

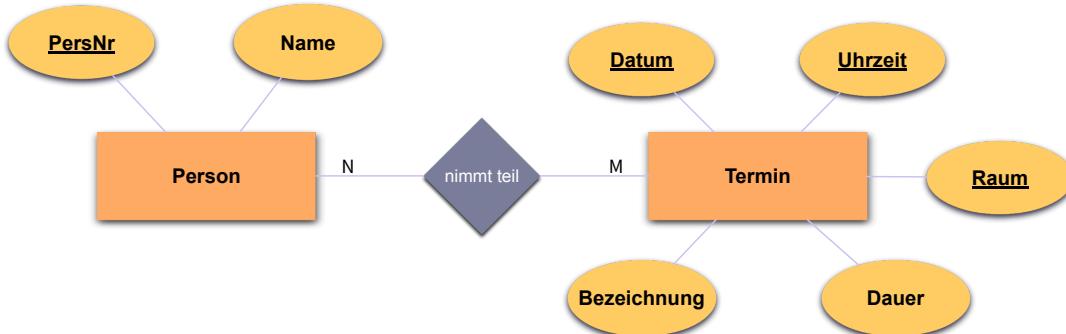
# Datenbanken

## Kapitel 2: ER-Modelle

# Datenmodellierung

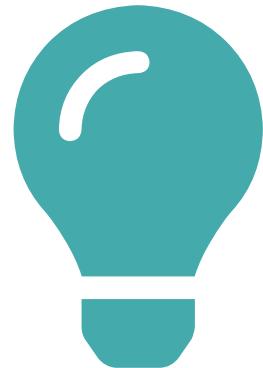
## Konzeptionelles Datenmodell

- Unabhängig vom eingesetzten DBMS
- **ER-Diagramme, UML, ...**



# DB-Entwurf

1. Wir stellen uns etwas aus der realen Welt vor
2. Modellierung einer *Miniwelt* als konzeptionelles Schema
3. Transformation in ein DB-Schema (Relational, XML, ...)

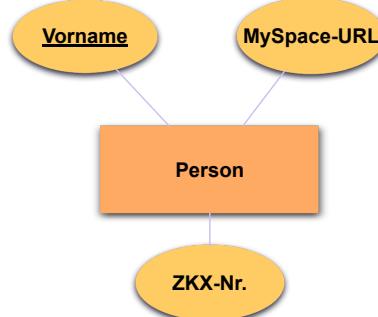


Die Miniwelt ist ein Ausschnitt der realen Welt. Beispiel: Personen haben eine Personalnummer und einen Namen. Dadurch reduzieren wir Personen auf lediglich diese beiden Eigenschaften, die in unserer Miniwelt relevant sind. Das konzeptionelle Schema kann manuell oder semi-automatisch in ein Datenbankschema transformiert werden, z. B. in CREATE TABLE-Befehle oder ein XML-Schema.

3

## Ziele des DB-Entwurfs

- Genaue Abbildung
- Aktuell
- Verständlich
- Simpel
- Redundanzfrei



Viele Fehler und Probleme fallen bereits in der Modellierungsphase auf. So können sie frühzeitig behandelt werden.

Im schlechten Beispiel-Diagramm auf dieser Folie fehlen wichtige Attribute, z. B. der Nachname. Außerdem ist ein Vorname kein sinnvoller Primärschlüssel. Die MySpace-URL ist veraltet und was die KNX-Nr. ist, ist unklar. Wird frühzeitig gemäß Kundenanforderungen ein solches konzeptionelles Modell angefertigt, kann Kundenfeedback erfragt und eingearbeitet werden und somit das Modell schrittweise den Anforderungen des Kunden gerecht werden.

4

# Entity-Relationship-Modell

Das ERM modelliert Dinge (Entities) und Beziehungen (Relationships) zwischen diesen.



Im ER-Diagramm werden Entitätstypen als Rechtecke dargestellt. Ein Entitätstyp ist so etwas wie eine Klasse. Instanzen eines Entitätstypen werden Entities genannt. Das ER-Modell beschreibt also, wie mögliche Entities aussehen können und wie sie in Beziehung zueinander stehen können.

5

## Entitätstypen und Entities

### Entitätstyp

- Rechteck im ER-Diagramm
- z. B. "Person"



### Entität

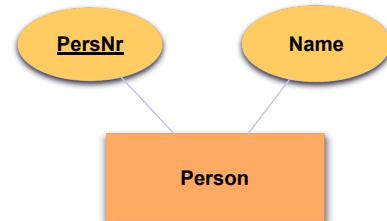
- Unterscheidbare Objekte der Miniwelt
- Ausprägungen von Entitätstypen
- z. B. "Peter mit der Personalnummer 5"
- Entitäten tauchen nicht im ER-Diagramm auf

ER-Diagramme stellen Metadaten dar. Sie beschreiben, was in einer Datenbank gespeichert werden kann. Sie beinhalten nicht die Daten selbst.

6

# Attribute

- Eigenschaften von Entitäten und Entitätstypen
- Haben Werte
- Ellipsen im ER-Diagramm



"Peter hat die Personalnummer 5"

"Personen haben eine Personalnummer und einen Namen"

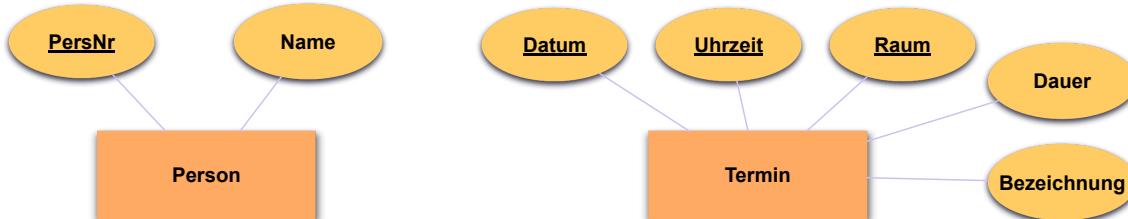
Bei der Erstellung von ER-Diagrammen hilft es immer, sich konkrete Ausprägungen der Entitätstypen vorzustellen, z. B. hier Peter mit der Personalnummer 5, auch wenn Peter nichts im ER-Diagramm zu suchen hat. Wenn man sich jedoch keine Ausprägungen vorstellen kann, hat man meist einen Fehler beim Erstellen des Entitätstyp gemacht. "Personalabteilung" oder "Datenbank" sind zum Beispiel keine sinnvollen instanziierbaren Entitätstypen. Attribute haben einen Datentypen, welcher jedoch im ER-Diagramm in der Regel nicht spezifiziert wird. Auch andere Eigenschaften, z. B. ob ein Attribut Pflicht oder optional ist, tauchen meist im ER-Diagramm nicht auf.

