

Vorlesung DB-Implementierung



DBMS, Architektur & Datenmodellierung

Lehrbeauftragter: Thomas Gutmann

Seminargruppe: TINF21AI1

Einführung



Lehrbeauftragter



Thomas Gutmann

- **Firma** Schwarz IT KG
- **Position** Executive Consultant
- **Bereich** Business Consultant Revision, Recht & Versicherung
- **Berufserfahrung** 19 Jahre
- **Studium** Diplom-Informatiker (FH), Wirtschaftsinformatik



Thomas.Gutmann.dhbw.ma@web.de



SCHWARZ



Lehrbeauftragter



Profil & Aktuelle Themen

▪ Profil

- Consulting
- Projektleitung
- Entwicklung

▪ Themen

- Revisionsprozesse „Filialeinsatzplanung“ & „Kapazitätsplanung“
- Versicherungs-Reporting, Umstellung von „SAP BW“ auf „Operatives Reporting“
- Open Source Software Development „OSSD“ - BIRT

Vorlesungsinhalt



Datenbanksysteme

▪ Grobgliederung

- Datenbankmanagementsysteme
- Datenbankarchitektur
- Datenmodellierung
- Structured Query Language „SQL“

▪ Datenbank, Schwerpunkt

- MS SQL Server mit T-SQL



Vorlesungen & Vorlesungsabschluss

▪ Vorlesungen

- 10 Termine
 - 9 Vorlesungen (2x 90min, 15min Pause)
 - 1 Klausur

▪ Klausur

- Dauer: 60min
- Hilfsmittel: keine

Einführung - Vorlesungskultur



Vorlesungskultur



Vorlesungskultur

 Die Vorlesung

- Ziel der Dozenten ist es, den Studenten eine optimale Lern- und Lehrsituat ion zu bieten.
 - Störungen jeglicher Art beeinträchtigen die Lern- und Lehrsituat ion und sind daher zu vermeiden.
 - Das Miteinander im Rahmen einer Vorlesung vereinbaren wir durch die „Vorlesungskultur“.

Vorlesungskultur



Vorlesungskultur

- Pünktlichkeit
(Vorlesungsbeginn, Pausenende)
- Nur eine Person redet während der Vorlesung
- Die private Nutzung des Internets während der Vorlesung/der Übungen ist untersagt.
- Kein Essen während der Vorlesung
- Handys ausschalten, keine Telefonate

Maßnahmen

- Ermahnung
- Ausschluss aus der Vorlesung
- Meldung und Vorsprache beim Studiendekan
- Meldung an das Ausbildungsunternehmen mit ggf. Arbeitsrechtl. Konsequenzen



Auszug aus aktuelle Projektthemen

„Das Operative Reporting“

/sit schwarz
IT

Übersicht



Inhalt der Vorlesungsreihe I



Datenbankmanagementsystem (DBMS)

- Von Daten zum Datenbankmanagementsystem
- Grundeigenschaften eines DBMS
- Grundbegriffe: „Daten“, „Informationen“ & Co.
- Grundthematik von Transaktionen
- Pessimistisches & Optimistisches Locking



Datenmodellierung

- Verfahren der Normalisierung
- Bedeutung der „Schlüssel“
- Datenmodellierung
- Entity-Relationship-Modell

Inhalt der Vorlesungsreihe II



SQL, Die „SQL“-Familie

- SQL, Historie & ISO-Standard
- Definition
- SQL als ISO-Standard
- SQL, „Abfragefamilie“
- OMR-Tools,
Umsetzung der SQL-Integration in Frameworks

Datentypen & Besonderheiten

- Datentypen
- Datentypen & Besonderheiten
 - Inhaltstyp „NULL“
 - Status-Spalten und ihr Wesen
- Compliance
 - Datenschutz
 - Informationssicherheit
 - Governance / IT-Governance
 - Compliance

Inhalt der Vorlesungsreihe III



SQL, Befehle der SQL-Familie

- DQL, SQL-Abfrage
- DDL, Tabellenerstellung, Änderung, Löschen
- DML, Datenerstellung, Änderung, Löschen
- DTL, Transaktionssicherung,
COMMIT & ROLLBACK
- DCL, Berechtigungskontrolle,
GRANT & REVOKE

Indexierung

- Index-Verwendung
- Definition, Index als Datenstruktur
- Indexierung, Index-Baum
- Indexsuchen, wann und wie wirkt der Index
- Zusammenfassung
- Hinweise zur Indexverwendung

Inhalt der Vorlesungsreihe IV



SQL, Advanced Feature

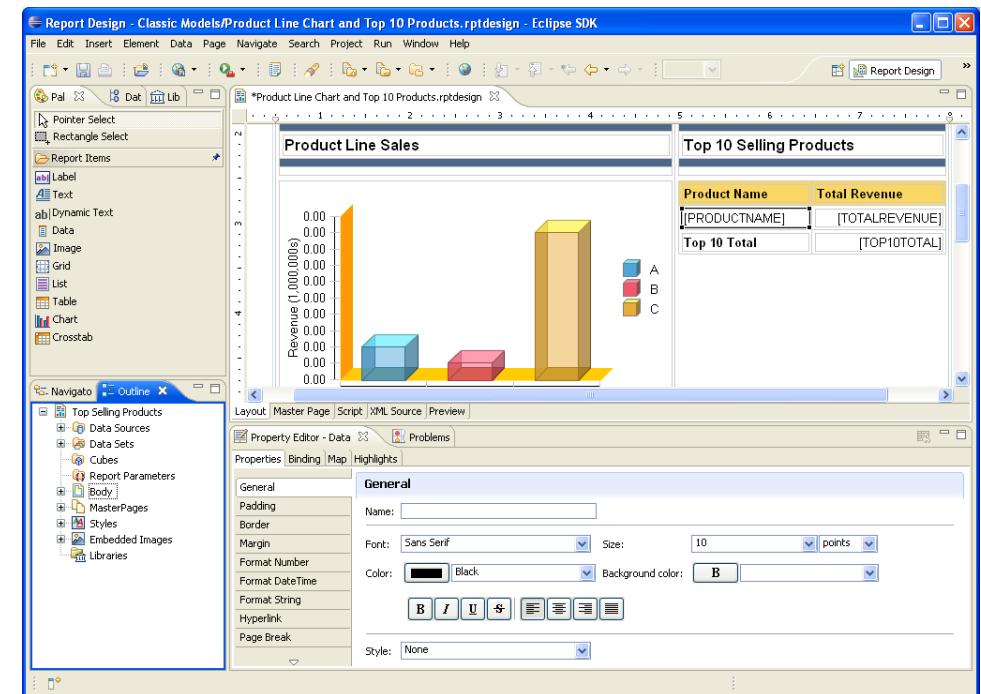
- Standardbenutzer, Datenbanken, Schema
- DDL, Tabellenerstellung, Änderung, Löschen
- Kommentare
- Parameter
- Datentyp-Konvertierungen
- Kontrollstrukturen
 - CASE
 - IF-ELSE
 - WHILE
- Computed-Columns
- Sequences
- Temporäre Tabellen
- Common Table Expression
- Funktionen (UDF) & Prozeduren
- String-Split & -Merge
- JSON-Funktionen
- Systemtabellen

Inhalt der Vorlesungsreihe V (optional)



BIRT, Business Intelligence Reporting Tool

- Kombination von Abfragen und Anwendung
- Einbindung von Datenbankabfragen
- Erstellung klassischer Reports
- Fallbeispiele



Datenbankmanagementsysteme (DBMS)

Der Ursprung, von der Antike bis Heute I/II



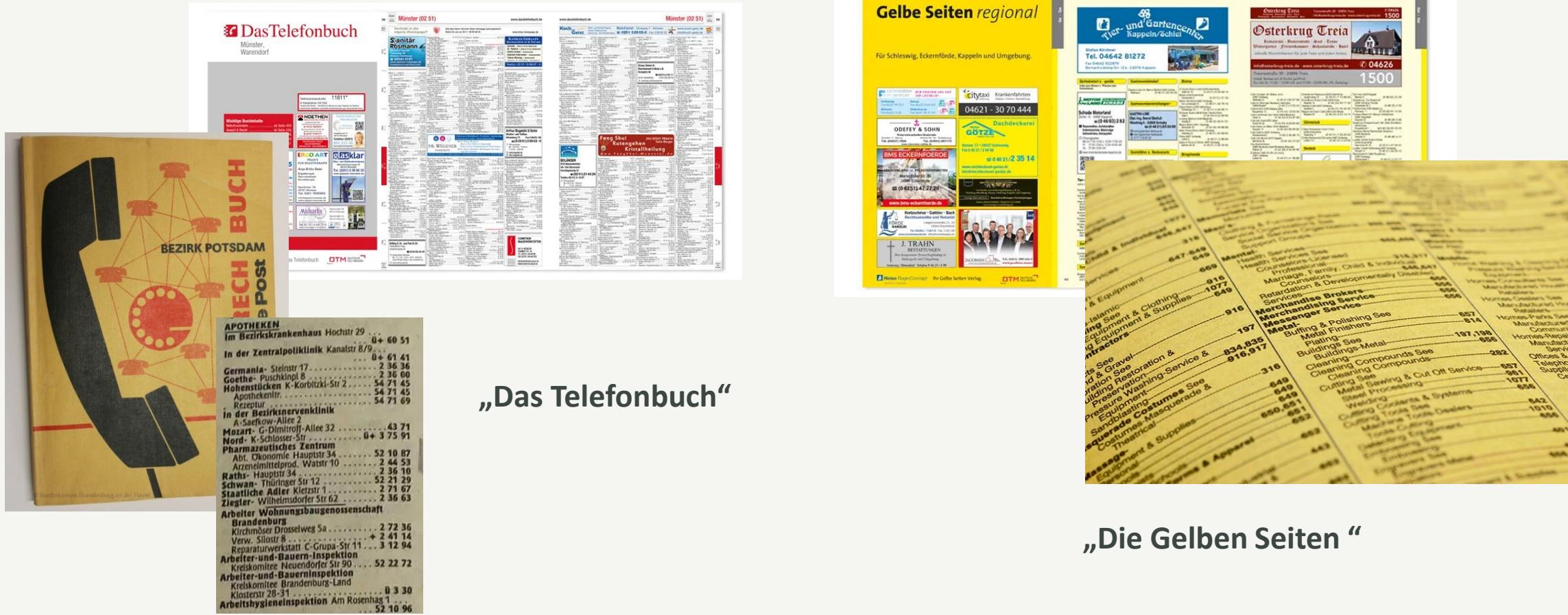
Eine unglaubliche Investition in Forschung und Kultur. Die Bibliothek von Alexandria zur Zeit ihres Höhepunktes.
Holzschnitt aus dem 19. Jh. Quelle: BILDAGENTUR-ONLINE / TH FOTO / SCIENCE PHOTO LIBRARY

Bibliothek von Alexandria



Bauern-Almanach
Almanach aus dem 19. Jahrhundert

Der Ursprung, von der Antike bis Heute II/II



„Das Telefonbuch“

„Die Gelben Seiten“



Warum Daten organisieren?



- Strukturierung von Informationen
- Verfügbarkeit von Informationen
- Verwaltung von großen Datenmengen
- Zugriff von verschiedenen Standorten
- Parallel Verwendung von Informationen
- Parallel Bearbeitung
- Übersichtlichkeit
- Steuerung wer hat wann, wie Zugriff

Begrifflichkeiten

• **Daten**

• **Informationen**

• **Skalare Daten**

• **Strukturierte Daten**

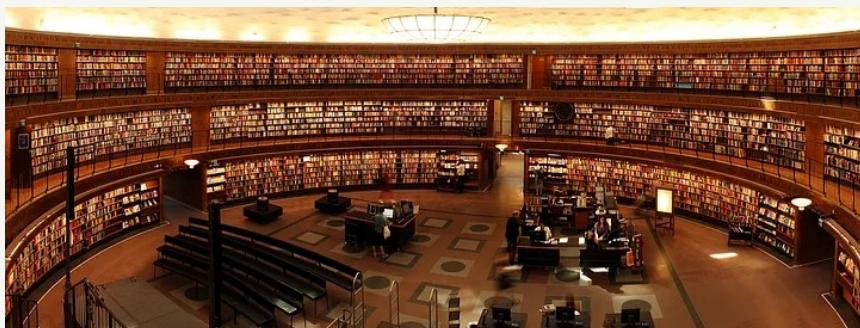
• **Unstrukturierte Daten**

- Atomare Informationen, welche in der Regel durch Beobachtungen oder Messungen erhoben werden.
- Daten die durch den Leser in einen Kontext gestellt werden.
- Daten besitzen eine feste Länge und einen definierten Datentyp, z.B. Zahlen, Zeichenketten, Datums-/Zeitwerte.
- Strukturierte Daten besitzen eine variable Länge bei einer fest strukturierten Vorgabe des inneren Aufbaus XML- & JSON-Dateien.
- In bestimmten Situationen erhöht die Normalisierung die Komplexität so, dass diese problematisch für das Laufzeitverhalten wird. Hier wird zugunsten der Systemstabilität die Normalisierung reduziert und Redundanz akzeptiert.

Datenbankmanagementsystem (DBMS)



Ein **Datenbankmanagementsystem** ist das zentrale Verwaltungssystem zur Organisation & Verwaltung von Daten, derer Organisation in unterschiedlichen Datenbanken bei gleichzeitiger Steuerung der Daten- & Benutzerberechtigungen auf die enthaltenen Datenbanken & Daten und verantwortet die Sicherstellung der Transaktionsausführung von Datenänderungen zur Gewährleistung der Datenkonsistenz.



DBMS vergleichbar mit einer digitalen Bibliothek

Harvard Library

(Quelle: <https://medium.com/@Velir/reimagining-the-harvard-library-experience-using-brand-archetypes-4335c1c69ccf>)

Typen von DBMS

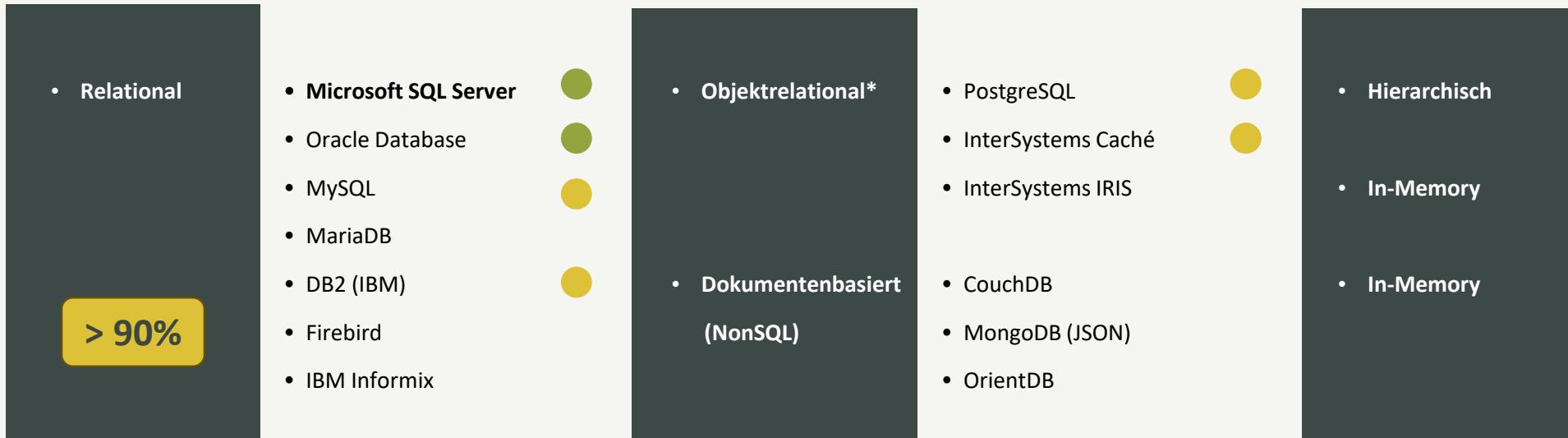


■ DBMS-Typen

- | | |
|----------------------|--|
| ▪ Relational | Organisation in Entitäten (Tabellen) und Relationen |
| ▪ Hierarchisch | Hierarchische Baumstruktur der Daten, Eltern-Kind-Beziehung |
| ▪ Netzwerkorientiert | Datenorganisation in einem Datennetz |
| ▪ Objektorientiert | Organisation der Daten als Objekt mit zugehörigen Attributen |
| ▪ Dokumentorientiert | Speicherung der Daten als Dokumentobjekt (XML, JSON, YAML) |
| ▪ NoSQL | Kombination aus netzwerk- & dokumentenorientierter Datenverwaltung |
| ▪ In-Memory-DBMS | Weiterentwicklung des klassischen DBMS mit vollständiger Datenhaltung im Arbeitsspeicher |



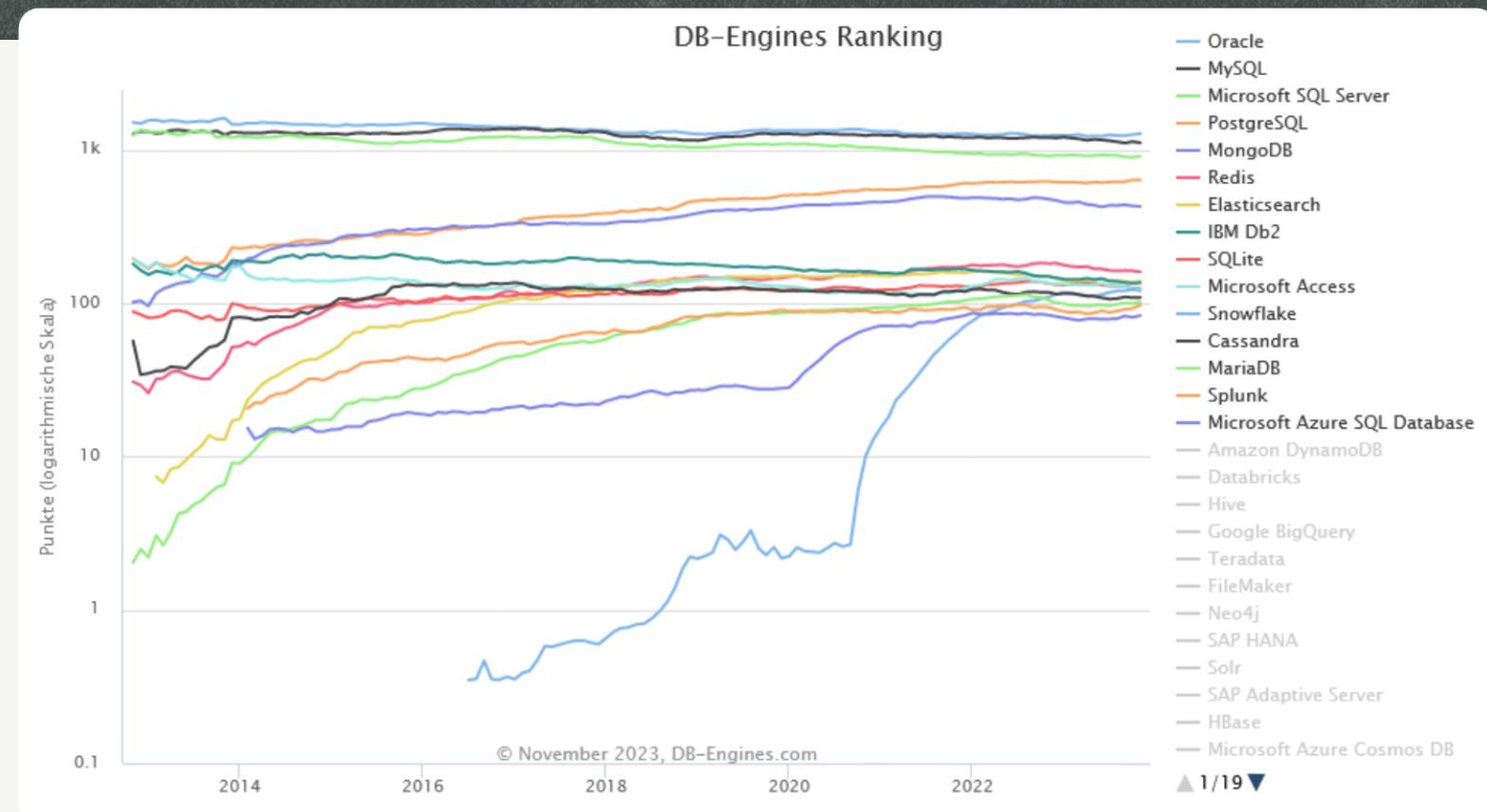
Typen von DBMS, Beispiele



*Kombination aus relationalem und objektorientiertem DBMS



DBMS im Ranking, 10-Jahre (Stand: November 2023)



