

Datenbanken und SQL

Datenbank-Typen, Hersteller

(hierarchische, relationale, objektorientierte,
objektrelationale, multimedia)

Datenbank-Typen

Aktuelles Ranking der DBMS

- zum DBMS-Ranking siehe auch:
 - <https://db-engines.com/de/ranking>
 - https://go.mariadb.com/22Q1-WC-EMEA-DE-DBaaS-Gartner-MQ-Cloud-DB-Management-System-DB1128_LP-Registration.html
- unterschiedliche Arten von Datenbanken
 - Relational DBMS
 - Document Store
 - Wide Column Store
 - Key-Value Store
 - Graph DBMS
 - (Multi-Model)
 - Time Series DBMS
 - Content Store
 - Multivalue DBMS
 - Navigational DBMS
 - ...

Einordnung MariaDB im DBMS-Markt

Figure 1: Magic Quadrant for Cloud Database Management Systems



Relationale Datenbanken

- alle Daten werden in **Relationen** = zweidimensionalen Tabellen gespeichert: Spalten = Attribute, Zeilen = Tupel; die Tabellen können aufeinander verweisen = referenzieren
- verwenden strenge, langlebige, statische Schemata, bei denen die Datensätze in den einzelnen Tabellen jeweils die gleiche Struktur (= Anzahl und Art von Attributen) benötigen
- verwenden SQL; SQL ist Datenbanksprache für relationale Datenbanken
- Nachteile:
 - (1) die Abbildung von sich häufig ändernden, netzartigen Beziehungen in sehr großen Datenmengen bringt Performance-Probleme
 - (2) Änderungen des Schemas in bestehenden Datenbanken sind aufwendig und riskant

Kunden-Nr.	Name	(Bezeichner)	Ort
K011	Krause	(Wert)	Berlin
K099	Meier	10117	Berlin
K003	Müller	10120	Berlin

Spalte = Eigenschaft = Attribut

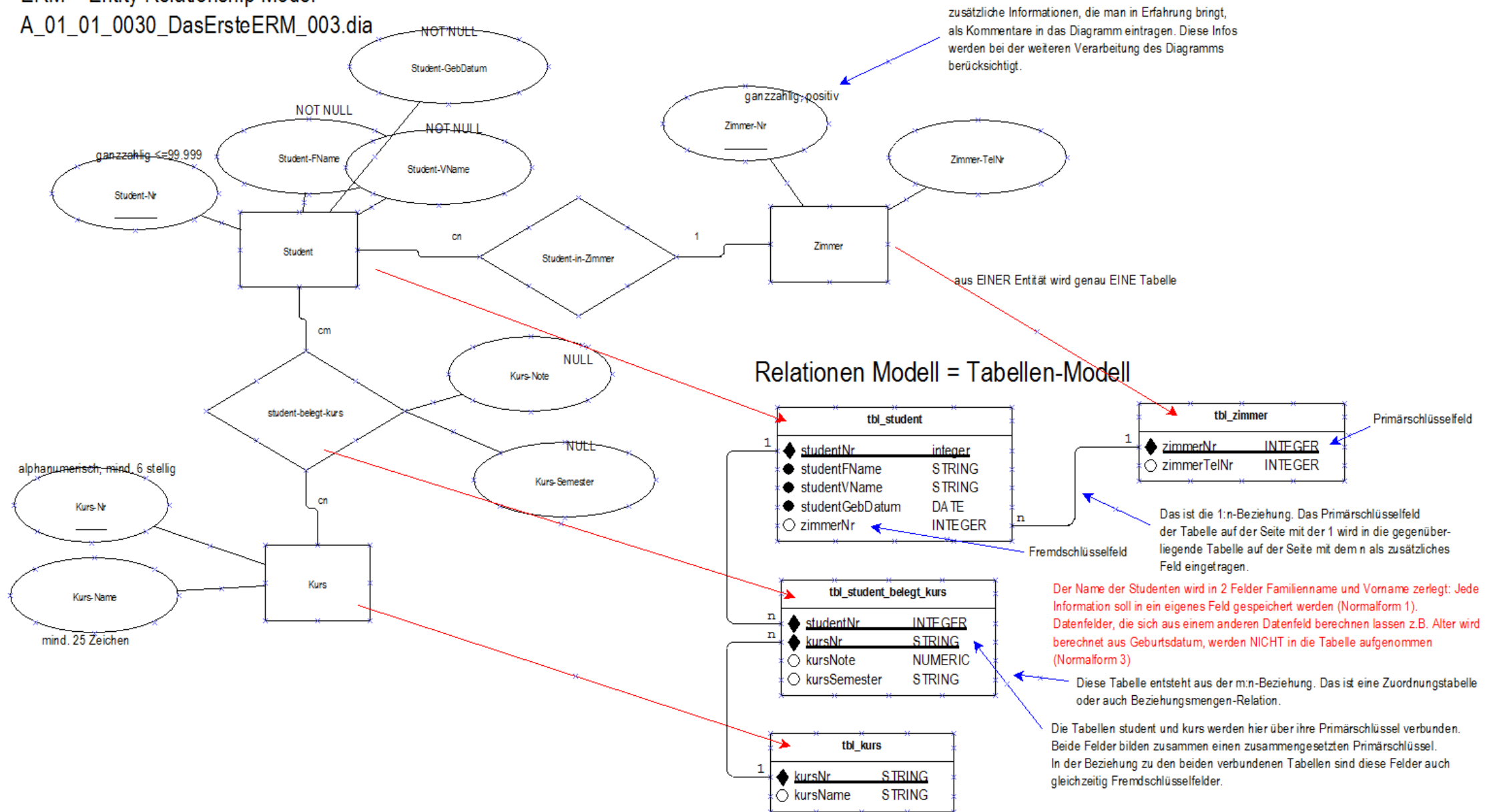
Kunden-Nr.	Name	PLZ	Ort
K001	Krause	10115	Berlin