7

8

## Primärschlüssel

- Entitäten müssen wohlunterscheidbar sein
- 1 oder mehr Attribute bilden den Primärschlüssel
- Beschreibt eine Entität eindeutig
- Im ER-Diagramm: Attribut(e) unterstreichen

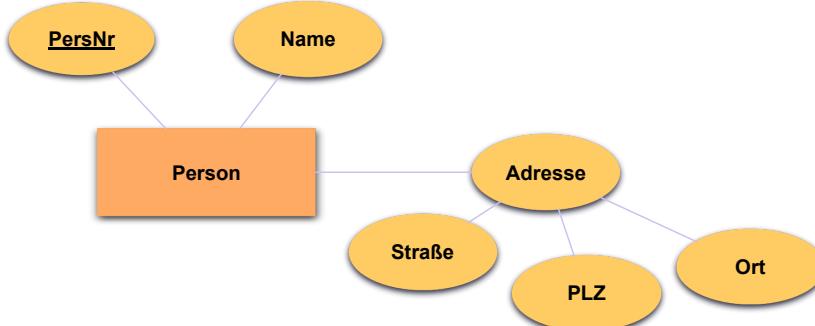


Der Personen-Entitätstyp hat als Primärschlüsselattribut die Personennummer. Dies kann zum Beispiel eine künstlich erzeugte laufende Nummer sein. Termine haben keine solche ID. Hier haben wir einen *zusammengesetzten Primärschlüssel* aus Datum, Uhrzeit und Raum. Dies bedeutet, dass es am gleichen Tag zur gleichen Zeit im gleichen Raum nur einen einzigen Termin geben darf. Manchmal gibt es mehrere Möglichkeiten, was der Primärschlüssel sein kann (sogenannte *Schlüsselkandidaten*). Dann muss man sich für einen entscheiden.

8

# Unterattribute

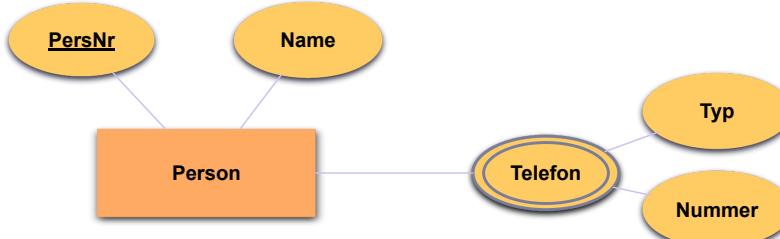
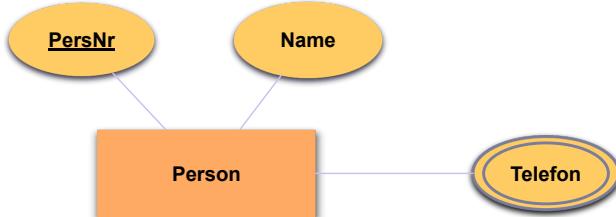
Attribute können aus mehreren Sub-Attributen zusammengesetzt sein.



Personen haben eine Adresse, die aus Straße, Postleitzahl und Ort besteht. Unterattribute können natürlich auch wieder mit Unterattributten versehen werden, um sie noch weiter hierarchisch zu strukturieren.

9

# Mehrwertige Attribute

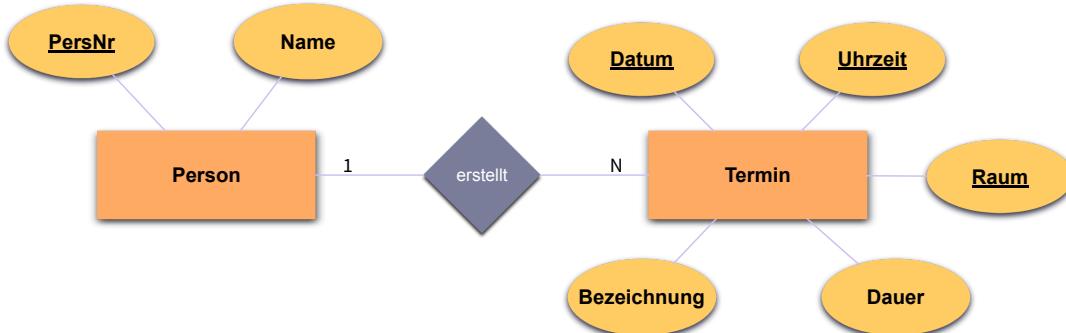


Eine Person hat eine Personennummer, einen Namen, aber mehrere (beliebig viele) Telefonnummern. Im unteren Beispiel hat eine Person mehrere Telefonnummern, die wiederrum aus Typ (z. B. "mobil") und der Telefonnummer bestehen. Es sind beliebige Schachtelungen möglich.

10

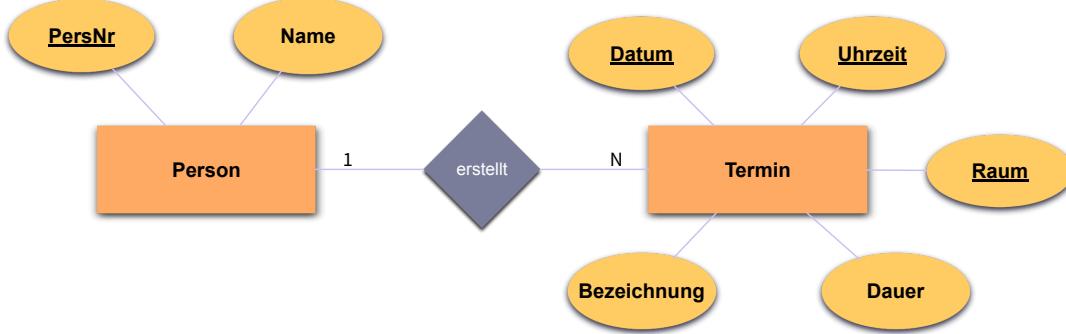
# Beziehungen

- Beteiligt sind mind. 2 Entity-Typen
- Rauten im ER-Diagramm
- Funktionalitäten: 1:1, 1:N / N:1, N:M



11

## 1:N-Beziehungen



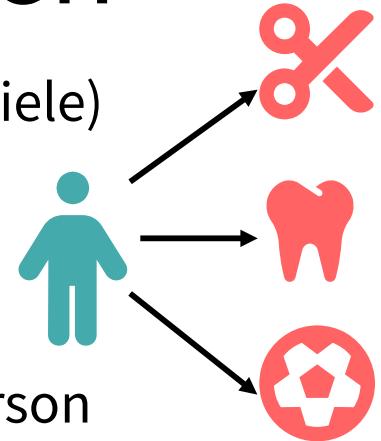
- Eine Person erstellt **N** Termine (beliebig viele)
- Ein Termin gehört immer zu **einer** Person

Man muss den Satz immer mit "Ein(e)... beginnen. Je nachdem, ob man den Satz fortfährt mit "... hat 1" oder "... hat N", kommt dementsprechend eine 1 oder ein N an den gegenüberliegenden Entitätstypen.