(Quelle: <https://db-engines.com/de/ranking>, Stand: 15.11.2023)

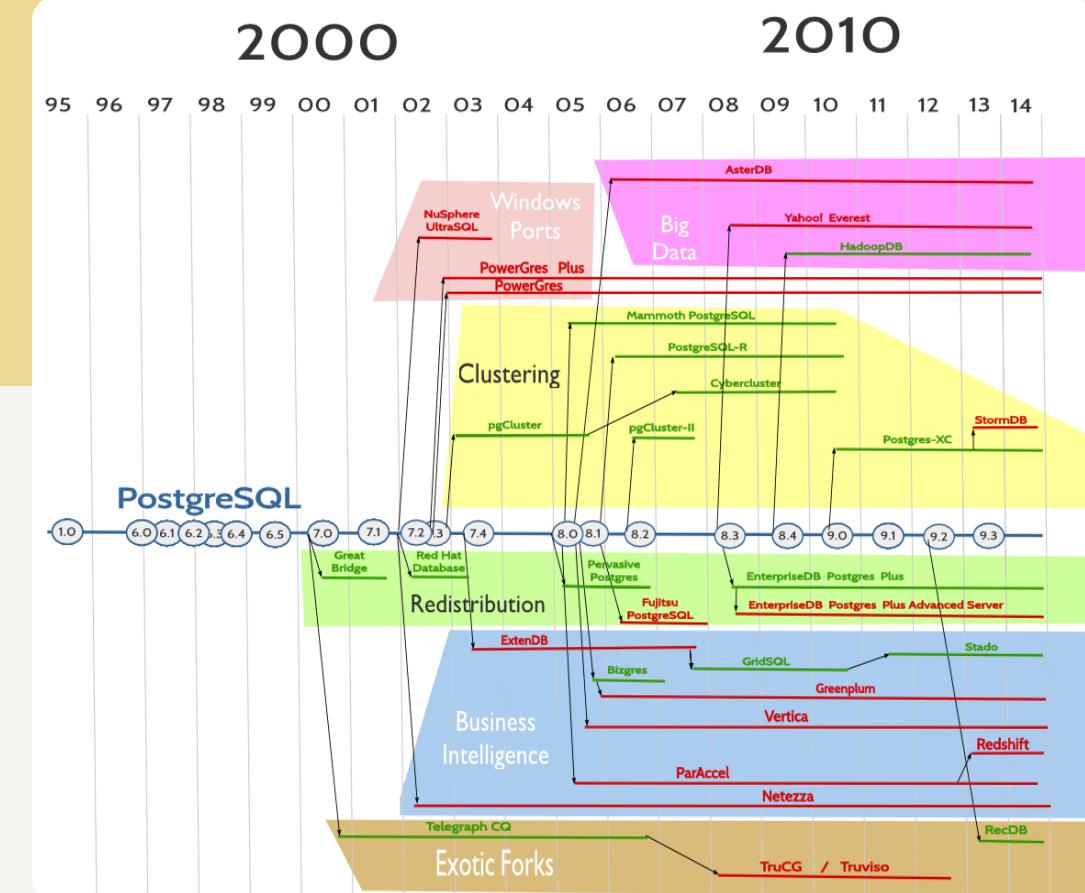
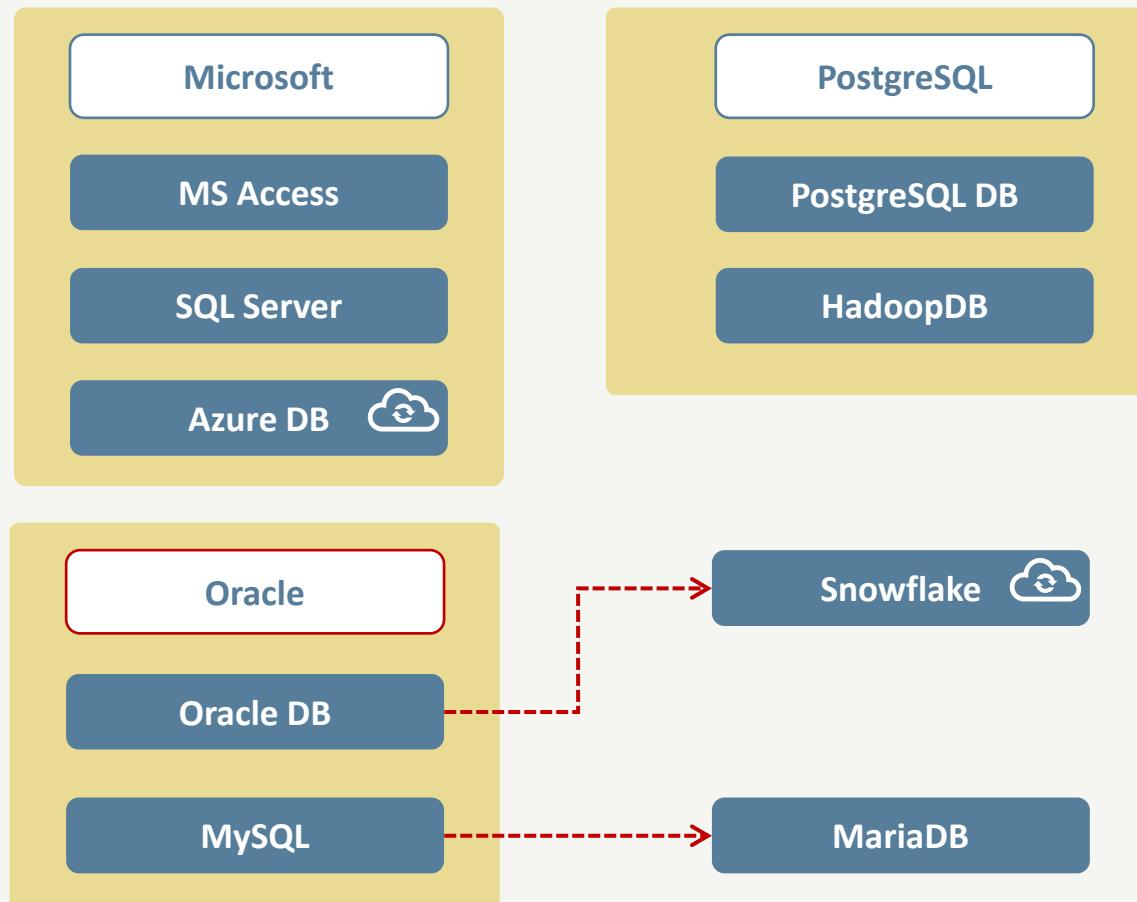


DBMS im Ranking, 12-Monatsbewegung (Stand: November 2023)

Rang			DBMS	Datenbankmodell
Nov 2023	Okt 2023	Nov 2022		
1.	1.	1.	Oracle	Relational, Multi-Model
2.	2.	2.	MySQL	Relational, Multi-Model
3.	3.	3.	Microsoft SQL Server	Relational, Multi-Model
4.	4.	4.	PostgreSQL	Relational, Multi-Model
5.	5.	5.	MongoDB	Document, Multi-Model
6.	6.	6.	Redis	Key-value, Multi-Model
7.	7.	7.	Elasticsearch	Suchmaschine, Multi-Model
8.	8.	8.	IBM Db2	Relational, Multi-Model
9.	9.	↑ 10.	SQLite	Relational
10.	10.	↓ 9.	Microsoft Access	Relational
11.	11.	↑ 12.	Snowflake	Relational
12.	12.	↓ 11.	Cassandra	Wide column, Multi-Model
13.	13.	13.	MariaDB	Relational, Multi-Model
14.	14.	14.	Splunk	Suchmaschine
15.	15.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-Model

(Quelle: <https://db-engines.com/de/ranking>, Stand: 15.11.2023)

DBMS, Datenbankfamilien, 12-Monatsbewegung (Stand: November 2023)



DBMS, das beste Datenbank-Managementsystem

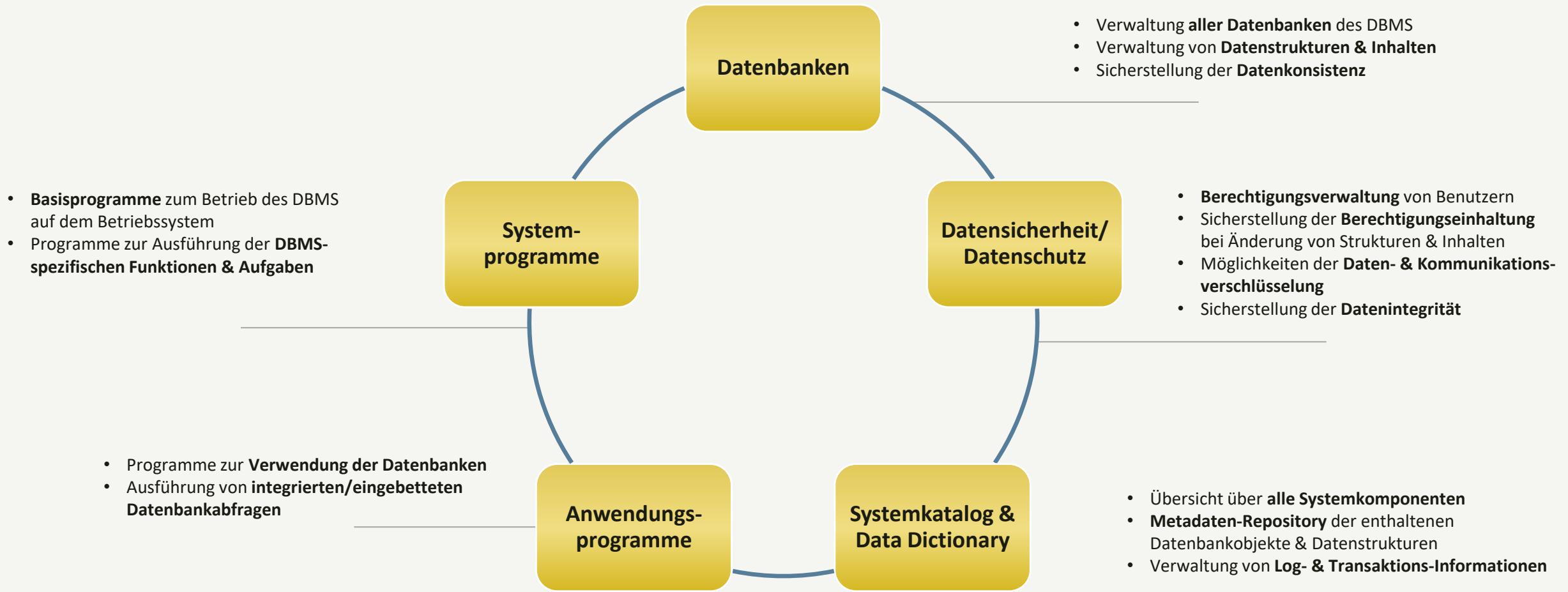


Das „beste“ Datenbankmanagementsystem

- Das „beste“ Datenbankmanagementsystem ist ein Mythos, die Ausrichtung erfolgt am „Anforderungsprofile“.
- DBMS-Neutralität kann mit ORM-Tools erreicht werden, bis die Anforderung „Reporting“ besteht.



Bestandteile eines DBMS

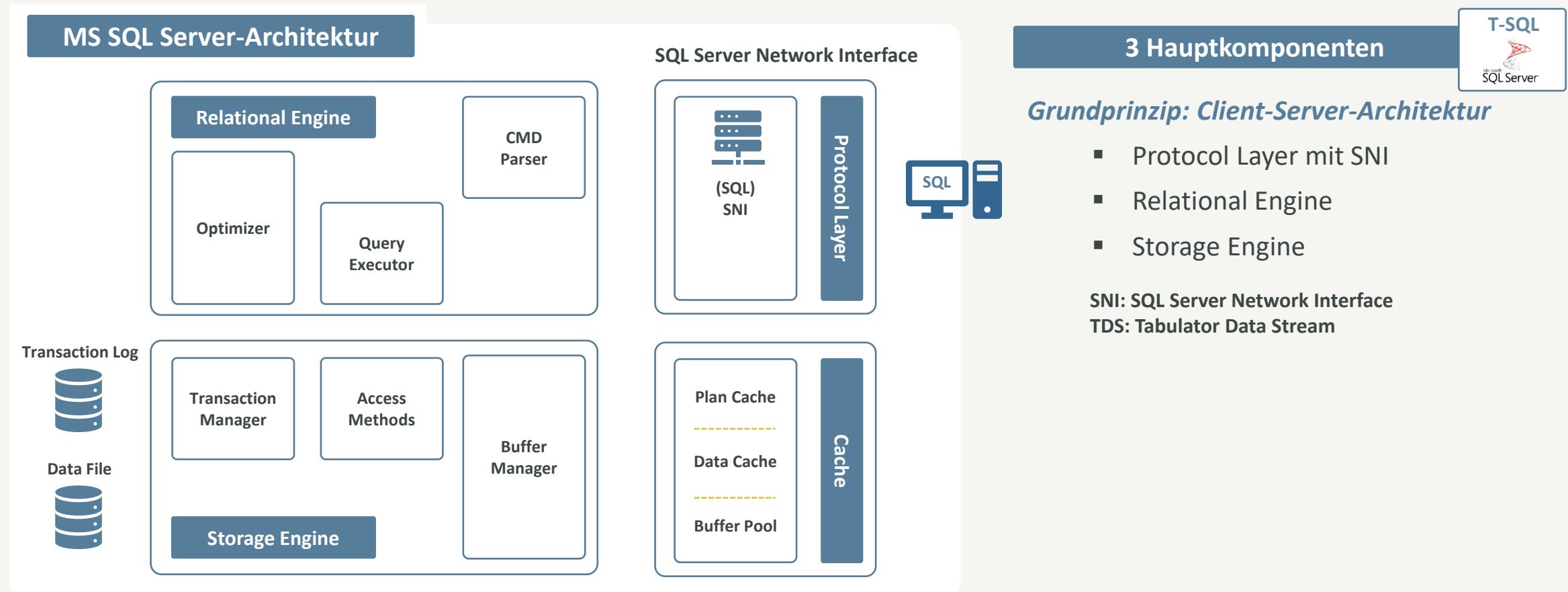


Datenbank- architektur

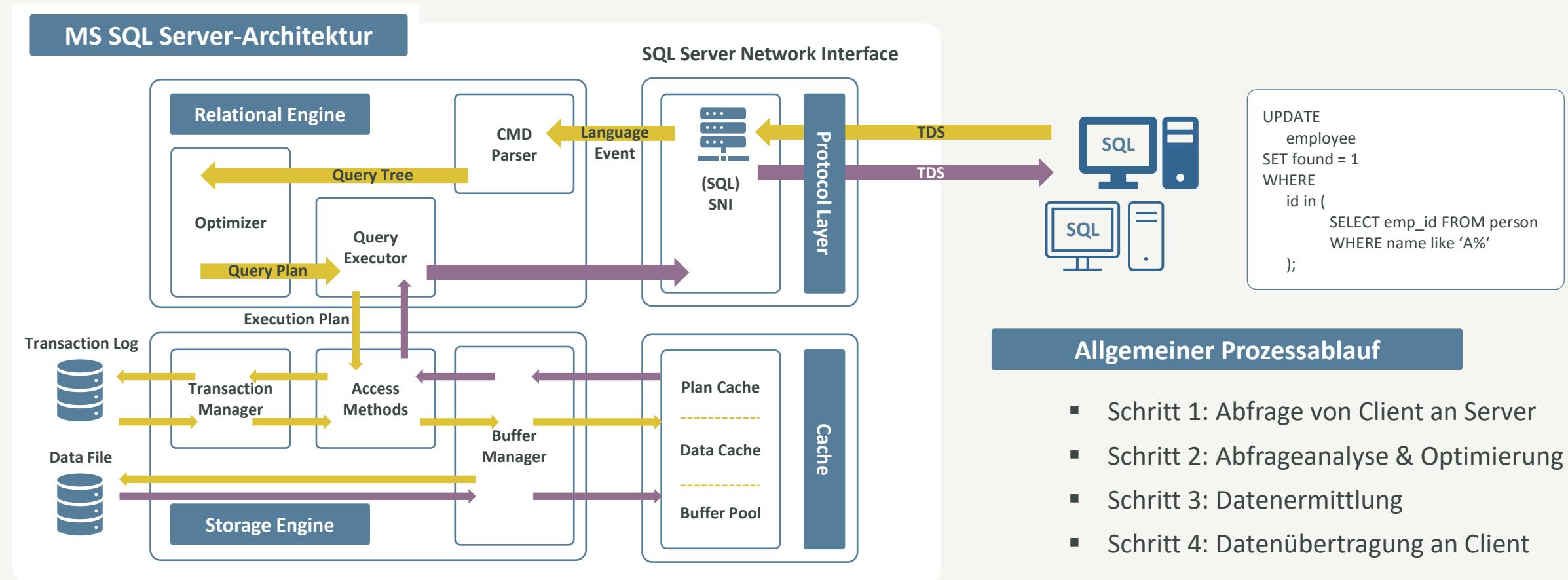


MS SQL Server

MS SQL Server-Architektur



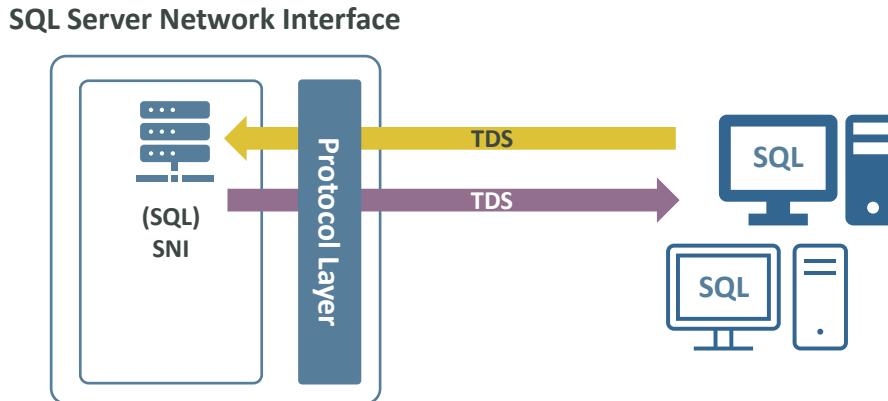
MS SQL Server-Architektur, Prozessablauf



Protocol Layer, SQL Server Network Interface



MS SQL Server-Architektur



Kommunikation von Client zu Server

- erfolgt per Tabulator Data Stream = TDS
- Kapselung der Netzwerkpakete
- Protokoll in Microsoft besitz (ursprünglich von Sybase)

