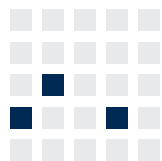




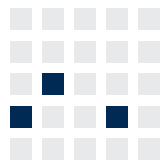
Einführung in die Wirtschaftsinformatik

Teil 3 - Von der Realwelt zum Datenmodell

Wintersemester 2025/2026



Lehrstuhl für Wirtschaftsinformatik
Prozesse und Systeme
Universität Potsdam



Chair of Business Informatics
Processes and Systems
University of Potsdam

Univ.-Prof. Dr.–Ing. habil. Norbert Gronau
Lehrstuhlinhaber | Chairholder

Mail August-Bebel-Str. 89 | 14482 Potsdam | Germany
Visitors Digitalvilla am Hedy-Lamarr-Platz, 14482 Potsdam
Tel +49 331 977 3322

E-Mail ngronau@lswi.de
Web lswi.de

Lernziele

- Wie wird die **reale Welt in ein Datenmodell** überführt?
- Welche **Abbildungsschritte** führen vom Original bis zur **physischen Datenbank**?
- Was sind **Datenmodelle und Datenstrukturen** und welche **Typen** werden unterschieden?
- Welche **Aufgaben und Bestandteile** besitzen **Datenbanken und Datenbankmanagementsysteme (DBMS)**?
- Wie funktioniert das **Entity-Relationship-Modell** und wie werden **Entities, Attribute und Beziehungen** dargestellt?



Modellierung - Abbildung des betrachteten Originals

Vom Original zum Modell - Abbildungsschritte

Datenmodelle und -strukturen

Datenbanken und Datenbankmanagementsysteme

Das konzeptionelle Schema: Prinzip des Entity Relationship Modells (ERM)

Hörsaal-Quiz - Einleitungsfragen

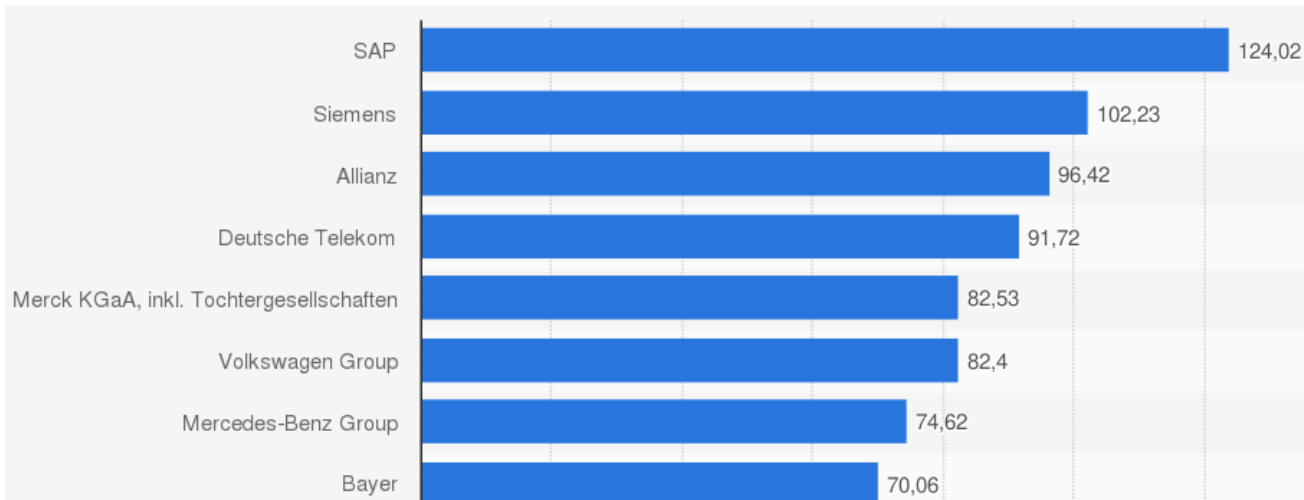
Öffnet die App über den QR-Code oder den Link:



<https://quiz.lswi.de/>

pwd: ewinf

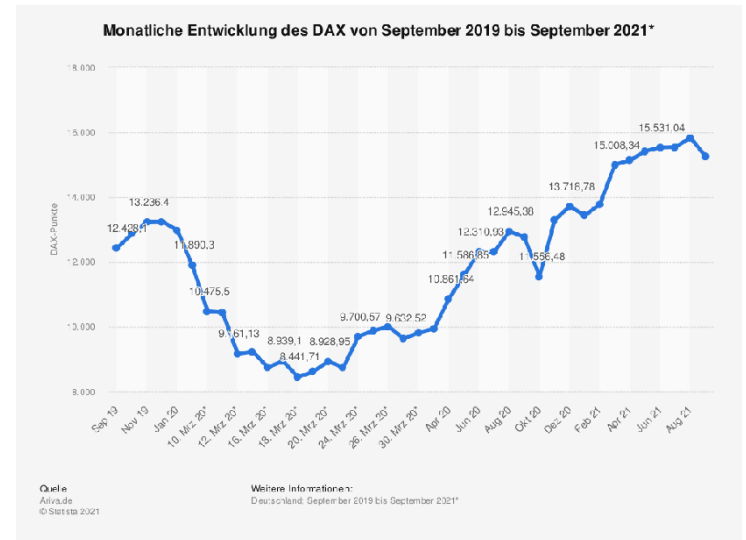
Daten



Größte dt. Unternehmen Stand 22.April 2022

Statisches Abbild der Anwendungswelt

- Personen - Fußballspieler, Politiker, Studierende
- Gegenstände - Bücher, Werkzeuge
- Künstliche Objekte - Bank- oder Versicherungskonten



Indirekter Ausdruck der Dynamik

- Kontostand vor und nach einer Transaktion
- Umsatzvolumen vor und nach einer Rechnungsstellung
- Spieltabelle (Turniertabelle) vor und nach einem Spieltag

Daten bilden die Basis für die Verknüpfung von Informationen. Werden Informationen von Personen interpretiert, entsteht Wissen.

Prinzipien der Kategorisierung von Daten

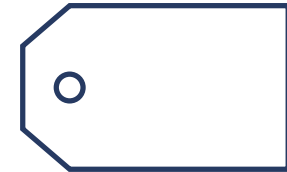
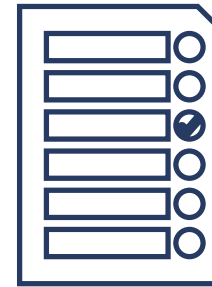


Kundendaten

Lagerbestände,
Kontostände, Aufträge



Lagerbestände,
Kontostände, Aufträge



Bruttopreis von
Produkten über
Mehrwertsteuer

... nach Inhalt

- **Stammdaten** und deren Änderungen --> Seltene Veränderungen
- **Bestandsdaten** und deren Änderung durch Bewegungsdaten --> Häufige Veränderungen

... nach Aufgabe

- **Ordnungsdaten** --> Identifikation (Personen, Dinge) oder Vergleiche (Preise, Gehälter)
- **Rechendaten** --> Berechnung oder Umrechnung von Werten

Aufgaben in der Datenhaltung

Aufgaben in betriebswirtschaftlichen Informationssystemen

- **Erfassung** von Geschäftsvorfällen (Bewegungsdaten)
- **Verarbeiten** der Geschäftsvorfälle (Verarbeiten mit Stamm- oder Bestandsdaten)
- **Pflege** der Stammdaten (Aktualisierung der Stammdaten - Änderungsdienst)
- Informations**abfragen** (Standard-/Ad-hoc-Abfragen)

Auswählbare Grundoperationen für Datenbearbeitung

- Auffinden - wird mehrfach benötigt
- Einfügen
- Ändern
- Entfernen

Die Operation Auffinden bestimmt häufig über das Antwortzeitverhalten des gesamten Informationssystems.