12

# 1:N-Beziehungen

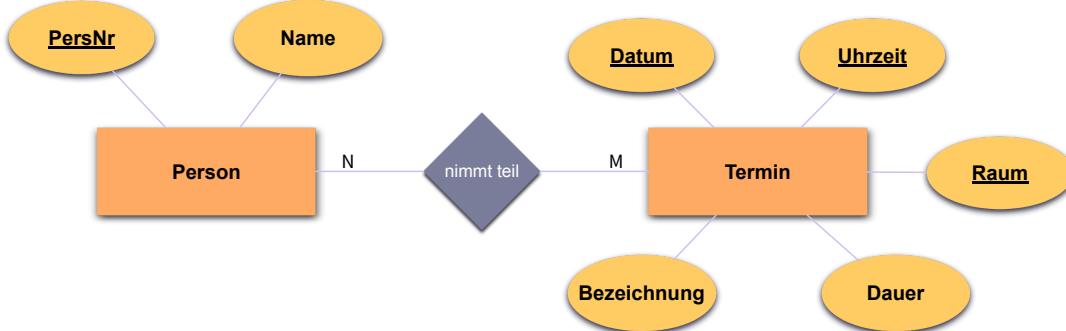
- Eine Person hat **N** Termine (beliebig viele)
  - Peter hat einen Friseurtermin
  - Peter hat einen Zahnarzttermin
  - Peter hat Fußballtraining
- Ein Termin gehört immer zu **einer** Person
  - Der Friseurtermin befindet sich in Peters Kalender



Der Primärschlüssel von Person ist die PersNr. Wenn wir sagen, dass Peter PersNr 5 hat, kann diese Person beliebig viele Termine haben (0, 1, 2, etc.). Primärschlüssel vom Termin ist (Datum, Uhrzeit, Raum). Da wir die Beziehung "Personen haben Termine" als 1:N definiert haben, darf es für einen bestimmten Termin (z. B. Montag 14:45 Uhr, Raum D14/404) nur eine einzige Person (oder keine!) geben, die diesen Termin hat.

13

# N:M-Beziehungen



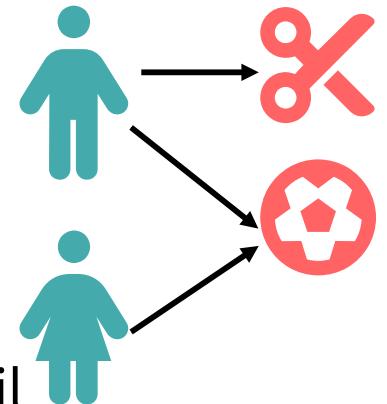
- Eine Person nimmt an beliebig vielen Terminen teil
- An einem Termin können beliebig viele Personen teilnehmen

Auch hier lesen wir wieder "Eine Person..." und "An einem Termin...". Da in beiden Sätzen im zweiten Teil "beliebig viele" steht, schreiben wir an die Linien ein N und ein M. Man schreibt nicht N:N, sondern N:M, da es sich in der Regel um unterschiedliche Zahlen handelt. Nur weil Peter 5 Termine hat, heißt das nicht, dass an jedem seiner Termine 5 Personen teilnehmen.

14

# N:M-Beziehungen

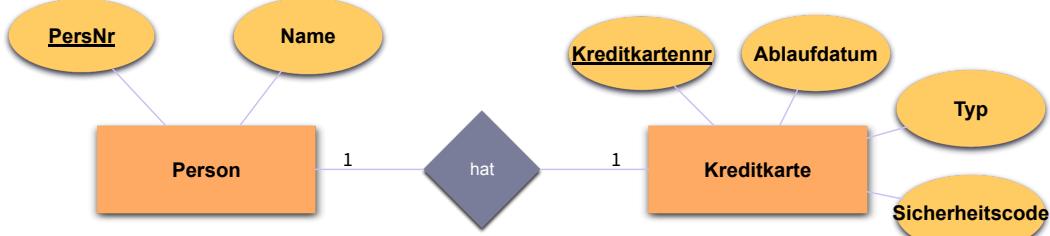
- Eine Person nimmt an beliebig vielen Terminen teil
  - Peter hat einen Friseurtermin
  - Peter hat Fußballtraining
- An einem Termin können beliebig viele Personen teilnehmen
  - Am Friseurtermin nimmt Peter teil
  - Am Fußballtraining nimmt Peter teil
  - Am Fußballtraining nimmt Katja teil



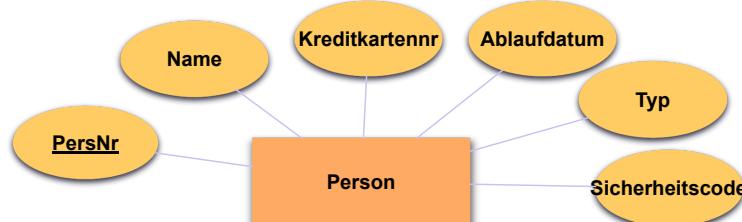
15

11

## 1:1-Beziehung



- Eine Person hat **eine** Kreditkarte
- Eine Kreditkarte gehört immer nur zu **einer** Person

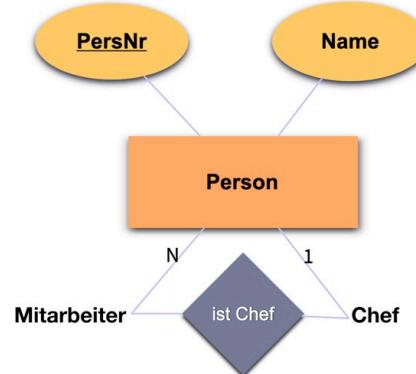


Statt einer 1:1-Beziehung kann man sich auch oft die Beziehung sparen und alles in einem Entitätstypen modellieren.

