

Handout

Themenfeld: Datenbanken und SQL

Abschnitt: 05.01. Modellierung von DB Fortsetzung

Autor: Thomas Krause

Stand: 14.11.2022 12:02:00

Inhalt

1	Kardinalität 1:1	2
2	Spezialisierung, Generalisierung	3
3	Rekursive Beziehungen	8
4	optional/ zur Info: Aggregation, Komposition	10



1 Kardinalität 1:1

- (1) ERM bilden die semantische **Struktur** von Datenobjekten und ihren Beziehungen untereinander ab → Struktur bildet etwas Statisches ab und keine zeitlichen Verläufe (im Unterschied z.B. zum ProgrammAblaufPlan) → praktisch bedeutet das: zu jedem Betrachtungszeitpunkt müssen in der aus dem ERM hervorgegangenen Datenbank die Beziehungen und speziell die Kardinalitäten eingehalten werden
- (2) Beispiel zur Kardinalität 1:1 (siehe Datei 'H.05.01.Modellierung_von_DB_Fortsetzung.dia') → zu jedem Zeitpunkt darf in der Datenbank zu diesem ERM jeweils nur 1 Student mit 1 Zimmer verbunden sein.
- (3) für die praktische Umsetzung im Relationen-Modell können verschiedene Varianten realisiert werden



2 Spezialisierung, Generalisierung

siehe Datei '***.Modellierung_von_DB_Fortsetzung.dia' → zu Kapitel 2

zum Begriff der Klassifizierung zur Wiederholung:

Konzeptioneller Entwurf

- **Ziel/ Ergebnis:**
 - Erstellung eines **Informationsmodells**
 - das wird auch **semantisches Modell** genannt (Semantik = Lehre von der Bedeutung)
 - es existieren verschiedene Arten von Informationsmodellen
z.B. **ERM = Entity Relationship Model**
 - das ERM ist die Grundlage für die Phase 3 (logischer Entwurf)
- **Inhalt:**
 - Grundlage des konzeptionellen Entwurfs sind das Lastenheft und das Pflichtenheft → Beschreibung des Ist-Zustands im Unternehmen und der angebotenen Lösung
 - Untersuchung der Situation im Unternehmen, wo diese Datenbank eingesetzt werden soll
 - Identifizieren und Beschreiben der beteiligten Objekte → in der Datenbank zu speichernde Objekte → werden Entities genannt
 - Eigenschaften = Attribute der Objekte ermitteln
Beziehungen, Abhängigkeiten = Relations zwischen den Objekten ermitteln
 - **die Objekte und die Relations bilden Klassen**
 - Objekte, ihre Eigenschaften und ihre Beziehungen untereinander **werden standardisiert abgebildet** → das ist das ERD
 - grafische Beschreibung und fachliche Strukturierung der Daten → bilden die Unternehmenstätigkeit ab
 - konzeptionelle Abbildung = unabhängig von der künftig verwendeten Hard- und Software

ein ERM = Entity Relationship Model enthält folgende 2 Hauptelemente:

- **Entity-Klassen**
 - Entity-Klasse hat einen eindeutigen Namen
 - Eigenschaften = **Attribute** der Entity-Klassen
 - Attribute haben einen **eindeutigen Namen** und einen **Wertebereich** = Domain
 - **die** Attribute, die die Objekte einer E.-Klasse eindeutig voneinander unterscheiden → **Primärschlüssel** → werden im ERM unterstrichen
- **Relationship-Klassen**
 - Eigenschaften = Attribute der Relationship-Klassen
 - gekennzeichnet durch die Anzahl, wie oft ein Objekt maximal an der Beziehung beteiligt sein kann → **Kardinalitäten** = eine wichtige strukturelle Integritätsbedingung



1.) Entitäten (zu speichernde Objekte) identifizieren

- Untersuchung der Situation im Unternehmen, wo die Datenbank verwendet werden soll
- beteiligte Objekte (was sind Objekte?) ermitteln
 - Entität (dt.)/ Entity/ Entities: allgemein der Gegenstand, Objekt in der realen Welt
 - E. müssen unterscheidbar sein und sind unterscheidbar durch ihre **Attributwerte** bzw. durch **Attributkombinationen**
 - gleichartige Entities (mit **gemeinsamen** Eigenschaften) werden zu **Entity-Mengen/ Objekttypen / Entity-Klassen** zusammengefaßt
- Identifizierung der Entities über Abstraktionskonzepte → Methoden:
 - **Klassifikation**
 - **Generalisierung/ Spezialisierung**
 - **Aggregation**

Was bedeutet der Begriff „Klassifikation“?