Die Logische Sicht auf Daten

Logische Sicht

- Hierarchische Dateiverwaltung -> Verzeichnisse (directories)
- Datei -> auf Datenspeicher abgelegte Datenmenge mit Zugriff über Dateiname

Dateiinhalt

- Datenobjekte
- Beschreibung durch ihre Eigenschaften (Attribute)

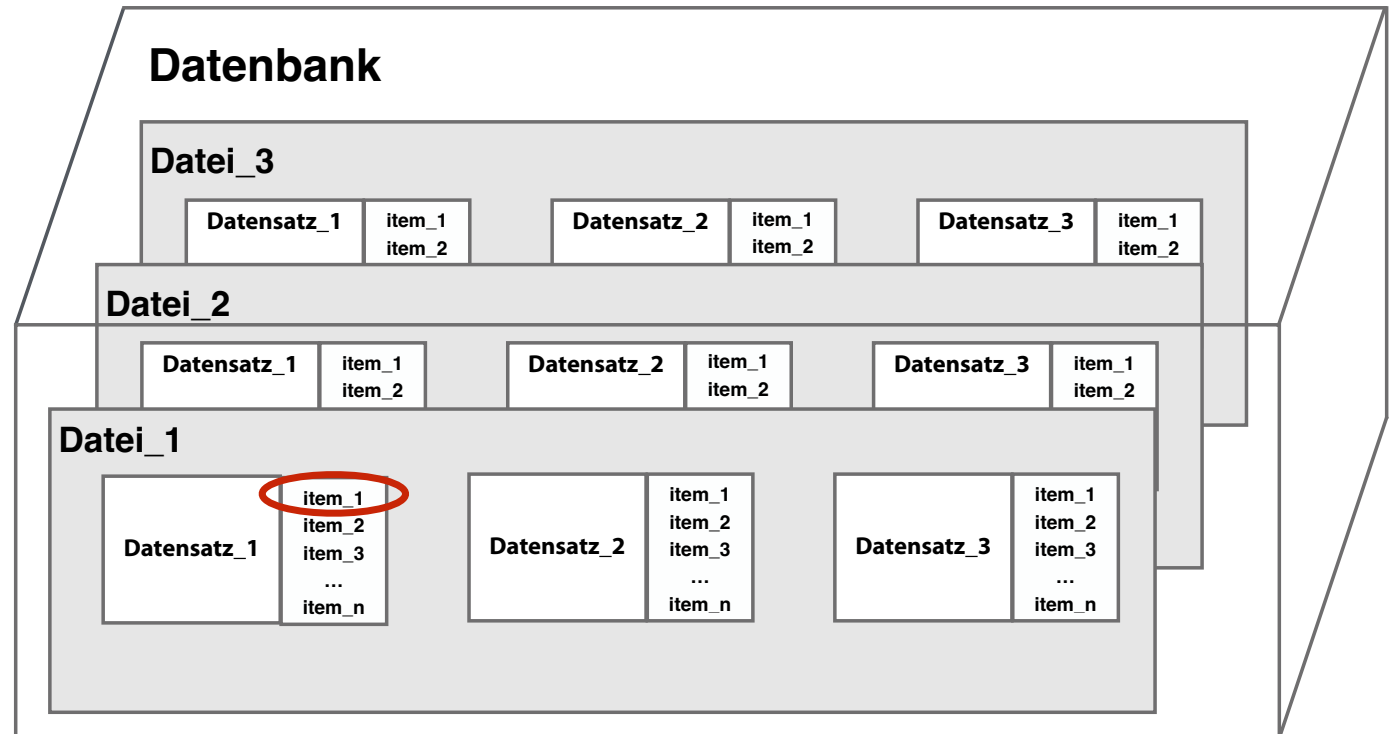
Beispiele - Objekte

- Personen: Lieferanten, Kunden, Mitarbeiter
- Gegenstände: Maschinen, Werkzeuge, Materialien
- Abstrakte: Konten, Buchungen, Rechnungen

Dazu werden Verfahren zur Strukturierung von Daten bzw. -beständen eingesetzt.

Datenstrukturen (Dateneinheiten)

- **Datenelement** (item) - kleinste logische Dateneinheit (z.B. Wohnort, Produktnummer, Nachname, etc.)
- **Datensatz** (record) - Bildung durch Datenelemente desselben Objekts (z.B. Kunde = Kundennummer, Nachname, Vorname, etc.)
- **Datei** (file) - gleichartige und logisch zusammengehörige Datensätze (z.B. Kunden, Produkte, Zulieferer, etc.)
- **Datenbank** (data base) - kann aus mehreren Dateien bestehen (Hinweis: Zwischen den einzelnen Dateien bestehen dabei immer logische Abhängigkeiten)



Datenstrukturen beschreiben, wie Daten inhaltlich miteinander verbunden werden.



Modellierung - Abbildung des betrachteten Originals

Vom Original zum Modell - Abbildungsschritte

Datenmodelle und -strukturen

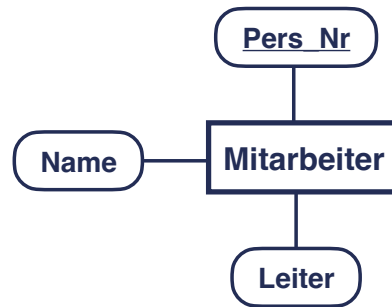
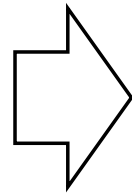
Datenbanken und Datenbankmanagementsysteme

Das konzeptionelle Schema: Prinzip des Entity Relationship Modells (ERM)

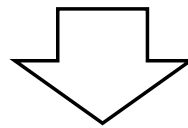
Schritte der Abbildung der realen Welt



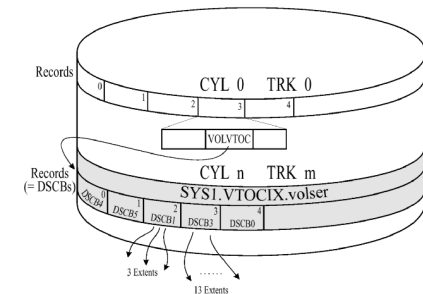
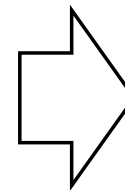
Ausschnitt der realen Welt



Konzeptuelles Modell



Logisches Modell



Physisches Modell

Abbildung der realen Welt

Merkmale

- **Ausgangspunkt** - Abstraktion der betrachteten (Diskurs)Welt
- **Betrachtungsfelder** - Vorgänge, Veränderungen oder statische Momentaufnahmen
- **Zweck** - strukturierte und geordnete Repräsentation als 1-zu-1 Abbildung mit hinreichender Genauigkeit
- **Ziel** - Beschreibung, Analyse, Entscheidung

Beispiele

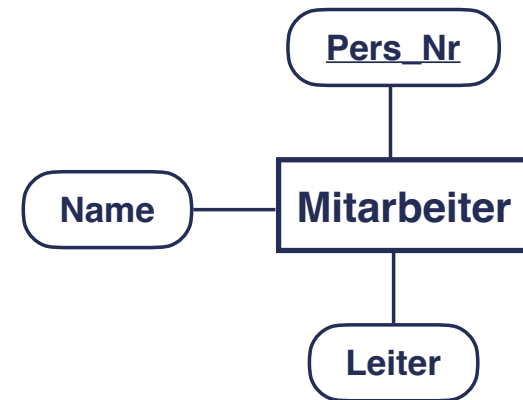
- Arbeitsgebiet des Einkäufers/Verkäufers
- Lager
- Buchhaltungsbereiche
- Produktionsprozesse



Konzeptuelles Schema



(Datenbankentwurf)



Aufgabe

- Erzeugung einer umfassenden Strukturierung der gesamten Informationsanforderungen
- Konzeptueller Entwurf der Miniwelt

