

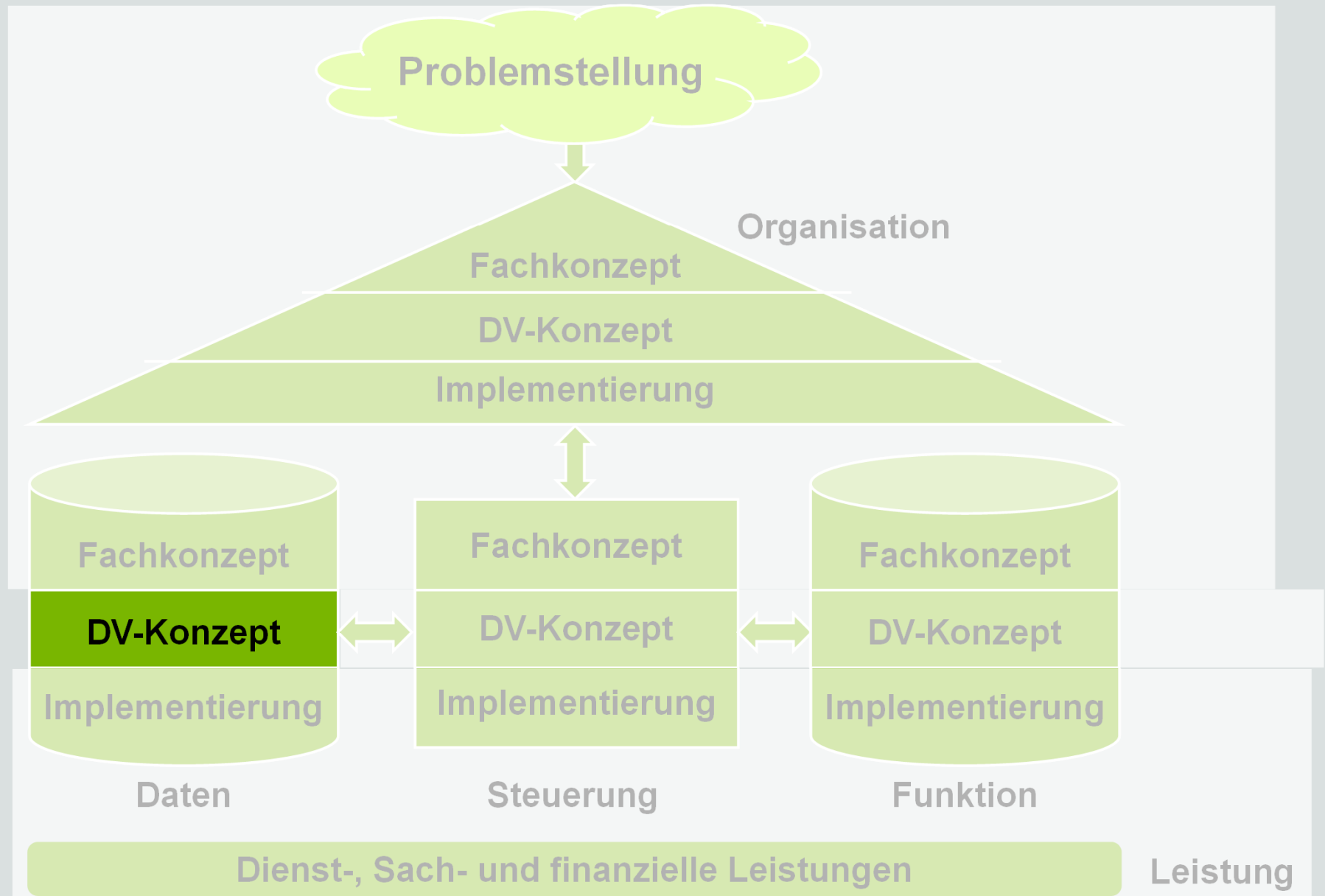
4 Datenbanken

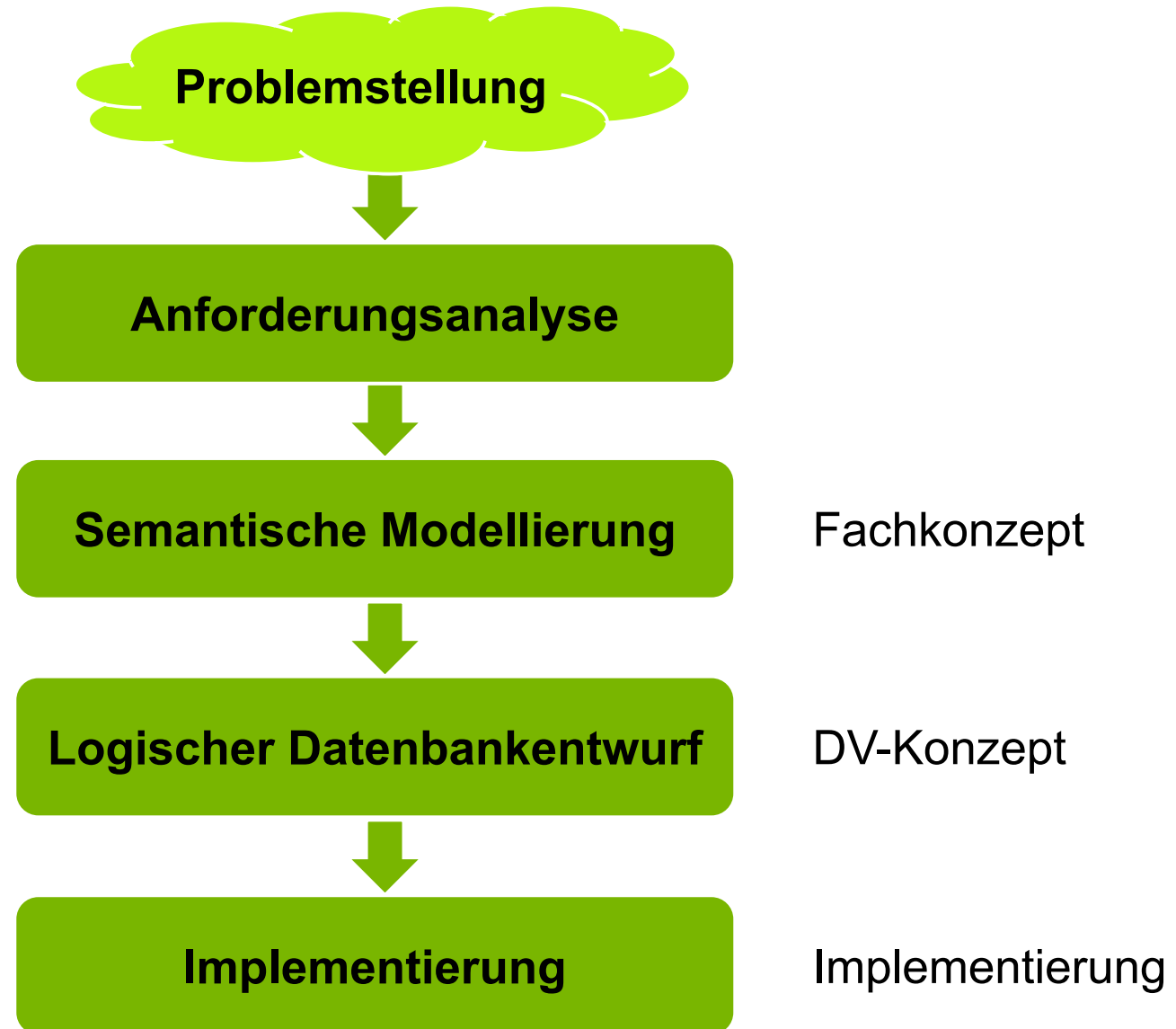
1. Motivation
2. Datenorganisation und Datenbankkonzept
3. Semantische Datenmodellierung
- 4. Umsetzung in Datenbanken**
5. Datenbanknutzung mit SQL
6. Transaktionsmanagement
7. Datenbankentwicklung
8. Datenbanken und IT-Sicherheit
9. Systemarchitektur
10. Verteilte Datenbanken
11. NoSQL und Entwicklungstrends

4 Lernziele

- Sie wissen, was „relationale“ Datenbanken sind.
- Sie können aus einem ER-Modell ein Relationenmodell erstellen.
- Sie kennen die Regeln für eine solche Erstellung.
- Sie können beurteilen, was ist ein „gutes“ Datenmodell ist.