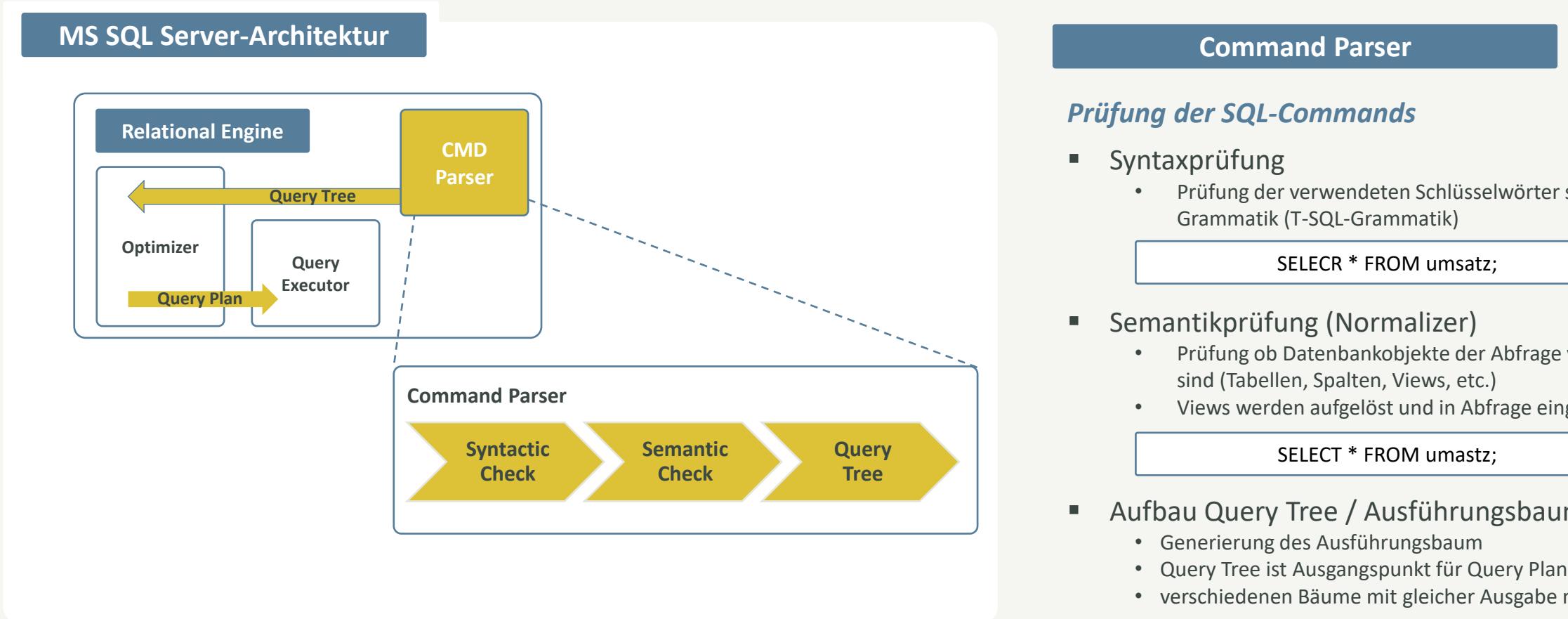
Alle Kommunikationsmöglichkeiten verwenden TDS

Client Server Kommunikation

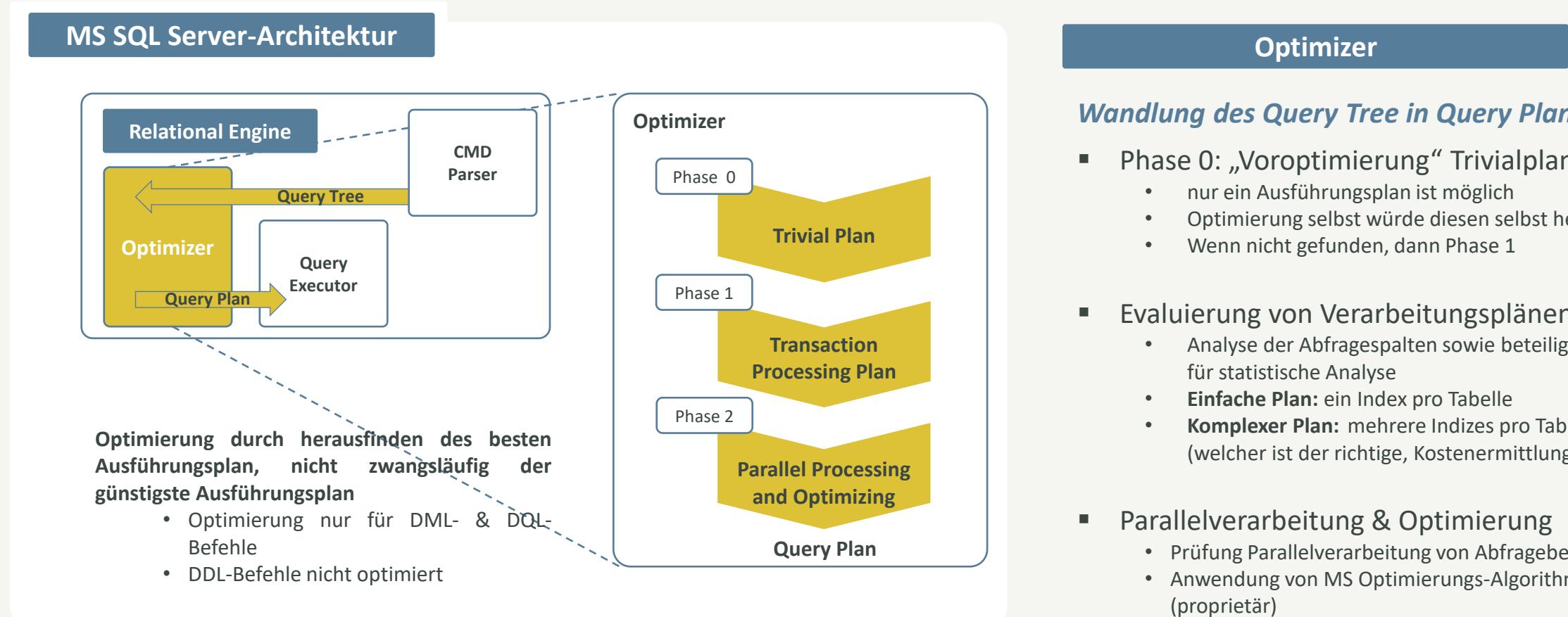
3 Kommunikationsmöglichkeiten

- Shared-Memory-Protokoll
 - Client & MS SQL Server auf dem gleichen System
 - Shared-Memory-Protokoll für Kommunikation
 - Servername: „.“,
- TCP/IP-Protokoll
 - Getrennte Installation von Client & Server
 - MS SQL Server läuft auf Port 1433 (Standard)
 - Servername: „Maschine\Instanz des Servers“
- Named Pipes
 - Kommunikation über benannte Queues/Pipes
 - Im Standard deaktiviert

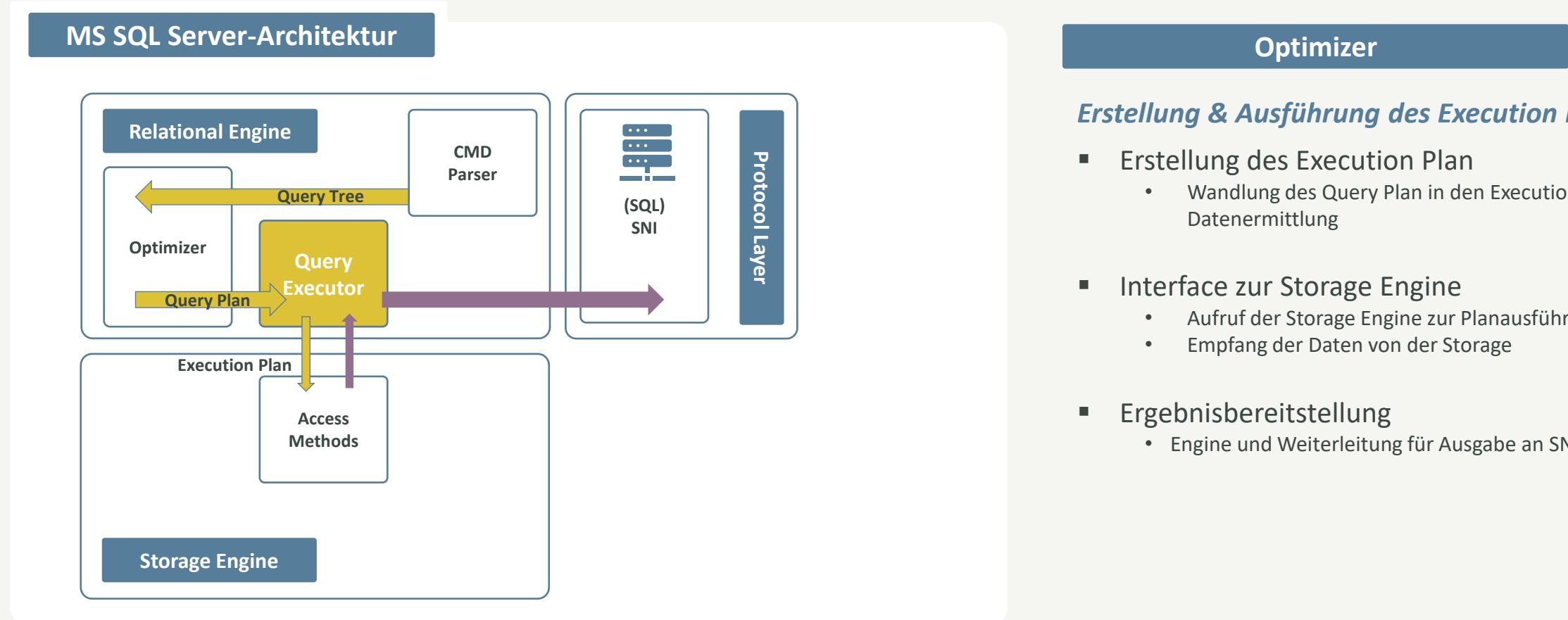
Relational Engine, CMD Parser



Relational Engine, Optimizer



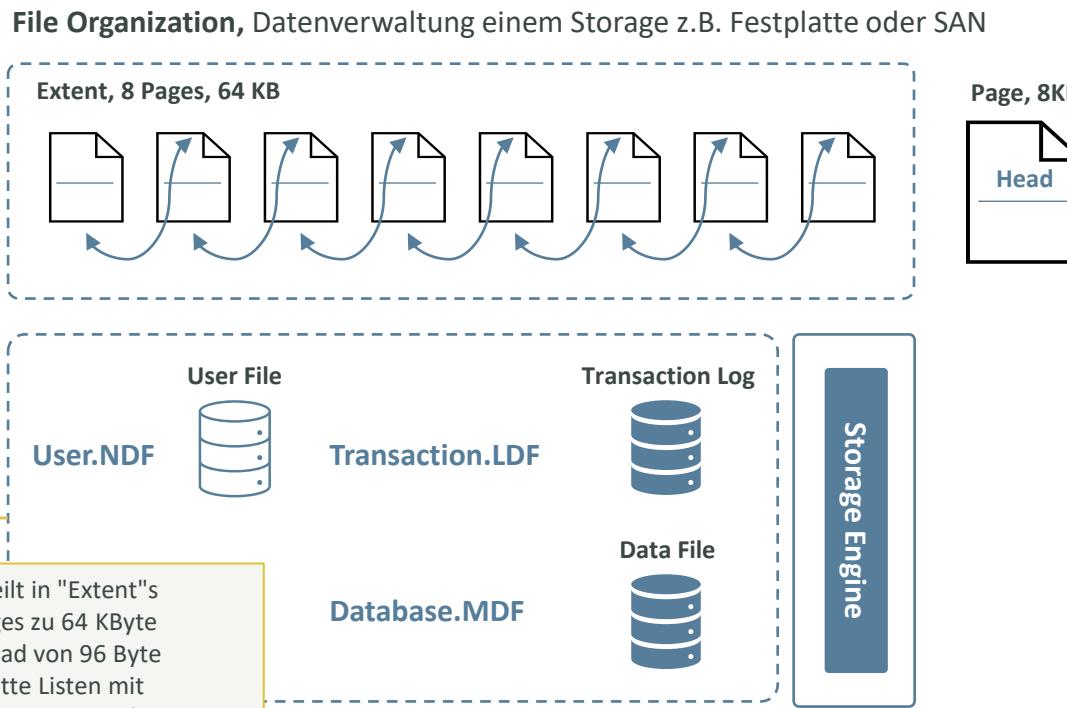
Relational Engine, Query Executor



Database Files



MS SQL Server-Architektur



Datenverwaltung (Speichern/Abrufen) der Datenbank auf dem Storage z.B. Festplatte oder SAN

Datenbank-Dateien

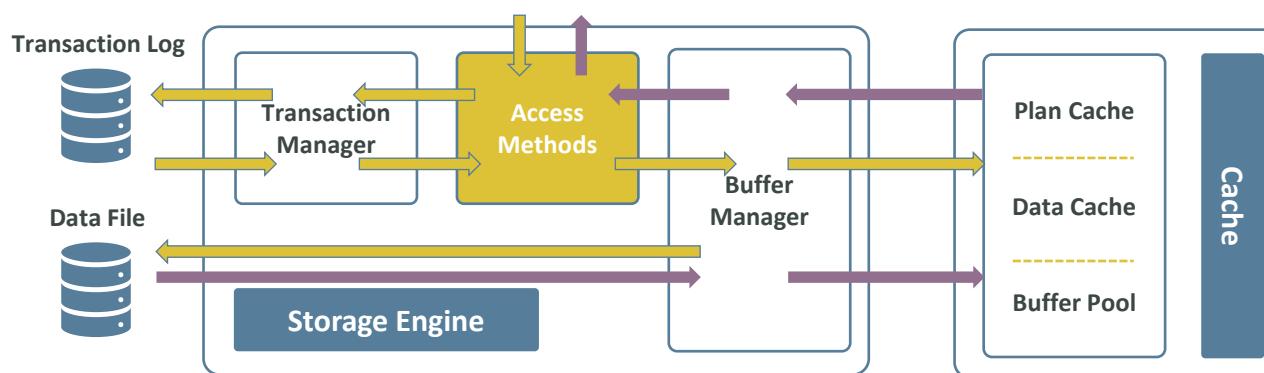
Identische technische Organisation

- Main Database File (Primary File)
 - File Extension: .MDF
 - Vorkommen: 1:N, 1 File je Datenbank
 - Speicherung der Datenbankobjekte sowie Daten (Tabellen, Ansichten, Trigger usw.)
- Secondary Database File
 - File Extension: .NDF
 - Vorkommen: 0:N
 - Speicherung benutzerspezifischer Informationen
- Ergebnisbereitstellung
 - File Extension: .LDF
 - Vorkommen: 1:N, 1 File je Datenbank
 - Enthält alle Transaktionsdaten zur Durchführung von Rollbacks nicht festgeschriebener Transaktionen "Write-Ahead-Protokolle"

Storage Engine, Access Methods



MS SQL Server-Architektur



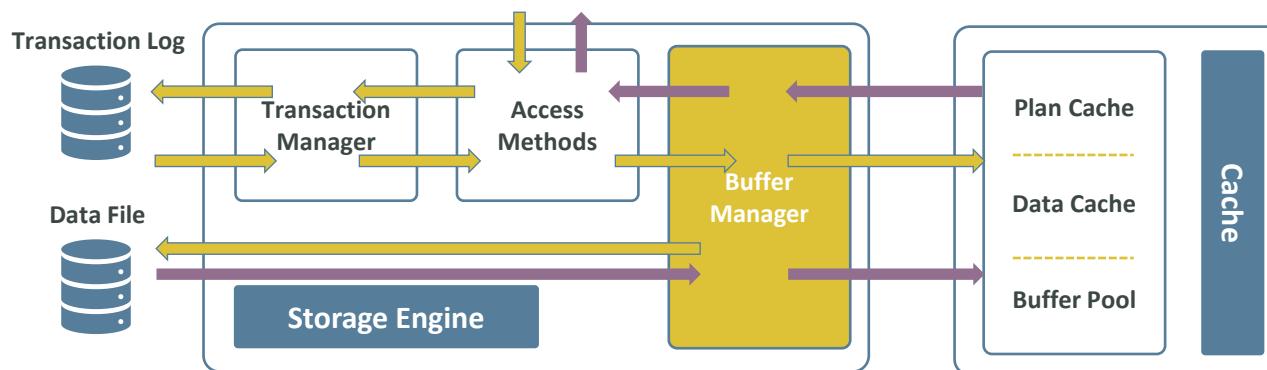
Access Methods

- **Interface Modul**
 - Koordination zwischen
 - Query Executor
 - Buffer Manager
 - Transaction Manager
- **Prüfung des Execution Plans**
 - Ausführung mit oder ohne Datenabfrage
 - Bei DQL: Call Buffer Manager
 - Bei DDL, DML: Call Transaction Manager (INSERT, UPDATE, DELETE)

Storage Engine, Buffer Manager



MS SQL Server-Architektur



Cache, "Dirty Pages"

- Caching Pages von Daten werden auch als "Dirty Pages" bezeichnet
- Direktfreigabe des Speichers von Dirty Pages bei hoher Serverlast

Buffer Manager

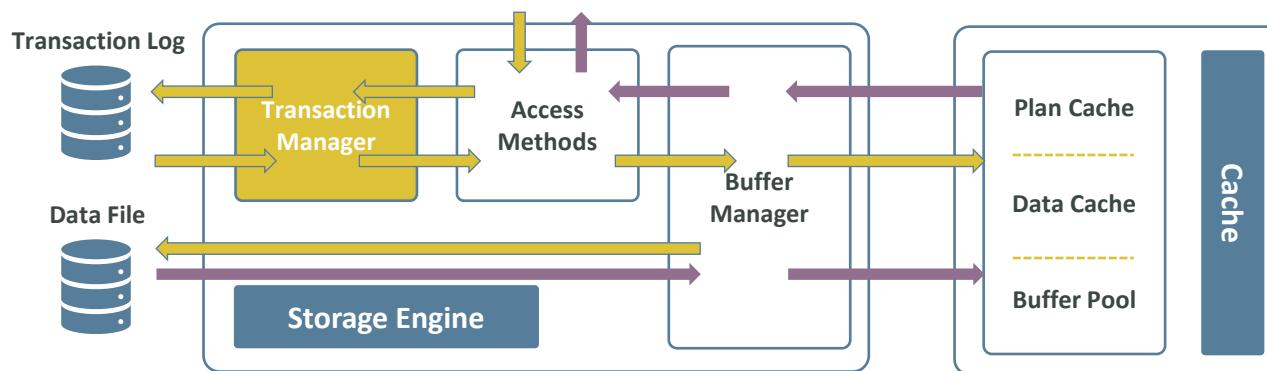
Data Load oder Cache Usage

- Plan Cache, Prüfung auf Execution Plan
 - Wenn Execution Plan vorhanden, dann verwenden
 - Wenn nicht vorhanden, zwischenspeichern
 - Executoin Plan beinhaltet vollständige Abfrage inkl. Ausführungslogik
- Data Cache, Datenbevorratung
 - Manager durchsucht nach verwendbaren Abfragedaten
 - Soft Parsing
 - Daten sind vorhanden, können verwendet werden
 - Hard Parsing
 - Daten sind nicht vorhanden, nachladen aus dem Database File
 - Daten im Cache zwischenspeichern

Storage Engine, Transaction Manager



MS SQL Server-Architektur



Transaction Manager

Aufruf bei None-DQL-Abfragen

- **Log Manager**
 - Verfolgung aller im System durchgeföhrten Aktualisierungen
 - Nachverfolgung der Transaktion bis zur Festschreibung (COMMIT) oder Zurücksetzen (ROLLBACK)
 - Log-Eintrag bestehend aus Reihenfolgennummer, der Transaktions-ID und dem Datenänderungsdatensatz
- **Lock Manager**
 - Transaktionsdaten erhalten Sperrstatus während der Transaktion
 - Entsperrung nach Festschreibung
 - Sicherstellung der Datenkonsistenz und -isolation (ACID)

MS SQL Server Integration



The screenshot shows a blog post from 'astera.com'. The title is 'MongoDB vs. SQL Server: Wie wählt man die richtige Datenbank aus?'. The post is dated March 9th, 2023. It features a sidebar with a navigation menu including 'Blogs', 'Inhaltsverzeichnis', and various links related to MongoDB and SQL Server. The main content area has a heading 'MongoDB vs. SQL Server: Programmier- und Abfragesprachen' with a yellow-bordered callout box containing text about the flexibility of MongoDB's programming languages compared to SQL Server's. Below this, there is more text about the query languages of each database.