Ergebnis

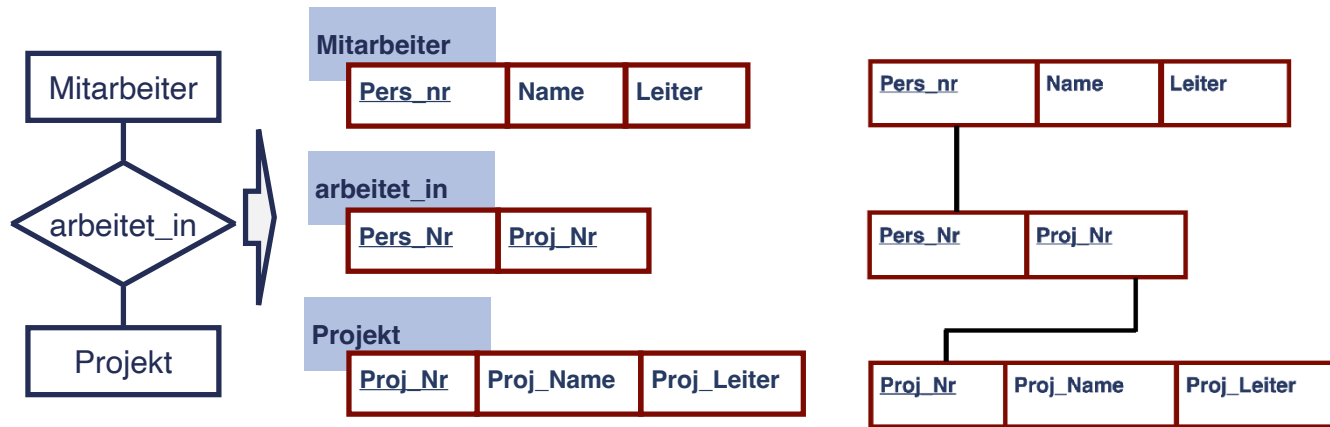
- Konzeptuelles Schema
- Umfassende Beschreibung der gesamt interessierenden Anwendungswelt (z.B. Schema einer Import-/ Exportfirma)

Darstellung

- Entity Relationship Modell

...stellt zu betrachtende Daten anwendungs- und speicherneutral dar.

Logisches Schema



Relationales Datenmodell

Relationstyp/Relationsformat

Attribut

Relation

Fremdschlüssel

Tupel

Attributwert

Weiterführung

- Logischer Entwurf
- Abbildung der Zusammenhänge des konzeptuellen Schemas in Relationsschemata

Ergebnis

- Logisches Schema
- Teil des zum Einsatz kommenden (relationalen) Datenbanksystems (DB-Realisierung)

Darstellung

- Relationenmodell

Die Logische Datenmodellierung stellt sicher, dass der Datenbestand vollständig anwendungsneutral und unabhängig von der späteren physischen Speicherung beschrieben wird.

Physisches Schema

Relationales Datenmodell

Relationstyp/Relationsformat

Attribut

Relation

Fremdschlüssel

Tupel

Attributwert

Realisierung

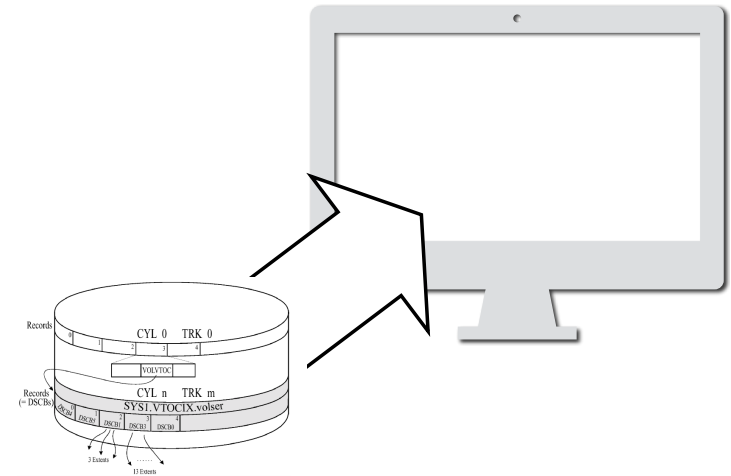
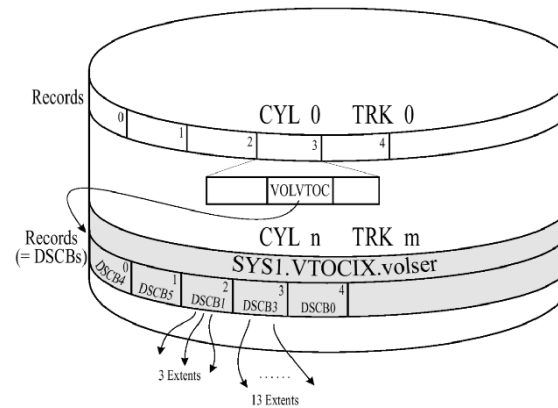
- Physischer Entwurf
- Speicherung von Relationen auf Speichermedien

Ergebnis

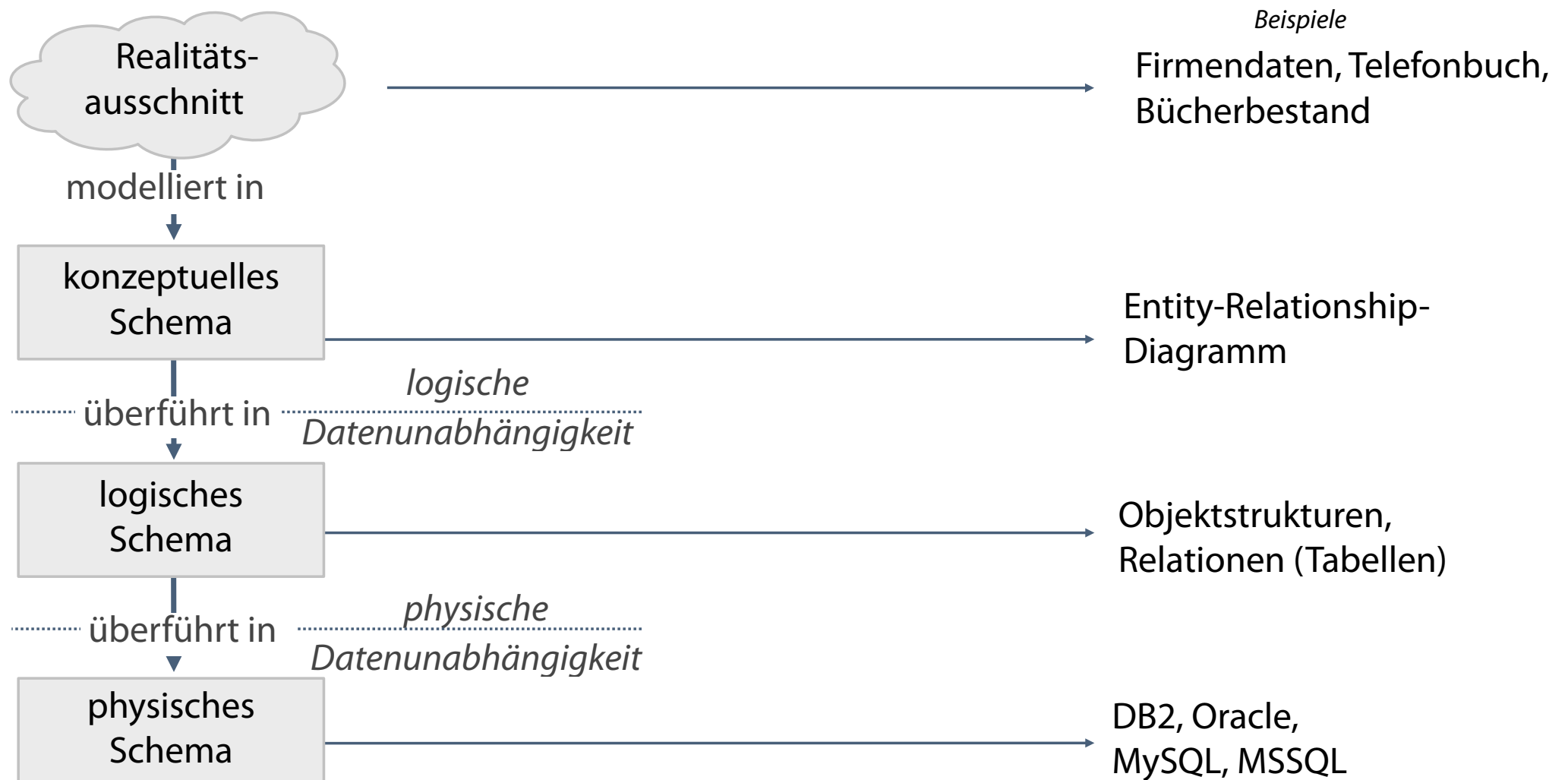
- Physische Ablage der Daten auf Speichermedium als Dateien, physische Records unsortiert
- Zugriffsbeschleunigung durch Indizierung der Records