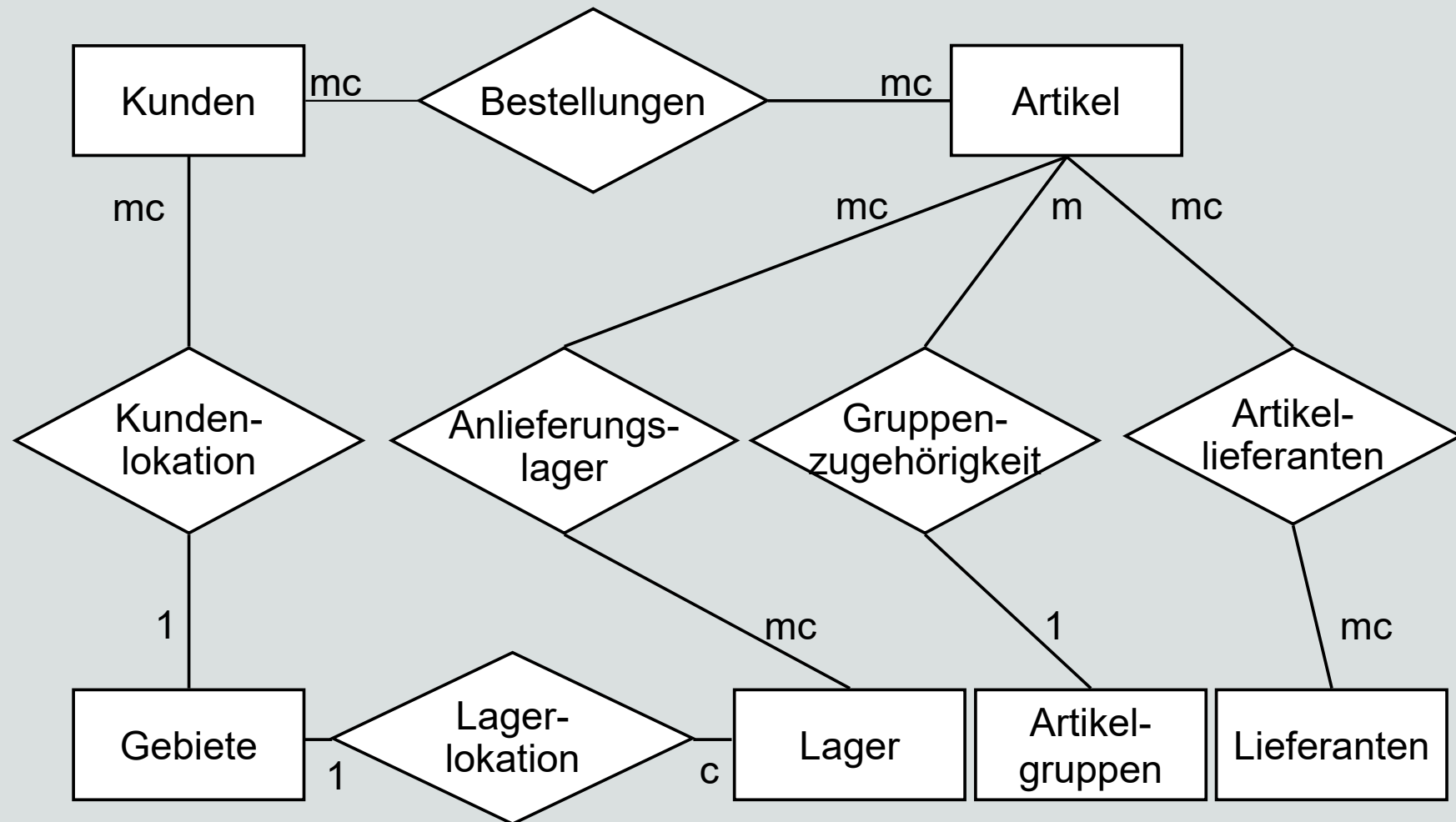
3 Einordnung ARIS-Konzept





- Datenbankabhängiger, modellhafter Entwurf
- Entwurfstypen
 - Hierarchisches Modell (verwendet bis ca. 1970, eingeschränkt)
 - Netzwerkmodell (nicht effizient realisierbar)
 - Objektmodell (nicht effizient realisierbar)
 - Relationenmodell
 - etc.

Beispiel ER-Diagramm: Nachfolgend in verschiedensten Entwurfstypen



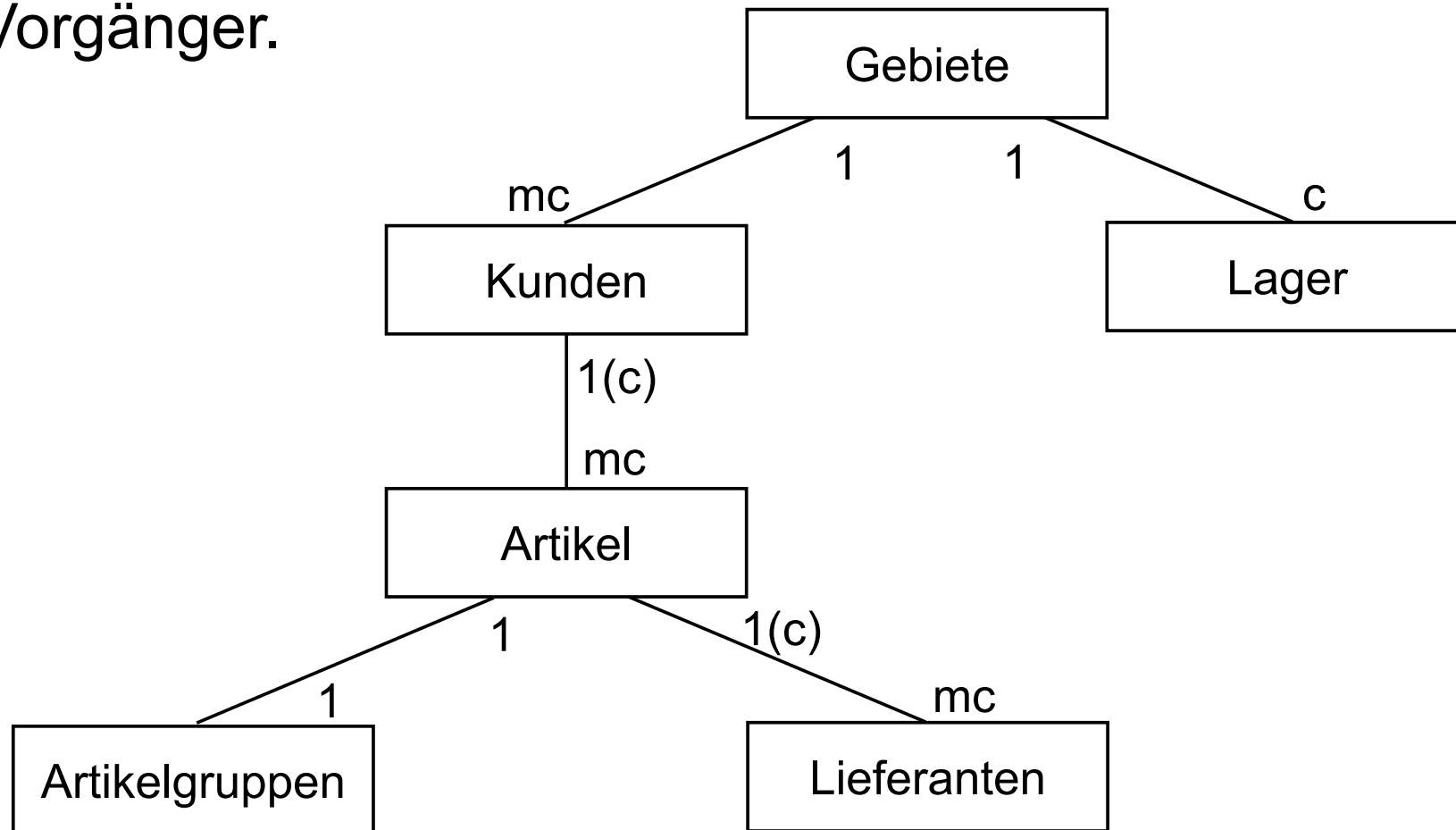
4 Hierarchisches Modell

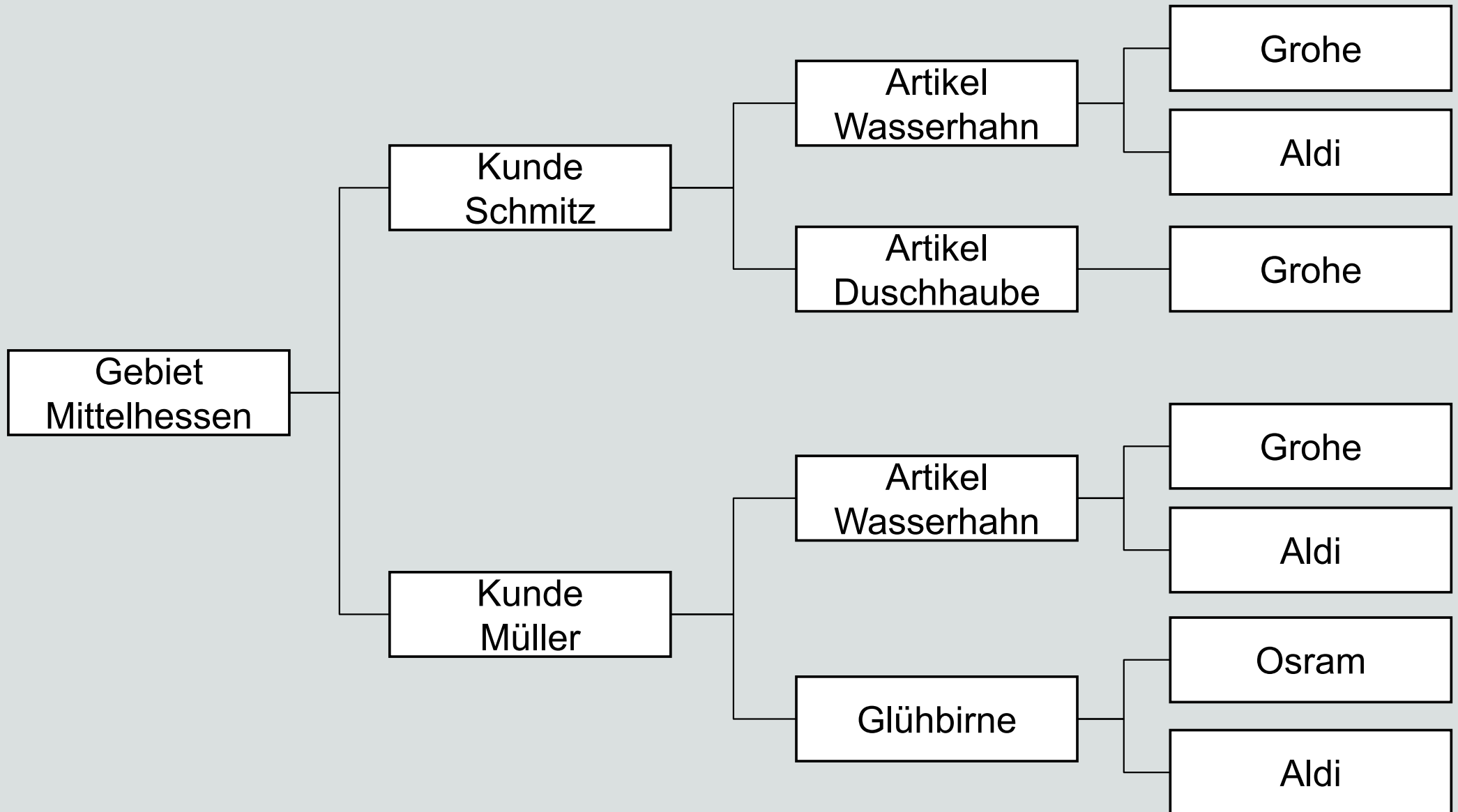
Verbreitung

- Grundlage der meisten Datenbanksysteme bis in die 80er Jahre
- z.B. IMS von IBM

4 Hierarchisches Modell

- Abbildung der Datenobjekte in hierarchischer Baumstruktur
- Jeder Datensatz (bis auf die Wurzel) hat genau einen Vorgänger.





