

1.3 Datenbankmodelle

Man unterscheidet meist 5 **Datenbankmodelle**. Dies sind die relationalen Datenbanken, die objektorientierten, die hierarchischen Datenbanken, die netzwerkartigen Datenbanken und NoSQL-Datenbanken. Die Unterschiede dieser 5 Modelle liegen in der Art des logischen Aufbaus der Datenbank.

1.3.1 Relationale Datenbanken

Eine **relationale Datenbank** besteht ausschließlich aus Tabellen. Ein Zugriff erfolgt immer über diese Tabellen. Da leicht neue Tabellen hinzugefügt oder gelöscht werden können, sind spätere Änderungen des logischen Datenbankaufbaus relativ leicht möglich. Zugriffe auf Tabellen sind einfach zu programmieren, was zu der großen Beliebtheit dieses Datenbankmodells führte.

Die Zusammenhänge zwischen den einzelnen Tabellen werden über Beziehungen hergestellt. Diese Beziehungen sind in den Tabellen mit abgespeichert. Der Aufbau von Datenbeständen über Tabellen und ihre Beziehungen zueinander sind mathematisch fundiert (Relationenalgebra).

Die relationalen Datenbanken besitzen aber auch Nachteile: Zugriffe erfolgen oft über mehrere Tabellen, was längere Laufzeiten und eine hohe Anzahl von Ein-/Ausgaben zur Folge haben kann.

1.3.2 Objektorientierte Datenbanken

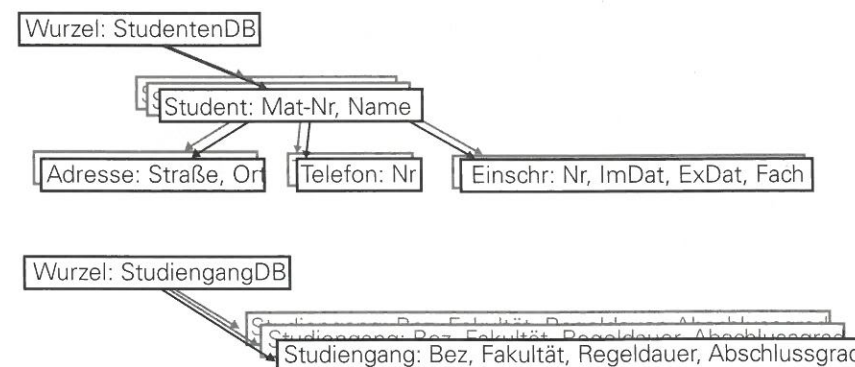
Eine **objektorientierte Datenbank** besteht ausschließlich aus Objekten. Ein Objekt ist entweder ein realer Gegenstand, z. B. ein Flugzeug, eine Person oder ganz allgemein ein abstrakter Gegenstand, etwa eine Adresse, eine Rechnung, ein Vorgang oder eine Abteilung einer Firma.

Da viele Objekte auch in Tabellenform gespeichert werden können, werden objektorientierte Datenbanken häufig als eine Erweiterung relationaler Datenbanken angesehen. Dies trifft allerdings nur teilweise zu. Schließlich gehören zu einer objektorientierten Datenbank auch objektorientierte Ansätze wie Klassen, Datenkapselungen oder Vererbungen.

Objektorientierte und objektrationale Datenbanken haben einen komplexeren Aufbau als relationale Datenbanken (fast beliebige Objekte statt einfacher Tabellen). Als Konsequenz müssen Datenbank-Designer und Anwendungsprogrammierer einen höheren Aufwand in Entwurf und Programmierung investieren. Auch die interne Verwaltung der Datenbank ist umfangreicher. Als Vorteil erhält man insbesondere bei technischen und multimedialen Anwendungen einen anschaulicheren Aufbau (komplexe Objekte müssen nicht zwangsweise auf Tabellen abgebildet werden). Dies kann erhebliche Laufzeitvorteile nach sich ziehen.