16

# Rekursive Beziehung

- "Beziehung mit sich selbst"
- Entitätstyp nimmt mehrfach an einer Beziehung teil
- Rollennamen zur Unterscheidung notwendig

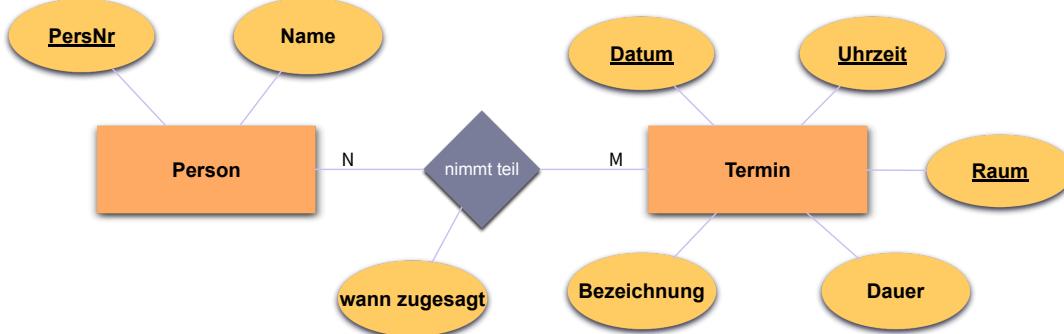


Hier wurden die Rollennamen "Mitarbeiter" und "Chef" an die Linien geschrieben. Eine Person hat nur einen Chef, aber sie kann mehrere Mitarbeiter unter sich haben, d.h. sie kann Chef von mehreren Personen sein.

17

11

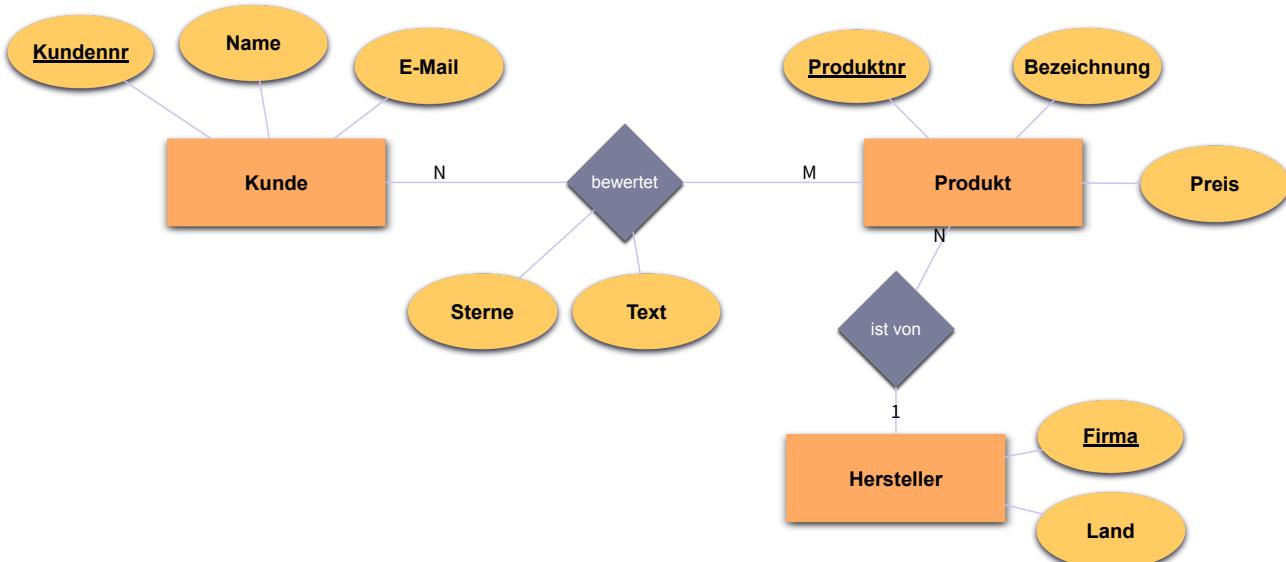
## Beziehungsattribute



- Peter nimmt am Fußballtraining teil und hat am Montag zugesagt
- Katja nimmt am Fußballtraining teil und hat am Dienstag zugesagt

18

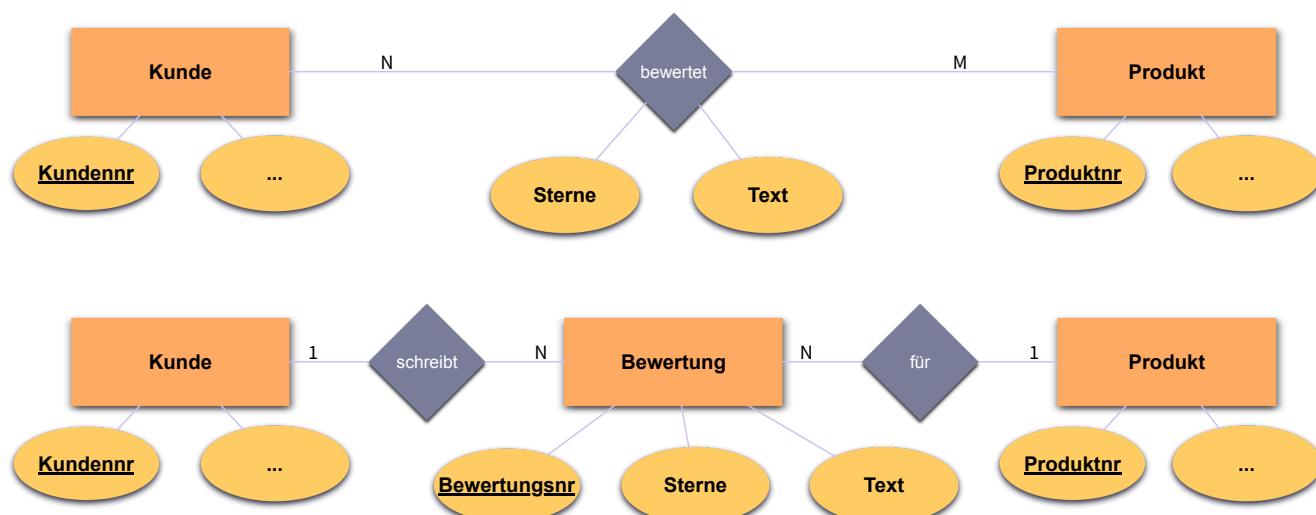
# Webshop-Beispiel



Kunden bewerten Produkte mit Sternen und einem Bewertungstext. Die Beziehungsattribute Sterne und Text werden direkt an die Beziehung geschrieben. Nicht Kunden haben Sterne, auch Produkte haben keine Sterne, sondern die Bewertung: Peter bewertet die Spülmaschinentabs mit einem Stern und schreibt: 'Mein Geschirr wird nicht sauber.'