Zeile = Tupel = zu speicherndes Objekt = Datensatz

Beispiel: Uni-Verwaltung

ERM = Entity Relationship Model

A_01_01_0030_DasErsteERM_003.dia



Key-Value Store

- speichert immer Paare von Schlüssel und Wert
- nicht für streng und statisch strukturierte Anforderungen geeignet
- ressourcenschonend und schnell
- Variante: mit sortierten Schlüsseln
- über Erweiterungen können sie Eigenschaften von Document Stores und Wide Column Stores* erhalten
- siehe auch:
 - <https://www.aerospike.com/what-is-a-key-value-store/>
 - <https://www.ionos.de/digitalguide/hosting/hosting-technik/key-value-store/>
- Vertreter: Redis, Google BigTable, Amazon DynamoDB, Memcached

* aka Spaltenorientierte DB, Datensätze mit jeweils unterschiedlich vielen dynamischen Spalten, schemafrei, ist quasi eine zweidimensionale Key-Value Store, Vertreter: Cassandra, HBase

Document Store

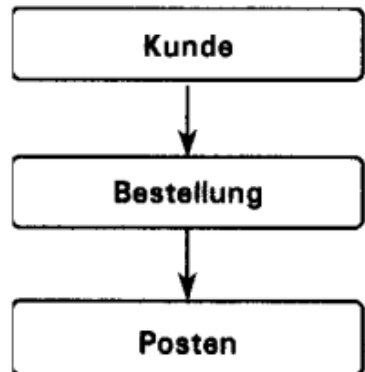
- dokumentenorientiert
- schemafrei
 - Datensätze haben keine einheitliche Struktur
 - für strukturierte Daten, aber mit sehr unterschiedlicher Anzahl von Datenfeldern, Informationen und Datentypen
 - die Daten eines Dokuments werden in Feldname(key)–Wert–Paaren gespeichert
 - je Datenfeld auch Aufzählung zulässig
 - Datensätze können untereinander eine Hierarchie haben
- Vertreter: MongoDB, CouchDB, Amazon DynamoDB

