

Wie funktionieren Datenbanken?

Sollen Informationen in einem Computersystem strukturiert gespeichert, optimal verwaltet und auf Anforderung schnell und richtig gefunden werden, kommen in der Regel Datenbanksysteme zum Einsatz. Dieser Artikel gibt einen Einstieg in die Thematik und erklärt die Funktionsweise.

Warum eine Datenbank?

Jeder Unternehmer weiß wie schwierig es ist, die Kundenkontakte, Produktlieferungen, Kostenvorschläge, Rechnungen usw. nur in Form von Papierkorrekt und zeiteffizient zu verwalten. Kompliziert und zeitaufwendiger wird es dann, wenn man z.B. Rechnungen in ganz bestimmten Abhängigkeiten sucht. Hier wird bereits eine optimale Ablage benötigt. Man kann sich ja leicht vorstellen, dass ein gut gehendes Unternehmen umfangreiche Datenbestände so natürlich nicht verwaltet, will man kein Datenchaos in einer weiter wachsenden Zettelwirtschaft erzeugen.

Um die Verwaltung umfangreicher Daten zu vereinfachen, wird dieser Prozess immer mehr durch Software unterstützt. Anfänglich wurden einfach die Daten in Dateien gespeichert, was dann natürlich zu ähnlichen Problemen wie bei der Zettelwirtschaft führte. Später dann wurden Softwaresysteme eingeführt, die den immer weiter wachsenden Datenbestand in einer optimalen Art und Weise strukturiert und persistent (dauerhaft und wiederverwendbar) speichert. Man spricht hier von einem Datenbanksystem oder auch Datenbank Management System (DBMS). Das DBMS speichert die Daten in der Regel auf einer Festplatte. Zu beachten ist, dass Daten ver-

facht bedeutet dies, dass die Unternehmensdaten zentralisiert sind und die Verwaltung der Daten schnell und einfach durchgeführt werden kann.

ANZEIGE

ZAHNWERK
Frästechnik GmbH

Ist Ihr CAD-CAM Partner:
ZIRKON-CoCr-Titan-Kunststoff Composite

Infos & Preise:
www.zahnwerk.eu

Datensätze (stl) an:
zeno@zahnwerk.eu

Welche Vorteile bietet eine Datenbank?

- Die Daten können zentral verwaltet werden.
- Eine Redundanz von Daten kann verhindert werden.
- Nutzer können gleichzeitig auf die Daten zugreifen und die Daten mittels Software verarbeiten.
- Unterschiedliche Anwendersoftware können auf die Datenbank zugreifen und die Daten je nach Anwendung unterschiedlich verarbeiten.
- Die Daten können einfach und schnell aktualisiert werden.
- Es besteht eine einfache und schnelle Suchmöglichkeit mit Filterung der Daten.
- Nicht mehr benötigte Daten können einfach und schnell gelöscht werden.

(Festplatte) gespeichert. In der Regel arbeitet ein DBMS in einer sogenannten „Client-Server-Architektur“, d.h. dass das DBMS von der Anwendersoftware getrennt ist und der Anwendersoftware einen Service anbietet.

Der Client stellt prinzipiell eine Anfrage an den Server und bekommt vom Server ein Ergebnis geliefert. Die Anwendersoftware, also der Client, wird auch als „FrontEnd“ und das DBMS, also der Server, als „BackEnd“ bezeichnet. Die Anwendersoftware kann somit nicht direkt auf die Datenbank zugreifen. Das DBMS fängt die Anfragen des Clients ab und kann z.B. dafür sorgen, dass im Falle von Mehrfachzugriffen verschiedener Nutzer im Mehrbenutzerbetrieb auf dieselben Daten keine unerwünschten Änderungen auftreten. Das DBMS muss also dafür sorgen, dass zu einem bestimmten Zeitpunkt nur ein Benutzer auf ein Datum zugreifen und die Datenänderung durchführen kann. Diese Steuerung wird auch als „Transaktionssteuerung“ bezeichnet. Die Client-Server-Architektur eines DBMS hat den großen Vorteil, dass FrontEnd und BackEnd in einem Netzwerk miteinander kommunizieren können, da beide eigenständige Prozesse sind. Ein Mehrbenutzerbetrieb des DBMS erfordert dies natürlich auch. Durch die Verbreitung des World Wide Web können durch

bankmodell und speichert die Daten in einer Baumstruktur ab. Ein Datensatz bildet einen Knoten und hat immer einen Vorgängerknoten. Die Baumstruktur beginnt mit dem sogenannten Wurzelknoten.

- **Netzwerk-Datenbankmodell:** Dieses Modell ist eine Verallgemeinerung des hierarchischen Datenbankmodells und ein Datenknoten kann mehrere Vorgänger haben.
- **Objektorientiertes Datenbankmodell:** Hierbei werden Datenobjekte gespeichert, die in der objektorientierten Soft-

Der Schlüssel wird in der Regel automatisch vom DBMS generiert. In unserem kleinen Beispiel kann nun ein Zahnarztkunde mehrere Patienten haben. Wir benötigen nun eine zweite Tabelle, in der die Patienten gespeichert werden. Jeder Patienten-Datensatz enthält nun zusätzlich eine Kunden-ID, mit der dieser Datensatz dem Zahnarztkunden eindeutig zugewiesen werden kann (siehe Tabelle Patienten).