MongoDB vs. SQL Server: Programmier- und Abfragesprachen

Bei den Programmiersprachen ist MongoDB flexibler als SQL Server. Sie können MongoDB mit verschiedenen Programmiersprachen wie JavaScript, Python, Java, PHP, C++, C, Ruby und Perl verwenden. SQL Server ist nur mit den Sprachen C, C++ und .NET kompatibel.

Die SQL Server-Datenbank verwendet leistungsstarkes SQL (Standard Query Language) zum Definieren und Bearbeiten von Daten. Die Abfragesprache von MongoDB hingegen basiert auf JavaScript, einer Sprache, mit der man leicht arbeiten kann. Es ermöglicht Ihnen, verschiedene Funktionen für MongoDB-Daten auszuführen, einschließlich Gruppieren, Überspringen, Aggregieren, Sortieren und mehr.

Wenn Sie beide Datenbanken vergleichen, kann SQL Server komplexe Abfragen übernehmen, während MongoDB aufgrund des Fehlens von Standardrückschlüssen Einschränkungen hat.

ODBC, ODBC-Treiber, Open Database Connectivity

- Standard-API zur unabhängigen Kommunikation mit Datenbankmanagementsystem (DBMS)

(Quelle: <https://www.astera.com/de/type/blog/mongodb-vs-sql-server>, Stand: 13.01.2024)

(ODBC & Java, Quelle: [https://wiki.byte-welt.net/wiki/Datenbankverbindung_\(Java\)#Access_Datenbank](https://wiki.byte-welt.net/wiki/Datenbankverbindung_(Java)#Access_Datenbank), Stand: 13.01.2024)

JDBC, JDBC-Treiber, Java Database Connectivity

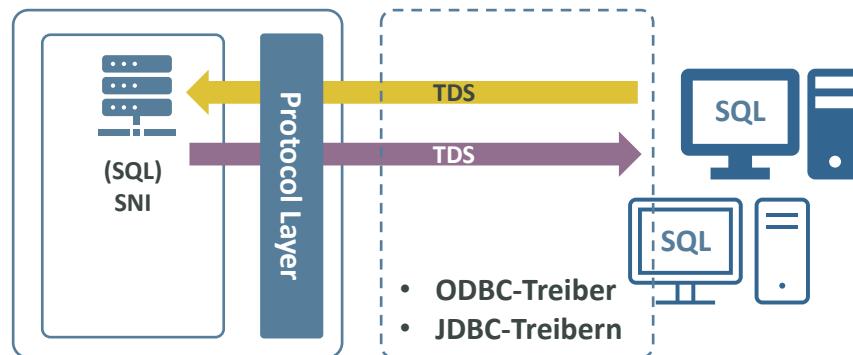
- Java-basierte Datenbank-API zur unabhängigen Kommunikation mit Datenbankmanagementsystem (DBMS)

Protocol Layer, SQL Server Network Interface



MS SQL Server-Architektur

SQL Server Network Interface



Treiber-Kommunikation zwischen Client und Server

- ODBC, ursprünglich von Microsoft entwickelt
 - Basis des Call Level Interface von X/Open und ISO/IEC
- JDBC, Umsetzung mit Java 1.1 von Sun Microsystems
- Bereitstellung der ODBC- & JDBC-Treiber durch DBMS-Hersteller

ODBC- & JDBC-Treiber

ODBC	JDBC
1. ODBC Stands for Open Database Connectivity.	1. JDBC Stands for Java database connectivity.
2. Introduced by Microsoft in 1992.	2. Introduced by SUN Micro Systems in 1997.
3. We can use ODBC for any language like C , C++ , Java etc.	3. We can use JDBC only for Java languages.
4. We can choose ODBC only Windows platform.	4. We can use JDBC on any platform.
5. Mostly ODBC Driver is developed in native languages like C, and C++.	5. JDBC Stands for Java database connectivity.
6. For Java applications it is not recommended to use ODBC because performance will be down due to internal conversion and applications will become platform-dependent.	6. For Java applications it is highly recommended to use JDBC because there are no performance & platform dependent problems.
7. ODBC is procedural.	7. JDBC is object-oriented.

Transaktions- sicherung & ACID



Transaktionen



■ Transaktion

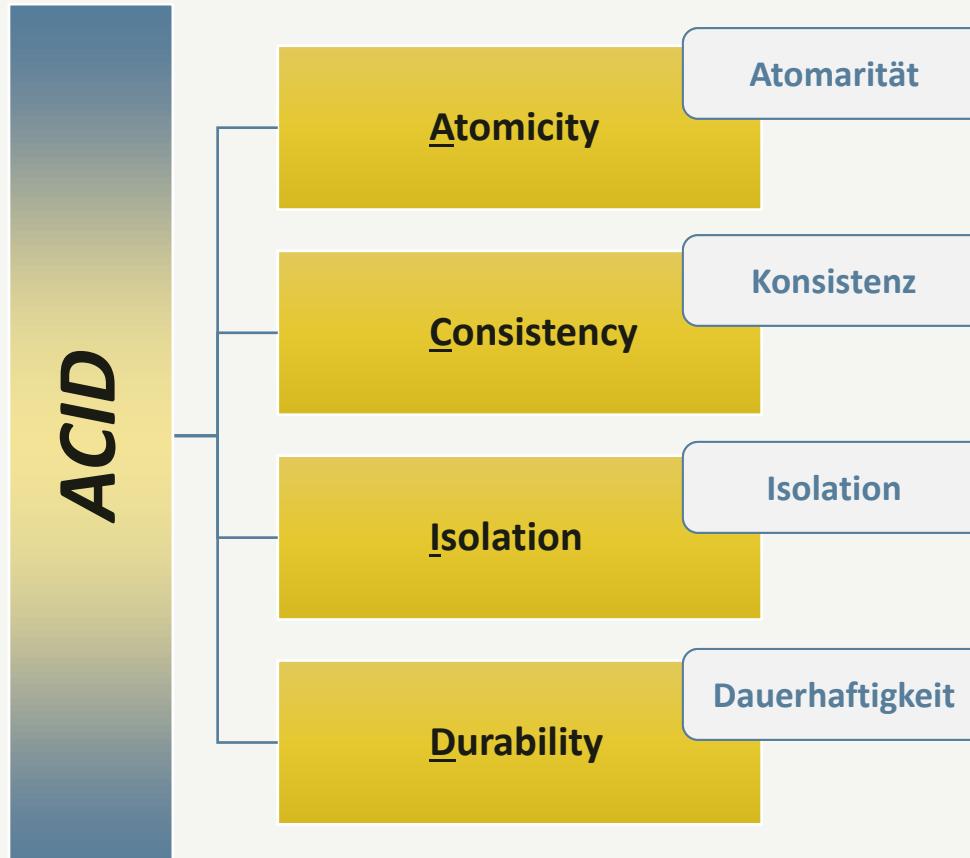
Eine Abfolge von Ausführungsschritten zur Bearbeitung von Daten nach dessen Beendigung ein fehlerfreier und vollständig konsistenten Datenzustand besteht.

Eine Transaktion folgt dem Prinzip entweder vollständig und fehlerfrei (**COMMIT**) oder gar nicht (**ROLLBACK**) ausgeführt zu werden.



Zur Bewahrung der Datenintegrität und Systemstabilität muss das Transaktionssystem eines jeden (R)DBMS die **ACID-Eigenschaften** garantieren.

ACID-Eigenschaften



- Transaktionen sind untrennbar und werden ganz oder gar nicht ausgeführt. Bei Abbruch einer atomaren Transaktion ist das System unverändert.
- Der Datenbestand weist eine konsistente Form nach der Transaktionsausführung auf.
- Parallel gleichzeitig ausgeführten Transaktionen dürfen sich nicht beeinflussen.
- Das Transaktionsergebnis muss dauerhaft persistent sein. Eine Verschachtelung von Transaktionen ist wegen dieser Eigenschaft nicht möglich. Durch das Zurücksetzen einer äußeren Transaktion könnten persistierte Ergebnisse von inneren Transaktionen widerrufen werden.



Transaktionsisolation, mögliche Anomalien

Ursache: mangelnde Transaktionsisolation

- **Lost Updates**

- 2 Transaktionen verändern gleichzeitig den identischen Datensatz.
Nach Ablauf der Transaktionen wird nur die Änderung einer Transaktion übernommen.
 - Stapelverarbeitung, Transaktions-Stack

- **Dirty Read**

- Daten einer noch nicht abgeschlossenen Transaktion werden von einer anderen Transaktion gelesen.
 - Daten erst nach Transaktions-Commit für Lese-Transaktionen freigeben

- **Non-Repeatable Read**

- Wiederholte Lesevorgänge liefern unterschiedliche Ergebnisse.
 - Sperren von Tabellen oder Datensätzen

- **Phantom Read**

- Suchkriterien treffen während einer Transaktion auf unterschiedliche Datensätze zu, weil eine andere Transaktion Datensätze hinzugefügt, entfernt oder verändert hat.
 - Leseaktionen erst nach allen Änderungsaktionen aus dem Transaktions-Stack



Transaktionsisolation, Isolation Level

DBMS: Transaktions-System

niedrig

- **Read uncommitted**

- Alle Daten können gelesen werden unabhängig vom Commit-Status
- Die Lese-Transaktionen werden freigegeben, wenn eine aktuell vorhandene Änderungs-Transaktion abgeschlossen ist.
- Alle Änderungs-Transaktionen (Schreiben, Ändern, Löschen) besitzen Ausführungsvorrang vor Lese-Transaktionen.
 - DBMS: meist genutzter Standard
- Alle Transaktionen werden von paralleler auf sequentiell Abarbeitung umgestellt, um Ergebnisverfälschungen grundsätzlich zu vermeiden.

hoch

- **Read committed**

- **Repeatable Read**

- **Serializable**



Transaktionsisolation, Locking

Ursache: mangelnde Transaktionsisolation

Pessimistic Locking

- Table-Level Locking

- Row-Level Locking

- Record-Level, Counter

- Lock-Merge

Optimistic Locking

- Sperren einer Tabelle vor der Transaktionsausführung.
Die Tabelle ist für Transaktionen so lange gesperrt bis der Unlock erfolgt.
- Datensätze während einer Transaktion sperren,
damit keine parallelen Änderungen erfolgen.
- Merkmal auf Datensatzebene zur Dokumentation der durchgeföhrten Änderungen
auf Basis einer Update-Counter-Spalte / Zeitstempel-Spalte.
 - Vor nächster Änderungstransaktion Abgleich des Wertes der Änderungsspalte
mit vorher gelesenem Wert, bei Gleichheit Transaktion ausführen bei Abweichung verwerfen
- Merkmal auf Datensatzebene zur Dokumentation der durchgeföhrten Änderungen
auf Basis einer Update-Spalte.
 - Vor nächster Änderungstransaktion Abgleich des Wertes der Änderungsspalte
mit vorher gelesenem Wert, bei Gleichheit Transaktion ausführen bei Abweichung
Änderungen auswerten und Abweichungen speichern

Methoden der Datenmodellierung

Begrifflichkeiten



- **Daten**

- Atomare Informationen, welche in der Regel durch Beobachtungen oder Messungen erhoben werden.

- **Informationen**

- Daten die durch den Leser in einen Kontext gestellt werden.

- **Redundanz**

- Mehrfache Führung einer identischen Information.

- **Normalisierung**

- Die Abbildung von gegenständlichen Sachverhalten in ihren atomaren Entitäten und ihren zugehörigen Eigenschaften / Die Modellierung von Informationen in Form von Tabellen und Relationen.

- **De-Normalisierung**

- In bestimmten Situationen erhöht die Normalisierung die Komplexität so, dass diese problematisch für das Laufzeitverhalten wird. Hier wird zugunsten der Systemstabilität die Normalisierung reduziert und Redundanz akzeptiert.



Normalisierung, Hintergrund

Beispiel als Grundlage für die Normalisierung I/II

Übersicht: DHBW-Mannheim, Organisation, Fachbereich Informatik	
• Prof. Dr. Max Müllerhausen Studiendekan Studiengangsleiter, Professor Informatik	Coblitzallee 1-9, 68163 Mannheim
• Müller, Claudia Sekretariat Informatik	Coblitzallee 1-9, 68163 Mannheim
• Gutmann, Thomas Dozent	Coblitzallee 1-9, 68163 Mannheim

Welche Informationen können aus dieser Darstellung gewonnen werden?

- Personen mit Vor- & Nachname
- Funktion(en) der Person
- Akademischer Grad (teilweise)
- Dienstanschrift

Welche Informationen können nicht aus dieser Darstellung gewonnen werden?

- Akademischer Grad (teilweise)
- Details zur Erreichbarkeit, Telefon, Email, etc.
- Öffnungszeiten des Sekretariats
- Anstellungsverhältnis an der DHBW (Anstellung, freie Dozententätigkeit, etc.)



Normalisierung, Hintergrund

Beispiel als Grundlage für die Normalisierung II/II

• Übersicht: DHBW-Mannheim, Organisation, Fachbereich Informatik

• Prof. Dr. Max Müllerhausen Studiendekan Studiengangsleiter, Professor Informatik	Coblitzallee 1-9, 68163 Mannheim
• Müller, Claudia Sekretariat Informatik	Coblitzallee 1-9, 68163 Mannheim
• Gutmann, Thomas Dozent, Nebentätigkeit, Hauptberuflich, Schwarz IT KG	Coblitzallee 1-9, 68163 Mannheim
• Gutmann, Thomas Dozent, Nebentätigkeit, Hauptberuflich, Schwarz IT KG	Coblitzallee 1-9, 68163 Mannheim

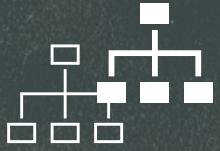
Executive Consultant, BC Revision, Recht, Versicherung

Teamleiter, BC BI & Data

• Welche inhaltliche Problemstellung aus informationstechnischer Sicht ergeben sich?

- Keine einheitliche Namensform der Personen
- Keine korrekte Trennung zwischen Namensgebung & Fachfunktion
- Grundsätzlich verschiedenen Informationen gemischt
- Mehrfachnennung der Anschrift, Redundanz
- Keine direkte & eindeutige Referenz auf eine Zeile, Datensatz möglich

Normalisierung, Definition



■ Normalisierung



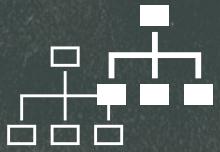
- Normalisierung ist die Abbildung von gegenständlichen Sachverhalten in ihren **atomaren Entitäten** und ihren **zugehörigen Eigenschaften**.
- Dabei erfolgt die **Modellierung** von Informationen in Form von **normalisierten Tabellen** und Relationen.

!

Alle Normalisierungsformen bauen aufeinander auf. Somit gilt, eine Normalform ist nur dann gültig, wenn deren Vorgänger gültige Normalformen sind.

Die Normalisierung wird durch 5+1Normalformen abgebildet.

Normalisierung, Hauptziele



■ Hauptziele

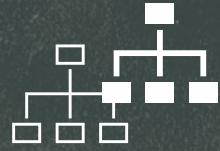


- Strukturierung von unorganisierten Informationen
- Vermeidung von Redundanzen
- Vermeidung von Inkonsistenzen
- Vermeidung von inhaltlichen Anomalien

! Durch die Normalisierung erhöht sich die **Verwendbarkeit** von Informationen, führt zur **Reduzierung** von allgemeinen **Verwaltungsaufwendungen** derer. Eine hohe **Redundanzvermeidung** wirkt gleichzeitig **reduzierend auf den Speicherbedarf** des DBMS und somit kostenvermeidend.



Normalisierung, Normalform 1 „Atomar“



■ Normalform 1

Alle Attribute müssen **atomar** (einfach und einwertig) sowie durch einen Schlüssel eindeutig bestimmbar sein

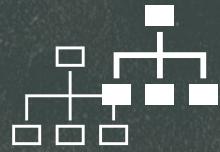
- Ein Wert ist genau einem Datenfeld zugewiesen
- Mehrfachnennungen sind durch Datensatzdopplungen aufzulösen
- Jeder Datensatz muss durch einen eindeutigen Schlüssel bestimmbar sein

• Übersicht: DHBW-Mannheim, Organisation, Fachbereich Informatik

• Prof. Dr. Max Müllerhausen Studiendekan Studiengangsleiter, Professor Informatik	Coblitzallee 1-9, 68163 Mannheim
• Müller, Claudia Sekretariat Informatik	Coblitzallee 1-9, 68163 Mannheim
• Gutmann, Thomas Dozent, Nebentätigkeit, Hauptberuflich, Schwarz IT KG	Coblitzallee 1-9, 68163 Mannheim
• Gutmann, Thomas Dozent, Nebentätigkeit, Hauptberuflich, Schwarz IT KG	Coblitzallee 1-9, 68163 Mannheim



Normalisierung, Normalform 1 „Atomar“



• Normalform 1, Variante A

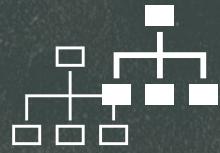
Personen-ID	Vorname	Nachname	(akadem.) Grad	Fachbereich	Funktion	Straße	Hausnummer	Postleitzahl	Ort	Ländercode	Land
1	Max	Müllerhausen	Prof. Dr.	Informatik	Studiendekan	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
2	Max	Müllerhausen	Prof. Dr.	Informatik	Studiengangleiter	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
3	Claudia	Müller		Informatik	Sekretariat	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
4	Thomas	Gutmann	Dipl.	Informatik	Dozent	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland

• Normalform 1, Variante B)

Personen-ID	Vorname	Nachname	(akadem.) Grad	Fachbereich	Funktion	Leitungsfunktion	Straße	Hausnummer	Postleitzahl	Ort	Ländercode	Land
1	Max	Müllerhausen	Prof. Dr.	Informatik	Studiendekan	Studiengangleiter	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
2	Claudia	Müller		Informatik	Sekretariat		Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
3	Thomas	Gutmann	Dipl.	Informatik	Dozent		Coblitzallee	1-9	68163	Mannheim	DE	Deutschland



Normalisierung, Normalform 2 „Sinneinheit“



■ Normalform 2

Die 1. Normalform muss erfüllt sein.
Jeder Themenbereich ist für sich in einer eigenen Tabelle abzubilden.