Darstellung

- Interne Speicherorganisation einer Festplatte
- Zugriffe auf Hardware von Datenbank (Oracle u.a.) über Betriebssystem



Abbildungsschritte von der Realität zur physischen Datenbank



Schemata sind zentrale Ergebnisse der Datenmodellierung!



Modellierung - Abbildung des betrachteten Originals

Vom Original zum Modell - Abbildungsschritte

Datenmodelle und -strukturen

Datenbanken und Datenbankmanagementsysteme

Das konzeptionelle Schema: Prinzip des Entity Relationship Modells (ERM)

Hörsaal-Quiz - Recap erste Hälfte

Öffnet die App über den QR-Code oder den Link:



<https://quiz.lswi.de/>

Pwd: ewinf

Datenmodelle

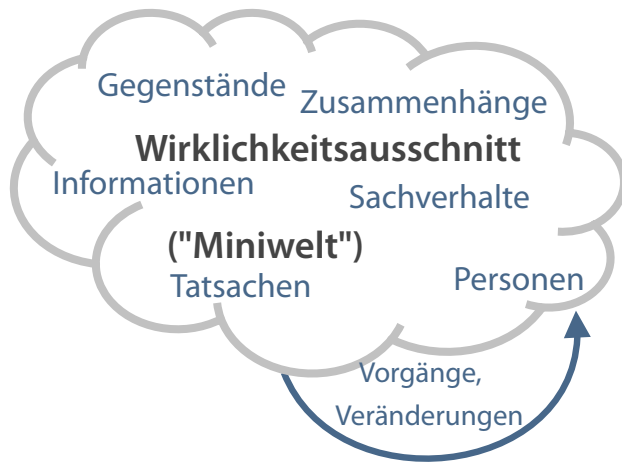
Inhalte

- Beschreibung einer Ordnungsvorstellung zur Strukturierung der Daten in einer Datenbank
- Festlegung der elementaren Datentypen und Datenstrukturen
- Beachtung eines Diskursbereichs (klar nach außen abgegrenzte Miniwelt)
- Vorgabe der Hilfsmittel, mit denen ein Diskursbereich hinsichtlich seiner Struktur und Inhalte modelliert wird
- Beschreibung der statischen Aspekte eines Systems

Beispiele

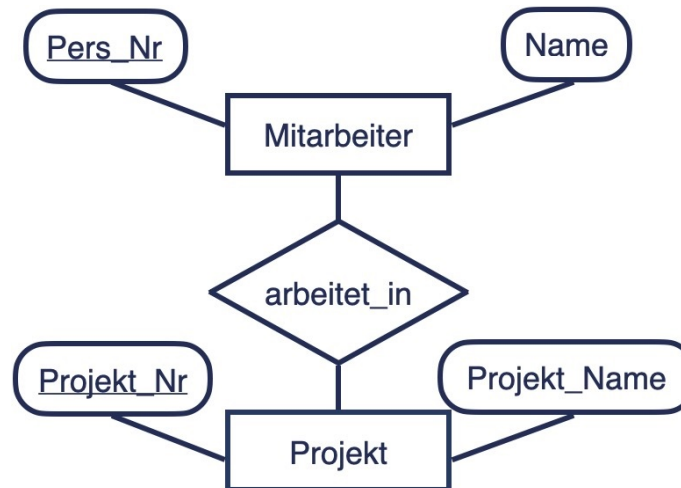
- Entity-Relationship-Modell (ERM, viele Erweiterungen)
- Relationales Modell
- Objektorientiertes Datenmodell
- Objekt-relationales Datenmodell (Kombination aus den beiden vorgenannten)

Datenmodellierung für statische Systeme



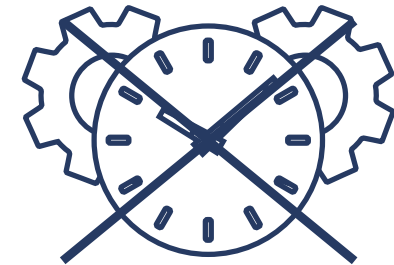
Ziel

- Modellierung einer Miniwelt
- Beschreibung der Struktur großer Datenmengen
- Beispiele: Kundendaten, Bibliotheksbestand, ...



Gegenstand der Beschreibung

- Beteiligte Objekte (Entitäten, Datensätze, ...) und deren Eigenschaften (Attribute)
- Statische Beziehungen zwischen den Objekten



NICHT Gegenstand der Beschreibung

- Abläufe resp. zeitliches Verhalten etc.
- Datenflüsse oder Interaktionen



Modellierung - Abbildung des betrachteten Originals

Vom Original zum Modell - Abbildungsschritte

Datenmodelle und -strukturen

Datenbanken und Datenbankmanagementsysteme

Das konzeptionelle Schema: Prinzip des Entity Relationship Modells (ERM)

Datenorientierte Software

Datenbankmanagementsystem (DBMS)

- Leistet technische Verwaltung einer oder mehrerer Datenbanken
- Bsp.: ORACLE, MySQL, DB2, Sybase, SAP HANA,...

Datenbanksystem (DBS)

- DBMS sowie von ihm verwaltete Datenbanken
- Bsp.: MySQL mit Datenbank der Import-/Export- Firma

(DB)-Anwendung, Anwendungsprogramm

- Greift auf DBS zu - realisiert Funktionen für Anwender
- Bsp.: Auftragserfassung der Import-/Export-Firma

Informationssystem (IS)

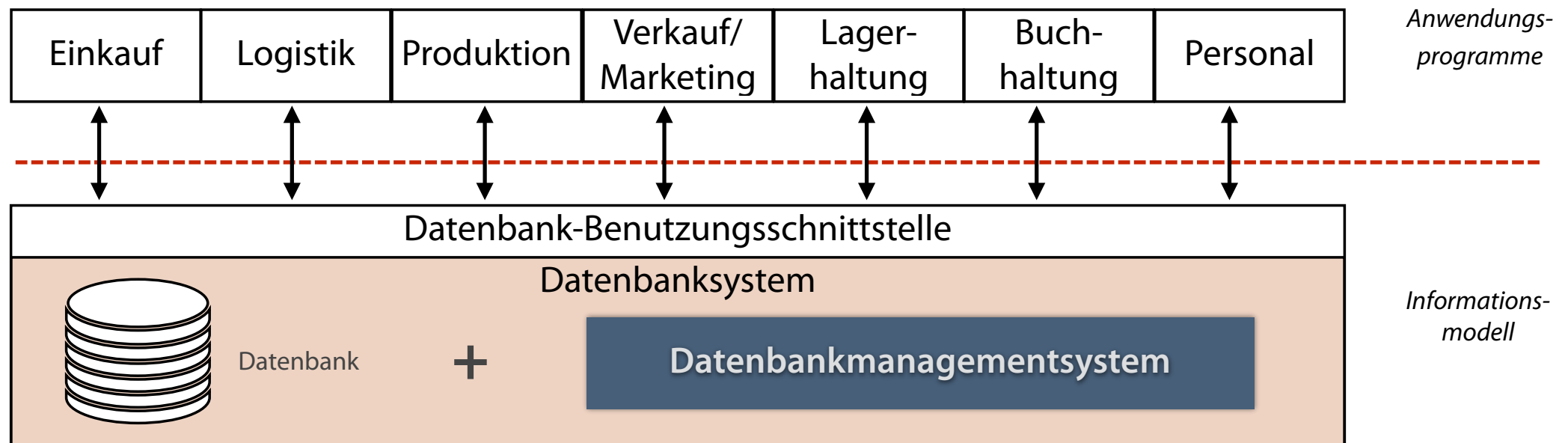
- Anwendungen und DBS sind zusammengefasst
- Bsp.: Gesamte, integrierte operative SW Import-/Export-Firma

Aufgabe und Ziel der Software besteht in der Strukturierung und Speicherung der Daten nach festgeschriebenen Ordnungskriterien.