- Folge der Baumstruktur:
m:m-Beziehungen müssen in 1:m Beziehungen umgewandelt werden

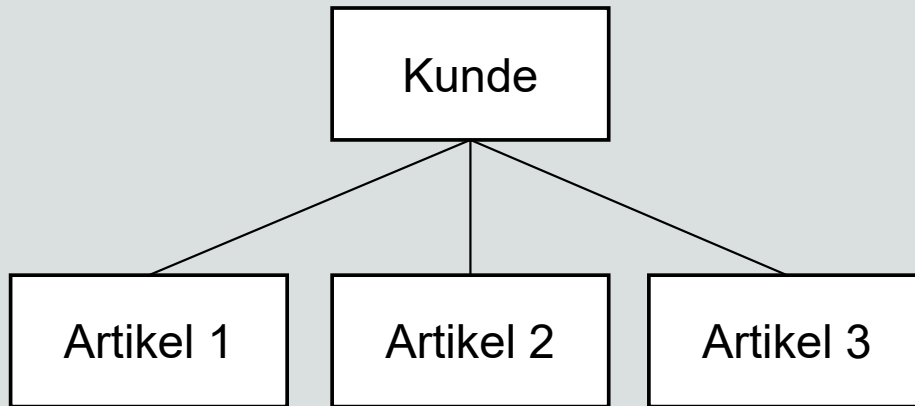


wird zu

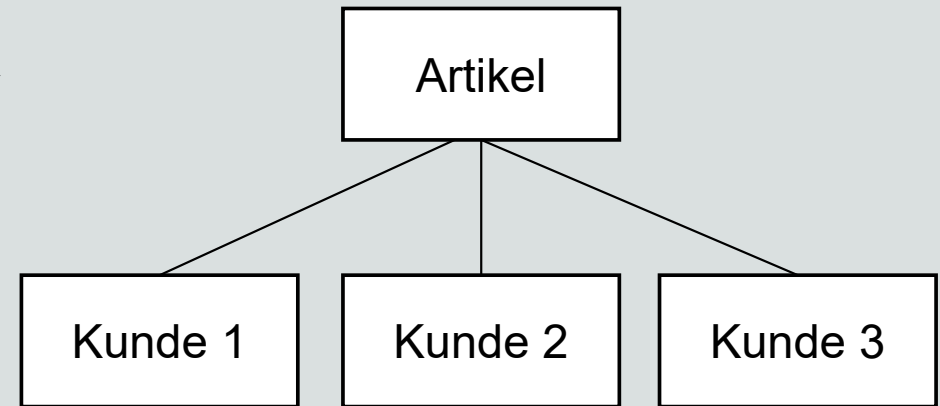


oder





oder



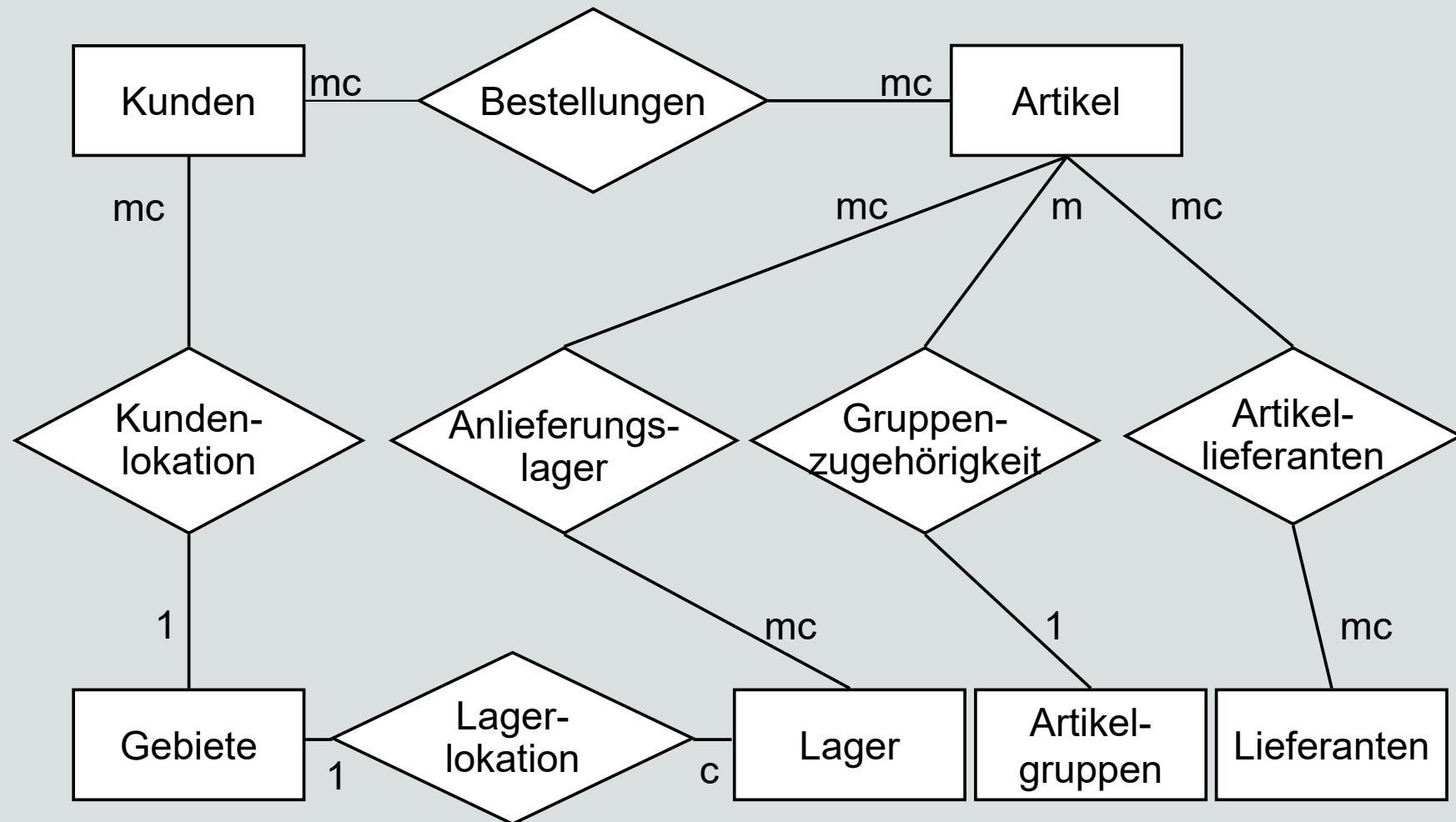
- Vorteile
 - Einfache, sequentielle Speicherung
 - Effiziente Auswertungen bei hierarchiekonformen Anfragen
- Nachteile
 - Gefahr von Informationsverlusten gegenüber konzeptionellem Modell und Realsystem
 - Redundanz von Daten
 - Eingeschränkte Auswertungsflexibilität
 - Hoher Retrievalaufwand bei quer zur Hierarchie verlaufenden Auswertungen

