



#### Weitere Datenbankmodelle

Ein Datenmodell oder auch Datenbankmodell bestimmt, auf welche Art und Weise Daten in einem Datenbanksystem gespeichert und bearbeitet werden, also wie die gespeicherten Daten untereinander in Beziehung stehen. Für die logische Organisation des Datenbestands wurden im Laufe der Zeit außer dem relationalen Datenbankmodell noch weitere Modelle entwickelt.

#### Dokumentenorientiertes Datenbankmodell (NoSQL<sup>1</sup>)

In diesem Datenmodell gibt es Schlüssel zu denen jeweils ein Wert gehört. Dokumente sind verschachtelte Strukturen. Für gewöhnlich handelt es sich dabei um JSON-Objekte (JavaScript Object Notation), da sie ein weitverbreitetes Datenformat für Webanwendungen und Apps sind.

```
myObj = {
  "name": "John",
  "age": 30,
  "cars": [
    { "name": "Ford", "models": [ "Fiesta", "Focus", "Mustang" ] },
    { "name": "BMW", "models": [ "320", "X3", "X5" ] },
    { "name": "Fiat", "models": [ "500", "Panda" ] }
  ]
}
```

Abb. 01: Verschachteltes JSON-Objekt

Im dokumentenorientierten Datenbankmodell werden keine vordefinierten Felder verlangt. Außerdem kann man ohne Probleme nachträglich bei einem Eintrag neue Felder hinzufügen, die bisher nicht existiert haben, oder andere weglassen, die bei manchen Einträgen verwendet wurden. Dieses flexible Modell findet heute besonders bei Web-Applikationen Anwendung.

MongoDB (abgeleitet vom engl. *humongous*, „gigantisch“) ist beispielsweise eine aktuell sehr populäre dokumentenorientierte NoSQL-Datenbank für Web-Anwendungen. Da die Datenbank dokumentenorientiert ist, kann sie Sammlungen von JSON-ähnlichen Dokumenten verwalten. So können viele Anwendungen Daten auf natürlichere Weise modellieren, da die Daten zwar in komplexen Hierarchien verschachtelt werden können, dabei aber immer abfragbar bleiben.

#### Hierarchisches Modell

Dieses Modell ist dadurch gekennzeichnet, dass die Daten wie ein auf den Kopf gestellter **Baum** dargestellt sind. Stell dir einen kleinen Ausschnitt aus einem Familienstammbaum bildlich vor. Die Eltern stehen hier ganz oben. Eltern können viele Kinder haben, Kinder jedoch nur ein Elternpaar. So verzweigt sich der Stammbaum mit jeder neuen Generation immer weiter nach unten; es entstehen viele neue Äste. Das hierarchische

<sup>1</sup> NoSQL (englisch für *Not only SQL* deutsch: „Nicht nur SQL“) bezeichnet Datenbanken, die einen nicht-relationalen Ansatz verfolgen. Diese Datenspeicher benötigen keine festgelegten Tabellenschemata. Sie werden häufig als „strukturierte Datenspeicher“ (engl. *structured storage*) bezeichnet.



## 2.3.2 Datenbanksysteme II

### Lerninhalte 232-02 Weitere Datenbankmodelle

Modell ist im Bereich der Datenbanksysteme heute weitgehend von anderen Datenbankmodellen abgelöst worden. Die hierarchische Datenspeicherung jedoch wird beispielsweise bei vielen PC-Betriebssystemen als **Dateimanagementsystem** benutzt.

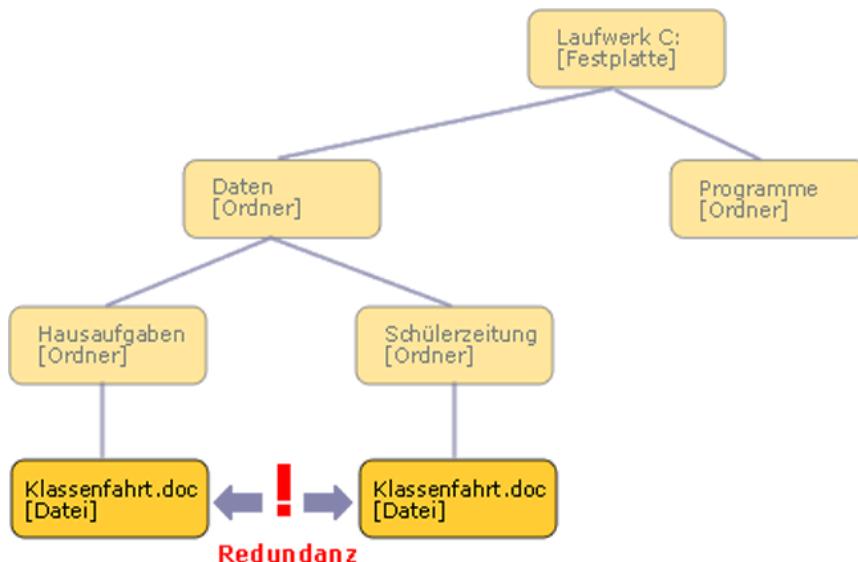


Abb. 02: Hierarchisches Datenmodell

Nachteilig wirkt sich bei diesem Modell aus, dass ein einzelner Dateneintrag eventuell mehrfach erscheinen muss, da er in verschiedenen hierarchischen Zusammenhängen stehen kann. Außerdem ist es sehr **umständlich**, eine einmal bestehende und weit verzweigte Struktur im Nachhinein nach neuen und jetzt wahrscheinlich sinnvolleren Gesichtspunkten **umzuorganisieren**. Ein Ablegen der gleichen Daten an verschiedenen Stellen des Baumes ist in diesem Modell ohne weiteres möglich und bedeutet deshalb eine **Verschwendungen von wertvollem Speicherplatz**.