Datenbanken

Inhalt der Datenbank

- Datenbankschema
- Eine Menge von konkreten Daten



Datenbanken beschreiben genau einen Zustand der im Schema modellierten Anwendungswelt.

Anforderungen durch Datenbanken

Anforderungen an die Hardware

- Große Speicherkapazität
- Effiziente Verarbeitung, kurze Antwortzeiten
- Niedrige Kosten

Anforderungen an die Software

- Datensicherheit und –schutz
- Vermeidung von Redundanz
- Konsistenz (Widerspruchsfreiheit)
- Persistenz (Robustheit gegenüber Hardwarefehlern)

Anforderungen an den Aufbau

- Datenunabhängigkeit: Änderungen auf einer Ebene wirken sich nicht auf andere Ebenen aus
- siehe auch Architekturmodell: 3-Schichtenkonzept

Anforderungen an die Bedienung

- Einhaltung von Standards
- Benutzerfreundlich, strukturierte Ablage (logisch, physisch)
- Mehrbenutzerbetrieb

Datenunabhängigkeit ist die wesentlichste Anforderung. Sie beinhaltet die Trennung von Daten und Programmcode.

Vorteile von Datenbanken

Einheitliches Konzept

- Verringerung von Erstellungs- und Verwaltungsaufwand
- Zusammenfassung mehrfach benötigter Funktionen

Vermeidung von Datenredundanz

- Zu jedem in DB gespeicherten Objekt genau ein Satz von Daten
- Verminderung mangelnder Übereinstimmung (z.B. bei Änderungen)

Kollisionsfreier paralleler Datenzugriff

- Gleichzeitiger Zugriff auf Daten durch mehrere Anwender
- Datenformat unabhängig von Bezug nehmenden Programmen
- Möglichkeit spontaner Abfragen abweichend von Programmen

Zentrale Datensicherheit und -schutz

- Vertraulichkeit
- Integrität der Daten

...gegenüber der Einzelablage im Dateisystem oder in der einzelnen Anwendung.

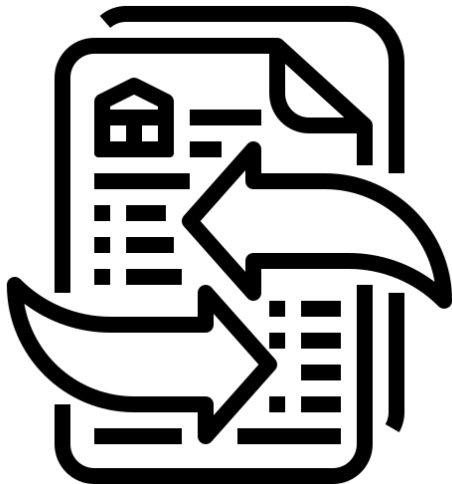
Schutz der Daten vor Verlust oder ungewollter Veränderung

Ziele

- **Datensicherheit** - Schutz vor Verlust von Datenbeständen durch technische Ausfälle
- **Datenintegrität** - Maßnahmen zur Gewährleistung unbeschädigter Daten in einem System während der Verarbeitung

Transaktionen als Konzept

- "Übergang der Datenbank von einem in einen anderen konsistenten Zustand"
- In sich abgeschlossener Verarbeitungsschritt innerhalb der Anwendungen der betreffenden Miniwelt
- Kann aus mehreren Teilschritten bzw. Vorgängen bestehen



ACID-Prinzip

Atomare Transaktionen (*Atomicity*)

- Änderung passiert ganz oder gar nicht
- Auch bei mehreren Schritten

Isolierte Transaktionen (*Isolation*)

- Unabhängig von eventuell parallel laufenden Prozessen
- Verarbeitung nur konsistenter Daten

Konsistente Transaktionen (*Consistency*)

- Datenbank wird in einem konsistenten Zustand gehalten
- War vor Ausführung der Änderung in einem solchen

Dauerhafte Transaktionen (*Durability*)

- Permanente Erhaltung geänderter Daten in der Datenbank
- Auch: Pufferdaten

Das Transaktionsprinzip realisiert die Sicherung vor Datenverlust.

Logische und physische Datenunabhängigkeit

Logische Datenunabhängigkeit (Anwendungsunabhängigkeit)

- Änderungen bzw. Erweiterungen von Anwendungsschnittstellen → keine Auswirkung auf DB-Funktionen
- Änderungen in der Systemumgebung (Datenbank-tuning, Erweiterung von Speicherstrukturen) → keine Auswirkung auf Funktionen in Anwendungsprogrammen

Beispiel:

- Erweitern einer Tabelle um neue Spalte für \$-Werte für Internationalisierung des Vertriebs

Physische Datenunabhängigkeit (Implementierungsunabhängigkeit)

- Beziehung zwischen logischer und physischer Ebene
- Wahl der Datenstruktur für die Speicherung der Daten → keine Auswirkung auf die benutzte konzeptuelle Sicht auf den Datenbestand
- Organisation von Datenstruktur und Zugriffspfaden ist nicht Programmaufgabe
→ Transparenz der physischen Organisation der Daten für Dialoge und Programme

Beispiel:

- „Umzug“ auf neuen Datenbank-Server

Einsatzbeispiele für Datenbanken

Traditionelle Bereiche

- Kaufmännische informationsverarbeitende Aktivitäten in Verwaltungsabteilungen großer Organisationen
- Beispiele: Versicherungen, Banken, Versand- und Telekommunikationsunternehmen etc.

Software

- Modulbibliotheken (Cross References)
- Repositories (z. B. Datenbanken zur Versionsverwaltung von Dokumenten)