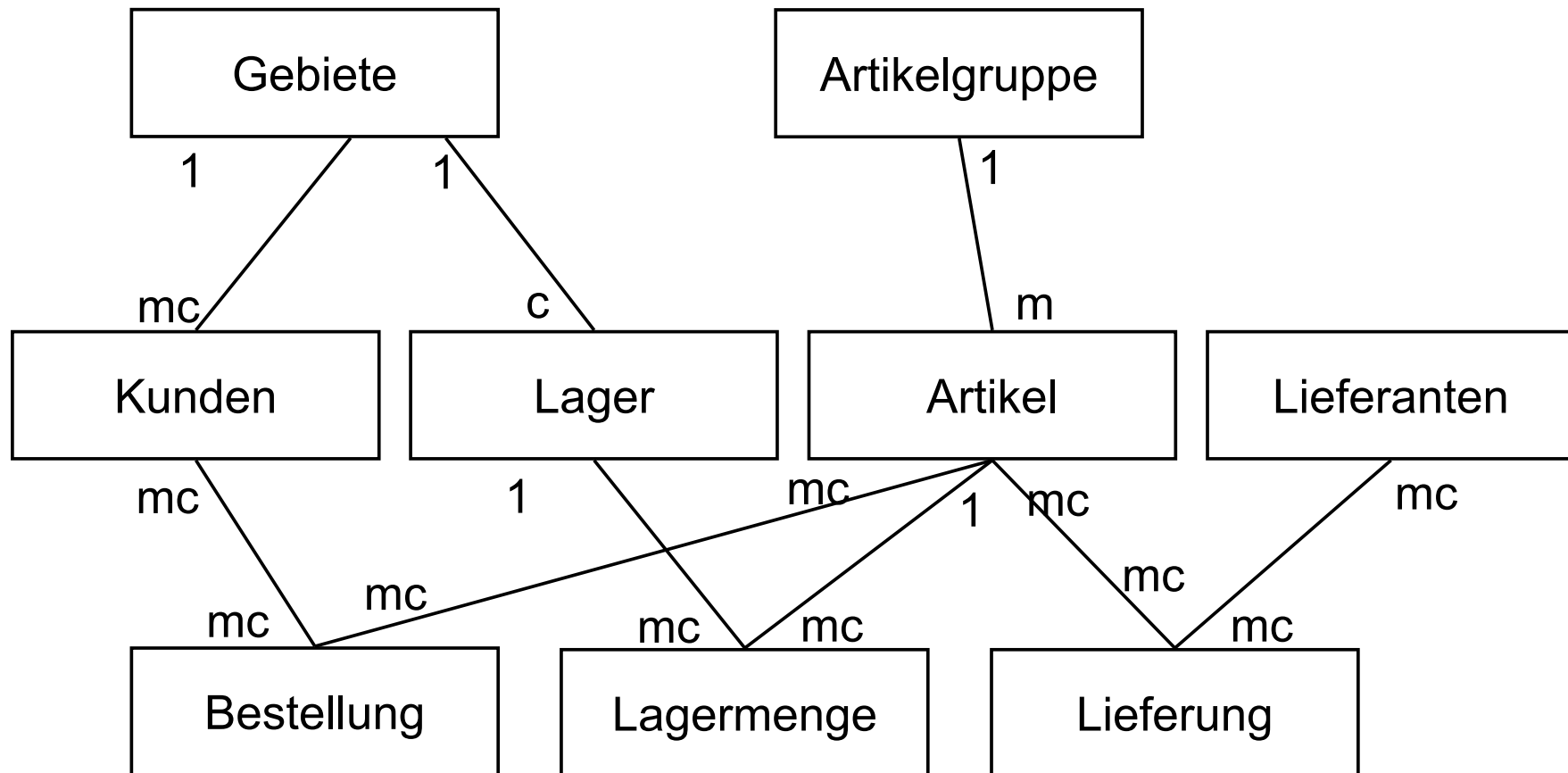
4 Netzwerkmodell

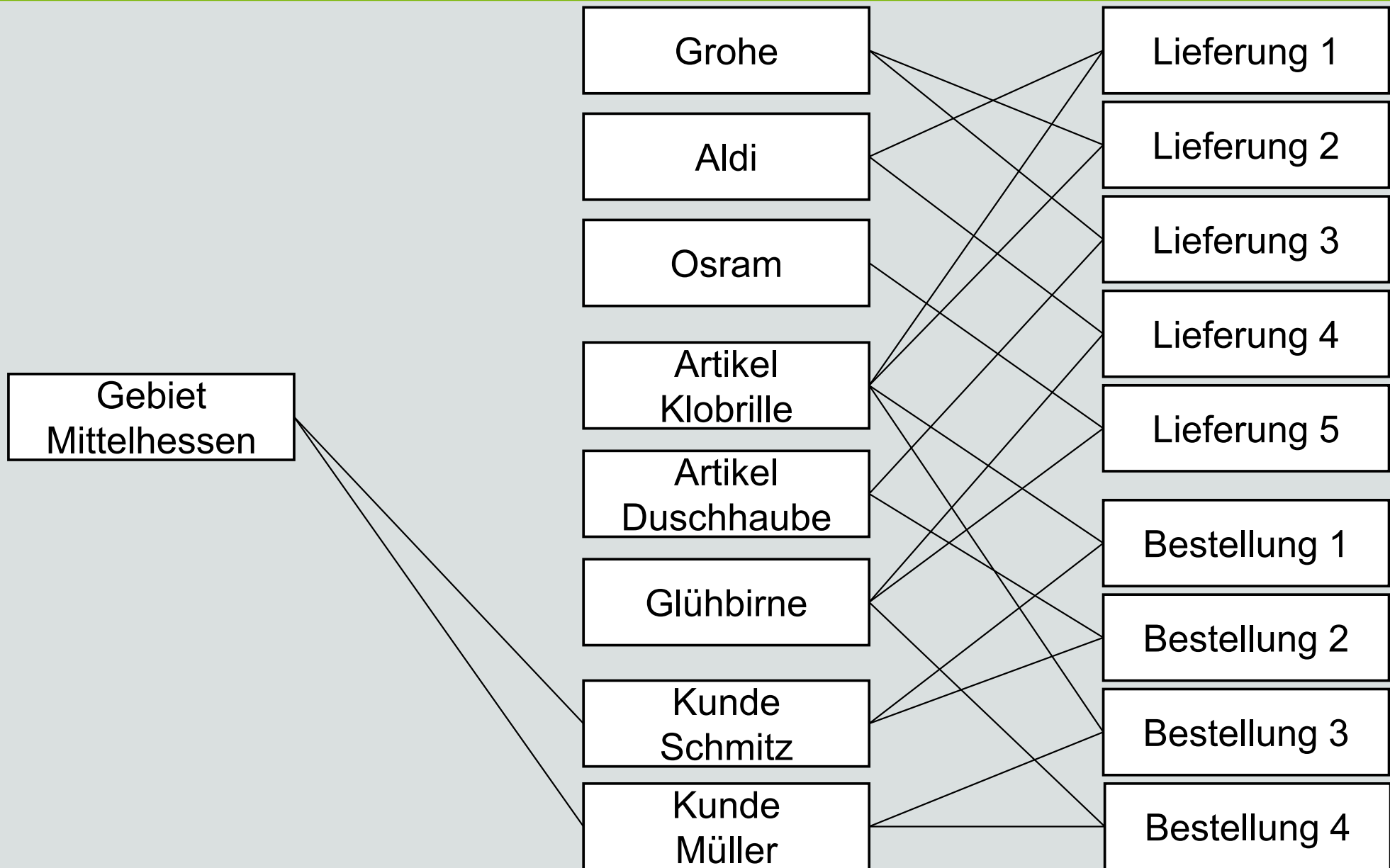
Verbreitung

- Standardisierung Anfang der 70er Jahre
- Keine signifikante Verbreitung in der Praxis
- z.B. UDS von Siemens

- Aufbrechen der strengen Hierarchie des Hierarchischen Modells
- Ein Datensatz kann mehr als einen Vorgänger haben.
- Dadurch Abbildung von m:m-Beziehungen möglich







- Vorteile
 - Flexibler Zugriff auf Daten
 - Realisierung von komplexen Datenbanken möglich
- Nachteile
 - Manipulation und Auswertung sind sehr aufwendig, d.h. dauern lange.

4 Objektorientiertes Modell

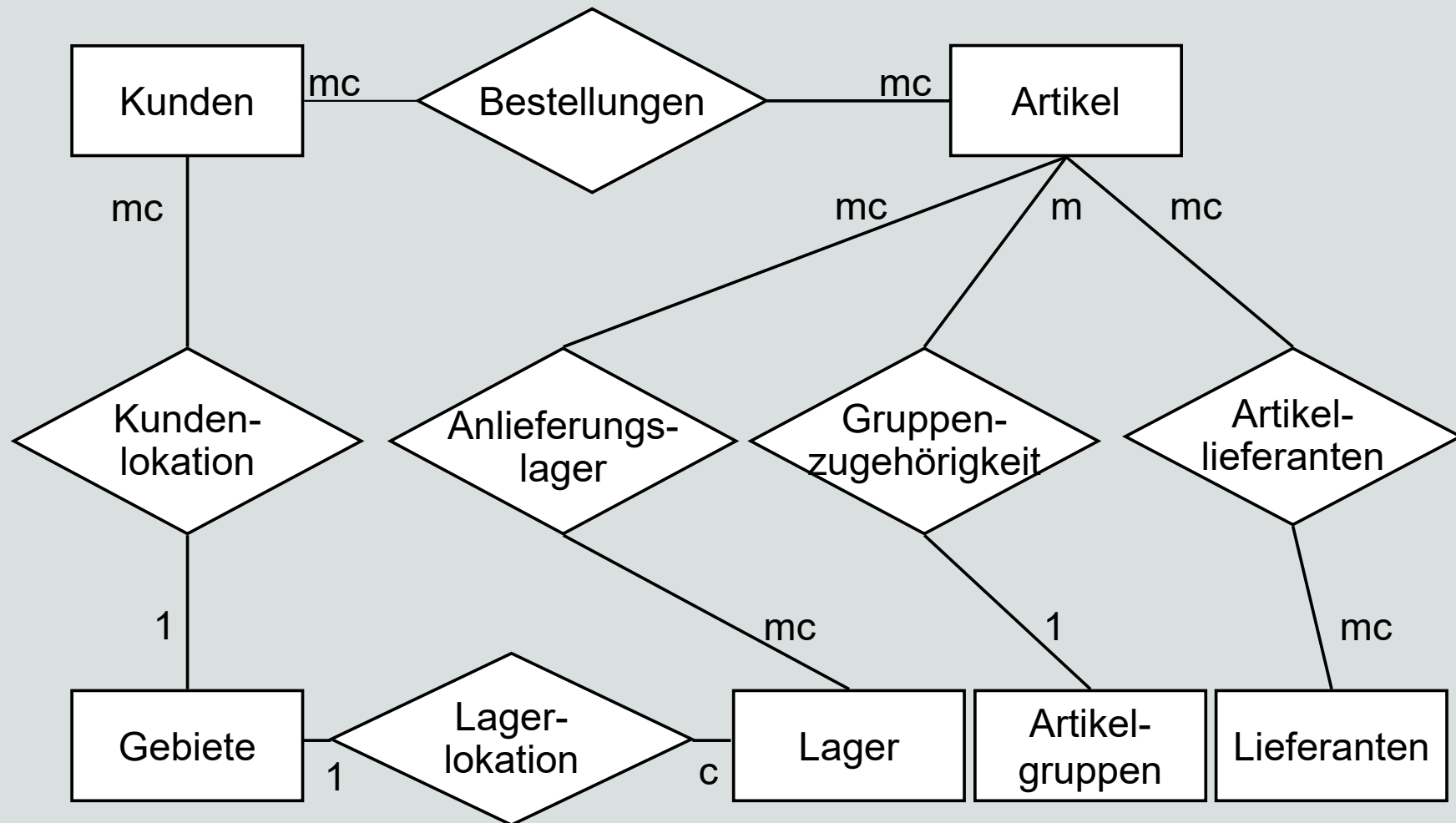
Verbreitung

- z.B. Poet
- Nach 20 Jahren noch nicht durchgesetzt
- Es gibt begründete Zweifel, ob es jemals effiziente Realisierungen hierfür geben kann (Queries können zu komplex werden).
- Eher theoretisch interessant für technologische und wissenschaftliche Betrachtungen

