

3 Datenbanksystem

3.1 Komponenten eines Datenbanksystems

Eine Datenbank (DB, engl. Data Base) ist eine systematische Sammlung von Daten. Zur Nutzung und Verwaltung der in der DB gespeicherten Daten benötigt der Anwender ein Datenbank-Verwaltungssystem (DBMS, engl. Data Base Management System). Die Kombination aus DB und DBMS ist das Datenbanksystem (DBS, engl.: Data Base System), das jedoch häufig fälschlicherweise als Datenbank bezeichnet wird.

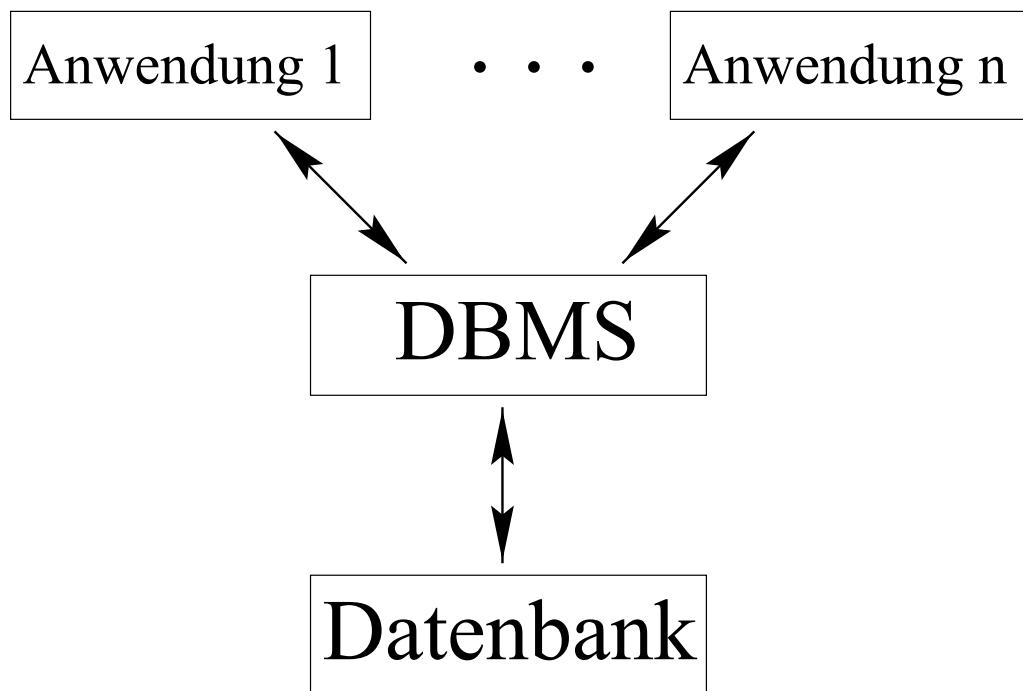


Abbildung 3.1: Struktur eines Datenbanksystems

Das DBMS besteht aus einer Vielzahl von Werkzeugen und Generatoren („Erzeugern“). Auf der einen Seite stellt es dem Entwickler die Instrumente zu Verfügung, mit denen er das Datenmodell beschreiben und einrichten kann. Auf der anderen Seite bietet es die Funktionen an, mit denen die einzelnen Anwender Daten eingeben, verändern, abfragen und ausgeben können.

Alle Funktionen des DBMS werden durch „was“ und nicht mehr „wie“ spezifiziert; soll heißen: Der Entwickler teilt dem Programm die Datenlogik mit und der Anwender formuliert seine Abfrage. Wie die Daten zu speichern und zu verwalten sind, ist Sache des

DBMS. Dieses ist also zuständig für die technische Umsetzung der Anforderungen des Entwicklers und der Anwender.

