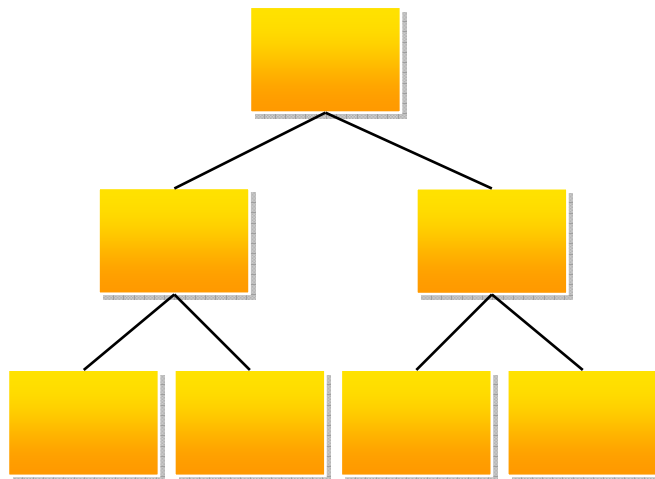


## Hierarchische Datenbanken



**Struktur**

hierarchische Baumstruktur

**Beispiele**

IMS/VS (IBM)

**Einsatz-  
gebiet**

Banken und Versicherungen

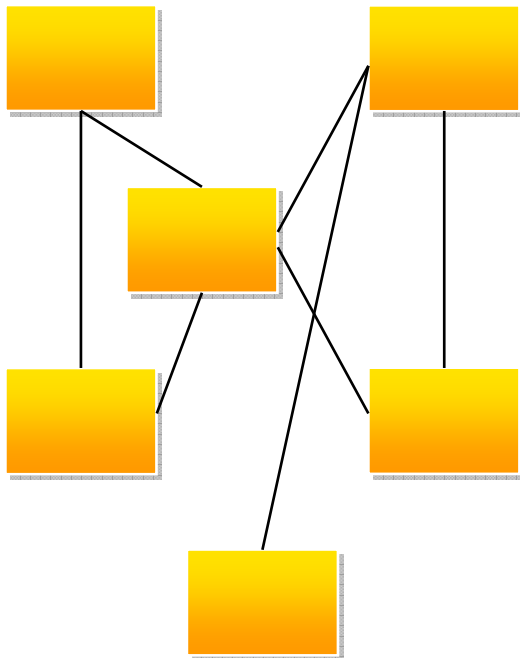
**Vorteile**

schneller Zugriff  
hohe Performance

**Nachteile**

redundante Daten  
Strukturänderungen sehr  
aufwändig

## Netzwerk-Datenbanken



**Struktur**

netzwerkförmige Struktur

**Beispiele**

IDMS (General Electric)

**Einsatz-  
gebiet**

Finanzbranche

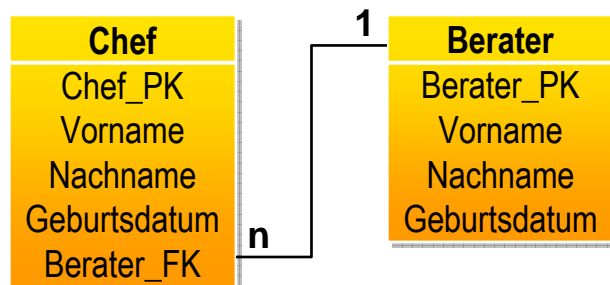
**Vorteile**

hohe Verfügbarkeit  
hoher Datendurchsatz

**Nachteile**

komplexer Aufbau  
relativ unübersichtlich

## Relationale Datenbanken (RDB)



### Struktur

DB-Typ mit Tabellen und Beziehungen

### Beispiele

Oracle, SQL-Server, MySQL, Ingres, PostgreSQL, Access

### Einsatzgebiet

zurzeit am weitesten verbreitete Datenbanktyp

### Vorteile

übersichtlich  
schnelle Datenabfrage

### Nachteile

bei Datenänderung muss Index nachgeführt werden

## SQL-Teilsprachen

**DDL**

Data Definition Language  
Befehle: CREATE, ALTER, DROP

**DML**

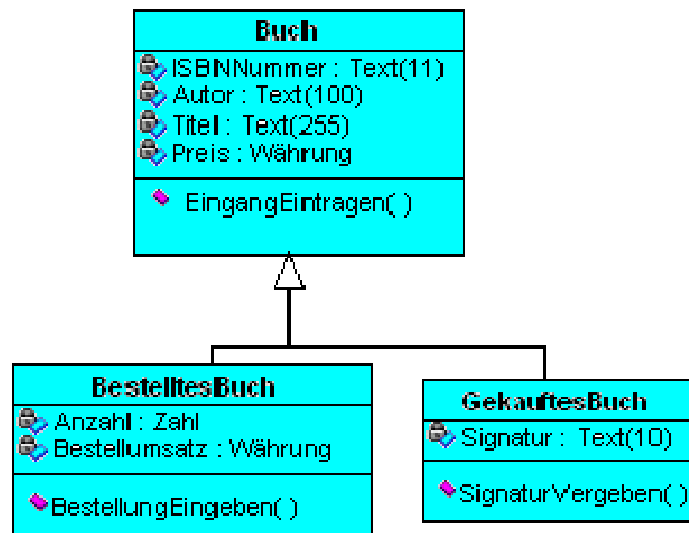
Data Manipulation Language  
Befehle: INSERT, UPDATE, DELETE