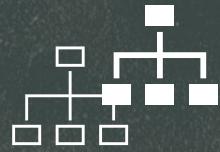
- Aufteilung in mehrere Tabellen nach Themen/ Informationsgebieten.
- Jedes Datenfeld muss eindeutig durch das Schlüsselfeld identifizierbar sein.
- Ein Datenfeld darf selbst nicht Bestandteil des Schlüssels sein.

• Normalform 1, Variante B)

Personen-ID	Vorname	Nachname	(akadem.) Grad	Fachbereich	Funktion	Leitungsfunktion	Straße	Hausnummer	Postleitzahl	Ort	Ländercode	Land
1	Max	Müllerhausen	Prof. Dr.	Informatik	Studiendekan	Studiengangleiter	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
2	Claudia	Müller		Informatik	Sekretariat		Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
3	Thomas	Gutmann	Dipl.	Informatik	Dozent		Coblitzallee	1-9	68163	Mannheim	DE	Deutschland



Normalisierung, Normalform 2 „Sinneinheit“

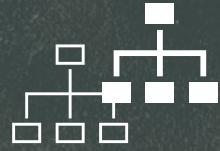


• Tabelle Personen						
PK	Personen-ID	Vorname	Nachname	(akadem.) Grad	Fachbereich	Funktion
	1	Max	Müllerhausen	Prof. Dr.	Informatik	Studiendekan
	2	Claudia	Müller		Informatik	Sekretariat
	3	Thomas	Gutmann	Dipl.	Informatik	Dozent

• Tabelle Adresse						
PK	Adress-ID	Straße	Hausnummer	Postleitzahl	Ort	Ländercode
	1	Coblitzallee	1-9	68163	Mannheim	DE
	2	St. Blasius Weg	31a	56458	Mühlhausen	DE
	3	Favoritenstraße	1-9	1100	Wien	AT



Normalisierung, Normalform 3 „Indirektion“



■ Normalform 3

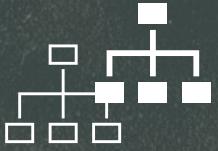
Die 2. Normalform muss erfüllt sein.
Transitive/Indirekte Abhängigkeiten müssen aufgelöst werden.

- Aufteilung von indirekten Abhängigkeiten in eigenständige Tabellen.
- Die Referenzierung muss durch einen eindeutigen Datensatzschlüssel erfolgen.

• Tabelle		Adresse						
	Adress-ID	Straße	Hausnummer	Postleitzahl	Ort	Ländercode	Land	
	1	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland	
	2	St. Blasius Weg	31a	56458	Mühlhausen	DE	Deutschland	
	3	Favoritenstraße	1-9	1100	Wien	AT	Österreich	



Normalisierung, Normalform 3 „Indirektion“



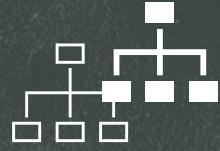
• Tabelle Adresse						
Adress-ID	Straße	Hausnummer	Postleitzahl	Ort	Ländercode	Land
1	Coblitzallee	1-9	68163	Mannheim	DE	Deutschland
2	St. Blasius Weg	31a	56458	Mühlhausen	DE	Deutschland
3	Favoritenstraße	1-9	1100	Wien	AT	Österreich





Normalisierung, Boyce-Codd-Normalform

Weiterentwicklung der NF3



■ Normalform 3,5

Die 3. Normalform muss erfüllt sein.
Schlüsselkandidaten müssen ersetzt oder aufgetrennt werden,
wenn direkte Abhängigkeiten zu Datenfelder bestehen.

- Eine Verletzung der BCN liegt potenziell vor, wenn mehrere Schlüsselkandidaten mit überlappenden Attributen existieren.
- Ein Attribut oder eine Gruppe von Attributen von dem andere voll funktional abhängen, heißt **Determinante**.
- Änderungsanomalien können bei der BCN-Verletzung auftreten.

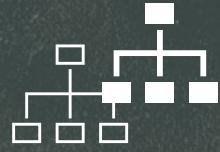
Zusammengesetzter
Primärschlüssel



• Tabelle		Lieferant		
Lieferanten-Nr. Name		Artikel-Nr.	Preis	Artikelbezeichnung
PK-Part-01		PK-Part-02		
10001	Baumwoll-Produktion GmbH	45.4578.01	40,00 EUR	Pullover (kurzarm)
10002	Creative Shirt GmbH	45.6678.02	50,00 EUR	Pullover (langarm)
10001	Baumwoll-Produktion GmbH	88.6678.03	70,00 EUR	Rollkragenpullover



Normalisierung, Boyce-Codd-Normalform



Weiterentwicklung der NF3

• Tabelle		Lieferant		
		PK		
Lieferanten-Nr.	Name	Artikel-Nr.	Preis	Artikelbezeichnung
PK-Part-01		PK-Part-02		
10001	Baumwoll-Produktion GmbH	45.4578.01	40,00 EUR	Pullover (kurzarm)
10002	Creative Shirt GmbH	45.6678.02	50,00 EUR	Pullover (langarm)
10001	Baumwoll-Produktion GmbH	88.6678.03	70,00 EUR	Rollkragenpullover

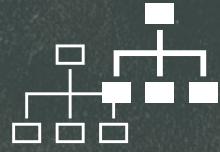
- Lieferant: ermittelbar mit Schlüssel-Teil 1
- Primärschlüssel Lieferant für Artikel nicht ausreichen
 - also zusammengesetzter Schlüssel, dadurch bedingt mehrere abhängige Gruppen
- Artikelpreis: ermittelbar mit Schlüssel-Teil 1 & -Teil 2

Änderungsanomalie

• Tabelle		Lieferant		
		Lieferanten-Nr.		Artikel-Nr.
Lieferanten-Nr.	Name	PK-Part-01	PK-Part-02	Artikelbezeichnung
10001	Wintersachen GmbH & Co.KG		45.4578.01	40,00 EUR
10002	Creative Shirt GmbH		45.6678.02	50,00 EUR
10001	Baumwoll-Produktion GmbH		88.6678.03	70,00 EUR



Normalisierung, Boyce-Codd-Normalform



Weiterentwicklung der NF3

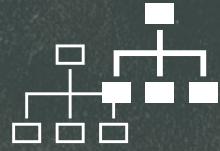
• Tabelle		Lieferant			
PK		Lieferanten-Nr.	Name		
		10001	Baumwoll-Produktion GmbH	• NF1: Erstellung von <u>eindeutigen</u> Schlüsseln	
		10002	Creative Shirt GmbH	• NF2: Aufteilung in Sinneinheiten	
		10003	Wintersachen GmbH & Co.KG	• NF3: Aufteilung von <u>indirekten</u> Abhängigkeiten	

• Tabelle		Artikel			
PK		Artikel-ID	Artikel-Nr.	Lieferanten-Nr.	Preis
		1	45.4578.01	10001	40,00 EUR
		2	45.6678.02	10002	50,00 EUR
		3	88.6678.03	10001	70,00 EUR

FK



Normalisierung, 4. Normalform „Funktional-Relation“



■ Normalform 4

Die BCN-Normalform muss erfüllt sein.

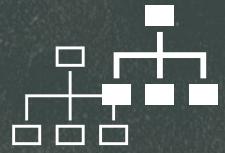
Alle funktionalen Attribute sind durch Relationen und somit nicht funktionale Attribute abzubilden.

- Redundanzen von funktionalen Attributen sind aufzulösen.
- Die Auflösung erfolgt durch die Definition von zugehörigen Definitionstabellen.
- Somit entsteht eine neue triviale Funktions-Relation zwischen Definitions- & Zieltabelle.

• Tabelle	Adresse					
	Adress-ID	Straße	Hausnummer	Postleitzahl	Ort	Land-ID
	1	Coblitzallee	1-9	68163	Mannheim	1
	2	St. Blasius Weg	31a	56458	Mühlhausen	1
	3	Favoritenstraße	1-9	1100	Wien	2



Normalisierung, 4. Normalform „Funktional-Relation“

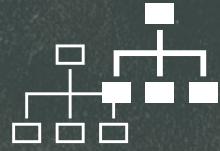


• Tabelle		Adresse			FK	
PK		Adress-ID	Straße	Hausnummer	Ort-ID	
	1	Coblitzallee	1-9		1	
	2	St. Blasius Weg	31a		2	
	3	Favoritenstraße	1		3	

• Tabelle		Ort			FK	
PK		Ort-ID	Postleitzahl	Ort	Land-ID	
	1	68163		Mannheim	1	
	2	56458		Mühlhausen	1	
	3	1100		Wien	2	



Normalisierung, 5. Normalform „Mehrwertigkeit“



■ Normalform 5

Die 4. Normalform muss erfüllt sein.

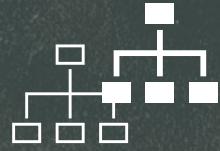
Alle mehrwertigen Attribute sind in eigenständige triviale Relationen zu überführen.

- Alle Attribute sind auf die grundlegendste Ebene zerlegt.
- Eine weitere Zerlegung in weitere Relationen ist nicht möglich.

• Tabelle		Ort				
		Ort-ID	Postleitzahl	Ort	Standteil	Land-ID
1	68163	Mannheim	Mannheim-Neuostheim	1		
2	68159	Mannheim	Mannheim-Mühlau	1		
3	68199	Mannheim	Almenhof, Neckarau	1		
4	56458	Mühlhausen	Mühlhausen	1		
5	1100	Wien	Favoriten	2		



Normalisierung, 5. Normalform „Mehrwertigkeit“

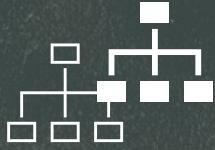


• Tabelle		Ort		FK
PK	Ort-ID	Ort		Land-ID
	1	Mannheim		1
	2	Mühlhausen		1
	3	Wien		2

• Tabelle		Stadtteil			FK
PK	ST-ID	Postleitzahl	Standort	Ort-ID	
	1	68163	Mannheim-Neustadt	1	
	2	68159	Mannheim-Mühlau	1	
	3	68199	Almenhof, Neckarau	1	
	4	56458	Mühlhausen	2	
	5	1100	Favoriten	3	



Normalisierung & Normalformen



- **NF1** Alle Attribute müssen **atomar** sowie durch einen **eindeutigen Schlüssel** bestimbar sein.
- **NF2** Aufteilung in mehrere Tabellen nach **Sinneinheiten**.
- **NF3** Aufteilung von **indirekten Abhängigkeiten** in eigenständige Tabellen.
- **BCN** Primärschlüssel müssen inhaltlich **unabhängig** sein.
- **NF4** Alle mehrwertigen Attribute sind in eigenständige **triviale Relationen** zu überführen.
- **NF5** Redundanzen von funktionalen Attributen sind **aufzulösen**.



In der Praxis meist genutzte Normalformen



Komplexitätserhörend & Performance relevant

- *Konzeption & Strukturierung von Datenbanken*
- *Redesign & Refactoring von bestehenden Datenbanken*

Die „Schlüssel“



Bedeutung von „Schlüsseln“



■ Hauptziele



- Schlüssel dienen zur eindeutigen Identifikation eines Datensatzes
- Abbildungsmerkmale von Relationen zwischen Entitäten
- Effizientes Auffinden von Informationen

!

Jede Tabelle ohne eine Primärschlüssel ist eine verlorene Tabelle.

Im Umkehrschluss muss **jede Tabelle für jeden Datensatz einen eindeutigen Schlüssel** besitzen,

Ansonsten wird grundsätzlich gegen die 1. Normalform verstößen.

Schlüssel, Abbildung von Relationen



■ Primärschlüssel, Primary Key (PK)

PK

- Jede Tabelle besitzt einen Primärschlüssel (NF1)
- Primärschlüssel müssen eindeutig sein
- Spalten der Primärschlüssel dürfen keine NULL-Werte beinhalten
- Der Primärschlüssel bestimmt die physische Speicherreihenfolge.

■ Fremdschlüssel, Foreign Key (FK)

FK

- Fremdschlüssel müssen immer auf einen existierenden Primärschlüssel der in Relation befindlichen Entität verweisen
- Fremdschlüssel können sich wieder holen
- NULL-Werte sind in der Fremdschlüsselspalte zulässig

Schlüssel, alternative Schlüssel



■ Alternativschlüssel, Alternative Key (AK)

AK

- Eindeutiges Attribut zur Identifikation eines Datensatzes
- Spalten der AK dürfen keine NULL-Werte beinhalten
- AK sind optionale natürliche Bestandteile einer Tabelle
- AK können als Ersatzwert zum PK verwendet werden

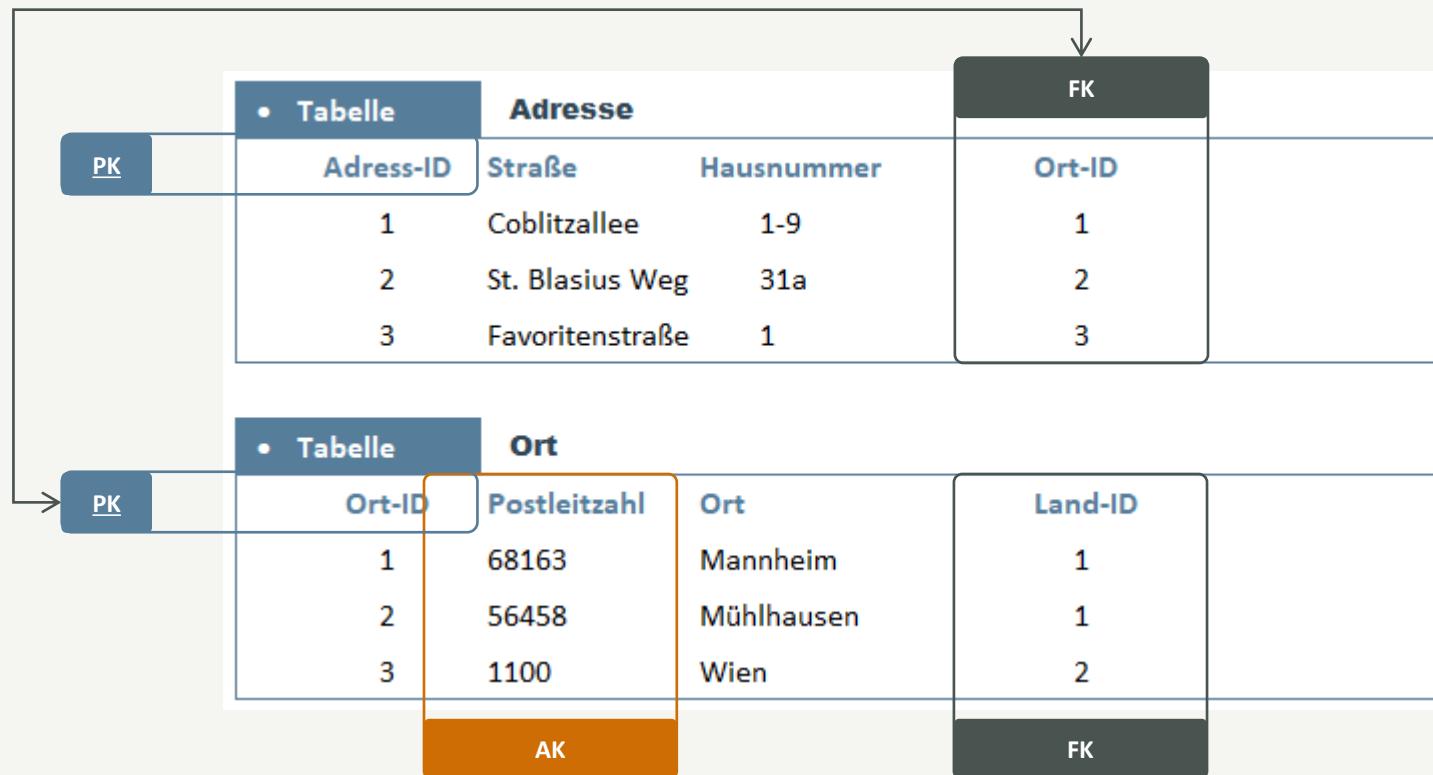
■ Sekundärschlüssel, Secondary Key (SK)

SK

FK

- Attribut das zur Erkennung eines Datensatzes unterstützend verwendet werden kann
- Keine Eindeutigkeit erforderlich
- Attribut ist optional und kann NULL-Werte enthalten
- Oftmals durch stützenden Index optimiert

Schlüssel, aus der „4. Normalform“



Schlüssel, Einzelschlüssel



■ Einzelschlüssel

- Ein Schlüssel definiert durch ein Attribut (eine Tabellenspalte)
- Kann aus bestehenden Attributen oder ebenso durch generische Attribute gebildet werden.

• Natürlicher Schlüssel, Natural Key

• Sprechender Schlüssel, Business Key

- Können sich aus vorhandenen Attributen der Entitäten ergeben
- Können durch eigene Syntax-Strukturen abgebildet werden
- Inhalte der Schlüssel selbst sind inhaltlich lesbar
- Inhalte der Schlüssel stellen fachlichen Kontext dar

• Surrogatschlüssel, Surrogate Key

• Stellvertreter-, Ersatzschlüssel

- Generisch erstellte Schlüssel
- Dienen ausschließlich der eindeutigen Identifikation von Datensätzen
- Beinhalten keine Fachinformationen
- Standardverfahren zur Erstellung sind:
 - numerische Auto-Inkrementoren
 - Generierung von Universal Unique Identifier (UUID)

„Der scheinbar beste Freund“

„Der Neutralisator & Problemvermeider“

Schlüssel, Einzelschlüssel - „Tabelle Ort“



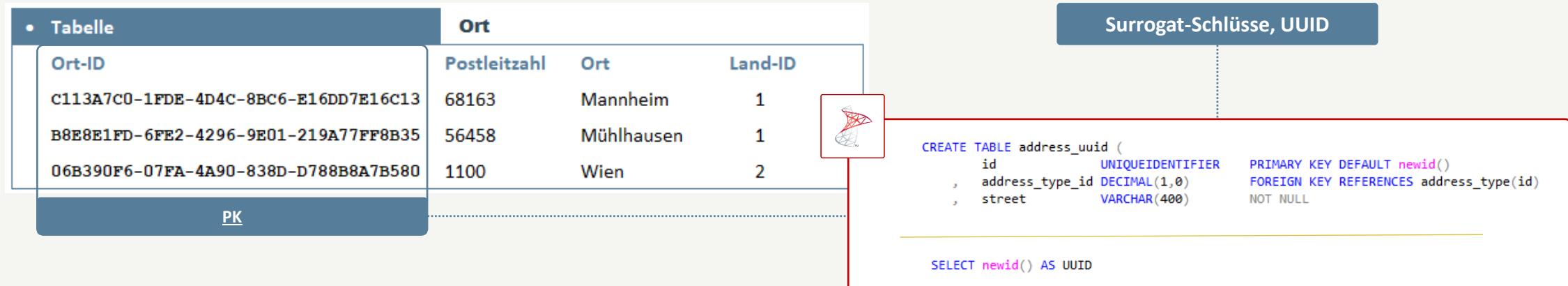
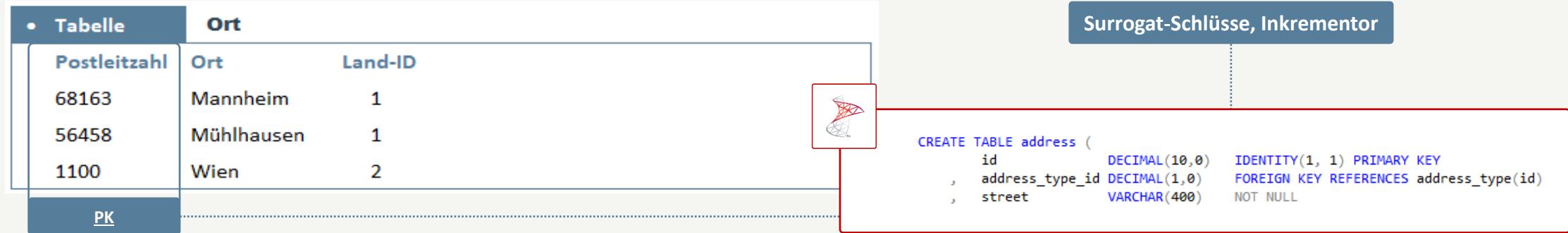
• Tabelle		Ort	
	Postleitzahl	Ort	Land-ID
	68163	Mannheim	1
	56458	Mühlhausen	1
	1100	Wien	2