3.2 Ebenen eines Datenbanksystems

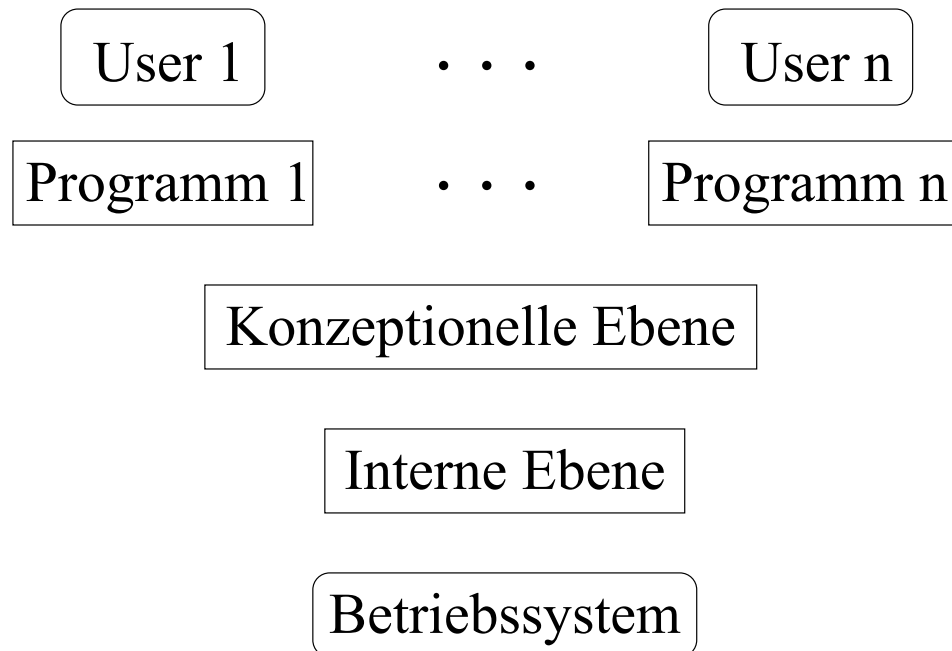


Abbildung 3.2: Die vier Ebenen eines DBS

Ein Datenbanksystem (DBS, engl.: Data Base System, = DB+DBMS) besteht aus den vier Ebenen:

3.2.1 Betriebssystem/Hardware

Dies ist die unterste Ebene, auf der jede Computeranwendung basiert. Neben dem DBS bauen auch alle anderen Programme auf dieser Ebene auf. Man kann diese Ebene aber noch weiter unterteilen: Zum einen ist da die Hardware als absolut unterste Ebene, deren Möglichkeiten vom Betriebssystem (BS) verwaltet werden. Das Betriebssystem zum anderen bietet Programmen die Hardwaremöglichkeiten an, ohne daß die Programme die Hardware direkt ansprechen müßten.

3.2.2 Interne Ebene

Auf der internen Ebene erfolgt die physische Speicherung der Daten. Die Speicherlogik, die dabei verwendet wird, hängt vom DBMS ab und kann dem Entwickler ziemlich egal sein, da er lediglich über die konzeptionelle Ebene auf die DB zugreift. Den Anwender

braucht weder die interne noch die konzeptionelle Ebene zu kümmern, da er erst über die oberste, nämlich die externe Ebene, auf die DB zugreift.

3.2.3 Konzeptionelle Ebene

Auf der dritten, der konzeptionellen Ebene, wird das Datenmodell beschrieben. Unter einem Datenmodell versteht man die datenmäßige Abbildung eines bestimmten Ausschnitts der realen Umwelt. Im Datenmodell sind die Strukturen der Daten und ihre Beziehung zueinander festgelegt. Nach der Art, wie die Beziehungen in dem Datenmodell geregelt werden, unterscheidet man zwischen hierarchischen, vernetzten, objektorientierten, objektrelationalen und relationalen Datenmodellen. Wir verwenden im Folgenden lediglich das relationale Datenmodell, da es (noch) die größte Verbreitung besitzt.

3.2.3.1 Tabellenstruktur

Beim relationalen Datenmodell werden die Daten in zweidimensionalen Tabellen angeordnet.

Jede Tabelle hat einen eindeutigen Relationsnamen. Alle Zeilen der Tabelle (ohne die Spaltenüberschriftszeile) werden als Relation, jede einzelne Zeile davon als Tupel bzw. Datensatz, die Spaltenüberschriften als Attributnamen oder Attribute und alle Attributnamen zusammen werden als Relationsschema bezeichnet.

Allgemein wird in jeder Zeile eine Entität abgebildet.

