird. Das Fundament des Datenbankmodells besteht aus drei Elementen: Tabellen, Attributen und Beziehungen.

Vorteile: sehr einfach und flexibel zu erstellen, hohe Flexibilität, leichte Handhabung

Nachteil: Effizienzprobleme bei großem Datenvolumen

Die wichtigsten Operationen mit Relationen (relationale Algebra), die ein Datenbankmanagementsystem zur Verfügung stellen muss, sind folgende:

- Auswahl von Zeilen
- Auswahl von Spalten
- Aneinanderfügen von Tabellen
- Verbund von Tabellen

Entität

Als Entität wird in der Datenmodellierung ein eindeutig zu bestimmendes Objekt bezeichnet, über das Informationen gespeichert oder verarbeitet werden sollen. Ein Entitätstyp beschreibt die Ausprägungen eines Objektes durch die Angabe von Attributen. Durch Typisierung (Erkennen gleicher Attribute von Entitäten) können Entitätstypen abgeleitet werden – aus mehreren Personen werden z.B. Kunden. Eine Entität kann materiell oder immateriell, konkret oder abstrakt sein. Beispiele für Entitäten: Fahrzeug, Konto, Person.

Attribute in einer Entität

Jede Entität besitzt eine bestimmbare Anzahl an Attributen (Ausprägungen bzw. Eigenschaften), die sich eindeutig von anderen Entitäten des gleichen Entitätstyps abgrenzen. Eine Eigenschaft ist ein konkreter Attributwert, den ein zuvor definiertes Attribut annehmen kann. Die Attribute stellen einen „Bauplan“ dar, der eine abstrakte Abbildung der Wirklichkeit ist.

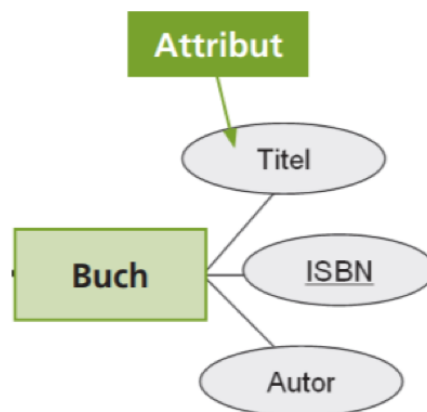


Figure 1.5: Attribute einer Entity

Die Attribute in einer Entität können unterschiedlich aufgebaut sein. Man unterscheidet zwischen *zusammengesetzte*, *mehrwertige* und *abgeleitete* Attribute.

Zusammengesetzte Attribute bestehen aus der Kombination mehrerer Attribute, die inhaltlich zusammengehören. Anhand einer Firmenadresse wird dies deutlich. Die Firmenadresse selbst ist ein Attribut der Firma, sie enthält aber die Attribute Straße, Hausnummer, Postleitzahl und Ort.

Mehrwertige Attribute können einen oder mehrere Attributwerte aufnehmen. So könnte ein Student gleichzeitig in zwei Studiengänge eingeschrieben sein.

Abgeleitete Attribute werden aus anderen Attributen oder Entitäten berechnet. Bezogen auf eine Datenbank wäre das z.B. die Bildung einer Summe aus mehreren Spalten einer Tabelle.

Chapter 2

Normalisierung

Chapter 3

SQL - DDL

Chapter 4

SQL DCL

Chapter 5

SQL - DML

Chapter 6

PL/SQL

Chapter 7

Gleichzeitigkeit

Chapter 8

Indizes

Chapter 9

Portieren von Projekten

Chapter 10

Datenbank Tuning

Chapter 11

Data Warehousing

Chapter 12

Struktur und Aufbau einer Datenbank anhand von Oracle

Chapter 13

Datenintegrität

Chapter 14

Trigger

Chapter 15

JPA

Chapter 16

Apex

Chapter 17

Verteilte Datenbanken

Chapter 18

OracleNet

Chapter 19

XML

Chapter 20

Objektorientierte und Objektrelationale Datenbanken

Chapter 21

Datenbank-Entwurfstools

Chapter 22

Data Analytics

Chapter 23

Docker

Chapter 24

Bäume

Chapter 25

NoSQL