Dienstleistungen

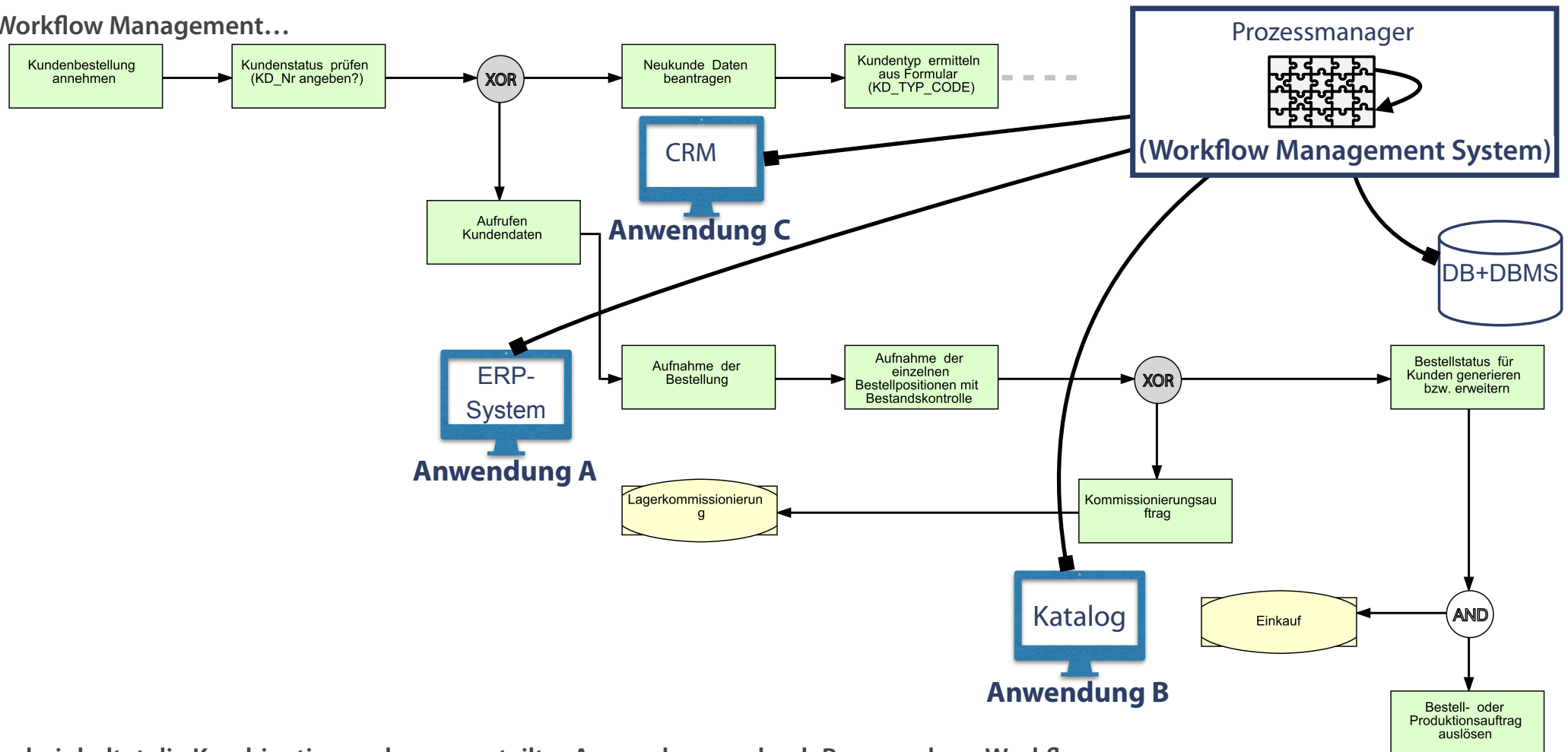
- Abrechnungs- und Auskunftssysteme
- Stadtkarten, Stadtplan/Fahrtrouten-Webdienste
- Strukturierte digitale Bibliotheken, z. B. Bilddatenbanken
- Kinoprogramm, Kleinanzeigenmarkt, ...

Unternehmen

- Warenwirtschaftssysteme
- Unternehmensplanungs- und -steuerungssysteme (ERP, PPS, ...)
- Kundenbetreuungssysteme (CRM)
- Verwaltung und Bearbeitung medialer Daten wie z. B. Webseiten (CMS)

Anwendungsbeispiele – Verteilte Systeme (Anwendungen)

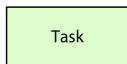
Workflow Management...



... beinhaltet die Kombination mehrerer verteilter Anwendungen durch Prozesse bzw. Workflows

Legend

Control Flow



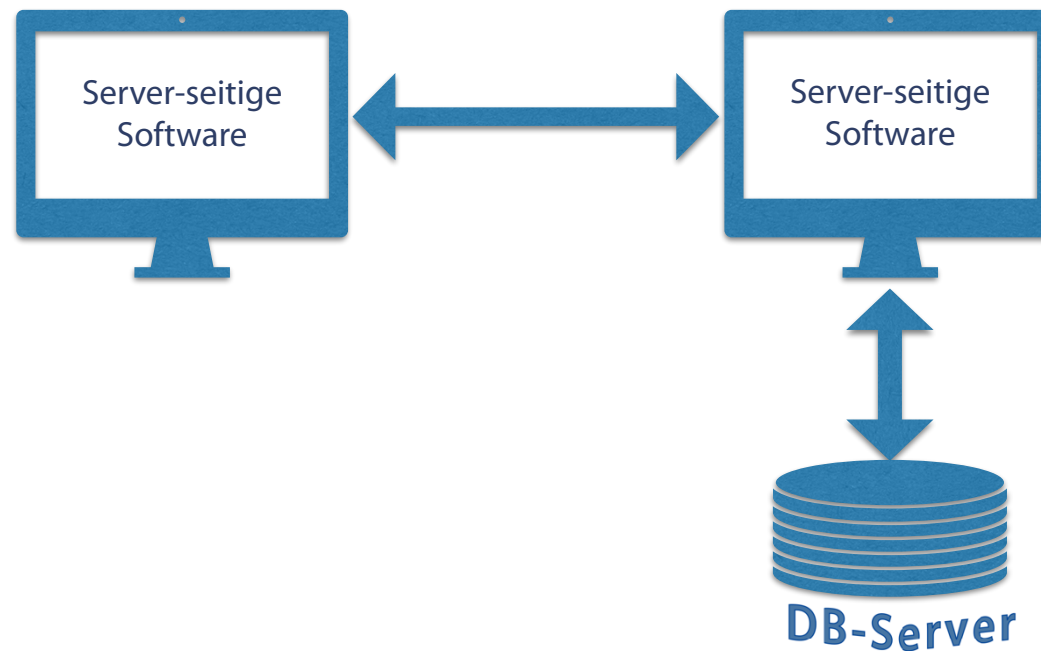
Architekturparadigmen – Client/Server-Systeme

Client/Server-System – zweischichtig

- Physikalische und räumliche Trennung von Client und Server
- Einfachste Variante eines verteilten Systems
- Verteilung der Komponenten über Rechnergrenzen