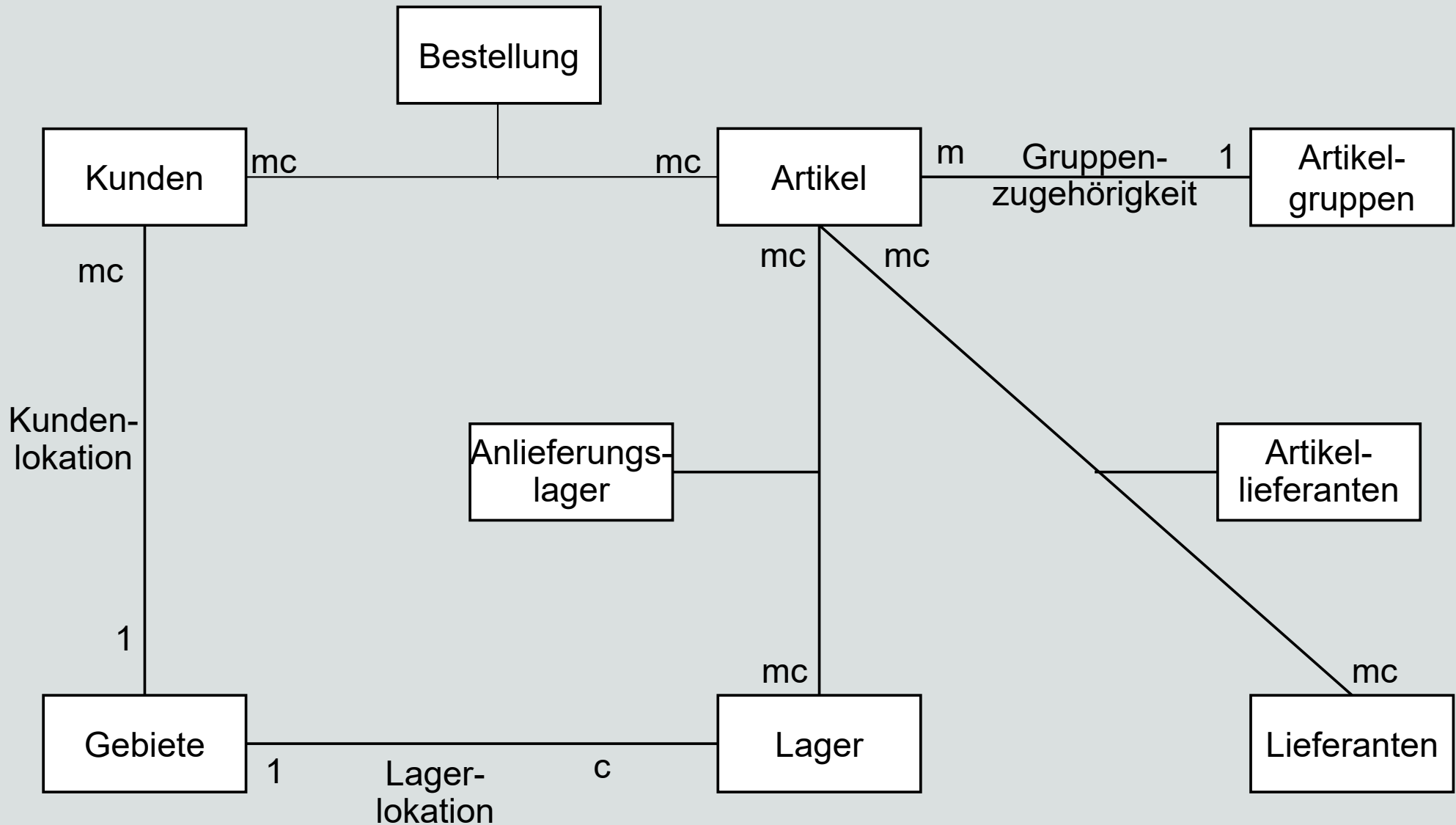
4 Objektorientiertes Modell

- Erweiterung der Entitäten um Methoden und Beziehungen
- Keine Atomizität von Attributen
 - Tupel
 - Listen
 - Mengen
- Vererbung

4 Beispiel ER-Diagramm



4 Objektorientiertes Modell



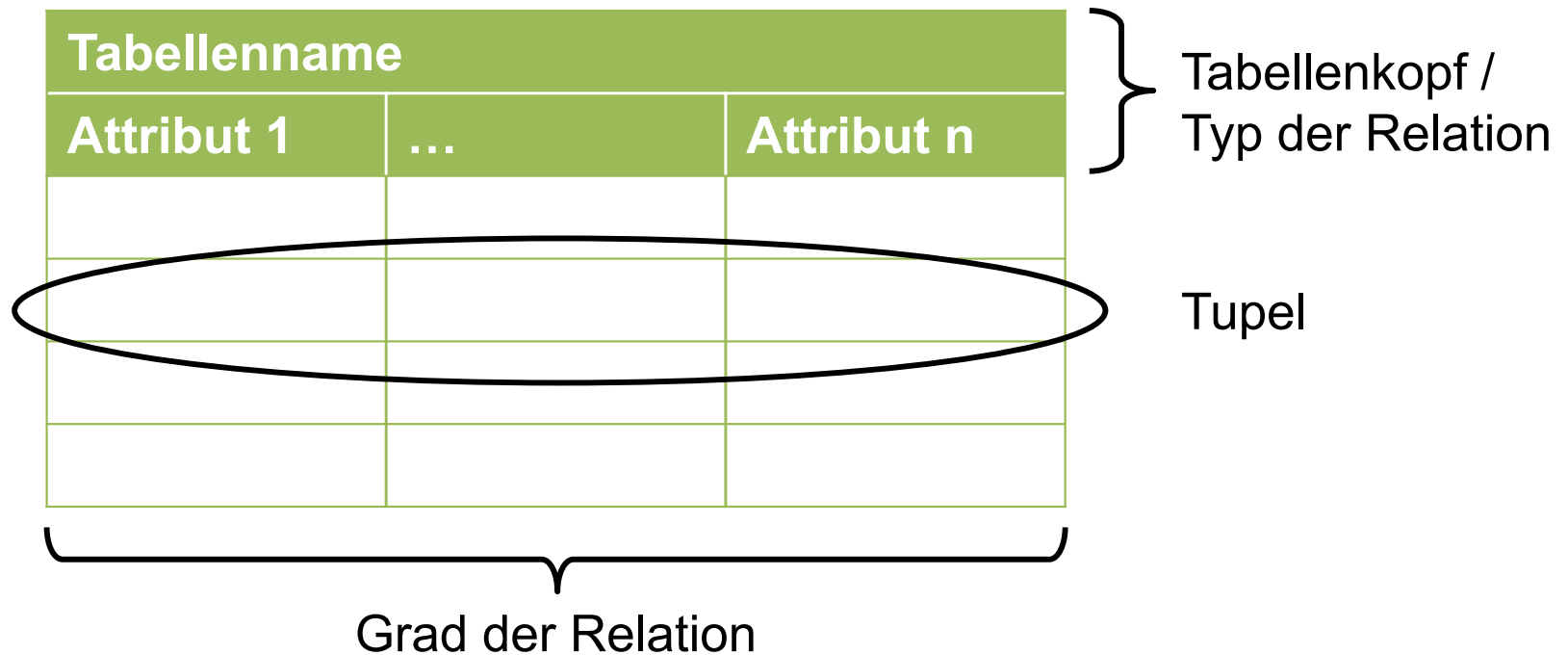


Relation

Eine Relation R ist eine Teilmenge des kartesischen Produkts von Wertebereichen (W_1, W_2, \dots, W_N) .

$$R \subseteq W_1 \times W_2 \times \dots \times W_N$$

Der Grad einer Relation ergibt sich aus der Anzahl der betrachteten Wertebereiche und ist in dem dargestellten allgemeinen Fall N .



4 Eigenschaften des Relationenmodells

- Jede Relation ist eine 2-dimensionale Tabelle und entspricht einem Entitytyp (z.B. Student) oder einer Beziehung.
- Jede Zeile (das Tupel) der Tabelle entspricht einem Entity (z.B. Datensatz: Müller) des durch die Relation erfassten Entitytyps.
- Die Spalten entsprechen den Attributen (z.B. Nachname), durch die die Entitys beschrieben werden.
- Die Anzahl der Attribute heißt Grad der Relation.
- Die Zusammenfassung aller möglicher Attributswerte eines Attributs wird als Domäne (z.B.: Müller, Meier, Pan = Domäne) bezeichnet.
- Der Aufbau der Tabelle ergibt sich aus einer festen Anzahl von Spalten (Attribute) und einer beliebigen Anzahl von Zeilen (Datensätze).

4 Eigenschaften des Relationenmodells

- **Die Spalten können vertauscht werden.**
Da die Attribute durch ihren Namen identifiziert werden, ist ihre Reihenfolge ohne Bedeutung,
- **Die Zeilen können vertauscht werden.**
Da alle Tupel gleichberechtigt sind, ist ihre Reihenfolge ebenfalls ohne Bedeutung
- **Die Werte der Attribute sind atomar.**
Für jedes Attribut eines Tupels existiert genau ein Wert und nicht eine Folge mehrerer Werte.
- **Identische Tupel (Zeilen) kommen nicht vor.**
Je zwei Tupel müssen sich durch mindestens einen Attributwert unterscheiden.
 - ⇒ Es existieren ein oder mehrere (im Extremfall die Kombination aller) Attribute, deren Werte die Tupel eindeutig identifizieren.

- **Einfachheit des Modells**

Das Modell ist einfach und leicht verständlich. Es basiert auf Mengenoperationen, die auf Relationen ausgeführt werden.

- **Geschlossene Systematik**

Das Modell ermöglicht eine systematische Analyse der relevanten Daten und bietet eine sinnvolle Hilfe für die Gruppierung der Merkmale.

- **Redundanzarme Modellierung**

Es ist ein logisches Modell und ein Modell zur redundanzfreien physischen Speicherung der Daten.

- **Darstellbarkeit und Übertragbarkeit**

Das entworfene Datenmodell lässt sich einfach darstellen und ist mit vernünftigen Aufwand auf heute vorhandenen Datenbanksystemen übertragbar/realisierbar.