19

## N:M-Bez. vs. eigener Entitätstyp

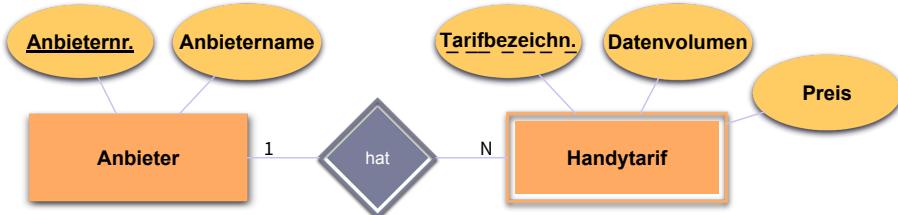


Die beiden hier dargestellten Lösungen ist fast äquivalent. Da Entitätstypen stets einen Primärschlüssel benötigen, muss es im unteren Beispiel, in dem Bewertungen als Entitätstyp modelliert wird, eine Bewertungsnummer oder ähnliches geben. Mit schwachen Entitätstypen (siehe nächste Folie) lässt sich dies jedoch auch ohne Primärschlüssel lösen (siehe übernächste Folie).

20

# Schwache Entitätstypen

- Kein eigener Primärschlüssel, stattdessen wird der Primärschlüssel von einer anderen Entität vererbt und erweitert
- ⇒ Existenzabhängig von einem anderen Entitätstypen
- Entitätstyp und Beziehung doppelt umrahmen

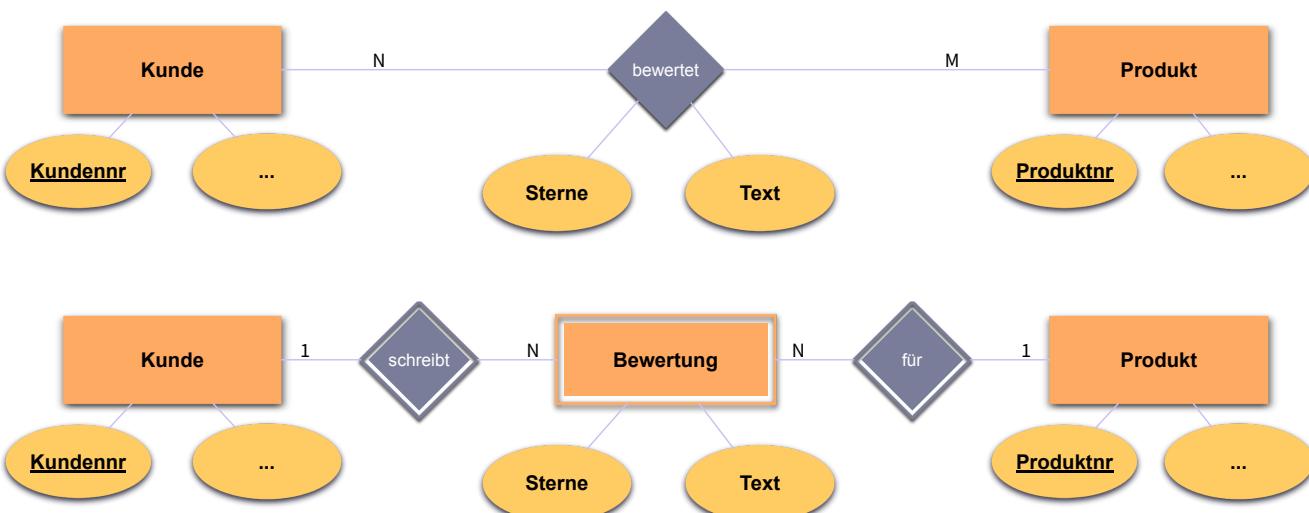


Die Kombination aus Anbieternummer und Tarifbezeichnung identifiziert einen Handytarif eindeutig, z. B. Vugafon "Data Extreme".

Handytarif ist ein schwacher Entitätstyp (*Weak Entity Type*). Die Tarife sind existenzabhängig vom jeweiligen Anbieter des Tarifs. Dies ist daran zu erkennen, dass sowohl Entitätstyp als auch die Beziehung zum Anbieter doppelt umrahmt sind. Zu einem Tarif muss es einen Anbieter geben. Die Tarifbezeichnung ist ein *erweiternder Primärschlüssel*. Sie ist eindeutig für einen bestimmten Anbieter. Es kann einen "Data Extreme"-Tarif bei mehreren Anbietern geben, aber bei jedem nur einmal.

21

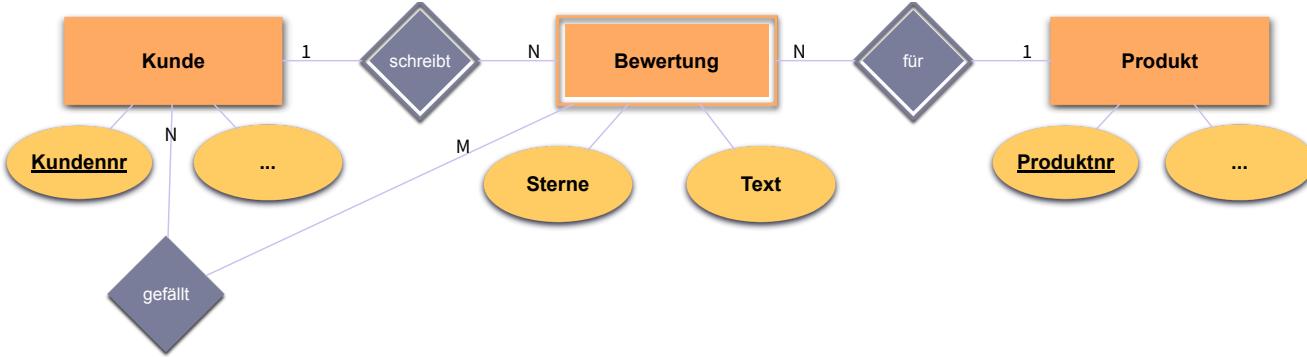
## N:M-Bez. vs. schwacher Entitätstyp



Der unten dargestellte schwache Entitätstyp "Bewertungen" hat keine Primärschlüssel-erweiternden Attribute, aber sie ist existenzabhängig vom bewertenden Kunden und dem bewerteten Produkt. Eine Bewertung wird also eindeutig identifiziert durch die entsprechende Kundennummer- und Produktnummer-Kombination. Die beiden auf dieser Folie dargestellten ER-Diagramme sind in ihrer aktuellen Version absolut äquivalent und modellieren genau das gleiche. Die untere Version bietet jedoch mehr Erweiterungsmöglichkeiten (siehe nächste Folie).