- planmäßiges Erkennen, Analysieren, Einteilen und Zusammenfassen von Objekten mit **gemeinsamen** Eigenschaften zu **Klassen**
- Identifizierung und Abgrenzung **unterscheidbarer** Objekte
- gleiche Eigenschaften, gleiche Struktur/ Attribute, gleiche Operationen führen zur **gleichen Klasse**
- Objekte, die sich durch **dieselben Eigenschaften** unterscheiden lassen, bilden eine Klasse, manchmal auch Typ genannt
- zu jeder Klasse/ zu jedem Typ gehören also ein oder mehrere Objekte
- Beispiele für Klassifikation:
 - Welche gemeinsamen Eigenschaften haben folgende Objekte?
 - Baupläne: Nummer, Maßstab, Vorschrift, Status, Projektname, Ersteller
 - Kraftfahrzeuge: Typ, Nutzlast, Farbe, Kraftstoff, E10-Verträglichkeit
 - Projektionstechnik: Auflösung, Projektionsabstand, ...
 - Zu welcher Klasse könnten diese Objekte zusammengefasst werden?
 - Bildband, Reiseführer, Roman: Bücher → Seitenzahl, Titel, ISBN, Verlag, Autor
 - ct → Magazin → Druckerzeugnis → Papier

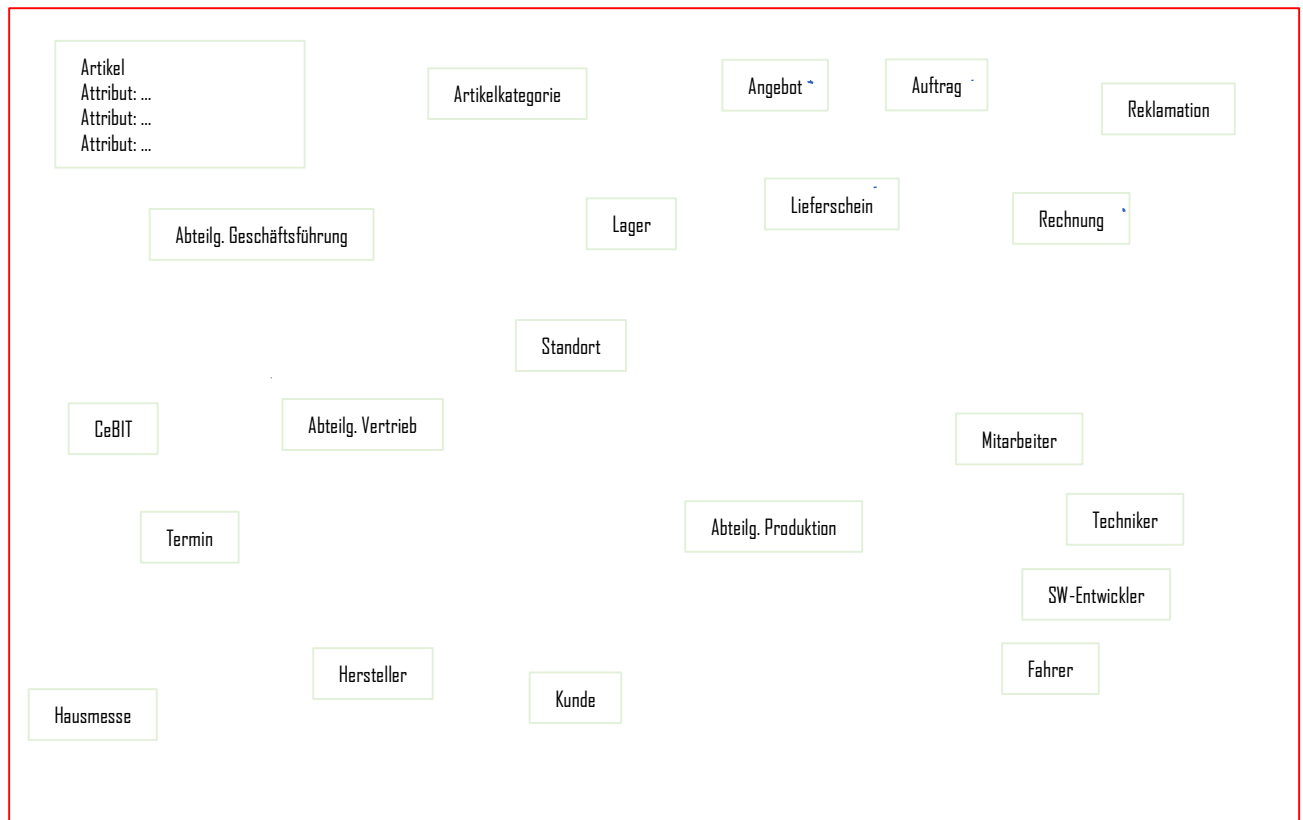


mögliche Klassifikation: Beispiel EDV-Großhandel

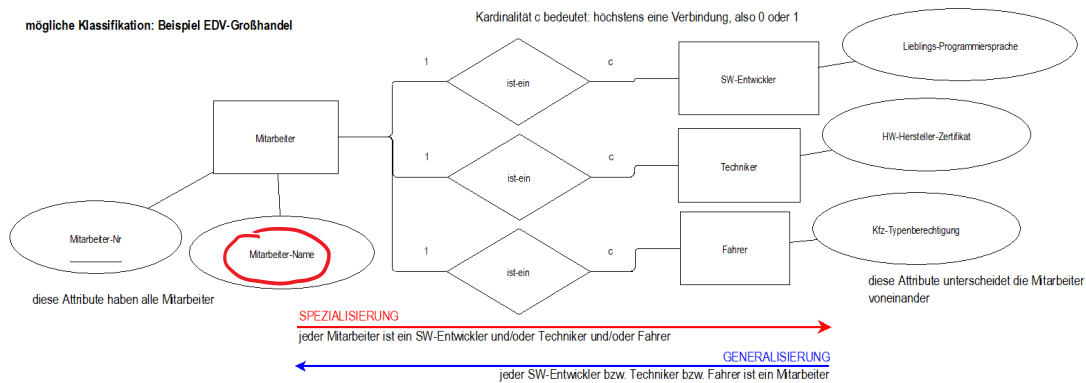
Durch welche wichtigen Eigenschaften = Attribute werden die aufgeführten Objekte beschrieben?

Welche Objekte verfügen über die gleichen Attribute?

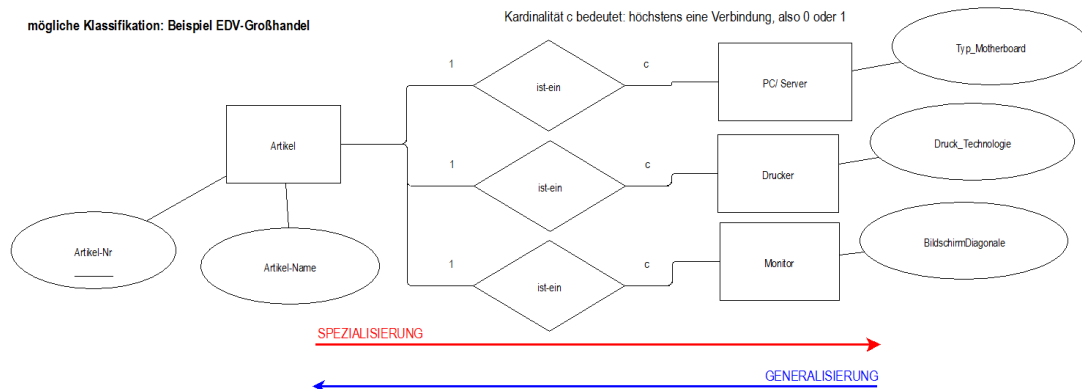
Wie lassen sich Objekte mit gleichen wichtigen Attributen zu Gruppen zusammenfassen?