- (1) **Integration:** Daten müssen in einer einheitlichen Struktur ohne Redundanz abgelegt werden.
- (2) **Operationen:** In einer Datenbank müssen Daten gespeichert, geändert und gesucht werden können.
- (3) **Katalog:** Im Katalog werden Informationen abgelegt, die die Daten in einer Datenbank beschreiben.
- (4) **Benutzersichten:** Für unterschiedliche Anwendungen brauchen wir eine unterschiedliche Sicht auf den Datenbestand.
- (5) **Integritätssicherung:** Die Korrektheit des Datenbankinhalts muss gewährleistet werden, also kann man Regeln angeben, die jeder Datenbankeintrag erfüllen muss.

- (6) **Datenschutz:** Nur berechtigte Benutzer und Programme dürfen Zugriff auf die Datenbank haben.
- (7) **Transaktionen:** Bündelung mehrerer Anweisungen zu einer einzigen Transaktion, welche als eine funktionale Einheit ausgeführt werden – entweder ganz oder gar nicht.
- (8) **Synchronisation:** Parallel ausgeführte Transaktionen müssen den gleichen Datenbankzustand hervorrufen wie irgendeine serielle Ausführung der Transaktionen.
- (9) **Datensicherung:** Das Datenbanksystem muss nach einem Systemfehler in der Lage sein, den letzten konsistenten Datenbankzustand mittels automatischer Datensicherungs- und Wiederherstellungsmechanismen herzustellen.