Spaltenorientierte DB, Wide Column Stores

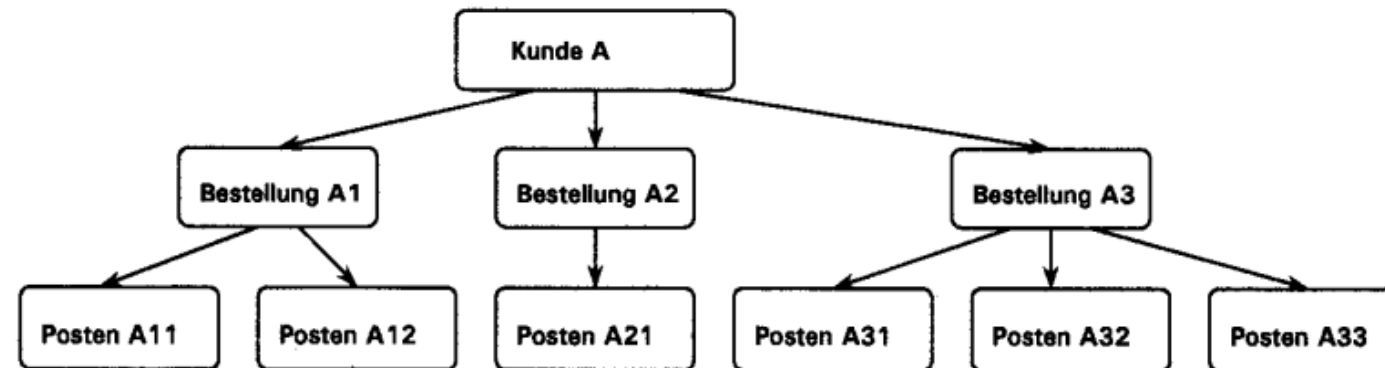
- Datensätze mit jeweils unterschiedlich vielen dynamischen Spalten
- Basis bilden Tabellen, aber die physische Speicherung erfolgt so, dass die Werte jeweils der Spalten zusammen geschrieben werden (im Gegensatz zu den Datensätzen bei relationalen Datenbanken)
- Performance-Vorteile bei analytischen Anwendungen, die nur Daten einer bestimmten Spalte (über sehr, sehr viele Datensätze) benötigen
- schemafrei
- Vertreter: Cassandra, HBase

Hierarchische Datenbanken

- unterschiedlich lange Datensätze
- baumartige Struktur mit „Knoten“ aus zusammengehörigen Datensätzen
- bilden thematische Hierarchien und Abhängigkeiten (Vater-Sohn)



Allgemeine Darstellung

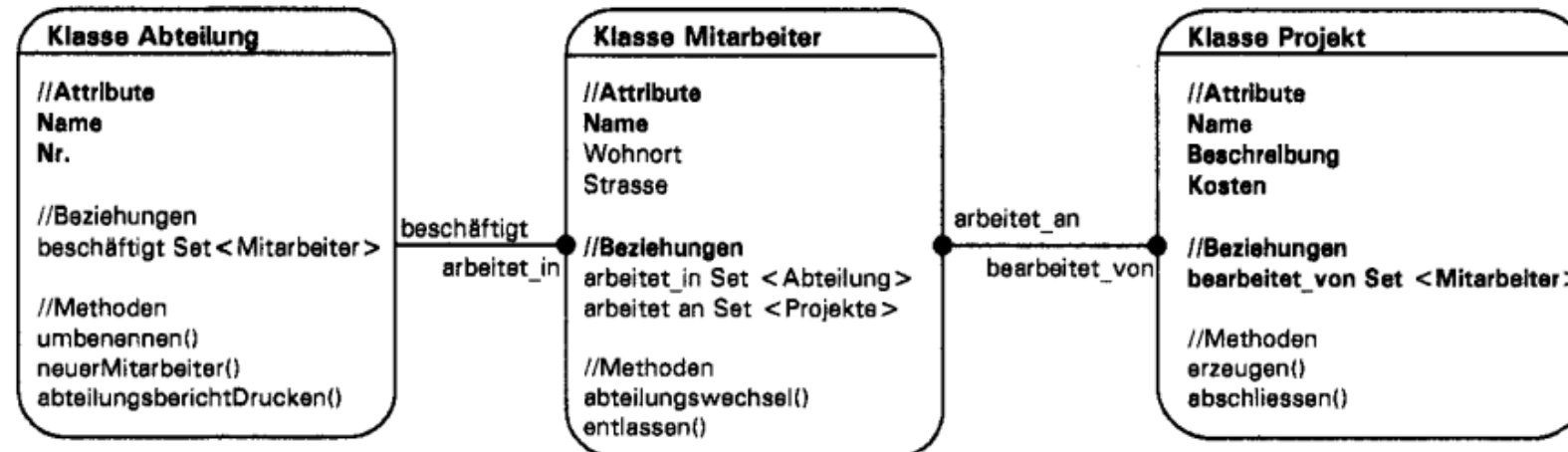


Darstellung mit Beispieldaten

Objektorientierte Datenbanken

Objekt-Relationale Datenbanken

- in Anlehnung an objektorientierte Programmiersprachen
- Objekte der realen Umwelt werden mit ihren Eigenschaften (Attribute) und ihrem Verhalten (Methoden) sowie Verweisen auf andere Objekte gespeichert (sind gekapselt). Gleichartige Objekte werden in Klassen zusammengefasst. Weitere Konzepte können realisiert sein: Vererbung, Überladung von Methoden, dynamische Bindung
- objekt-relationale DB haben Eigenschaften aus beiden Konzepten