mögliche Klassifikation: Beispiel EDV-Großhandel



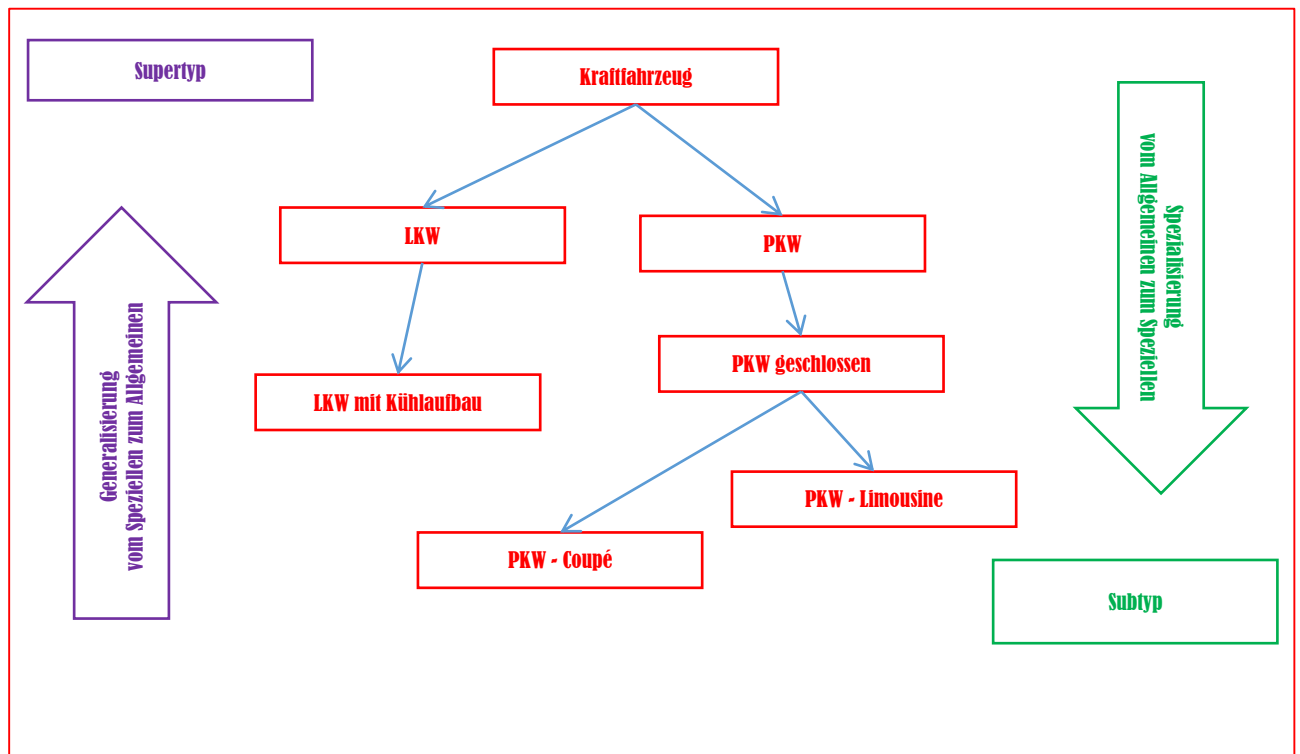
mögliche Klassifikation: Beispiel EDV-Großhandel