Natürlicher Schlüsse

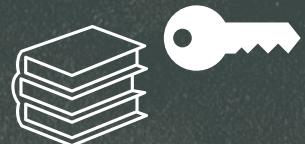
• Tabelle		Ort		
	Ort-ID	Postleitzahl	Ort	Land-ID
	1	68163	Mannheim	1
	2	56458	Mühlhausen	1
	3	1100	Wien	2

Surrogat-Schlüsse

Schlüssel, Einzelschlüssel - „Tabelle Ort“



Natürliche Schlüssel, Beispiele



Leseprobe

Gunter Saake, Kai-Uwe Sattler, Andreas Heuer
Datenbanken - Konzepte und Sprachen

Gebundenes Buch

Jetzt bewerten

Auf die Merkliste Bewerten Teilen Produkterinnerung

Weitere 2 Ausgaben: eBook, ePUB eBook, PDF

Detaillierte Einführung in die Datenbanktechnologie in der 6. aktualisierten Neuauflage Konzepte relationaler Datenbanken: Architekturen, Modelle, Entwurfsmethoden, Relationenalgebra Ausführliche Behandlung von SQL sowie Einblick in weitere Datenbanksprachen Datenbanken sind eines der wichtigsten Teilgebiete der Informatik und dank Big Data, Industrie 4.0, Internet of Things & Co. müssen immer mehr Daten effizient verwaltet, performant analysiert und sinnvoll ausgegeben werden. Das Buch behandelt systematisch und fundiert die wichtigsten Konzepte und Sprachen, die für den Einsatz von [„mehr“](#)

Produktdetails

mitp Professional
Verlag: MITP / MITP-Verlag
Artikelnr. des Verlages: 95845776
6., überarb. Aufl.
Erscheinungstermin: 4. Mai 2018
Deutsch
Abmessung: 246mm x 177mm x 50mm
Gewicht: 1469g
ISBN-13: 9783958457768
ISBN-10: 3958457762
Artikelnr.: 50633823

ISBN

Internationale Standardbuchnummer
International Standard Book Number

Natürlicher Schlüsse

Artikelnummer **95845776**

ISBN-10 **3-958457762**

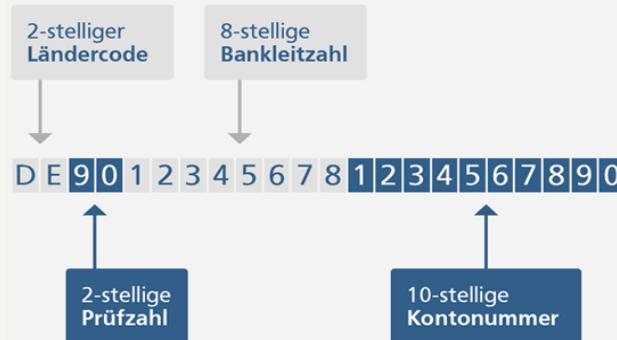
ISBN-13 **978-3-958457762**

- Mit dem Verwendungsverlauf reduziert verfügbarer Nummernkreis verfügbar
- Nummernkreis wurde von 10- auf 13-stellig umgestellt
- Alte ISBN-Nummern bekamen Standardnummer 978 vorangestellt, um Umstellung zu vermeiden

Natürliche Schlüssel, Beispiele II



IBAN, International Bank Account Number



Gesellschaftsnummern

Gesellschaft	buecher.de GmbH & Co. KG	1&1 Mail & Media GmbH
Handelsregisternummer	HRA 13309	HRB 7666
Umsatzsteuer-ID	DE 204210010	DE 243413002

SV-Nr., Sozialversicherungsnummer

SV Beispiel 01 33 160894 W 098

ISBN-13 65 170806 J 008

- 1 - 2 Bereichsnummer des Rentenversicherungsträgers
- 3 - 8 Dein Geburtsdatum
- 9 Anfangsbuchstabe deines Geburtsnamens
- 10 - 11 Seriennummer (00-49 = männlich, 50-99 = weiblich)
- 12 Prüfziffer

Schlüssel, Einzelschlüssel



■ Zusammengesetzter Schlüssel, Composite Key

- Ein Schlüssel bestehend durch eine Gruppe von Attributen (mehrere Tabellenspalten)

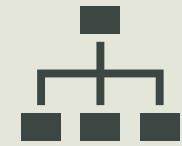


- Zusammengesetzter Schlüssel
- Composite Key, Combined Key

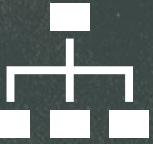
- Bestehen aus existierenden Attributen einer Entität
- Inhalte der Schlüssel selbst sind inhaltlich lesbar
- Inhalte der Schlüssel stellen fachlichen Kontext dar
- Komplexität erhöht bei Verwendung als FK

„Der einfach(e) trügerische Weg“

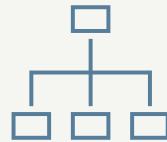
Datenmodellierung



Datenmodellierung



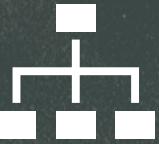
■ Prinzip



- Abbildung von Informationen eines konkreten Sachverhaltes
- Ermittlung der Kernentitäten eines Sachverhaltes
- Darstellung der Beziehungen zwischen den Entitäten
- Ausprägung der charakterisierenden Attribute einer Entität
- Definition der Schlüsselattribute

!

Die Datenmodellierung ist ein Verfahren zur formalen Abbildung von Objekten, deren Attribute sowie der Beziehungen zwischen den Objekten in einem spezifischen Kontext mit dem Hauptziel einer Definition für die datentechnische Verwaltung.



Rückblick

1976, DAS JAHR DER DATENMODELLIERUNG



APPLE I

Markteintritt von Apple in den PC-Markt mit seinem PC-Bausatz



APPLE II

Mit dem Apple II begann 1977 die Massenproduktion der Mikrocomputer



APPLE MACINTOSH

Führt Icons und Fenster ein sowie das Zeigergerät „Maus“



2007

1975

1976

1977

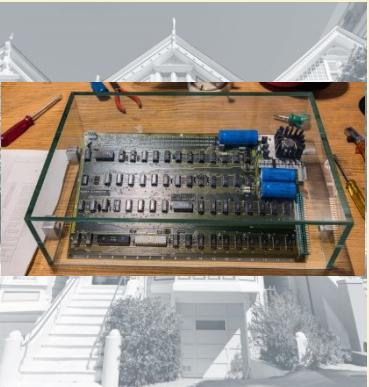
1977

1983

1984

MITS ALTAIR 8800

Erster „Personal Computer“, Selbstbausatz



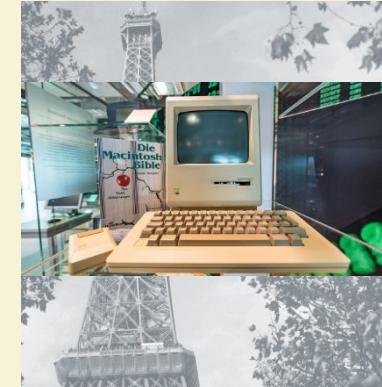
COMMODORE PET 2001

Commodore bringt seinen ersten PC mit Bildschirm auf den Markt



OSBORN 1

1. Laptop



APPLE IPHONE

1. Smartphone mit Touchscreen & App-Anwendungen



Datenmodellierung, Peter Pin-Shan Chen



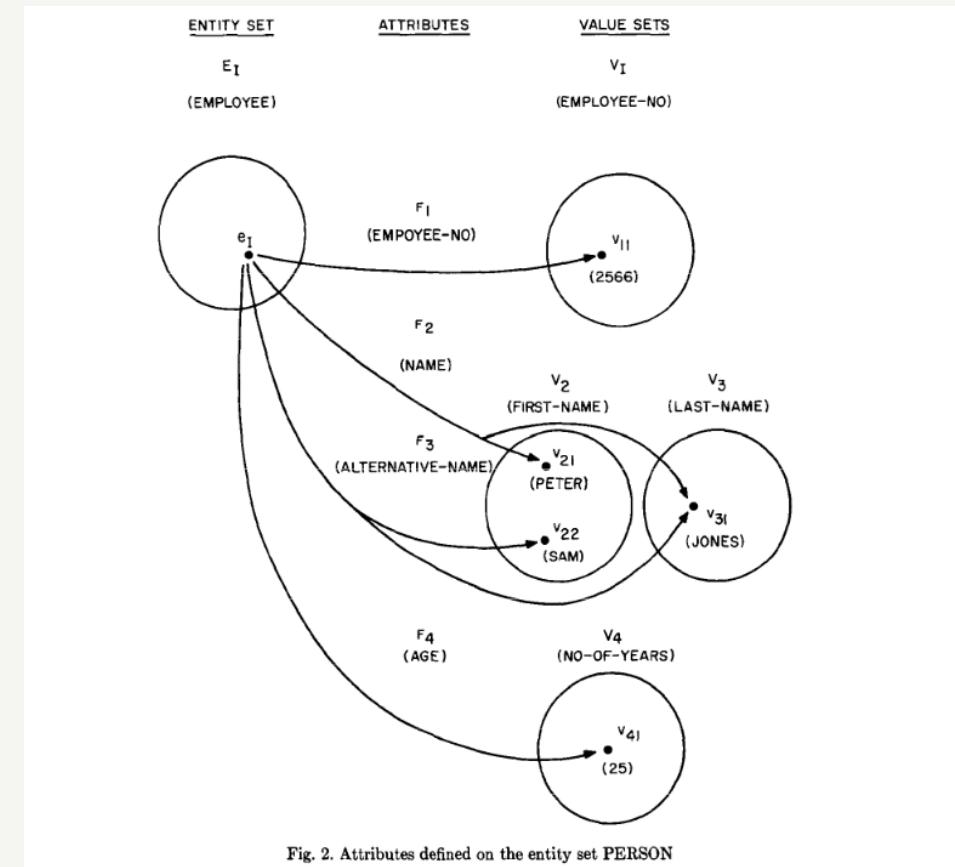
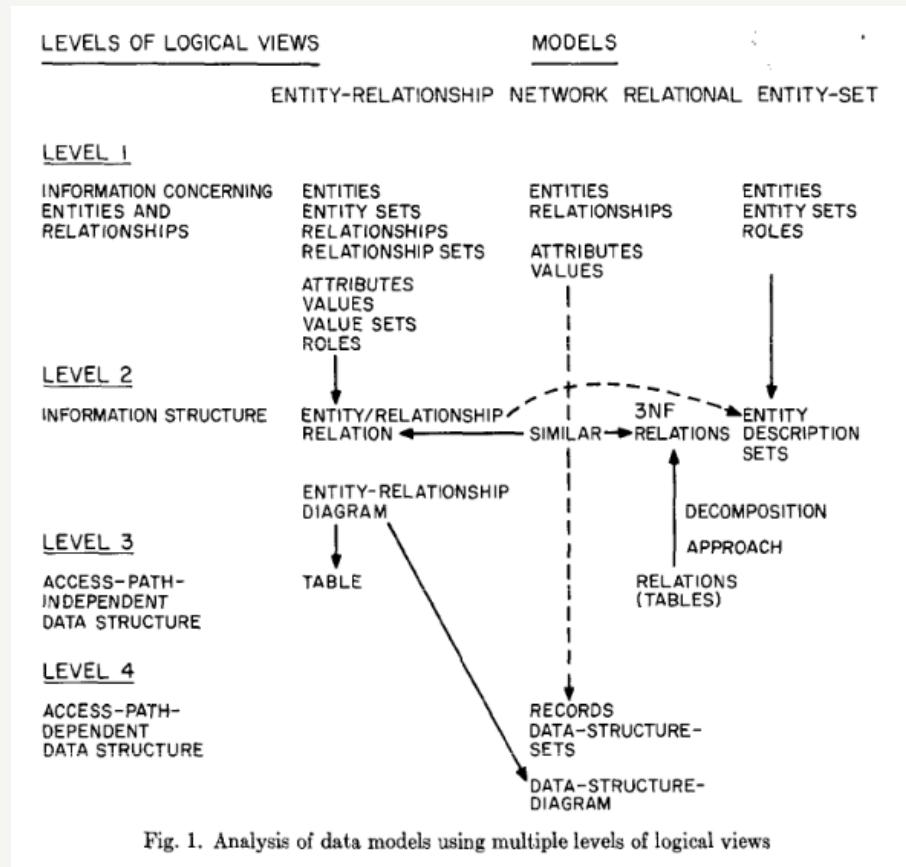
- **Peter Pin-Shan Chen**, chinesisch: Chen Pin-Shan
- **Geboren** 3. Januar 1947 in Taichung, Taiwan
- **Beruf** Informatiker
- 1968 - 70 Studium des Elektroingenieurwesens an der National Taiwan University
- 1970 - 73 Master, Doktor in Computer Science und angewandter Mathematik an der Harvard University
- 1974 - 78 Assistenzprofessor an der MIT Sloan School of Management
- 1978 - 84 Professor an der University of California, Los Angeles
- 1983 Professor der Informatik an der Louisiana State University

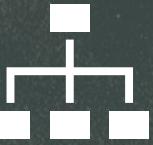
!

- **1976**, Veröffentlichung des Artikels: *The Entity Relationship Model – Toward A Unified View of Data*
- Entwicklung der Entity-Relationship-Modellierung für die Datenmodellierung
- Revolutionäre Ansatz in der **grafischen Darstellung der Datenelemente** sowie deren **Beziehung** zueinander



Datenmodellierung, ER-Modellierung (Ursprung I)





Datenmodellierung, ER-Modellierung (Ursprung II)

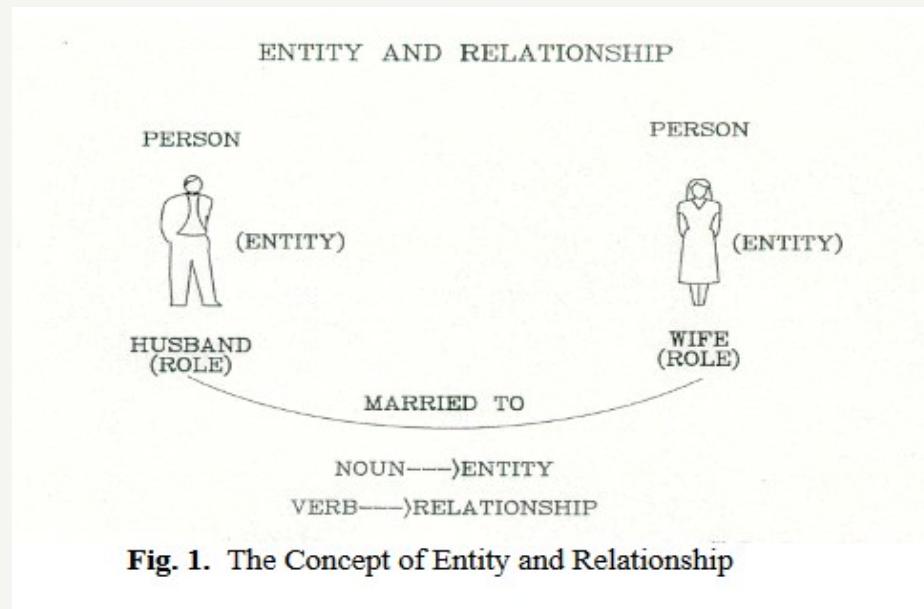


Fig. 1. The Concept of Entity and Relationship

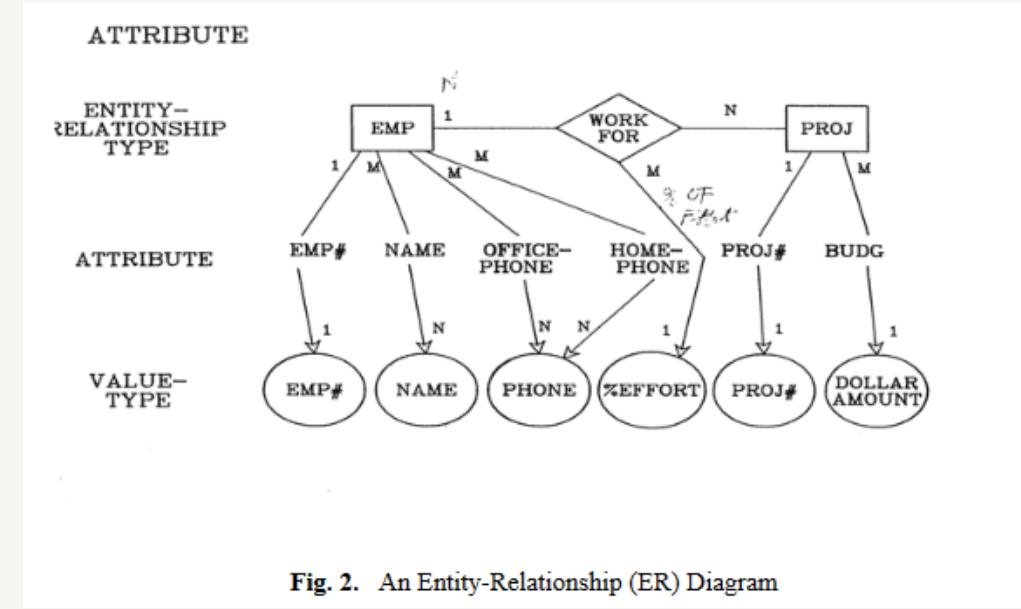


Fig. 2. An Entity-Relationship (ER) Diagram

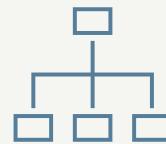
SET THEORY (DEFINITIONS)		A RELATIONSHIP SET IS DEFINED AS A "MATHEMATICAL RELATION" ON ENTITY SETS
ENTITY	e	
ENTITY SET	$E; e \in E$	
VALUE	v	
VALUE SET	$V; v \in V$	
RELATIONSHIP	r	$R = \{r_1, r_2, \dots, r_n\}$
RELATIONSHIP SET	$R; r \in R$	$r_i = [e_{i1}, e_{i2}, \dots, e_{in}] e_{in} \in E_1, \dots, e_{in} \in E_n$

Fig. 3. Formal Definitions of Entity and Relationship Concepts

Datenmodellierung, Entity-Relationship-Model



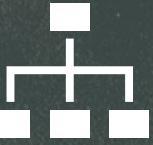
■ Modellierung



- Kerninhalte werden als Entität abgebildet
- Beziehungen als Relation dargestellt
- Relationen erhalten eine Beziehungsausprägung
- Entitätsmerkmale werden als Attribute abgebildet
- Primär- & Fremd-Schlüssel werden gesondert hervorgehoben

!