Aufgabenteilung zwischen Client und Server

- Datenverwaltung --> Server
- Präsentation --> Client





Modellierung - Abbildung des betrachteten Originals

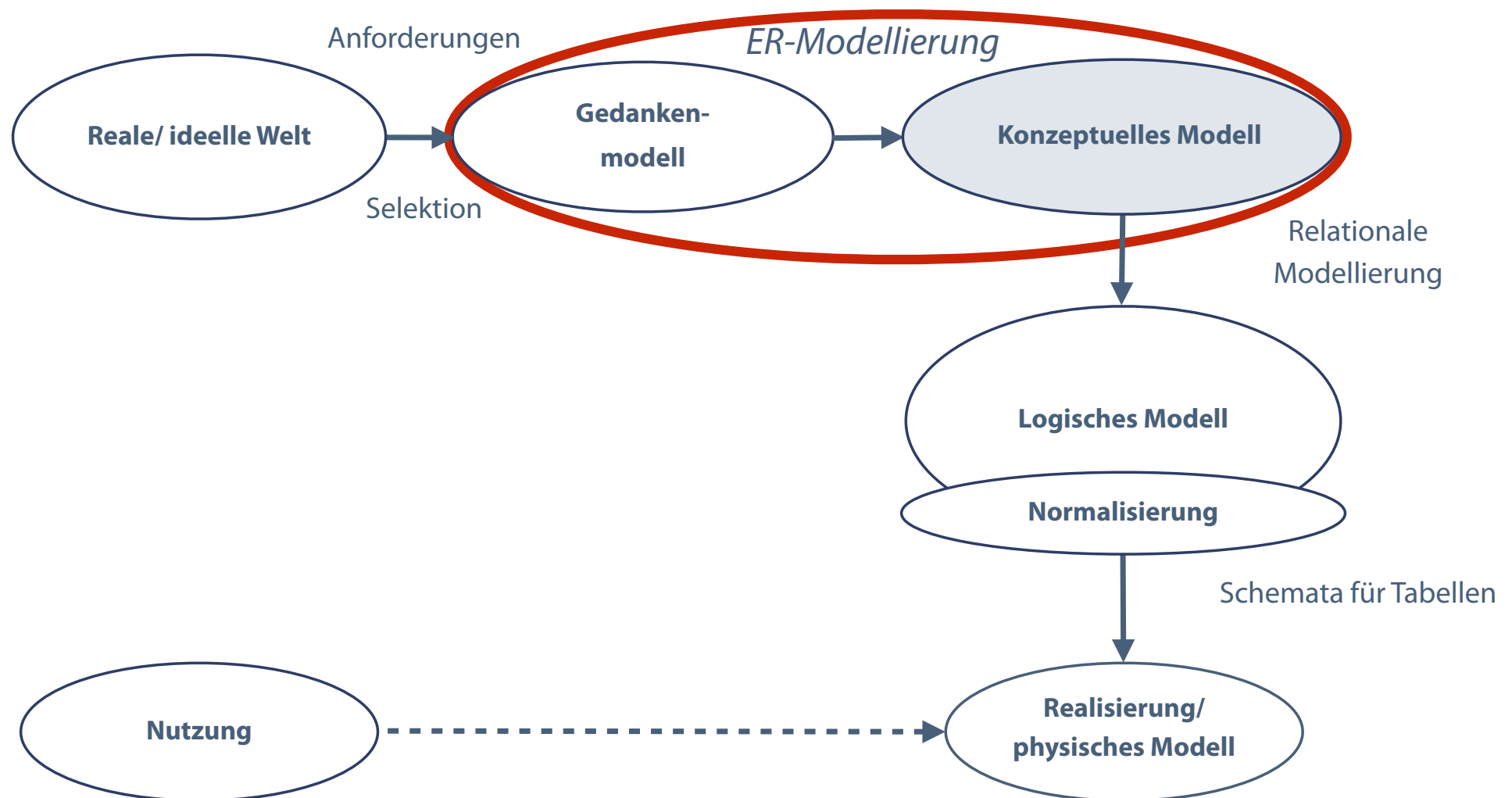
Vom Original zum Modell - Abbildungsschritte

Datenmodelle und -strukturen

Datenbanken und Datenbankmanagementsysteme

Das konzeptionelle Schema: Prinzip des Entity Relationship Modells (ERM)

Der Weg zur Datenbank - ER-Modellierung



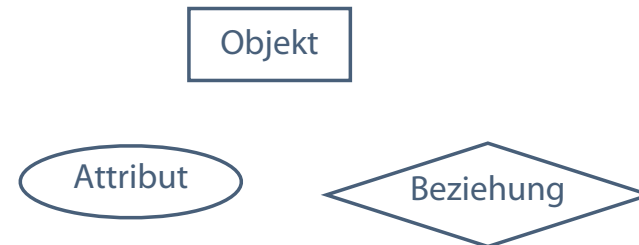
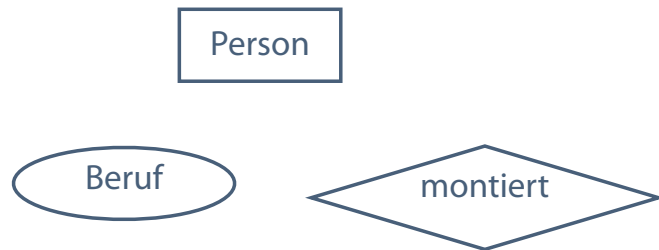
Das Entity Relationship Modell (ERM)

Objekte

- Entities - Informationsobjekte, Gegenstände
- Attribute - Eigenschaften dieser Objekte
- Beziehungen - Verknüpfung zwischen Entities

Symbole

- Entitytypen: Rechtecke
- Attribute: Ellipsen an Rechtecken oder Rauten
- Beziehungstypen: Rauten



Das ERM beschreibt Objekte als Namen im Singular und Beziehungen grafisch im ER-Diagramm.

Entity - Entitytyp

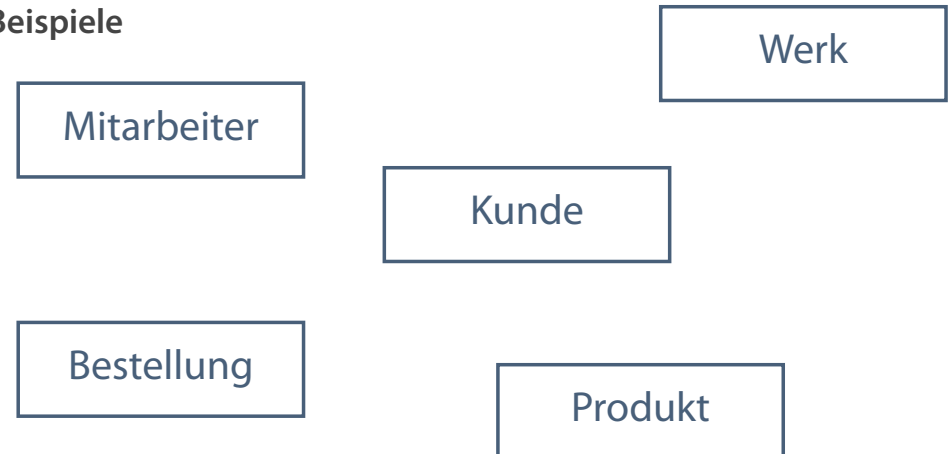
Entities (Objekte) - Informationseinheiten für das Modell

- Reale oder abstrakte Dinge, die für den zu betrachtenden Ausschnitt Relevanz besitzen
- Identifizierung eines Entity über ein eindeutig definierendes Merkmal (bzw. eine eindeutig definierende Kombination von Merkmalen)
- **Beispiel 1:** Mitarbeiter eines Unternehmens - Personalnummer
- **Beispiel 2:** Elektromotoren - Seriennummer
- **Beispiel 3:** Bestellung - Bestellnummer

Entitytyp - Zusammenfassung gleichartiger Entities

- Eindeutige Zuordnung jedes Entity zu einem Entitytyp
- Unterscheidung der Entities durch mindestens einen ihrer Attributwerte

Beispiele



Objekte bzw. Entitäten, die sich durch die selben Eigenschaften beschreiben lassen, werden zu einem Entitytyp zusammengefasst.

Das Attribut

Merkmal eines konkreten Objektes

- Zuordnung mehrerer Attribute (Merkmale) zu einem Objekt
- Ein oder mehrere Attribute (Attributkombination, Schlüssel) - Aufgabe der eindeutigen Identifizierung eines Entities
- Ermittlung der Beziehungskomplexität zwischen Entities

Eigenschaften eines Attributs

- Identifikation jedes Attributs über seinen Namen
- Vereinbarung eines Typs für Attributwerte
- Vereinbarung einer Optionalität

Das Attribut stellt eine Eigenschaft dar, die die Beschreibung einer Entität weiter ausführt.