22

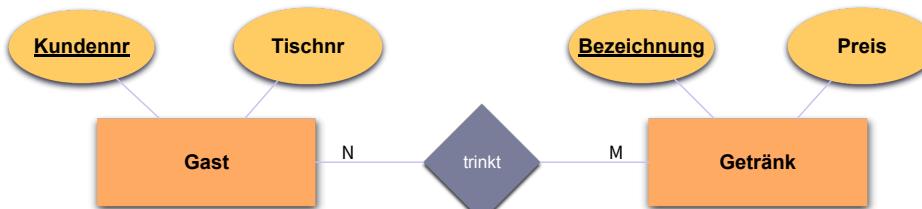
# N:M-Bez. vs. schwacher Entitätstyp



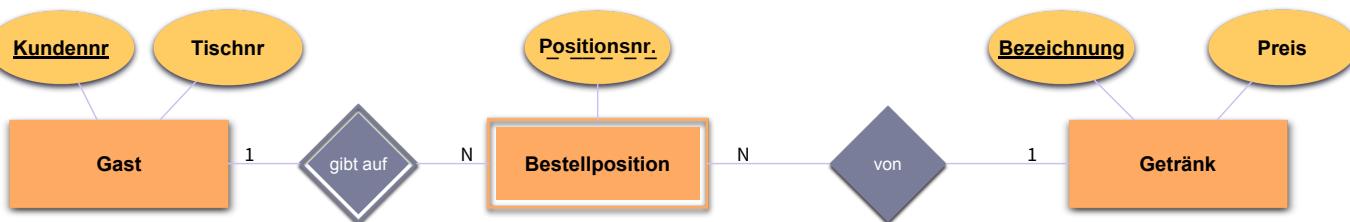
Dadurch, dass die Bewertungen nun ein Entitätstyp sind, kann dieser nun an Beziehungen teilnehmen und Subtypen (siehe später in diesem Kapitel) haben. Kunden können bei einer Bewertung auf "gefällt mir" klicken. Die "gefallen"-Beziehung ist natürlich nicht doppelt umrahmt. Nicht die Kunden, denen eine Bewertung gefallen identifizieren diese, sondern diejenigen, die sie schreiben (und das bewertete Produkt). Löscht ein Kunde seinen Account, darf es keine Bewertungen mehr geben, von denen der Kunde Autor ist, aber es darf weiterhin Bewertungen geben, für die er oder sie ein Like abgegeben hat. In der Modellierung mit der "bewertet"-Beziehung (vorherige Folie oben) ist es nicht möglich, Bewertungslikes zu modellieren.

23

## N:M- vs. zwei 1:N-Beziehungen



Besser:

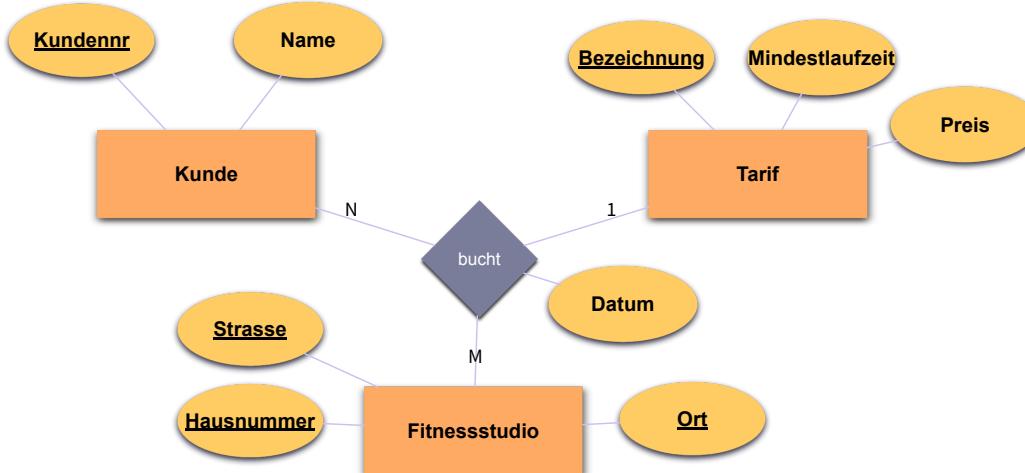


Gäste bestellen in einer Bar Getränke. Ein Gast kann mehrere Getränke trinken (Kunde Nr. 5 trinkt Rotwein und Mineralwasser) und ein Getränk kann von mehreren Gästen bestellt werden (Kunden 7 und 8 trinken Weißwein). Das Problem hier ist jedoch, dass niemand ein bestimmtes Getränk mehrfach trinken kann. Wir lösen das Problem, indem ein Kunde mehrere Bestellpositionen aufgibt. Jede Position erhält eine für den Kunden eindeutige Nummer und enthält nur ein Getränk. Bestellt ein Kunde 2x Rotwein und 1x Mineralwasser, werden dazu drei Bestellpositionen erfasst.

24

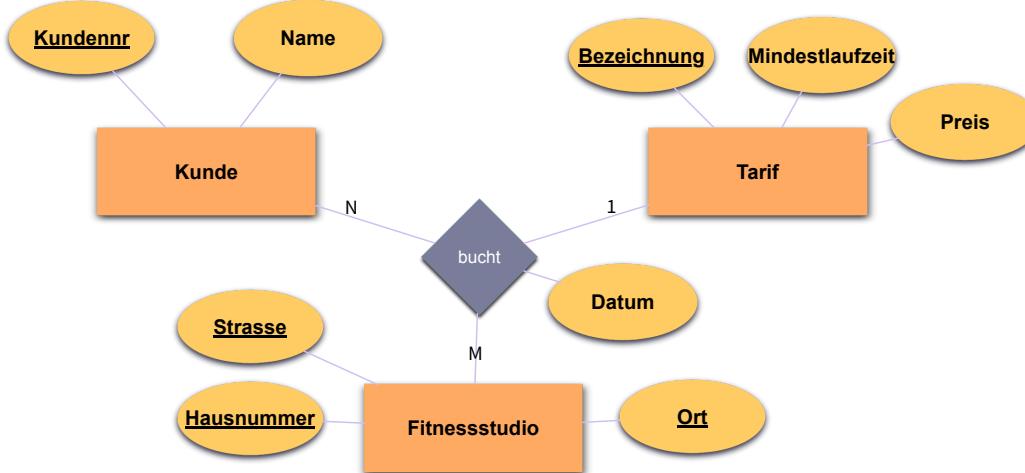
# Ternäre Beziehungen

- Bisher: Beziehungs-Grad = 2 (binäre Beziehung)
- Grad: Anzahl der teilnehmenden Entitätstypen
- Ternäre Beziehung: Grad = 3