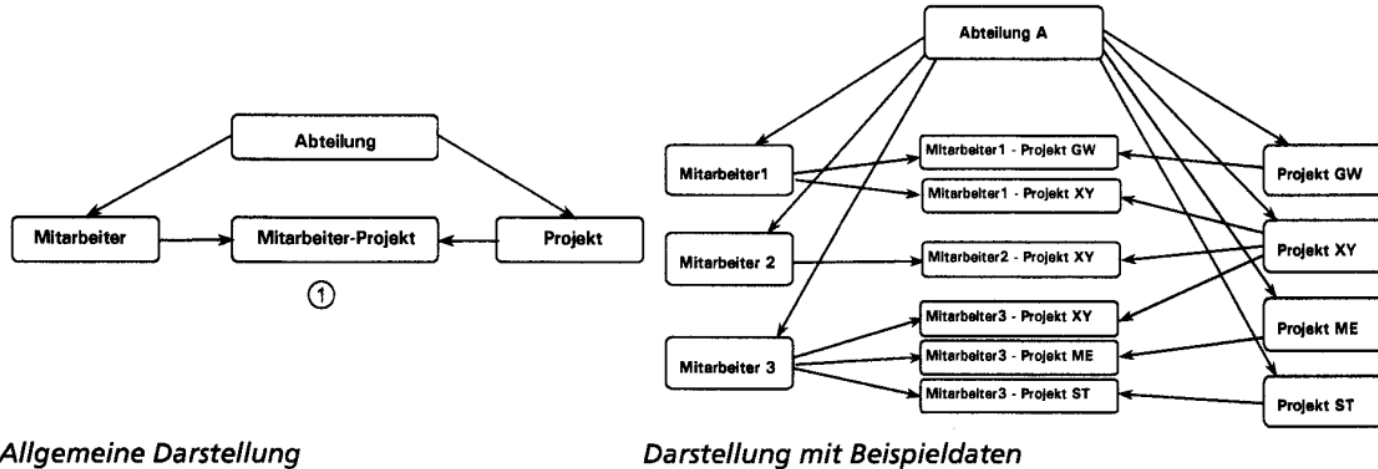
Darstellung im objektorientierten Datenbankschema

NoSQL-Datenbanken

- No ... **no** ORDER AUCH **not only** SQL-DB
- Datenbanken ohne Schema-Bindung wie bei relationalen Datenbanken
- relationale DB haben Probleme beim Verwalten von komplexen, netzartigen Beziehungen in sehr großen Datenmengen über viele Tabellen bzw. bei häufigen Schema-Änderungen
- NoSQL-DB sind:
 - dokumentenorientierte Datenbanken
 - Graphen-Datenbanken
 - Key-Value-Datenbanken

Weitere Datenbanken-Typen ...

- **Netzwerk-Datenbanken:** Speicherung gleichartiger Daten in sogenannten Recordsets, deren Records untereinander in (gerichteten) Beziehung stehen können. Die Beziehungen werden als Sets (Mengen) bezeichnet.



- **Multimedia-Datenbanken:** spezialisiert auf die Verwendung und Verwaltung beliebiger binärer (Massen-)Datentypen wie Bilder, Audio, Video; verfügen über spezielle Funktionen für inhaltliche Kennzeichnungen (z.B. über Metadaten) und das Durchsuchen (z.B. Anzeigen und Sichten), Import/ Export, Archivierung/ Versionierung

Relationale Datenbanken: Wichtige Datenbank-Hersteller

Welche Datenbanken sind in der Praxis wichtig, welche werden häufig eingesetzt? Welche Hersteller sind wichtig?

- Unterscheidung kommerzieller Systeme und OpenSource
- wichtige Hersteller von DBMS-Server
 - **Oracle:** Oracle Database 19c
 - **Oracle:** MySQL 8.0.x
 - **Microsoft:** SQL Server 2019
 - PostgreSQL 14.2
 - MongoDB 5.0.x
 - **IBM:** DB2 12.1 (2016)

siehe auch DBMS-Ranking: <https://db-engines.com/de/ranking>

- Produkte im PC-Bereich:
 - **Microsoft:** Access 2019
- von allen wichtigen kommerziellen Systemen existieren Express-Versionen (zum Lernen und Testen)