Relationship - Relationshiptyp

Merkmal einer konkreten Beziehung

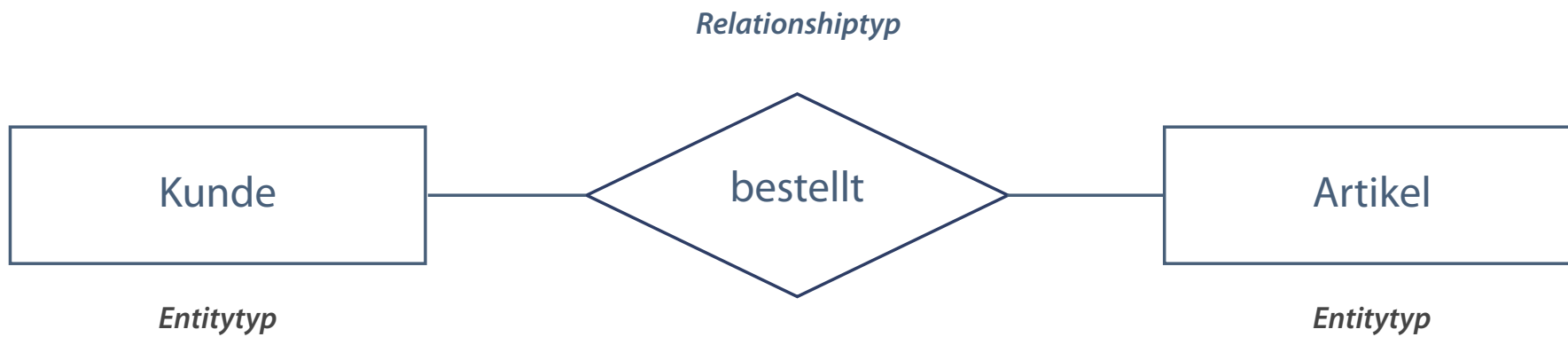
- Verknüpfung zwischen zwei (oder mehreren) Entitytypen
- Prinzip der Zuordnung beschreibender Merkmale (Attribute) wie bei Entitytypen
- Besonderheit: Zuordnung vom eindeutig beschreibenden Merkmalen von den miteinander zu verbindenden Entitytypen

Unterschied zu Entitytypen

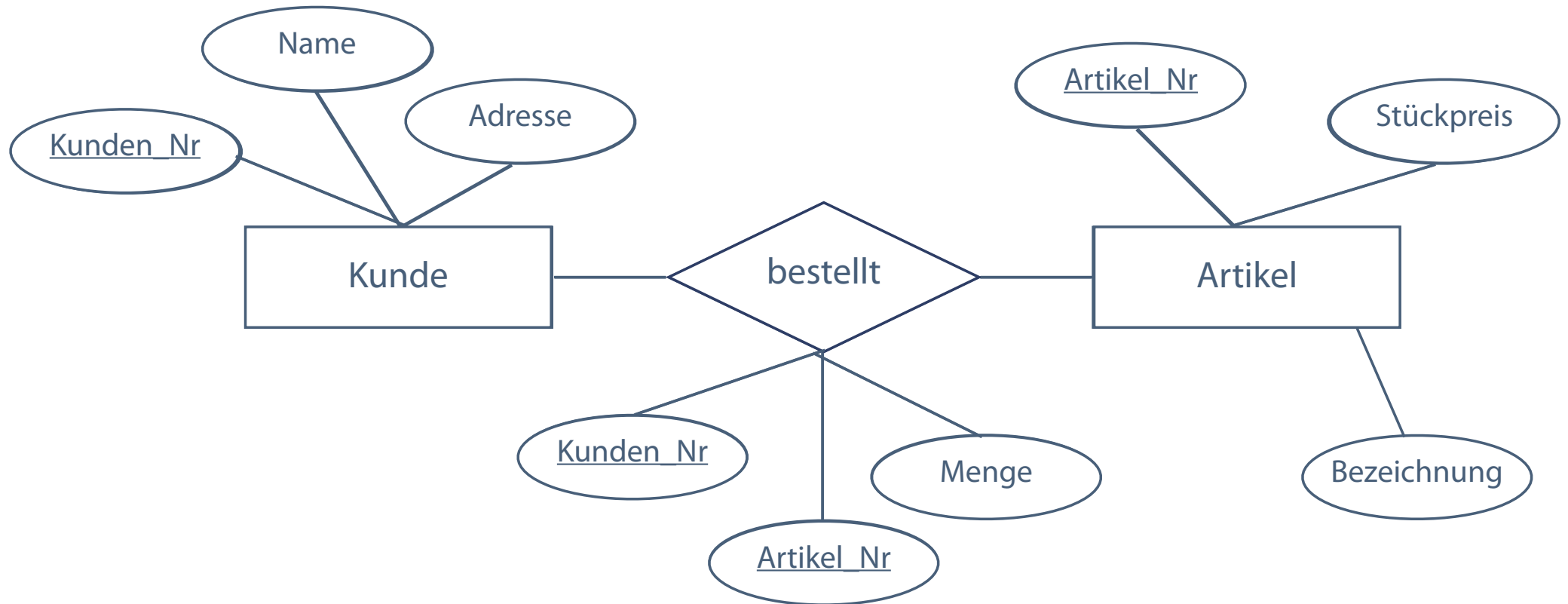
- Verknüpfungseigenschaften = Eigenschaften (Schlüsselwerte) aus den verbundenen Entities
- Identifizierung eines Relationships entweder über Kombination der Schlüsselwerte oder durch zusätzliches Attribut

Binäre Relationshiptypen setzen zwei Entitytypen miteinander in Verbindung.

Das Grundmodell



Beispiel eines Entity-Relationship-Diagramms (einfaches Prinzip)



Hörsaal-Quiz - Wissensvertiefung

Öffnet die App über den QR-Code oder den Link:



<https://quiz.lswi.de/>

pwd: ewinf

Kontrollfragen

- Wie geschieht der Übergang von der realen Welt zur Datenbank?
- Welche Aufgaben hat ein Datenbankmanagementsystem?
- Was ist eine Transaktion?
- Warum muss der normale Benutzer sich nicht um den Mehrbenutzerbetrieb kümmern?

Literatur

Elmasri, R./Navathe, S. B.: Grundlagen von Datenbanksystemen; 3. Auflage, 2002, Addison-Wesley

Stahlknecht, P./Hasenkamp, U.: Einführung in die Wirtschaftsinformatik; 11. Auflage, 2004, Springer Verlag

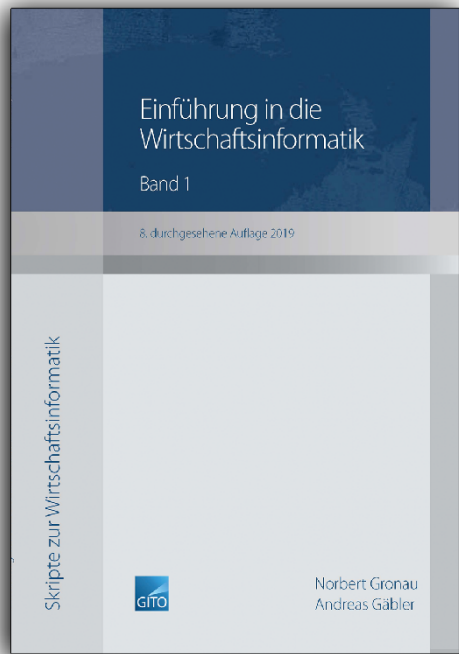
Mertens P. et. al: Grundzüge der Wirtschaftsinformatik; 12. Auflage; 2017, Springer Verlag

Laudon, Kenneth C./Laudon, Jane P./Schoder, Detlef: Wirtschaftsinformatik Eine Einführung; 3. Auflage, 2015, Pearson

Statista (2021): Monatliche Entwicklung des DAX 2021, Statista, [online] <https://de.statista.com/statistik/daten/studie/162176/umfrage/monatliche-entwicklung-des-dax/> [abgerufen am 01.10.2021].

Hilbert, M./P. Lopez (2011): The World's Technological Capacity to Store, Communicate, and Compute Information, in: Science, Bd. 332, Nr. 6025, S. 60–65, [online] doi:10.1126/science.1200970.

Zum Nachlesen



Gronau, N., Gäbler, A.:
Einführung in die Wirtschaftsinformatik, Band 1
8. überarbeitete Auflage
GITO Verlag Berlin 2019, ISBN 978-3-95545-233-9

Kontakt

Univ.-Prof. Dr.-Ing. Norbert Gronau
Center for Enterprise Research
Universität Potsdam
August-Bebel-Str. 89 | 14482 Potsdam
Germany
Tel. +49 331 977 3322
E-Mail ngronau@lswi.de