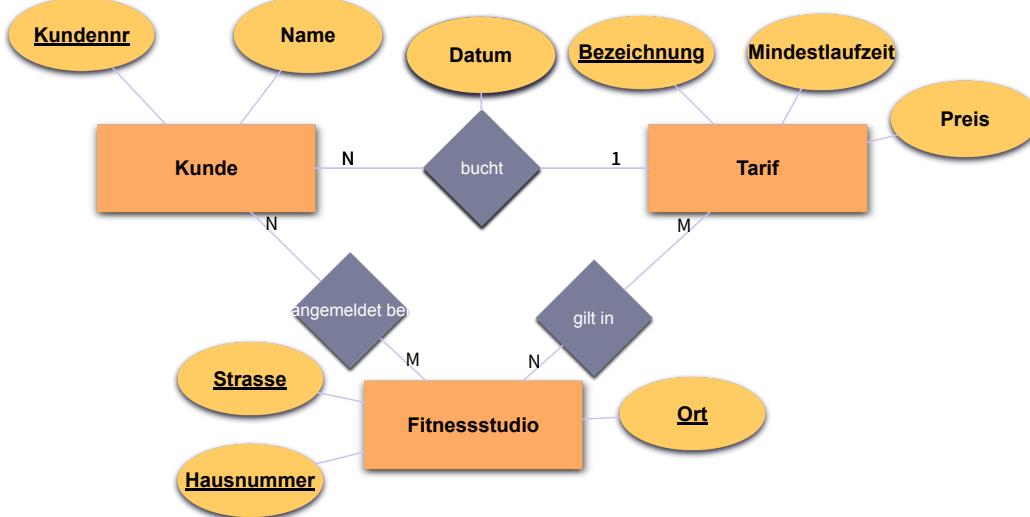
25

# Ternäre Beziehungen



- Ein Kunde kann in einem Fitnessstudio nur **einen** Tarif buchen
- Ein Kunde kann einen Tarif in **vielen** Studios buchen
- Ein Tarif kann in einem Studio von **vielen** Kunden gebucht werden

# Ternäre vs. drei binäre Beziehungen



- Ein Kunde bucht stets nur **einen** Studio-übergreifenden Tarif.
- Ein Kunde kann bei **vielen** Studios angemeldet sein
- Ein Tarif ist in **vielen** Fitnessstudios gültig

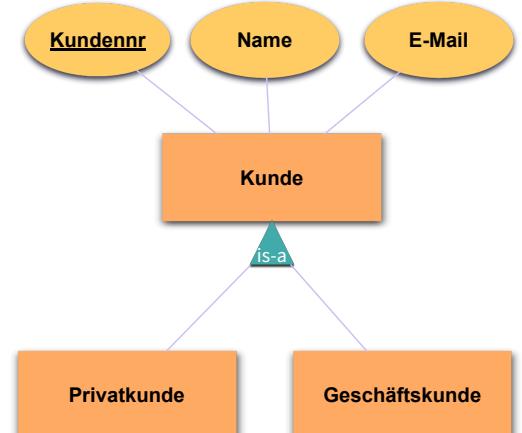
Diese Lösung ist nicht weniger oder mehr richtig als die Lösung mit der ternären Beziehung auf der vorherigen Folie, aber sie modelliert eine komplett andere Miniwelt.

27

13

## Generalisierung

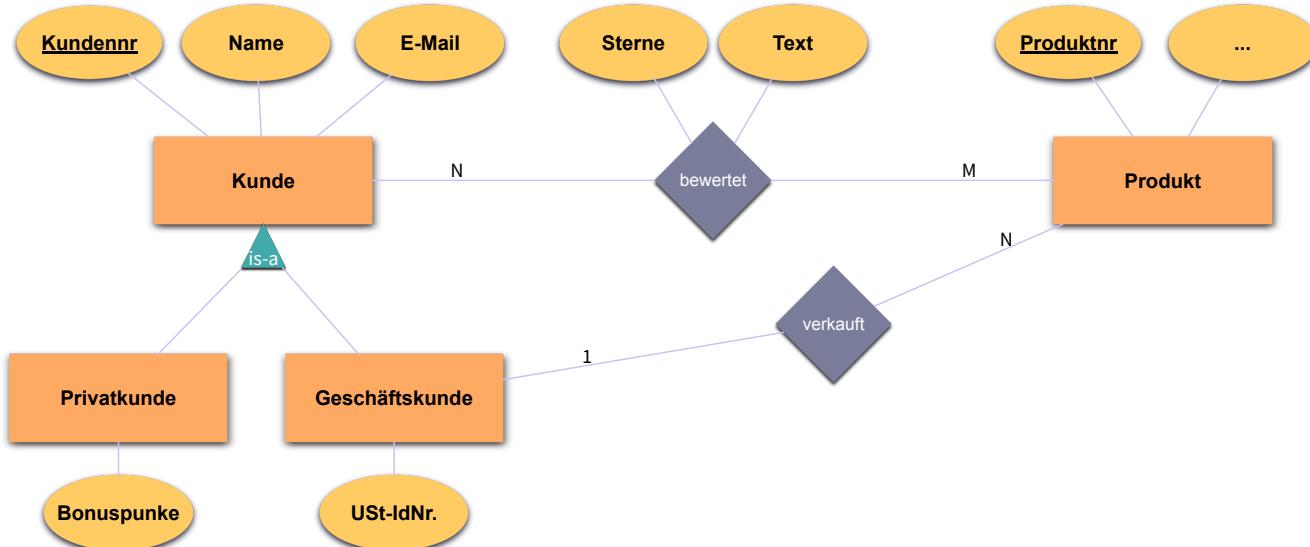
- Spezialisierung, Vererbung
- Attribute, Primärschlüssel und Beziehungen des Super-Typen werden an die Sub-Typen vererbt.
- "is-a"-Pfeil im ER-Diagramm



Der Entitätstyp "Kunden" hat zwei Sub-Typen "Privatkunden" und "Geschäftskunden". In unserer Datenbank können wir also für jeden Kunden speichern, ob er einfach nur Kunde ist, oder aber auch Geschäftskunde und Privatkunde. Es gibt Notationen, in der Überlappungen (Mehrklassenzugehörigkeit) und abstrakte Typen (nicht instanzierbar) modelliert werden können. In unserer Notation ist eine Entität stets nur eine direkte Instanz von einem Entitätstypen, und damit implizit auch Instanz von dessen Super-Typen. Jeder Privatkunde ist also auch Kunde. Subtypen können auch wieder Subtypen haben, sodass sich komplexe Typ hierarchien modellieren lassen.