In Abbildung 3.3 wurde versucht, die Zusammenhänge grafisch darzustellen.

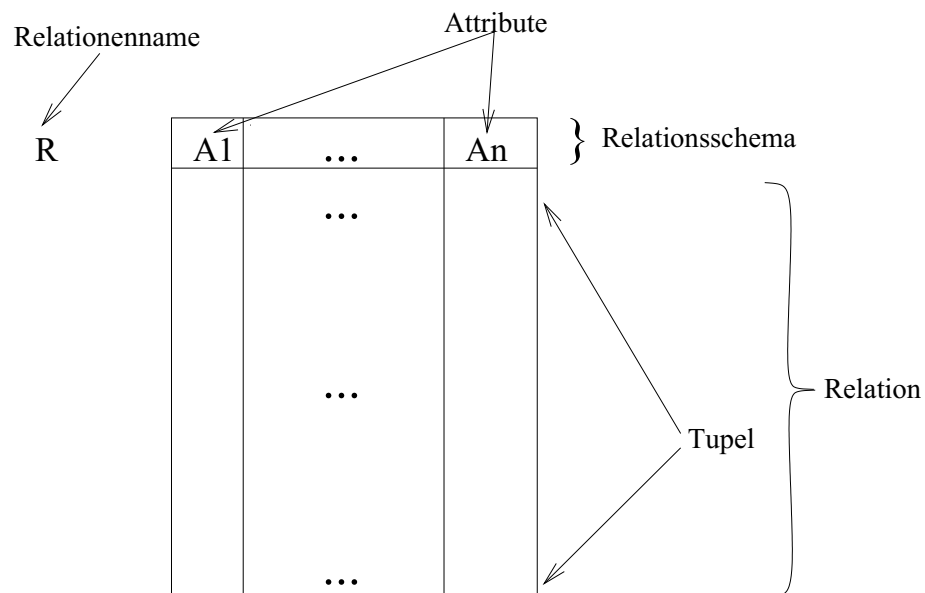


Abbildung 3.3: Tabellenstruktur

Um das Ganze etwas konkreter zu machen, habe ich in Tabelle 3.1 ein kleines Beispiel dargestellt.

Mitarbeiter				
MNr	AbtNr	Name	GebDat	Telefon
1	3	Christoph Reeg	13.5.1979	NULL
2	1	junetz.de	5.3.1998	069/764758
3	1	Uli	NULL	NULL
4	1	JCP	NULL	069/764758
5	2	Maier	NULL	06196/671797
6	2	Meier	NULL	069/97640232

Tabelle 3.1: Beispiel für Tabellenstruktur

Das Beispiel zeigt die Relation mit dem Namen ‘Mitarbeiter’. Jeder Mitarbeiter hat die Attribute ‘MNr’, ‘Name’, ‘GebDat’ und ‘Telefon’. In der Relation stehen 6 Datensätze bzw. Tupel.

3.2.3.2 Schlüssel

Damit man jede Zeile gezielt ansprechen kann, wird ein Schlüsselattribut eingeführt. Der Schlüssel muß immer eindeutig sein und wird auch als Primärschlüssel bezeichnet. Der Primärschlüssel muß nicht immer aus nur einem Attribut bestehen. Es ist auch möglich, mehrere Attribute zusammen als (zusammengesetzten) Primärschlüssel zu verwenden. Teilweise hat man in einer Relation mehrere Attribute, die eindeutig sind, d. h. Schlüssel sein könnten; in diesem Fall werden die anderen Attribute als Schlüsselkandidaten bezeichnet. Oder anders herum: Jeder Schlüsselkandidat kann jederzeit als Primärschlüssel benutzt werden. Es kann aber für eine Tabelle immer nur einen Primärschlüssel gleichzeitig geben.

Zum Einrichten der DB mit ihren Tabellen bedient man sich der Data Definition Language (DDL).

3.2.4 Externe Ebene

Auf der obersten Ebene befindet sich der Anwender, der auf das DBS mit einer Daten-Abfragesprache (DQL, engl.: Data Query Language), einer Daten-Manipulationssprache (DML, engl.: Data Manipulation Language) oder einer eigenen Anwendung, welche in unserem Beispiel die WWW-Seite ist, zugreift.