	c	1	m	mc
c				
1				
m				
mc				

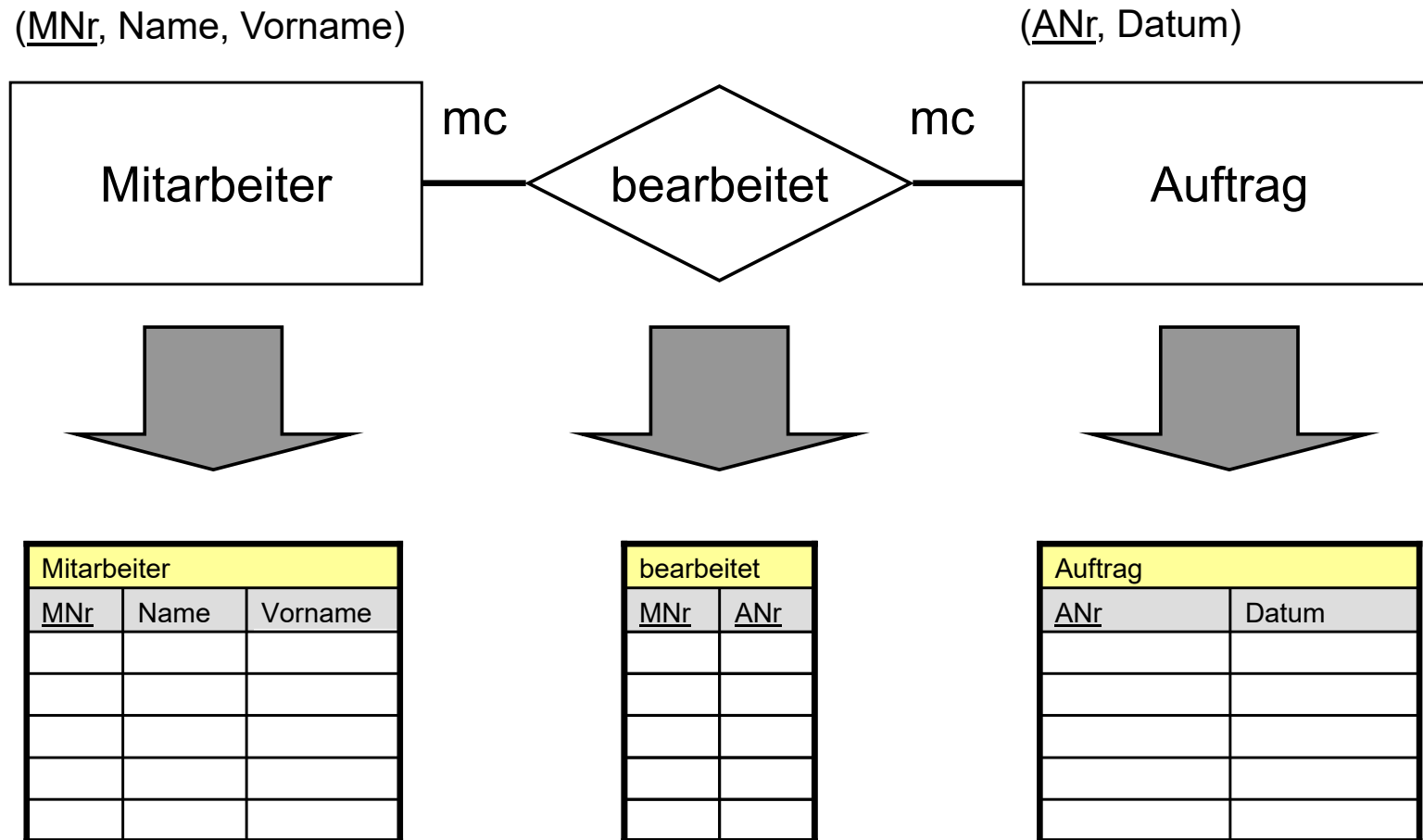
Einfache R-Typen ohne 1-Kante

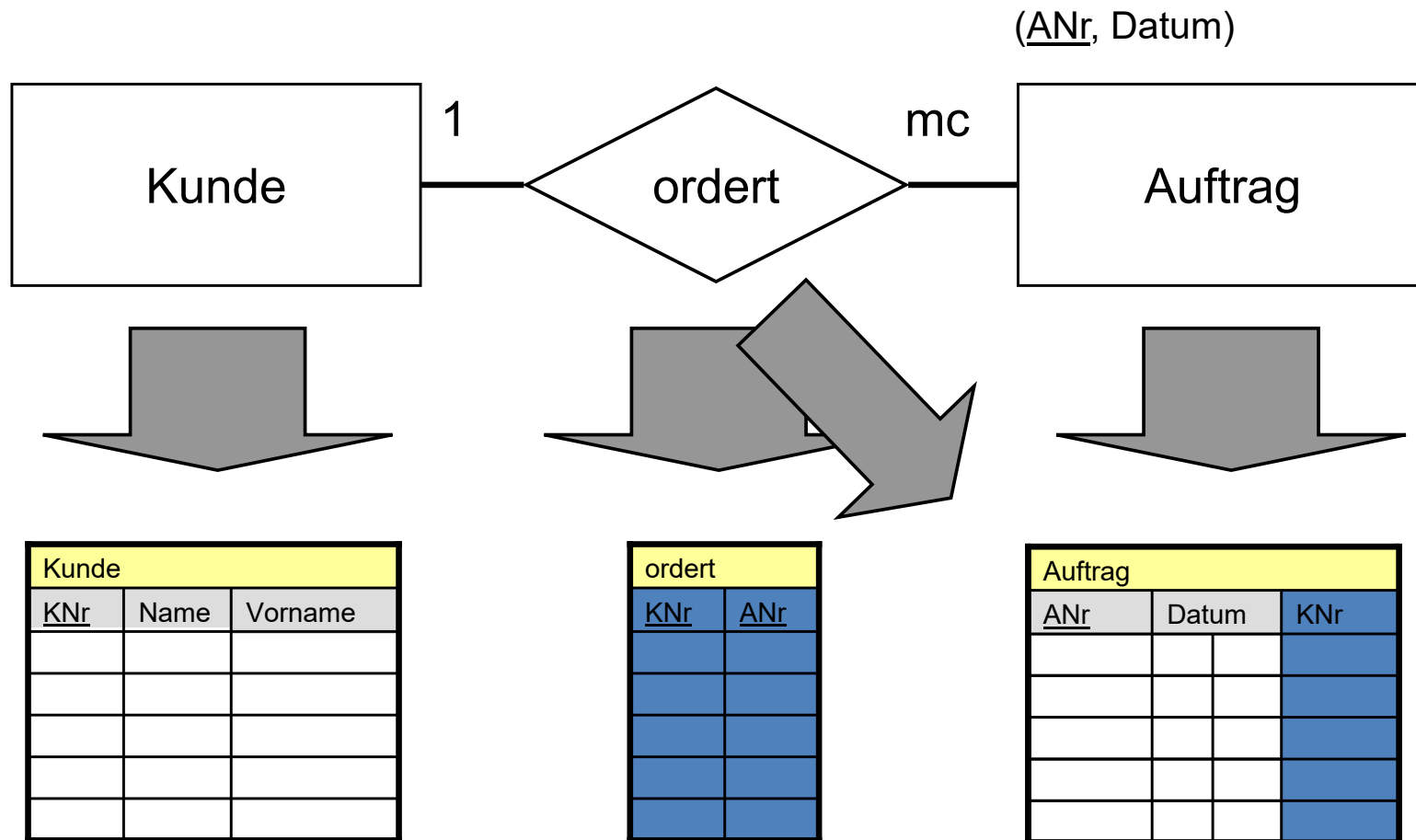
Einfache R-Typen mit 1-Kante

Komplexe R-Typen

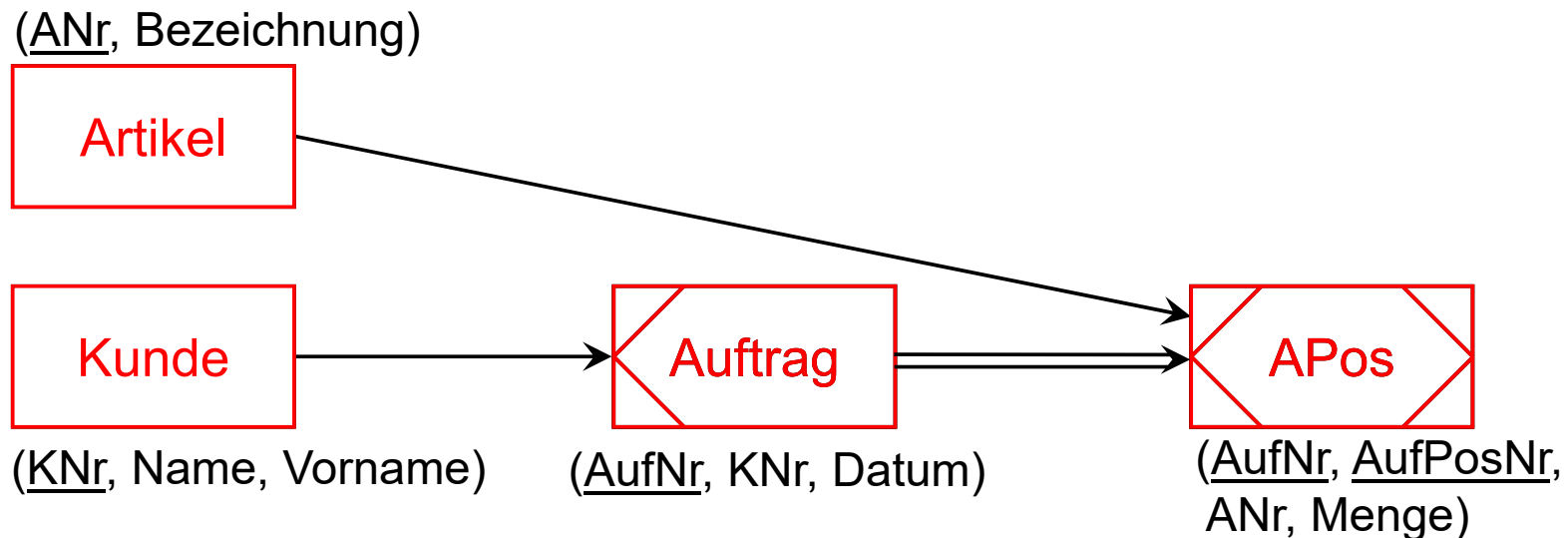
4 Vom ERM zum Relationenmodell

1. Für jeden E-Typ wird ein Relationstyp (Tabelle) gebildet. Bei einer Generalisierung erhalten die Subtypen den Primärschlüssel des Supertyps.
2. Für einen R-Typen wird in Abhängigkeit seiner Komplexität ein Relationstyp gebildet:
 - Für komplexe R-Typen wird ein Relationstyp gebildet.
 - Einfache R-Typen ohne 1-Kante werden als eigenständiger Relationstyp realisiert.
 - Einfache R-Typen mit einer 1-Kante werden in den Relationstyp des entsprechenden E-Typs integriert.





Bilde für jeden Objekttypen einen Relationstyp/eine Tabelle.



RT.Artikel(ANr, Bezeichnung)

RT.Kunde(KNr, Name, Vorname)

RT.Auftrag(AufNr, KNr, Datum)

RT.Auftragsposition(AufNr, AufPosNr, ANr, Menge)

4 Vom Klassenmodell zum Relationenmodell

- Vorgehen bei UML ähnlich zum ERM
- Bilde für jede Klasse einen Relationstypen / eine Tabelle.
- Bilde für attributierte Assoziationen eine Tabelle.
- Bilde für komplexe Assoziationen (ohne Kardinalität 0 oder 1) eine Tabelle.

4 Krähenfußnotation

Abstatt Kardinalitäten an die Kanten zu schreiben, verwendet man Symbole.



1 - 1 - Beziehung



c - m - Beziehung



1 - n - Beziehung



c - cm - Beziehung



1 - c - Beziehung



n - m - Beziehung



1 - cn - Beziehung



n - cn - Beziehung



c - c - Beziehung



cn - cm - Beziehung

4 Aufgabe 1: Anwendungsszenario – Banken

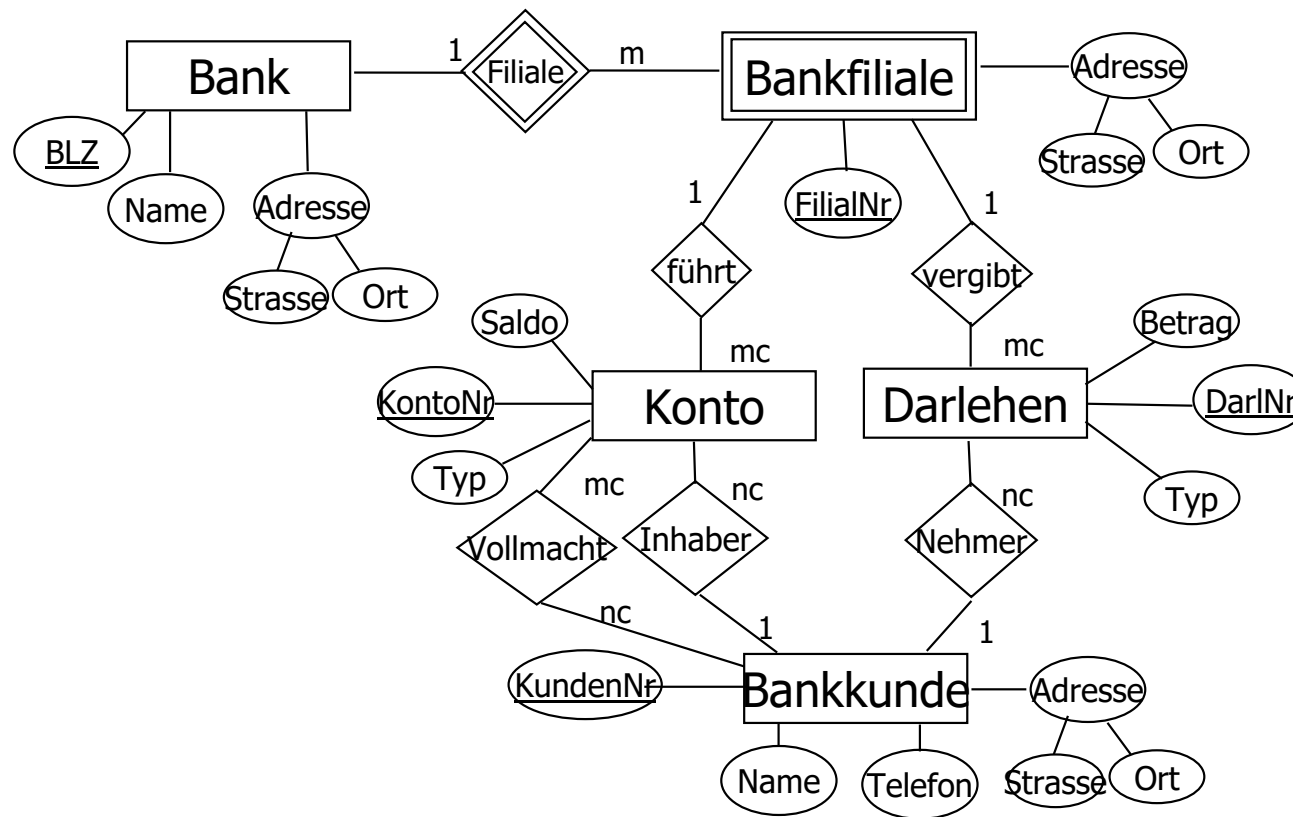
Jede **Bank** verfügt über mehrere **Filialen** in unterschiedlichen Städten. Die Bankfilialen führen **Konten** unterschiedlichen Typs (Giro, Festgeld, Spar) für Bankkunden.

An die **Kunden** werden zudem **Darlehen** unterschiedlichen Typs vergeben, welche durch eine bankweite eindeutige Darlehensnummer identifiziert werden. Es gibt verschiedene Typen von Darlehen.

Für ein Konto können außer dem Inhaber noch mehrere Kunden eingetragen werden, die über das Konto **verfügen** dürfen.

4 Aufgabe 1: Anwendungsszenario – Banken

Erstellen Sie ein Relationenmodell (mit PK und FK) in Krähenfußnotation zu dem gegebenen semantischen Modell.