Generalisierung/ Spezialisierung

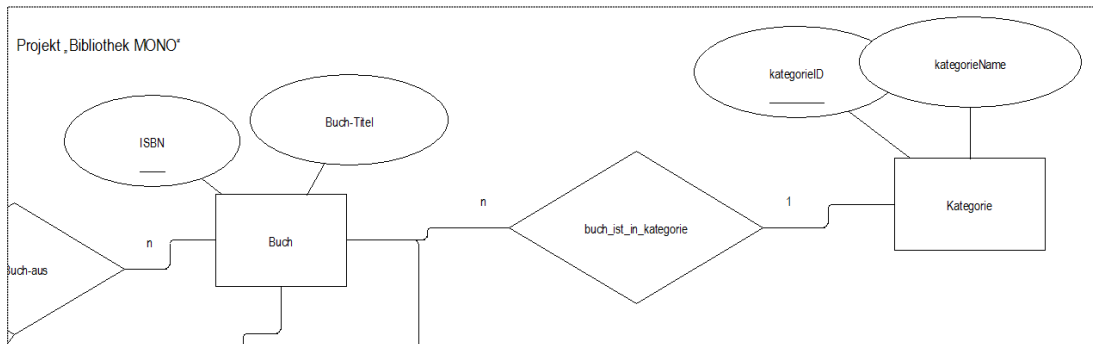
beschreibt das Verhältnis von Klassen/ Typen untereinander → Anordnung in Hierarchien, also untergeordnete und übergeordnete Klassen





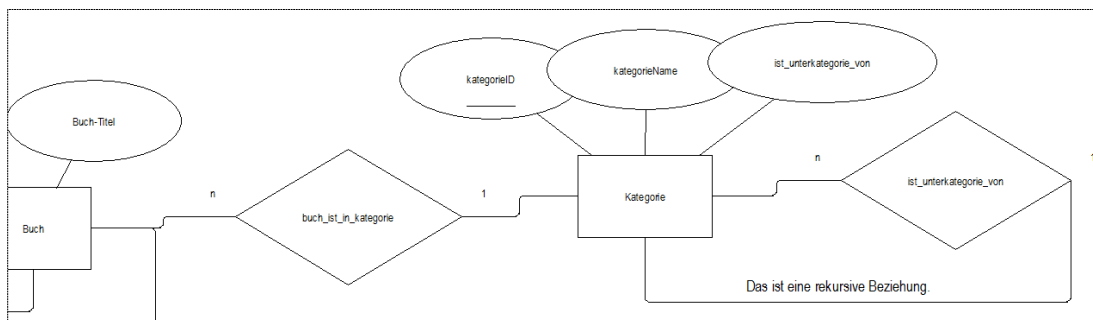
3 Rekursive Beziehungen

einfache Beziehungen



- in diesem Auszug eines ERMs wird einem Buch genau eine Kategorie zugewiesen
- die Kategorien werden in einer Tabelle mit 2 grundlegenden Attributen erfasst und stehen in keiner Beziehung zueinander
- alle Kategorien befinden sich auf einer Ebene und haben die gleiche Wertigkeit