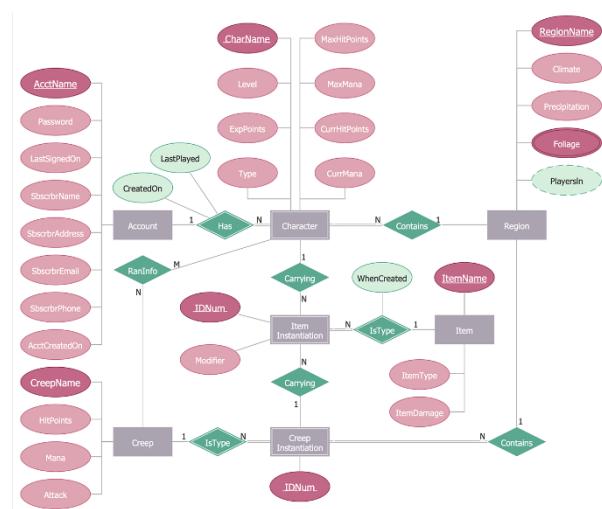
Für die semantische Datenmodellierung findet das Entity-Relationship-Modell (ERM) Anwendung. Hierfür erfolgt die modellhafte Abbildung eines relevanten Ausschnittes einer fachlich konkreten Situation. Das ER-Modellierung erfolgt im Schwerpunkt durch eine Grafikmodellierung der Kontext-Entitäten und deren Attribute.



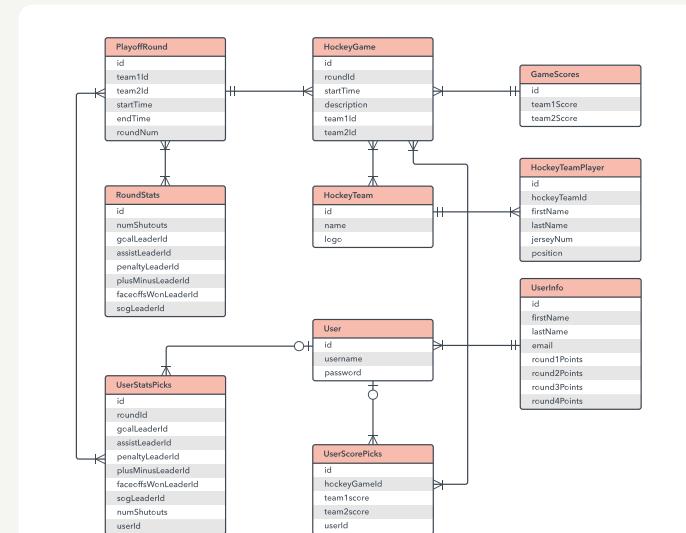
Datenmodellierung, Entity-Relationship-Model

Anwendung

- Verständigung zwischen Anforderungssteller & Entwicklern
- Unterstützung der Konzeptionsphase
- Grundlage in der Implementierungsphase für das Datenbankdesign



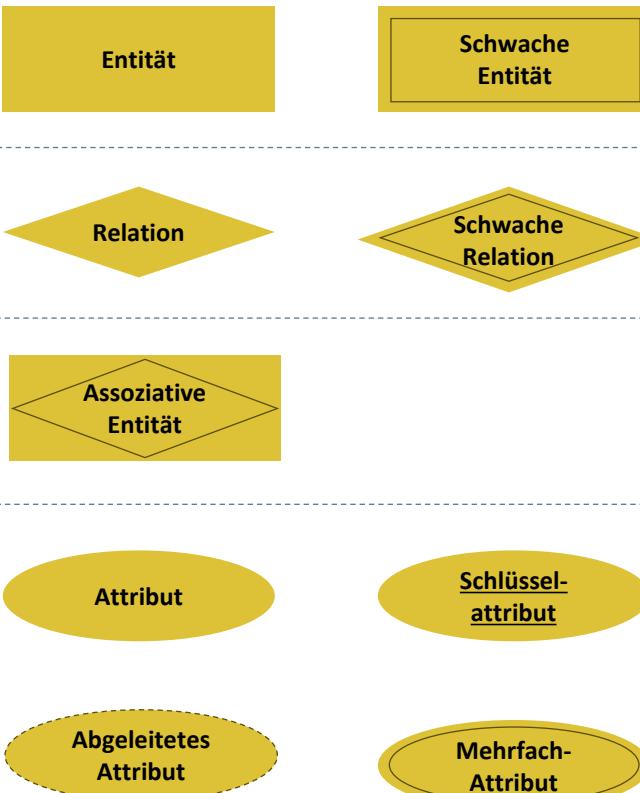
Chen-Notation (Peter Pin-Shan Chen)



Martin-Notation / Krähenfußnotation



ERM, Formen

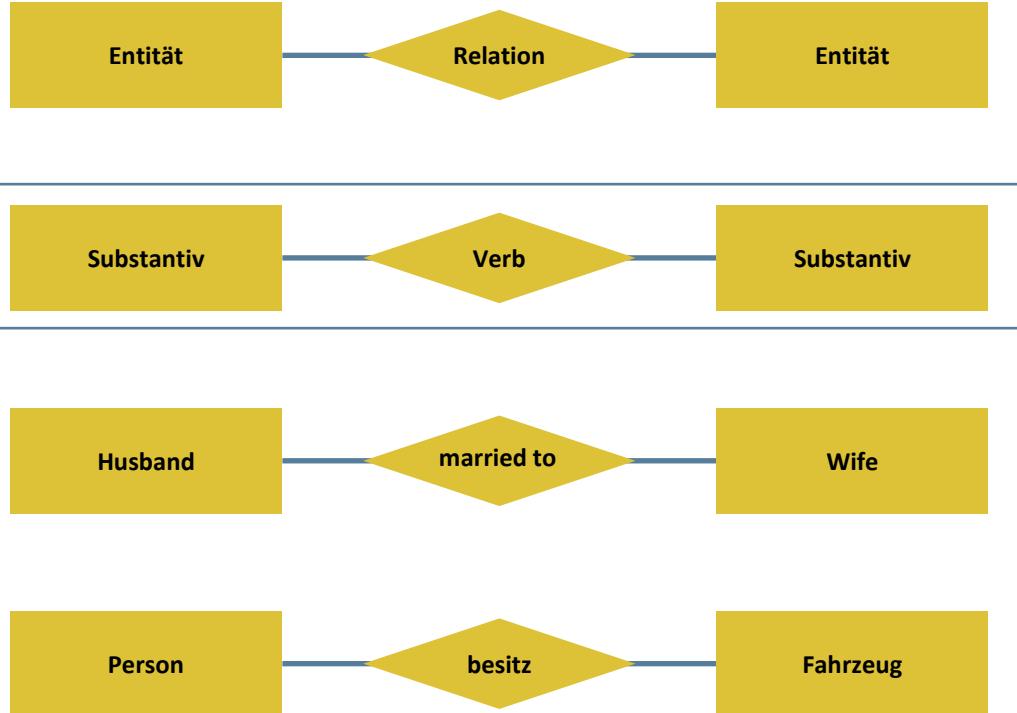


- **Entität**, Abbildung eines Objektes, Gegenstandes oder Person/Gruppe, geprägt durch ein „Substantiv“
 - **Schwache Entität**, eine Entität die von einer anderen (Haupt-)Entität abhängig ist
- **Relation**, Beziehung zwischen den Entitäten, geprägt durch ein „Verb“
 - **Schwache Relation**, eine Relation zwischen einer schwachen Entität & Haupt-Entität
- **Assoziative Entität**, aus einer Relation hervorgegangener Entität
- **Attribut** Charakterisierendes Merkmal einer Entität
 - **Schlüsselattribut**, Schlüssel einer Entität
 - **Abgeleitetes Attribut**, Attribut, welches aus einem Bestandsattribut abgeleitet ist
 - **Mehrfachattribut**, bestehendes Attribut, mit Mehrfachwerten (z.B. Telefonnummern)



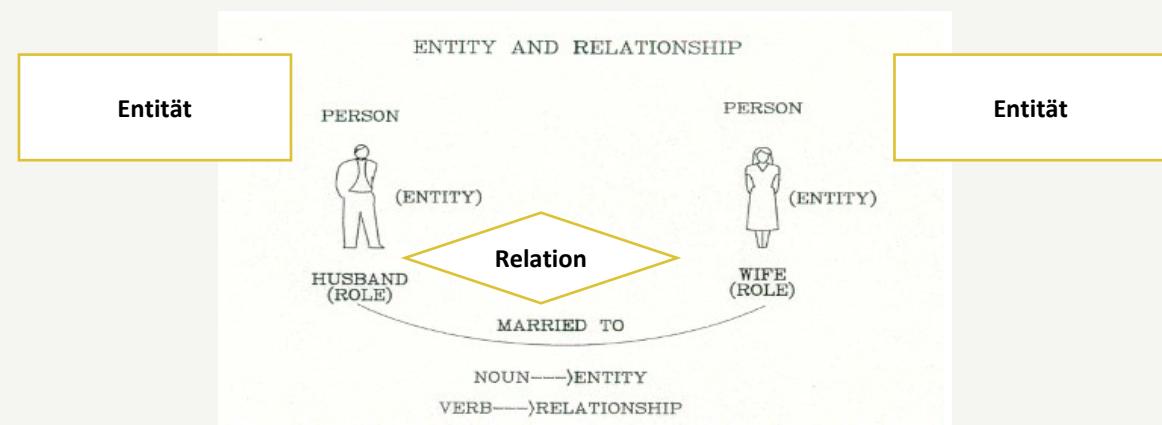
ERM, Modellierung, Sprachbildung

Diagrammmodellierung



Diagrammmodellierung

- Eine **Entität** wird durch ein **Substantiv** ausgeprägt
- Eine **Relation** wird durch ein **Verb** abgebildet
- Das Verb beschreibt verstärkend die Beziehungen zwischen den Entitäten und verstärkt den fachlichen Zusammenhang dieser.





ERM, Kardinalitäten (Max-Notation)

Kardinalitäten

- Max-Notation

1 : 1



1 : N



N : M



- **Kardinalität, beschreibt das Beziehungsmehrheitsverhältnis zwischen 2 Entitäten.**

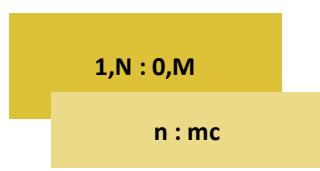
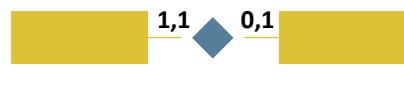
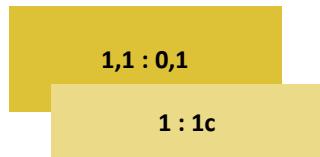
- **1 : 1** Direkte Beziehung zwischen 2 Entitäten, jede Entität ist eindeutig zueinander zugeordnet.
- **1 : N** Die Hauptentität wird von der Folgeentität mehrfach referenziert/verwendet.
- **N : M** Die Entitäten stehen zueinander in einer Mehrfachnennungsbeziehung.
- Die jeweiligen Entitätsmengen „1“ und „N“ beziehen auch die Möglichkeiten der Nicht-Existenz ein.
- Die Darstellung erfolgt in der [Max-Notation](#).



ERM, Kardinalitäten (Min-Max-Notation)

Kardinalitäten

- Min-Max-Notation



- Max-Notation

- Grundlegende für Modellierung der relevanten Relationen sowie deren Basis-Kardinalität
- Nachteilig für das Datenbankdesign, dass die Minimum-Kardinalität nicht explizit verfügbar
- Minimum- und Maximum-Kardinalität relevant für Bewertung des Datenbankdesigns

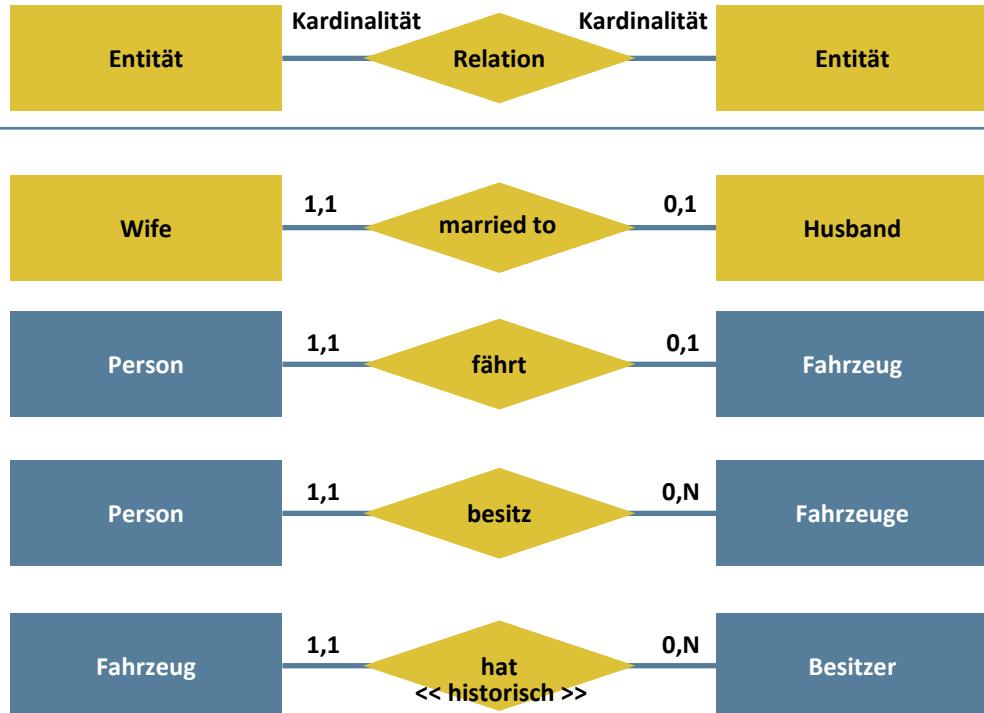
- Lösung

- Einführung der Min-Max-Notation
- „Minimum-Kardinalität“ erfolgt vorangestellt der Maximum-Kardinalität Komma-getrennt
- Alternativnotationsform „Conditioned“
- 0-Kardinalität dargestellt durch nachgeführten Buchstaben „c“ (=„conditioned“)



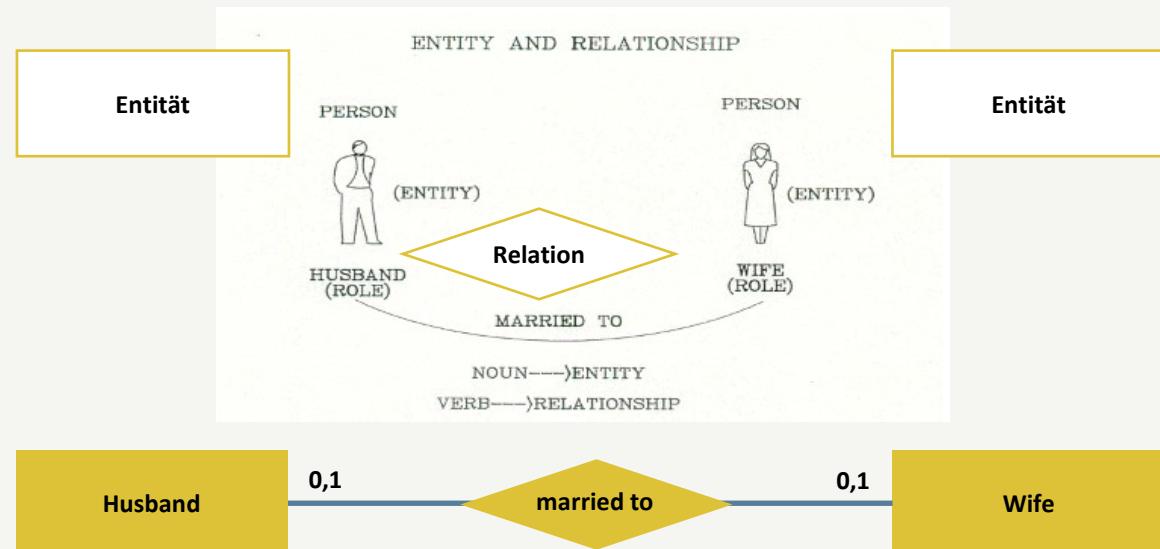
ERM, Modellierung, Kardinalität

Diagrammmodellierung



Diagrammmodellierung

- Die Kardinalität wird direkt an der Entität selbst angebracht
- Die Leserichtung ist (in der Regel) von links nach rechts und oben nach unten
- Zur Inversprüfung der Kardinalität in umgekehrter Leserichtung

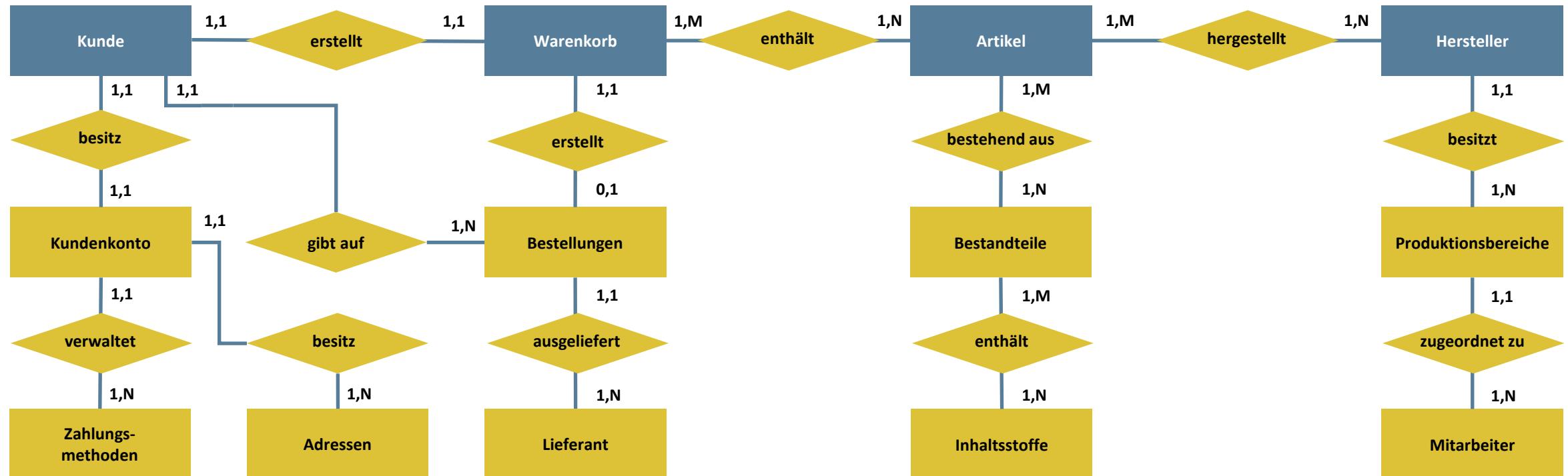




ERM, Modellierung, umfangreiches Beispiel I

Diagrammmodellierung

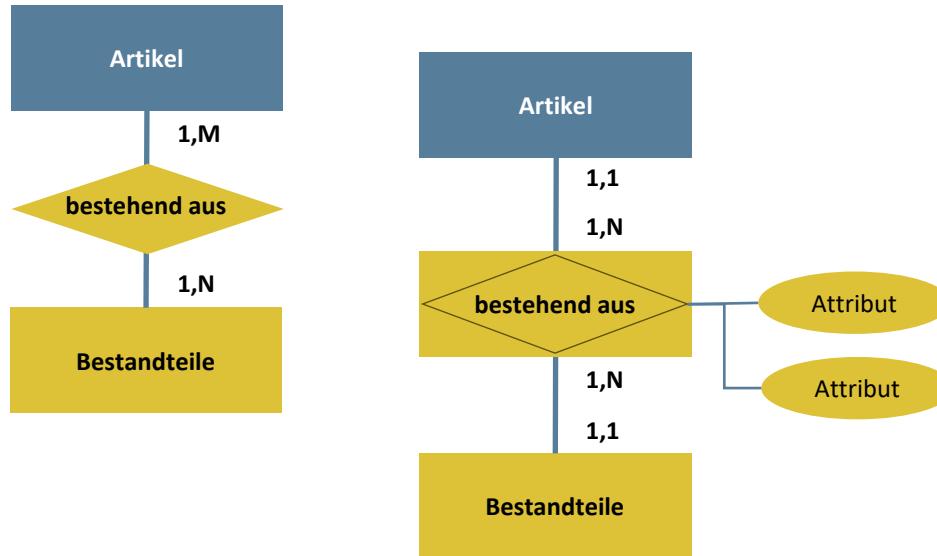
- Beispiel, Onlineshop – Kundenbestellung (inkl. Artikel- & Hersteller-Details)