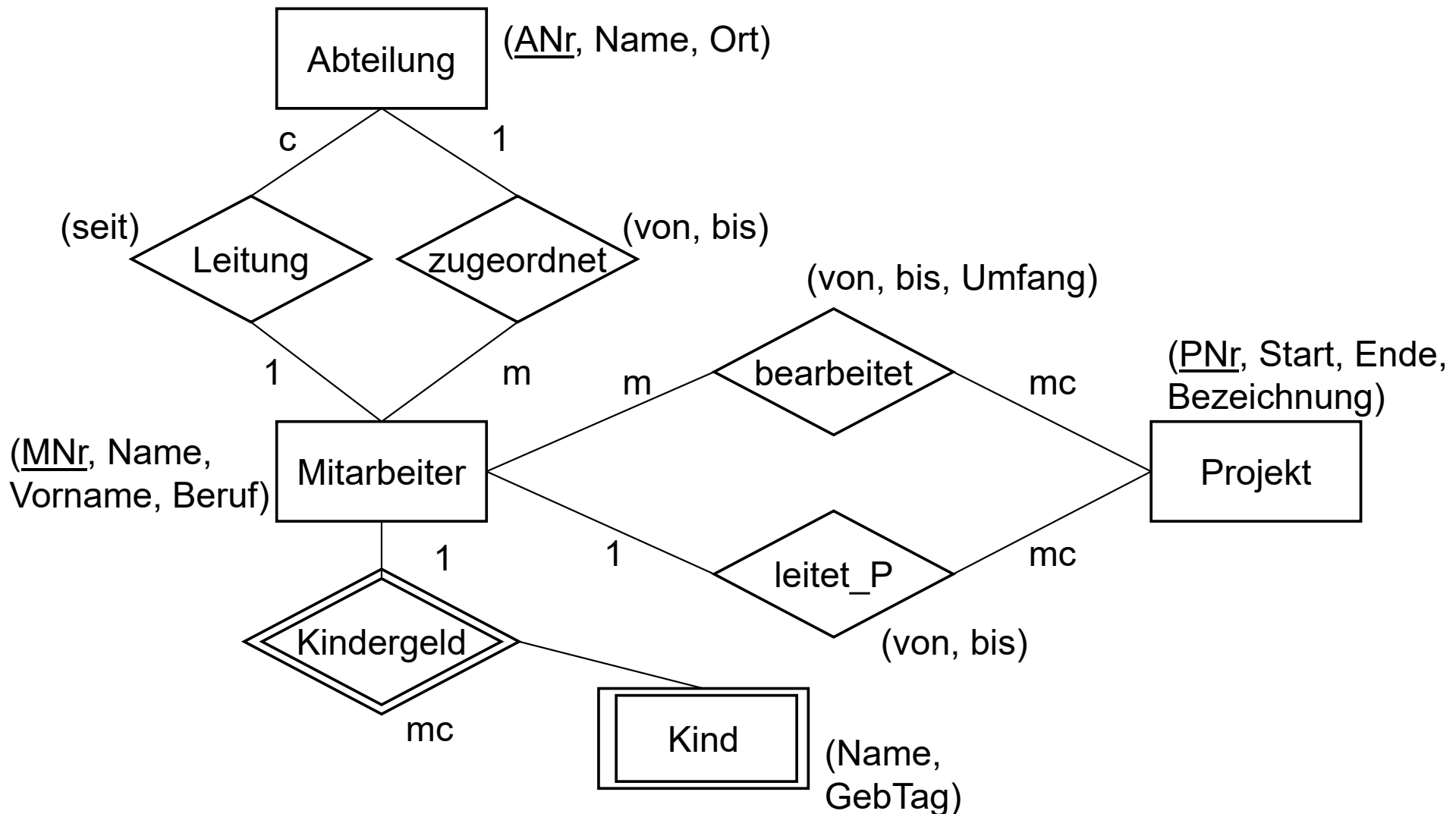
4 Aufgabe 2: Mitarbeiter und Projekte

Mitarbeiter arbeiten in Abteilungen mit Abteilungsleitern einer Firma. Die Abteilungen sind verstreut über das Firmengelände an verschiedenen Orten. Die Firma hat Projekte, die von Mitarbeitern in einem Zeitraum und bearbeitet und geleitet werden. Der Umfang der Mitarbeit eines Projekts ist zu erfassen.

Zudem können für jeden Mitarbeiter seine Kinder erfasst, damit Kindergeldzahlungen möglich sind.

4 Aufgabe 2: Mitarbeiter und Projekte

Erstellen Sie ein Relationenmodell (mit PK+FK) in Krähenfußnotation.



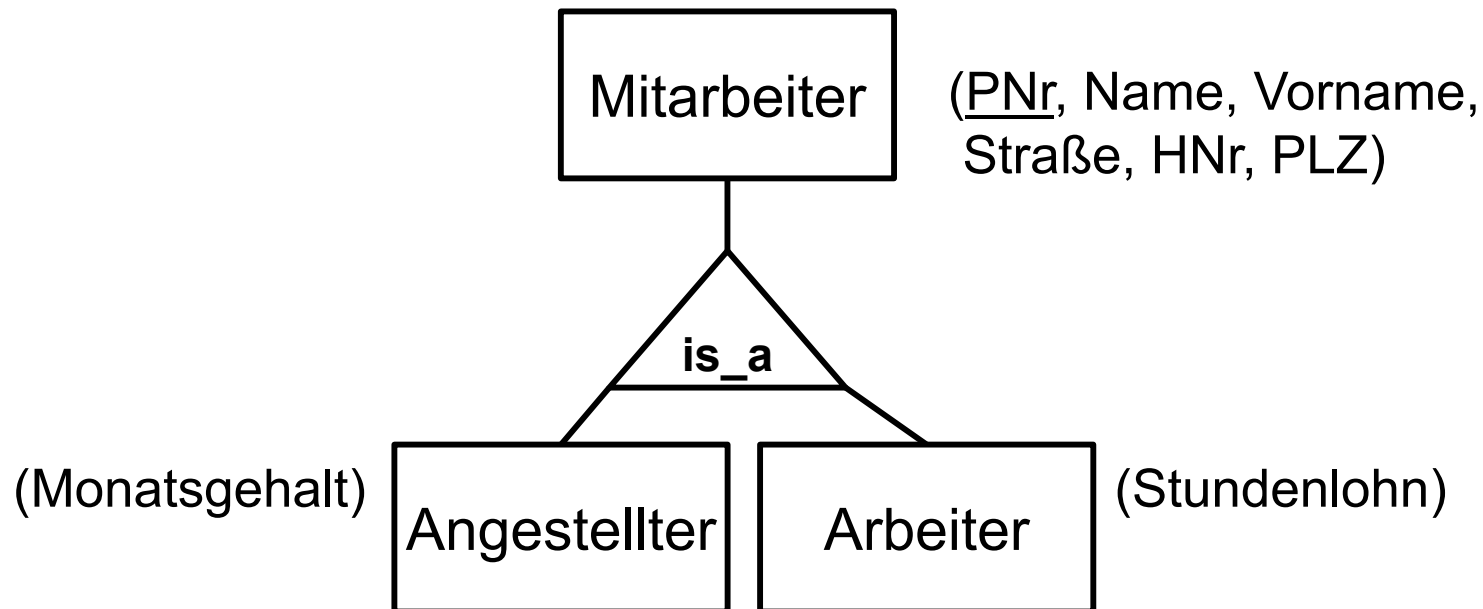
4 Aufgabe 3: Angestellte und Arbeiter

Ein Industrie-Unternehmen beschäftigt Angestellte, die ein festes Monatsgehalt bekommen, und gewerbliche Mitarbeiter (Arbeiter), die einen Lohn erhalten, der sich aus geleisteten Arbeitsstunden und dem Stundensatz zusammensetzen.

Der Personalstammsatz für alle Mitarbeiter enthielte z.B. die Pers-Nr. (als Primärschlüssel) sowie Name, Adresse usw. Nur für Angestellte ist das Attribut Monatsgehalt definiert, während nur für Arbeiter das Attribut Stundenlohn angegeben wird.

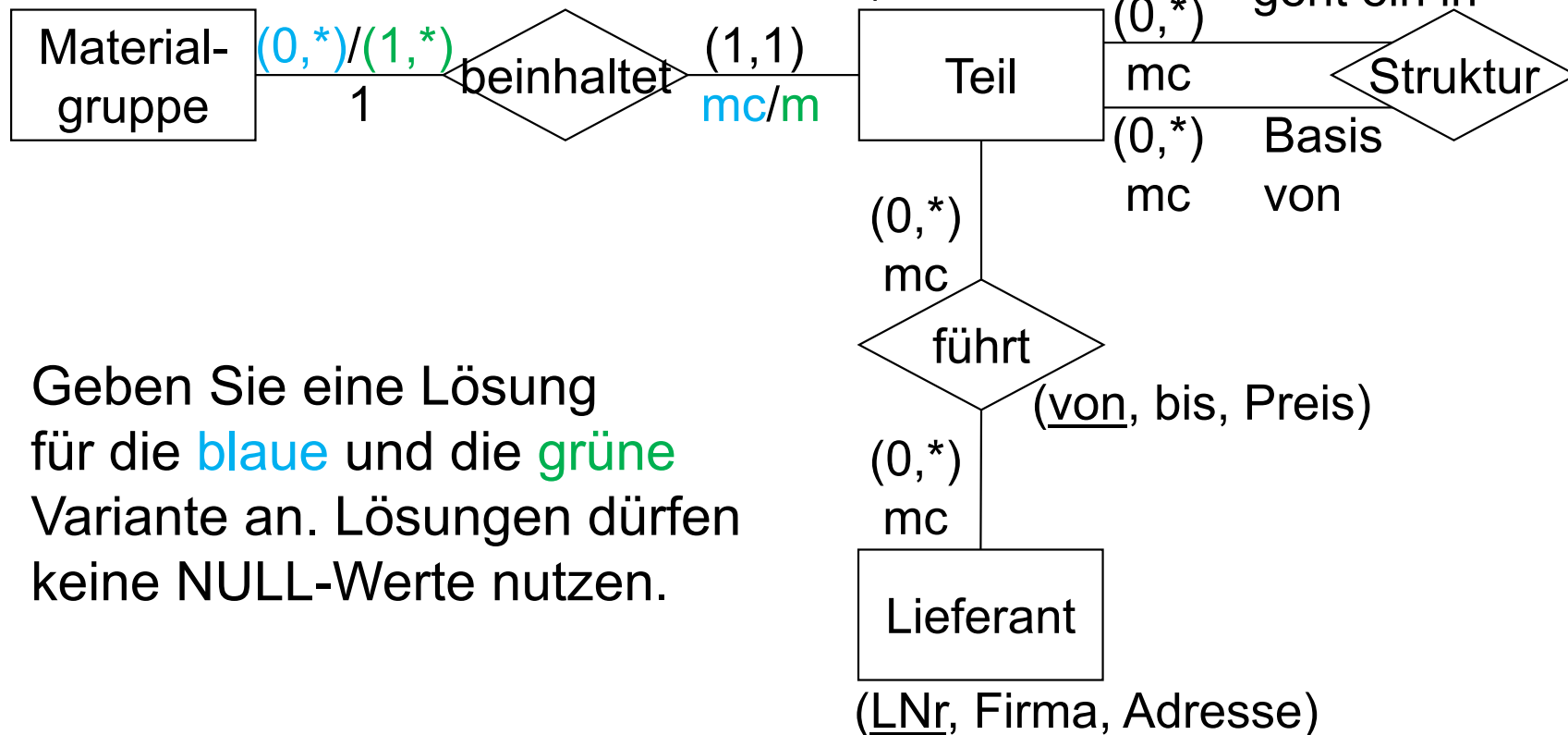
4 Aufgabe 3: Angestellte und Arbeiter

Erstellen Sie ein Relationenmodell in Relationenschreibweise $R(\dots, \dots, \dots)$ zum folgenden Modell.



Erstellen Sie ein Relationenmodell in Krähenfußnotation.

(MGNr, Bezeichnung)



Geben Sie eine Lösung für die **blaue** und die **grüne** Variante an. Lösungen dürfen keine NULL-Werte nutzen.

Erstellen Sie ein Relationenmodell in Krähenfußnotation.