Hier kann man sehr gut erkennen, dass die Tabelle „Zahnarztkunden“ mit der Tabelle

an den Standard und weichen von der SQL-Norm ab. Das macht es natürlich für Softwareentwickler nicht einfach und sie müssen die Abweichungen bei der Programmierung genau kennen und berücksichtigen.

SQL ist prinzipiell die Schnittstelle zwischen der relationalen Datenbank und der Anwendersoftware. SQL unterscheidet zwischen zwei Kategorien der SQL-Sprachelemente (siehe Abb. 2).

Die Zukunft der Datenbanksysteme

Unsere Informations- und Wissensgesellschaft speichert derzeit unglaublich viele Daten in weltweit installierte Datenbanken. Heute speichern wir Datenmengen, die um Größenordnungen umfangreicher sind als noch vor zehn Jahren. Die Anzahl nimmt weiter zu und es ist kein Ende abzusehen. Relationale Datenbanken sind derzeit die meist installierten weltweit und werden auch die nächsten Jahre die DBMS-Welt beherrschen. Wohin die Reise genau geht, ist noch nicht klar. Auf jeden Fall müssen die großen Datenmengen in Datenbanksysteme ohne Datenverlust weiter durch neue Softwaremethoden verdichtet werden (**Verdichtung von Daten**), um Hardware und somit Energie einzusparen. Des Weiteren müssen Teile einer Datenbank in anderen Datenbanken integriert werden können, um neue Businessanwendungen aus bestehenden unterschiedlichen Businessanwendungen in unterschiedlichen Datenbanken zu erhalten (**Integration von Daten**). **z**

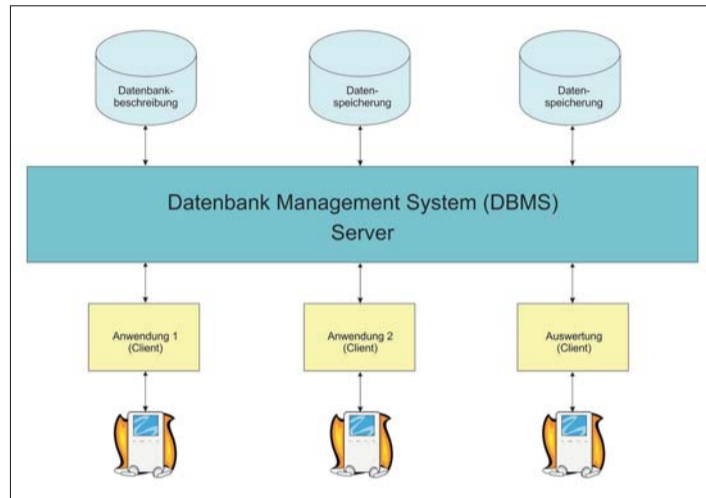


Abb. 1: Architektur eines Datenbank Management Systems (DBMS).

wareentwicklung einfacher und leistungsfähiger verarbeitet werden können.

- **Relationales Datenbankmodell:** Dieses Datenbankmodell ist das am meist verwendete und soll weiter betrachtet werden.
- **Objektrelationales Datenbankmodell:** Hier werden die Daten in der Anwendersoftware objektorientiert verarbeitet und in der Datenbank relational gespeichert.

„Patienten“ in einer 1:n-Beziehung steht, d.h. es kann jeder Zahnarzt „n“ Kunden besitzen. Der Zahnarztkunde Meyer besitzt die Patienten Huber und Obermayer.

Die Abfragesprache SQL

Die **Structured Query Language** (abgekürzt SQL) dient zur Definition, Abfrage und Manipulation von Daten für Software, die mit der verbun-

Kunden-ID	Kundenname	Straße	PLZ	Ort
1	Meyer	Musterstr. 10	12345	Musterstadt
2	Schmidt	Hinterweg 2	56789	Neustadt

Tabelle Zahnarztkunden.

Patienten-ID	Kunden-ID	Kundenname	Straße	PLZ	Ort
1	1	Huber	Blumenweg 5	22222	Unterstadt
2	1	Obermayer	Birkenweg 8	33333	Oberstadt
3	2	Meyerhuber	Eichenweg 2	44444	Oberstadt

Tabelle Patienten.

waltet werden und nicht Informationen. Das liegt daran, dass ein Datum z.B. einen Wert von 100 besitzt. Erst wenn die Semantik, also die Bedeutung eines Datums bekannt ist, kann daraus eine Information erzeugt werden. Ein kleines Beispiel soll dies verdeutlichen: Ein Datenbank-Nutzer bekommt die Information geliefert, dass von einem bestimmten Material nur noch 100 Gramm vorrätig sind. Das Datenbanksystem verwaltet dazu das Datum 100 und ein Datum mit dem Text „Menge in Gewicht“, das somit dem Datum 100 zugeordnet werden kann. Ein weiterer wesentlicher Grund für ein Datenbanksystem ist die Zentralisierung der Datenhaltung. Dies ergibt sich aus betriebswirtschaftlichen Gesichtspunkten, die Unternehmensdaten so zu organisieren, dass für die Daten eine verantwortliche Stelle vorhanden ist. Diese verantwortliche Stelle ist für die Organisation der Daten zuständig. Verein-

- Eine Datenbank kann nach Bedarf ausgebaut werden. Die Datenbank kann also mit den Anforderungen mitwachsen.
- Die Daten können einfach und schnell gesichert werden.