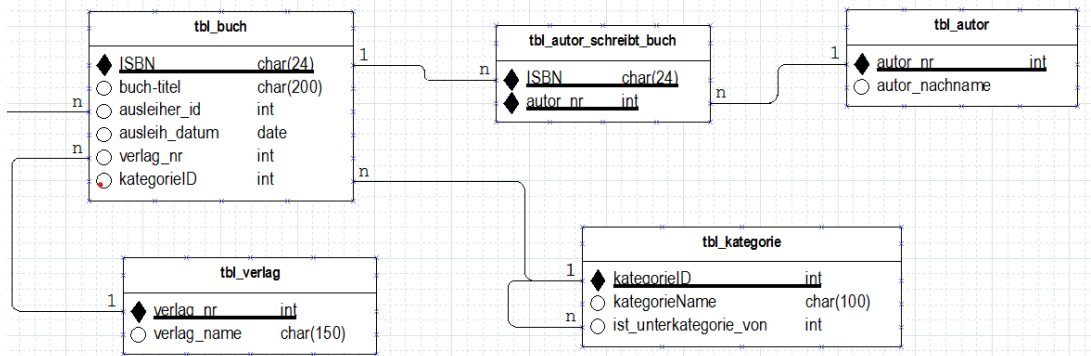
rekursive Beziehungen



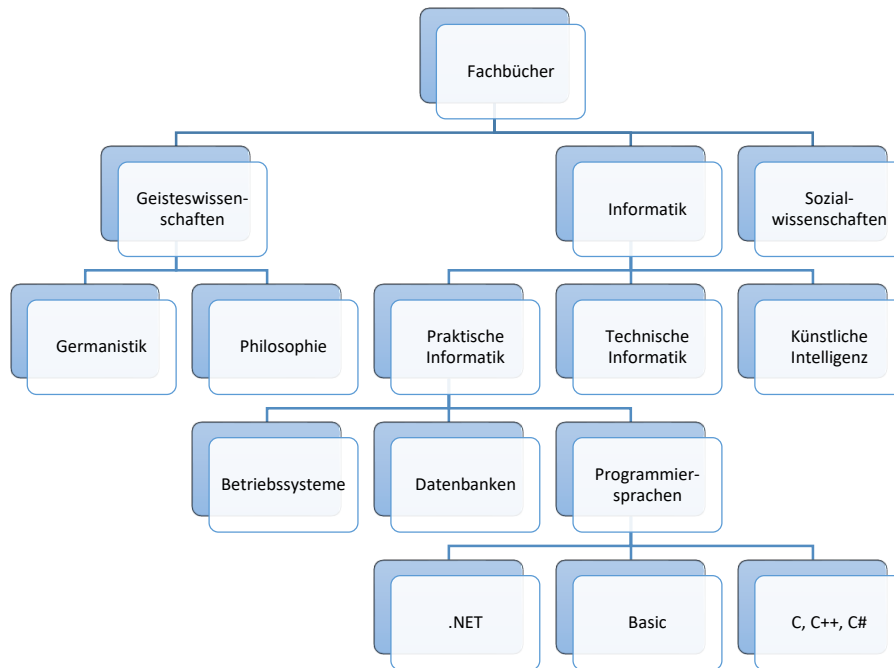
- die Entität wird über eine Beziehung mit sich selbst verknüpft
- über ein zusätzliches Attribut kann die Beziehung zu einem anderen Datensatz dargestellt werden:
in diesem Beispiel → **ist_unterkategorie_von**
- damit ist die Darstellung beliebig tiefer und beliebig verzweigter Baumstrukturen möglich



Beispiel Bibliothek:



Beispiel: buecher.de (Auszug)



4 optional/ zur Info: Aggregation, Komposition

Die Begriffe Aggregation und Komposition werden in weiteren Themenfeldern zur Programmierung nochmal eingeführt und praktisch vertieft. Hier zunächst nur einige grundlegende Informationen:

Aggregation

- Bildung von neuen Objekten durch Zusammensetzung einfacherer Objekte (zusammengesetztes Objekt)
 - ein Objekt ist Teil eines anderen, übergeordneten Objekts → Beziehung wird auch als 'is-part-of' bezeichnet
 - z.B.: Stücklisten
 - **WICHTIG:** die Teil-Objekte können auch ohne das übergeordnete Objekt unabhängig existieren
- Beispiele für Aggregation
 - Fahrgestell, Motor, Karosserie sind Teil-Objekte des Objekts Kraftfahrzeug
 - 1x Doppelbett, 1x Kleiderschrank, 2x Nachttische, 1x Kommode sind Teil-Objekte des Objekts 'Schlafzimmer Set VALIUM II Hochglanz schwarz'

Komposition

- Bildung von neuen Objekten durch Zusammensetzung einfacherer Objekte (zusammengesetztes Objekt)
 - ein Objekt ist Teil eines anderen, übergeordneten Objekts → Beziehung wird auch als 'is-part-of' bezeichnet
 - **WICHTIG:** die Teil-Objekte können NICHT ohne das übergeordnete Objekt unabhängig existieren
- Beispiele für Komposition
 - die einzelnen Räume sind Teil-Objekte des Objekts Gebäude
 - ???