Unter **Redundanz** versteht man mehrfach gehaltene Informationen. Eine Informationseinheit ist dann redundant, wenn sie ohne Informationsverlust weggelassen werden kann.

## Netzwerkartiges Datenbankmodell

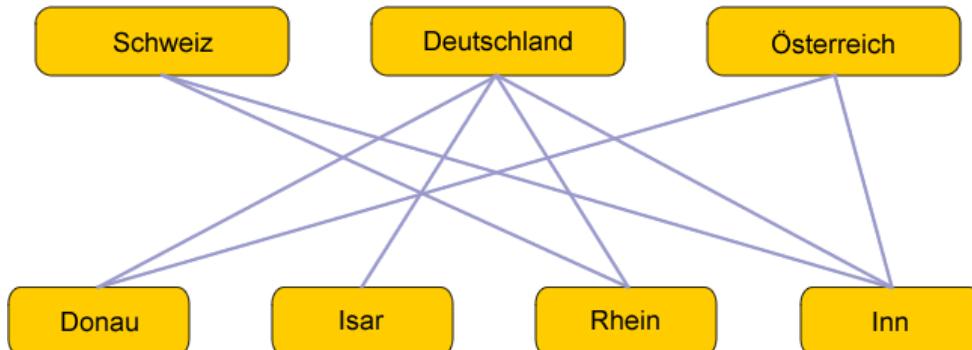


Abb. 03: Netzwerkartiges Datenmodell



## 2.3.2 Datenbanksysteme II

### Lerninhalte 232-02 Weitere Datenbankmodelle

Das Netzwerk-Datenbankmodell ist eine Weiterentwicklung des hierarchischen Modells. Im Netzwerkmodell ist es möglich, dass „Kinder“ viele „Eltern“ haben können.

Durch diese Erweiterung wird das Netzwerkmodell leider sehr schnell **unübersichtlich** und für den normalen Anwender in der Praxis **schwer zu handhaben**. Was man suchte, war ein einfaches Modell, das relativ leicht von jedermann genutzt werden konnte (siehe Relationale Datenbankmodell).

### Objektorientiertes Datenbankmodell

Ein objektorientiertes Datenbankmanagementsystem (DBMS) erlaubt es, komplexe Datenobjekte zusammen abzuspeichern. Dabei lehnt es sich an die objektorientierte Programmierung (OOP) an und vermeidet damit das wiederholte Zusammensuchen zusammengehörender Daten - eine Abbildung der Objekte auf die relationale Tabellenstruktur ist dadurch nicht mehr notwendig.

Ein komplexes Datenobjekt modelliert normalerweise einen Gegenstand oder Begriff der realen Welt und enthält nicht nur alle dazu gehörigen Attribute, sondern auch gleich die Methoden, die man zum Zugriff auf seine Daten braucht.