Das Datenbank Management System (DBMS)

Das DBMS ist, wie bereits oben beschrieben, eine komplexe Verwaltungssoftware. Dabei werden die Daten persistent auf ein Speichermedium

ANZEIGE

die Client-Server-Architektur die Benutzer ganz einfach über einen Web-Browser mit einer Datenbank kommunizieren. Hierbei liegt der eigentliche Client verborgen auf einem Webserver und hält die Kommunikationsverbindung zur Datenbank. Der Client, also die Webserver-Software, sendet nach erfolgreicher Anfrage zur Datenbank zum Schluss ein geliefertes Ergebnis (z.B. das Ergebnis einer Suchanfrage) dann zum Web-Browser über eine bestehende HTTP-Verbindung (Abb. 1).

Das relationale Datenbankmodell

Das Datenbankmodell bestimmt die Art der Datenspeicherung und Datenbearbeitung innerhalb der Datenbank. Derzeit existieren fünf unterschiedliche Datenbankmodelle:

- **Hierarchisches Datenbankmodell:** Ist das älteste Daten-

Die Funktionsweise des relationalen Datenbankmodells

Die weltweit am meisten eingesetzten Datenbanksysteme basieren auf dem relationalen Datenbankmodell:

- **Oracle Datenbank**
- **Microsoft SQL-Server**
- **IBM DB2**
- **MySQL** (früher SUN Microsystems, jetzt Oracle)

Im relationalen Datenbankmodell werden die Datensätze in Tabellen, auch als **Relationen** bezeichnet, gespeichert, wobei jeder Datensatz (Record) eine Tabellenzeile darstellt. Diese Zeile wird auch **Tupel** genannt. Jedes Tupel wiederum besteht aus mehreren Attributwerten, den Tabellenspalten. Das sogenannte **RelationsSchema** legt die Anzahl und den Datentyp der Attribute für eine Relation fest. Jedes Tupel enthält einen sogenannten **Schlüssel** oder auch mehrere Schlüssel, d.h. jeder Datensatz ist **eindeutig identifizierbar**. Mehrere Tabellen können auch miteinander in Beziehung stehen und bilden so die Abhängigkeiten. Das folgende kleine Beispiel mit gespeicherten Zahnarztkunden soll dies verdeutlichen (siehe Tabelle Zahnarztkunden).

Hier kann man gut erkennen, dass der Datensatz (Tupel) mit der Kunden-ID (Schlüssel) 1 eindeutig in der Tabelle ist.

• **Data Definition Language (DDL)**
Anweisungen zum Anlegen und Verwalten von Datenbank-Schemata. Beispiel: Anlegen von Tabellen mit den entsprechenden Attribut-Typen und Definition der Relationen. Außerdem können Benutzer mit entsprechenden Zugriffsrechten mittels DDL verwaltet werden.

Beispiel für das Anlegen einer neuen Tabelle in der Datenbank:

```
CREATE TABLE zahnarztkunden
  kunden_ID INTEGER NOT NULL PRIMA-RY KEY,
  name CHAR(30) NOT NULL,
  plz CHAR(5),
  ort CHAR(50) NOT NULL
```

INTEGER = Ganzer Zahlenwert (z.B. 10)
CHAR = Character (z.B. Heinz Mustermann)
NOT NULL = Wert darf nicht leer sein
PRIMARY KEY = eindeutiger Schlüssel

• **Data Manipulation Language (DML)**
Mit der DML können Anweisungen für die Eingabe, Änderung und Löschung von Daten in eine vorhandene Tabelle erstellt werden. Außerdem können mittels der DML Abfragen an das DBMS gestellt werden, wie z.B. eine Suchabfrage nach bestimmten Datensätzen.

Beispiel für das Auslesen aller gespeicherten Datensätze in der Tabelle „Zahnarztkunden“:

```
SELECT * FROM zahnarztkunde
```

Abb. 2

denen Datenbank arbeiten. Möchte also eine Anwendersoftware einen Datensatz in eine Tabelle hinzufügen, muss ein sogenannter SQL-Befehl an das DBMS geleitet werden. SQL ist weltweit standardisiert und ist speziell nur für relationale Datenbanken entwickelt worden, dennoch halten sich nicht alle Unternehmen, die Datenbanksysteme entwickeln,

ZT Adresse

Thomas Burgard Softwareentwicklung & Webdesign
Dipl.-Ing. (FH) Thomas Burgard
Bavariastr. 18b
80336 München
Tel.: 0 89/54 07 07-10
Fax: 0 89/54 07 07-11
E-Mail: info@burgardsoft.de
www.burgardsoft.de