**SQL**

Structured Query Language  
Befehle: SELECT



## Objektorientierte Datenbanken (OODB)



### Struktur

enthält Informationen direkt in Form von Objekten

### Beispiele

Object-Store, O2, Caché

### Einsatzgebiet

grosse Online-Shops

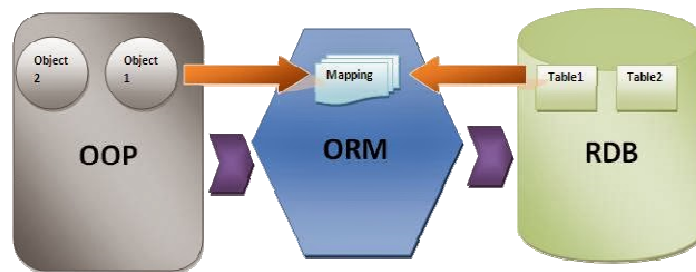
### Vorteile

kann komplexe Objekte und komplizierte Beziehungen verwalten

### Nachteile

schlechte Performance  
noch nicht sehr verbreitet

## Objektrelationale Datenbanken (ORDB)



OOP    object-oriented programming  
ORM    object-relational mapping  
RDB    relational database

### Struktur

relationale DB mit komplexen Objekten

### Beispiele

Oracle, DB2, Postgres, Informix

### Einsatzgebiet

Erfassung geographischer Daten (GIS)

### Vorteile

Kombination der Vorteile von relationalen und objekt-orientierten Datenbanken

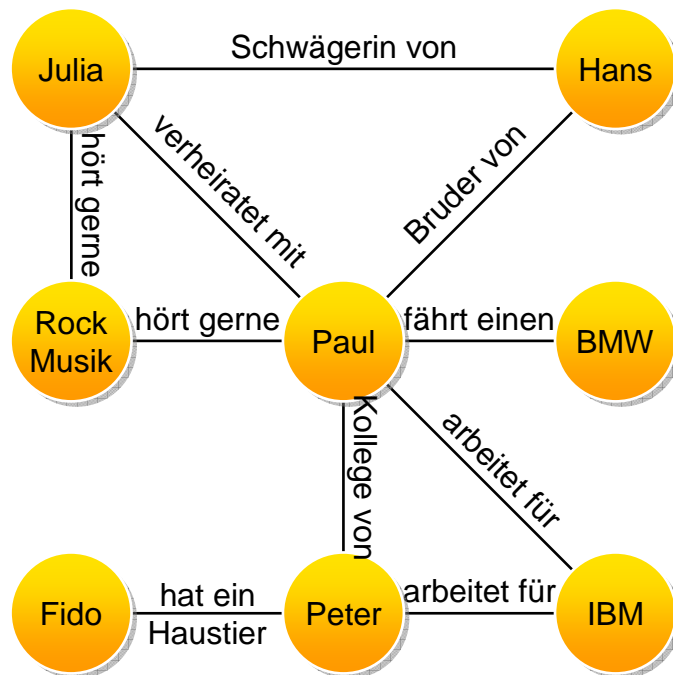
### Nachteile

Datenmodell relativ komplex, Schnittstellen noch unvollständig standardisiert

## Klassifikationsmatrix von Stonebraker

	einfache Daten	komplexe Daten
Abfragen	Relationale Datenbanken	Objektrelationale Datenbanken
keine Abfragen	Dateisysteme	Objektorientierte Datenbanken

## NoSQL-Datenbanken



### Struktur

strukturierter Datenspeicher anstelle von relationalen Tabellenschemas

### Beispiele

Apache Cassandra, Redis, CouchDB, MongoDB

### Einsatzgebiet

soziale Netzwerke, Betrugs-erkennung, Netzwerk- und Cloud-Management

### Vorteile

hohe Performance und Skalierbarkeit, für grosse Datenmengen geeignet

### Nachteile

nicht standardisiert, keine einheitliche Abfragesprache



## NoSQL-Datenbanken

### Dokumentenorientierte Datenbanken

Vorteile für die effiziente Verwaltung und Ablage von Dokumenten  
Beispiel: MongoDB

### Key-Value Datenbanken

Vorteile bei der Verwaltung von Datensätzen über einen Schlüssel bzw. "key"  
Beispiele: Oracle Coherence, Redis

### Spaltenorientierte Datenbanken

Vorteile für grosse Datenbestände, wie z.B. Data Warehouse  
Beispiele: Oracle Database Server, DB2, HBase

### Grafen- Datenbanken

Vorteile für Abbildung von Beziehungen, wie z.B. in sozialen Netzwerken oder in Fahrplänen  
Beispiel: Neo4j