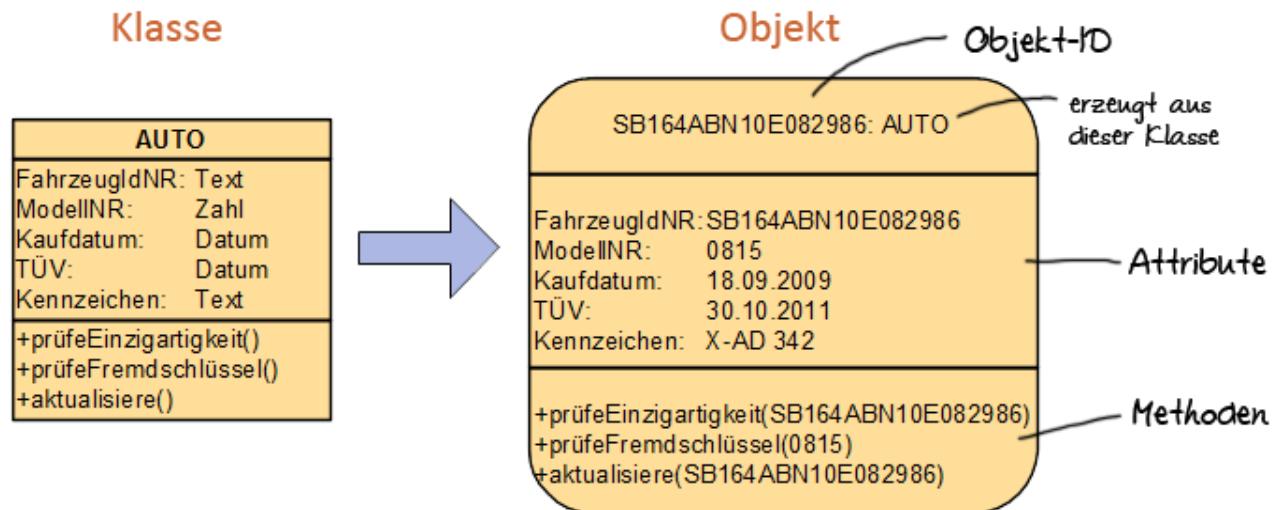


Abb. 04: Aus der Klasse AUTO erzeugtes komplexes Datenobjekt

Objektdatenbanken haben jedoch noch immer Nachteile gegenüber relationalen Datenbanken bei der Verarbeitung großer Datenmengen. Das liegt zum Teil am komplexen Zugriff auf die benötigten Datenobjekte, was vor allem bei Schreiboperationen zu schlechter Leistung (Zeitverbrauch) führt. Die Leistungsprobleme wurden in den **objektrelationalen Datenbanken** aufgegriffen, in denen nur die Teile aus objektorientierten Datenbanken mit niedrigerer Komplexität übernommen wurden.

### Fazit

Datenbankmanagementsysteme (DBMS) die auf dem **relationalen Datenbankmodell** beruhen, werden bis jetzt noch am meisten genutzt. Sie sind einfach zu handhaben und leistungsstark. Sowohl bei den serverbasierten ([MySQL](#), [PostgreSQL](#)), als auch bei den Desktop-DBMS ([OpenOffice Base](#) & [HSQLDB](#)) gibt es kostenlose Alternativen zu kommerziellen Produkten, was die Beliebtheit bei Datenbankentwicklern weiter steigert.



## 2.3.2 Datenbanksysteme II

Lerninhalte 232-02 Weitere Datenbankmodelle

Das **hierarchische Datenmodell** erlebt seit einigen Jahren jedoch ein Wiederaufblühen durch **XML**. Die Extensible Markup Language (XML), ist eine erweiterbare Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien. XML wird bevorzugt für den Austausch von Daten zwischen unterschiedlichen IT-Systemen eingesetzt, speziell über das Internet. (Anm.: Die Datei *werkzeuge.xml* findest du im Ordner Materialien)

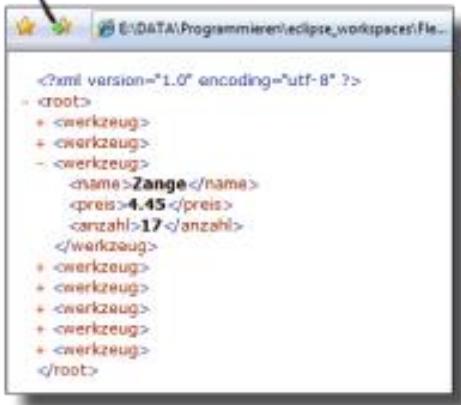
**Datei**

Name	Änderungsdatum	Größe	Typ
werkzeuge.xml	23.01.2008 17:41	1 KB	XML-Dokument

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <root>
3     <werkzeug>
4         <name>Hammer</name>
5         <preis>8.50</preis>
6         <anzahl>45</anzahl>
7     </werkzeug>
8     <werkzeug>
9         <name>Säge</name>
10        <preis>19.95</preis>
11        <anzahl>7</anzahl>
12    </werkzeug>
13    <werkzeug>
14        <name>Zange</name>
15        <preis>4.45</preis>
16        <anzahl>17</anzahl>
17    </werkzeug>
18    <werkzeug>
19        <name>Meißel</name>
20        <preis>11.75</preis>
21        <anzahl>23</anzahl>
22    </werkzeug>
23    <werkzeug>
24        <name>Hobel</name>
25        <preis>9.95</preis>
26        <anzahl>2</anzahl>
27    </werkzeug>
28    <werkzeug>
29        <name>Schleifpapier</name>
30        <preis>2.95</preis>
31        <anzahl>85</anzahl>
32    </werkzeug>
33    <werkzeug>
34        <name>Schraubendreher</name>
35        <preis>7.95</preis>
36        <anzahl>41</anzahl>
37    </werkzeug>
38    <werkzeug>
39        <name>Hacke</name>
40        <preis>29.95</preis>
41        <anzahl>12</anzahl>
42    </werkzeug>
43</root>
```

**XML-Dateien können von vielen Standardprogrammen geöffnet werden. Probier's mal mit einem Browser aus.**

**Internet Browser**



**Excel 2007**

A	B	C
1 name	preis	anzahl
2 Hammer	8,5	45
3 Säge	19,95	7
4 Zange	4,45	17
5 Meißel	11,75	23
6 Hobel	9,95	2
7 Schleifpapier	2,95	85
8 Schraubendreher	7,95	41
9 Hacke	29,95	12
10		

Abb. 05: XML-Daten in Standardanwendungen