

## **Tabelle: Vor- und Nachteile der verschiedenen Datenbankmodelle**

Eine Datenbank kann mithilfe unterschiedlicher Modelle organisiert werden. Die bekanntesten Datenbankmodelle dabei sind hierarchische, netzwerkartige, relationale und objektbasierte Datenbankmodelle.

- Die ersten Datenbanken, die es bereits seit den 1950er-Jahren gibt, bestanden aus einer großen Datei. Solche Datenbanken werden auch als flat-file-system bezeichnet und für viele Anwendungen reichen sie im Grunde genommen aus, beispielsweise für einen Katalog oder zum Erfassen und Verwalten einer Sammlung.
- Nachteilig ist aber, dass solche Datenbanken recht unflexibel sind. In den 1960er- und 1970er Jahren entstanden daraufhin die ersten hierarchischen Datenbankmodelle, die gleichzeitig auch die Grundlage für netzwerkartige Datenbankmodelle bildeten.
- In den 1980er-Jahren wurden dann die ersten relationalen Datenbankmodelle entwickelt, objektbasierte Datenbankmodelle gibt es seit den 1990er-Jahren. Nun hat jedes Datenbankmodell seine Vor-, aber auch seine Nachteile:

	<b>Vorteile</b>	<b>Nachteile</b>
Hierarchisches Datenbankmodell	<ul style="list-style-type: none"><li>• einfache Struktur</li><li>• schneller Zugriff</li><li>• Datenintegrität und -unabhängigkeit bleiben erhalten</li><li>• auch für große Datenmengen geeignet</li></ul>	<ul style="list-style-type: none"><li>• setzt Kenntnisse der Struktur voraus</li><li>• jede Beziehung erfordert eigene Definition</li><li>• pro Satz nur ein Feld und eine Verknüpfung</li><li>• nachträgliche Strukturänderungen kaum möglich</li></ul>

<p>Netzartiges Datenbankmodell</p>	<ul style="list-style-type: none"> <li>• flexibler als hierarchische Datenbankmodelle</li> <li>• leistungsfähiger als relationale Datenbankmodelle</li> <li>• gute Integrität</li> </ul>	<ul style="list-style-type: none"> <li>• Implementierung, Erweiterung und Wartung aufwändig und kompliziert</li> <li>• wird schnell unübersichtlich</li> <li>• Datenstruktur bestimmt über Aufbau</li> </ul>
<p>Relationales Datenbankmodell</p>	<ul style="list-style-type: none"> <li>• einfach umzusetzen</li> <li>• Daten bleiben weitgehend unabhängig voneinander</li> <li>• SQL-fähig</li> </ul>	<ul style="list-style-type: none"> <li>• weniger leistungsfähig als andere Datenbankmodelle</li> <li>• keine Gewährleistung der Datenintegrität</li> <li>• fehler- und störungsanfällig</li> </ul>
<p>Objektorientiertes Datenbankmodell</p>	<ul style="list-style-type: none"> <li>• Daten können flexibel repräsentiert werden</li> <li>• unterstützt mehrdimensionale Daten</li> <li>• mehrfache Verwendung von Objekten möglich</li> </ul>	<ul style="list-style-type: none"> <li>• Implementierung recht kompliziert</li> <li>• geringe Geschwindigkeit</li> </ul>