ERM, Modellierung, spezielle Entitäten

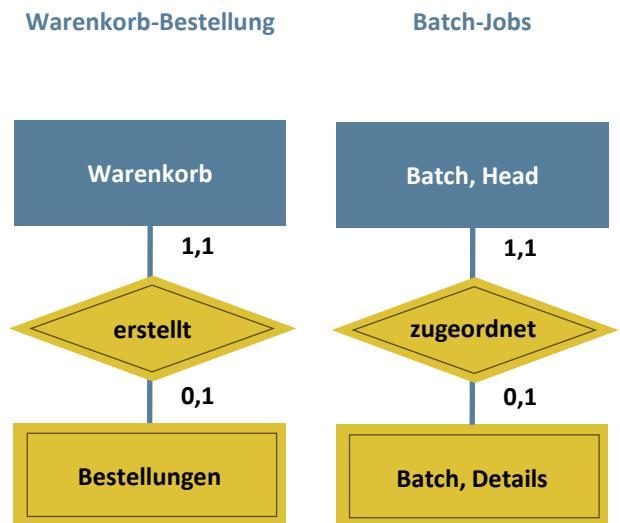
Assoziative Entität



- **Assoziative Entität**

- Wandlung der N:M-Beziehung in **1:N-Beziehungen**
- Möglichkeit der Zusatz-Attribute an der Mapping-Entität

Schwache Entität



- **Schwache Entität & Relation**

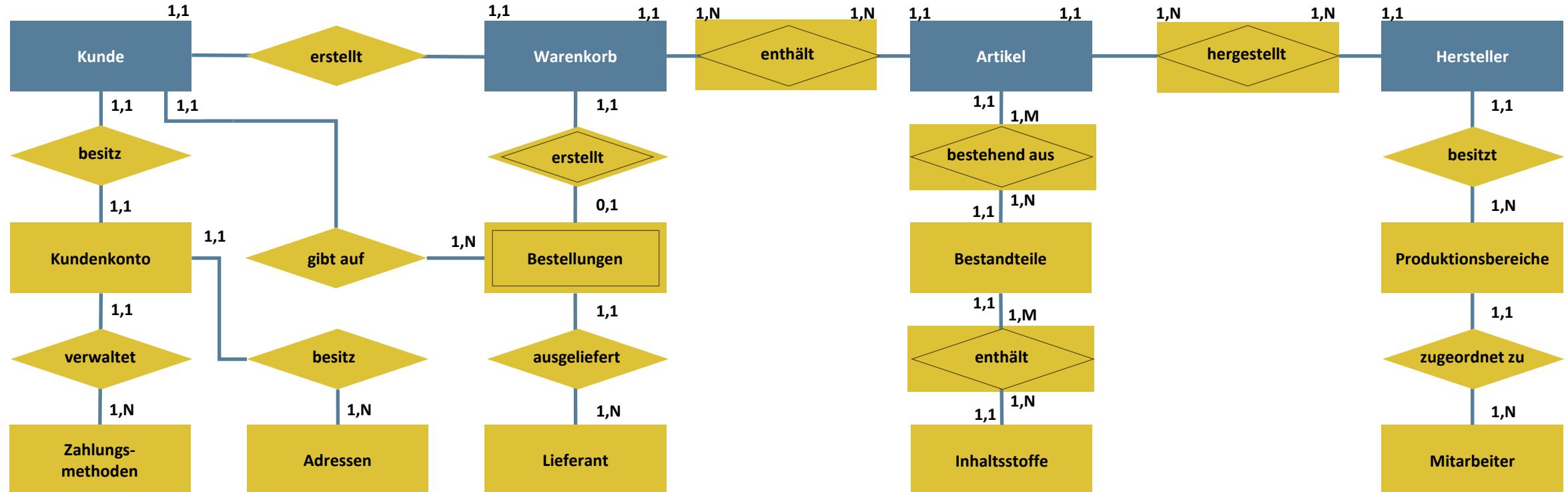
- Eine Schwache Entität kann ohne die Hauptentität nicht existieren
- Die Gewichtung ist einer normalen Entität gleichgestellt



ERM, Modellierung, umfangreiches Beispiel II

Diagrammmodellierung

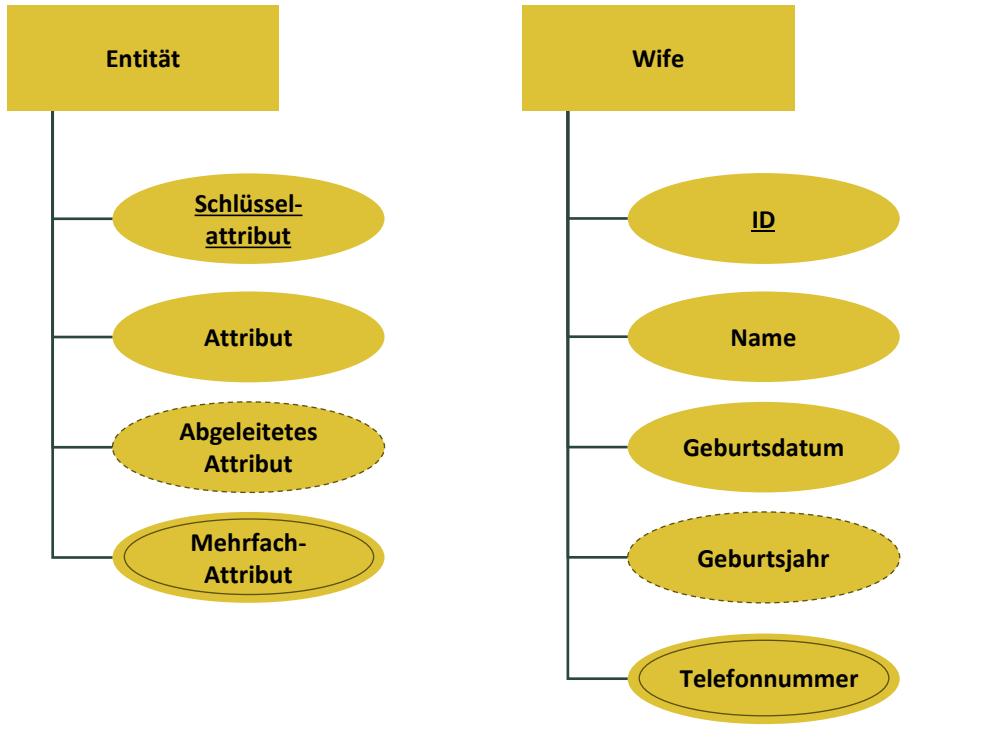
- Beispiel, Onlineshop - Kundenbestellung





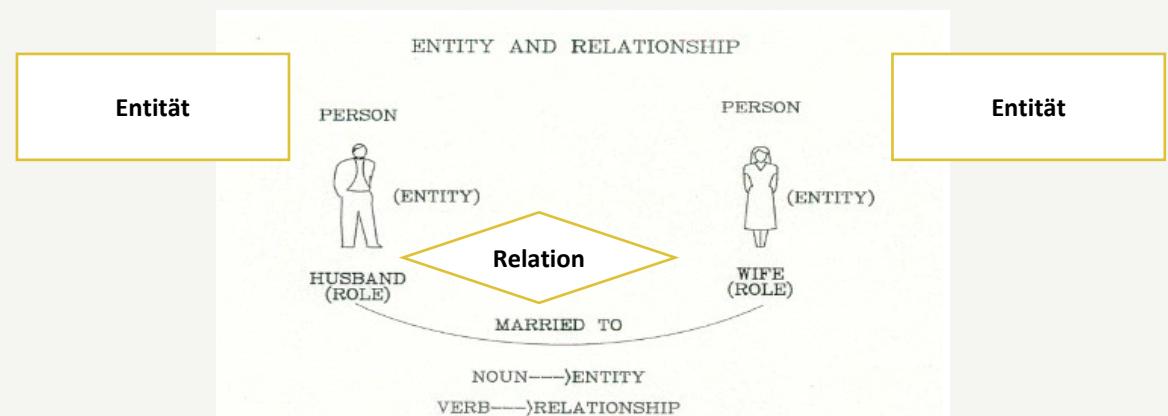
ERM, Modellierung, Attribute

Diagrammmodellierung



Diagrammmodellierung

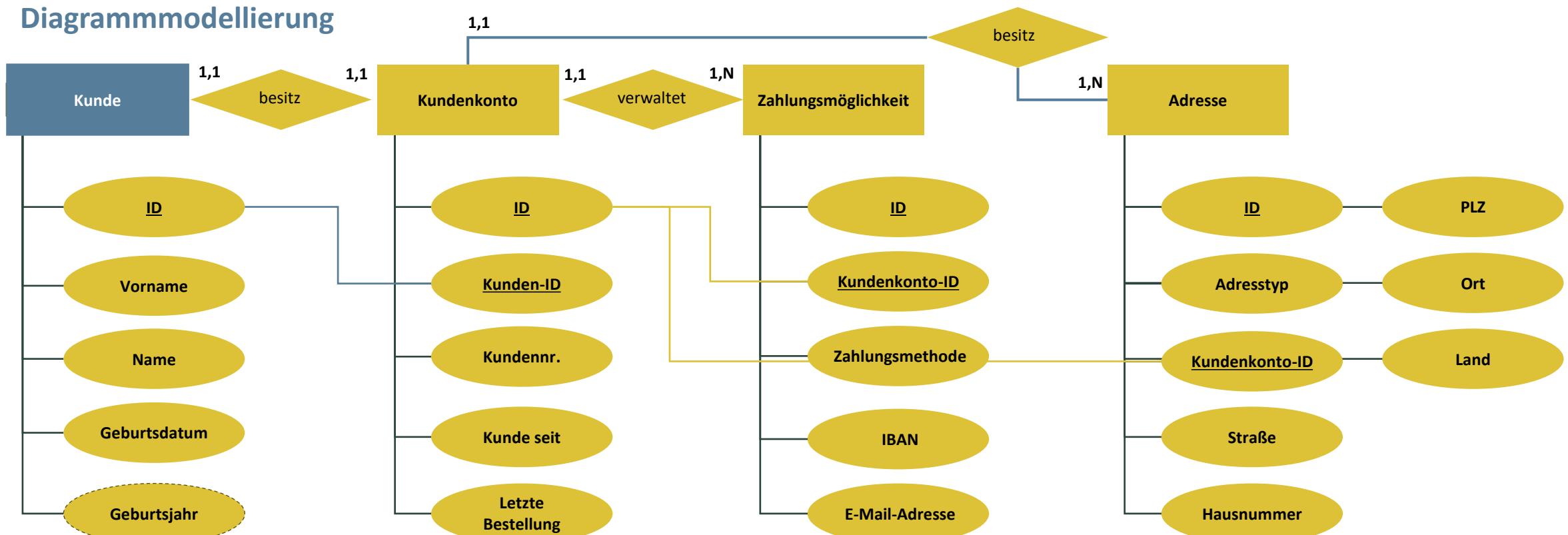
- Die Attribute werden als Substantive dargestellt.
- Jedes Attribut stellt eine wesentliche Charakterisierung der Entität dar.
- Attribute und Datentypen
 - Chen-Notation definiert keine Datentypen an den Attributen
 - Dient selbst als funktionale Erweiterung für den Designprozess im Schwerpunkt für das Datenbankdesign



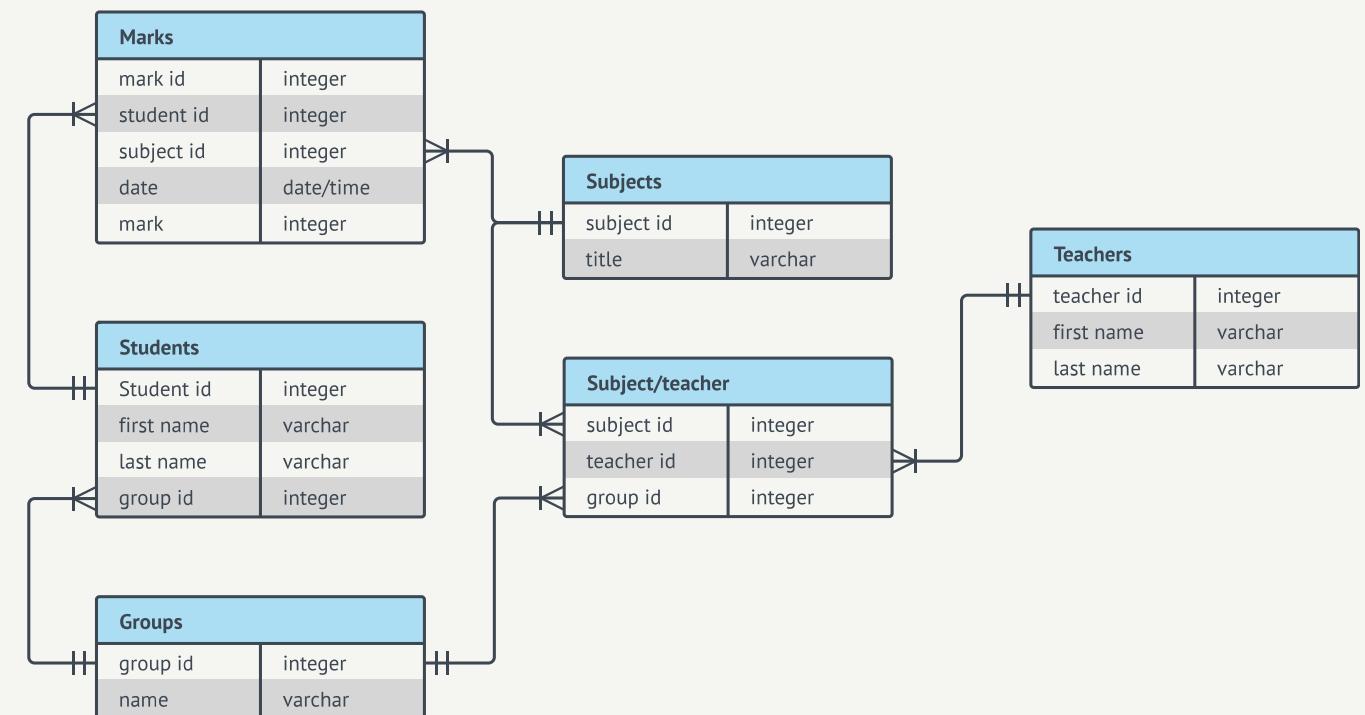
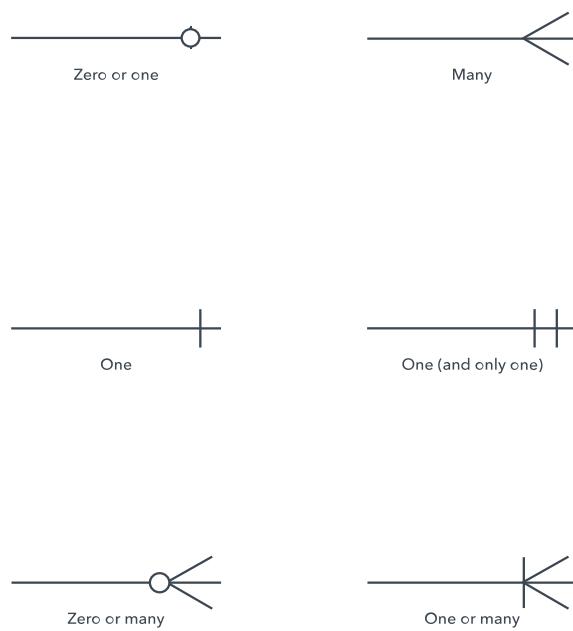


ERM, Modellierung, Attribute

Diagrammmodellierung



ERM, Variante: Martins- / Krähenfußnotation





Das „ideale“ Datenmodell

- Das „ideale“ Datenmodell ist ein Mythos bis man es erreicht.
- Ob ein Datenmodell „ideal“ ist, liegt am Zeitpunkt der Betrachtung sowie im Auge des Betrachters.

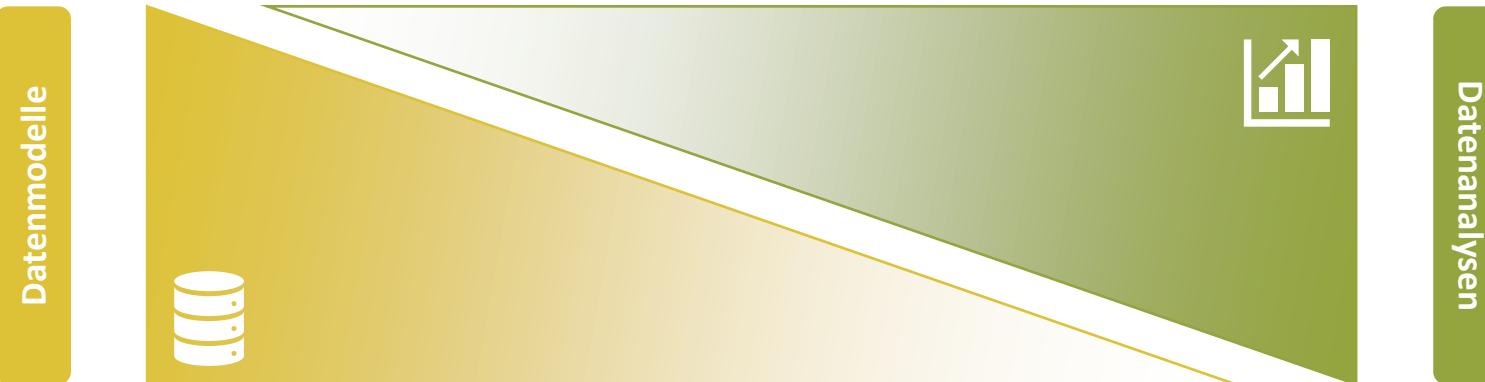
Datenmodelle, Knowledge-Skilling



Data Scientist



Data Analyst



- Auseinandersetzung mit den Datenmodellen & Datenstrukturen
- Überführung von relevanten Daten in das DataWarehouse (ETL-Prozesse)
- Erstellung von Datenmodellen für Analysen der Fachprozessanalytiker
- Durchführung von Vorhersageanalysen (Predictive Analytics).
- Nutzung von weiteren Advanced Analytics wie „Machine Learning“ & „Artificial Intelligence“
- Ableitung der Requirements aus dem Fachbedarf-Requests
- Kommunikation zwischen Fachanforderer & Data-Scientists
- Aufbereitung des Analysebedarfs für Datenresearch
- Analyse der bereitgestellten Bedarfsinformationen
- Aufbereitung der Informationen in grafischer Form sowie in KPIs

Datenmodelle



Reporting, ist die Umwandlung von Geschäftsdaten in Informationen



BI - Business Intelligence

- Operatives Reporting
- Statistiken
- Reports, Bericht
- Chart, Diagramme



Advanced Analytics

- Process Mining
- Predictive Analytics (ForeCast)
- AI - Artificial Intelligence
- Machine Learning



Heuristiken



DataWarehouse, DataLake,
SmartData, BigData, DataBase



Präsentation, Ausgabe & Darstellung