1.3.3 Hierarchische und netzwerkartige Datenbanken

Die ältesten Datenbanken sind **hierarchische Datenbanken**, eine Weiterentwicklung der konventionellen Dateiorganisation, wie sie beim PC intern genutzt wird. Der logische Aufbau dieser Datenbanken entspricht einer umgedrehten Baumstruktur. Der Zugriff erfolgt immer über die Wurzel in Richtung des gewünschten Knotens. Ein Objekt kann dabei stets nur zu einer Wurzel zugeordnet werden, was als Monohierarchie bezeichnet wird. Dies gewährleistet geringste Redundanz, da direkt über die Baumstruktur zugegriffen wird, und garantiert kürzeste Zugriffszeiten.



Beispiel für eine hierarchische Datenbank

Hierarchische Datenbanken verknüpfen die Daten über feste Beziehungen, wobei ein Datensatz auf den nächsten verweist.

Die Aufgabe von Datenbanksystemen, die Realwelt zu modellieren, ist mit dem hierarchischen Modell nur sehr begrenzt möglich. Die Beschränkung auf die Darstellung von Ober- und Unterbegriffsbeziehungen ist für eine realistische Abbildung von Alltagssituationen, in denen oft nur wenige rein hierarchische Beziehungen existieren, nicht geeignet.

Beispiel für ein hierarchisches Datenbanksystem ist IMS von IBM.

Bei **netzwerkartigen Datenbanken** besteht der logische Aufbau aus Daten, die nicht mehr rein hierarchisch, sondern über ein beliebig aufgebautes Netz miteinander in Verbindung stehen. Dies erhöht die Flexibilität erheblich, allerdings erhöht sich die Komplexität des Aufbaus.

Beide Modelle genügen den heutigen Anforderungen nicht mehr.

Die wichtigsten Vertreter netzwerkartiger Datenbanken sind IDMS (Computer Associates) und UDS (Siemens-Nixdorf).

1.3.4 NoSQL-Datenbanken

NoSQL-Datenbanken (von: *Not only SQL* = nicht nur SQL) bezeichnet Datenbanken, die einen nicht-relationalen Ansatz verfolgen. NoSQL Datenbanken verwenden keine Tabellen und versuchen Verbindungen zwischen den Daten (Joins) weitestgehend zu vermeiden.

Typische NoSQL Datenbanken sind: Cassandra von Apache, MongoDB oder Redis.

NoSQL-Datenbanken wurden zuerst für einfache Open-Source-Datenbanken verwendet, die keine SQL-Zugriffsmöglichkeit bereitstellen. Nicht relationale, verteilte Datenspeichersysteme werden häufig unter dem Begriff NoSQL aufgeführt. Die Nachteile des relationalen Datenbankmodells, z.B. Leistungsprobleme bei Indizierung großer Datenbestände oder Leistungsprobleme bei hohen Datenanforderungen/Datenänderungen können mit NoSQL-Systemen reduziert werden.

Die leistungsoptimierten NoSQL-Architekturen bieten meist nur geringe Konsistenzforderungen der Daten. Auch Transaktionen sind häufig nur auf wenige Datensätze eingeschränkt.

Typische NoSQL-Datenbanksysteme unterstützen verteilte Datenbanken mit redundanter Datenhaltung auf vielen Servern. Die Systeme können so einfach erweitert werden und Ausfälle einzelner Server überstehen.

Vorteile und Nachteile von Datenbankmodellen		
Modell	Vorteile	Nachteile
Relationale Datenbanken	leichte Änderbarkeit des Datenbankaufbaus, mathematisch fundiert, leicht programmierbar, einfache Verwaltung.	häufig viele Ein-/Ausgaben notwendig, erfordert bei großen Datenbeständen eine hohe Rechnerleistung.
Objektorientierte Datenbanken	universeller, objektorientierter Aufbau, noch relativ einfach programmierbar, einfach zu verwalten.	relativ viele Ein-/Ausgaben notwendig, komplexer Aufbau, erfordert eine relativ hohe Rechnerleistung.
Hierarchische und netzwerkartige Datenbanken	kurze Zugriffszeiten, geringe Redundanz.	Strukturänderung kaum möglich, komplexe Programmierung.
NoSQL-Datenbanken	Leistungsoptimiert für große Datenbestände und zahlreiche, zeitgleiche Zugriffe lesend und schreibend.	geringe oder keine Datenkonsistenz, hoher Aufwand für kontrollierte Transaktionen.