28

# Generalisierung



Das ER-Diagramm wurde erweitert um Subtypen-spezifische Attribute und Beziehungen. Nur Geschäftskunden können Produkte verkaufen, aber alle Kunden können sie bewerten. Privatkunden sammeln Bonuspunkte und für Geschäftskunden wird die Umsatzsteuernummer vermerkt.

29

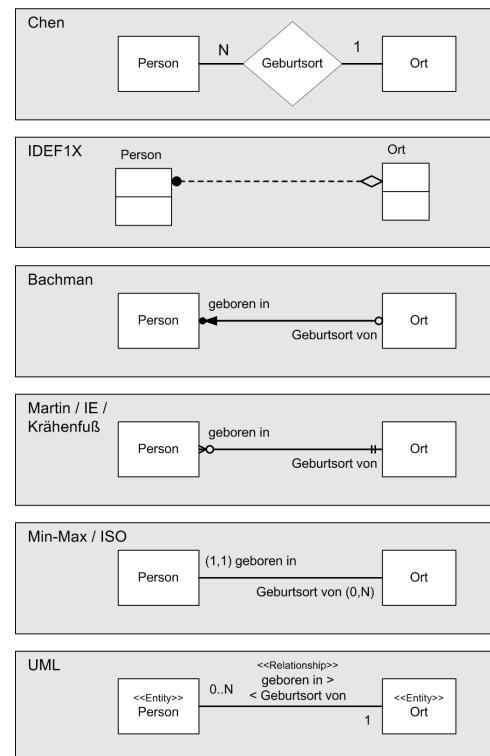
# Notationsformen

<https://de.wikipedia.org/wiki/Entity-Relationship-Modell>

Wir verwenden die *Chen-Notation*

Nicht anpassbar in Chen:

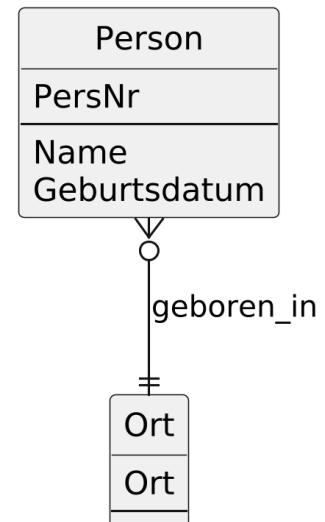
- Es gibt Orte ohne Personen
- Es gibt Personen ohne Ort



# Krähenfuß-Notation

Die Krähenfuß-Notation kommt in vielen Datenbank-Modellierungs-Tools zum Einsatz.

- $\rightarrow+$  Genau eins
- $0+$  Höchstens eins
- $\leftarrow*$  Mindestens eins
- $0\leqslant$  Mindestens keins (beliebig viele)



In der Krähenfuß-Notation kann erzwungen werden, dass jede Entität eines Entitätstypen mindestens einmal an einer Beziehung teilnehmen muss. Das Diagramm auf dieser Folie sagt, dass jede Person in einem Ort geboren sein muss. Würde man auf der Minimum-Seite statt dem | eine 0 schreiben, kann es Personen ohne Geburtsort geben. Dies entspräche der Chen-Notation N:1 auf der vorherigen Folie oben. Das  $0\leqslant$  im Diagramm bedeutet, dass es Orte mit vielen Personen, aber auch Orte ohne Personen geben kann. In der Chen-Notation gibt es nur 1 und N, was hier  $0+$  und  $0\leqslant$  entspricht.

31

## Kapitelzusammenfassung

- DB-Entwurf mit ER-Diagrammen
- Entitätstypen, Attribute, Beziehungen
- Primärschlüssel
- Schwache Entitätstypen
- Ternäre Beziehungen
- Generalisierung
- Krähenfuß-Notation

32

# Kochrezept: ER-Diagramm erstellen

1. Text lesen; beim Lesen markieren;  
wichtige Zusatzinfos unterstreichen

- Rosa: Entitätstypen
- Gelb: Attribute
- Grün: Beziehungen



2. Das Markierte in ER-Diagramm überführen
3. Nochmals Text lesen und ER-Diagramm überprüfen
4. Letzte Überprüfung mittels Checkliste (nächste Folie)

33

## Checkliste: ER-Diagramm erstellen

- Sind alle Kardinalitäten an Beziehungen? ✓ 
- Kardinalitäten richtig herum? ✓ 
- Gibt es überflüssige Entitätstypen? ○ 
- Hat jeder Entitätstyp einen Primärschlüssel?
- Müssen manche Entitätstypen schwach sein?
- Ist die entsprechende Beziehung doppelt umrahmt?
- Sind erweiternde Primärschlüssel unterstrichelt?
- Sind Generalisierungen korrekt modelliert?

34

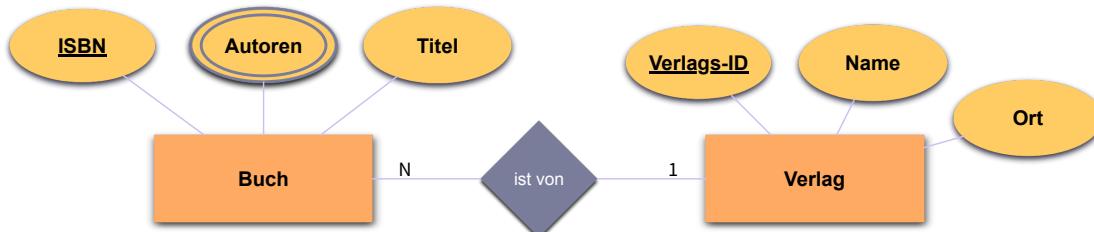
# Beispielaufgabe

Eine Bibliothek möchte eine Datenbank einsetzen, um Bücher zu speichern. Jedes Buch hat eine eindeutige ISBN, mehrere Autoren, und einen Titel. Bücher sind von einem Verlag. Von einem Verlag kann es mehrere Bücher geben. Jeder Verlag wird durch eine Verlags-ID identifiziert und hat einen Namen sowie einen Ort.

35

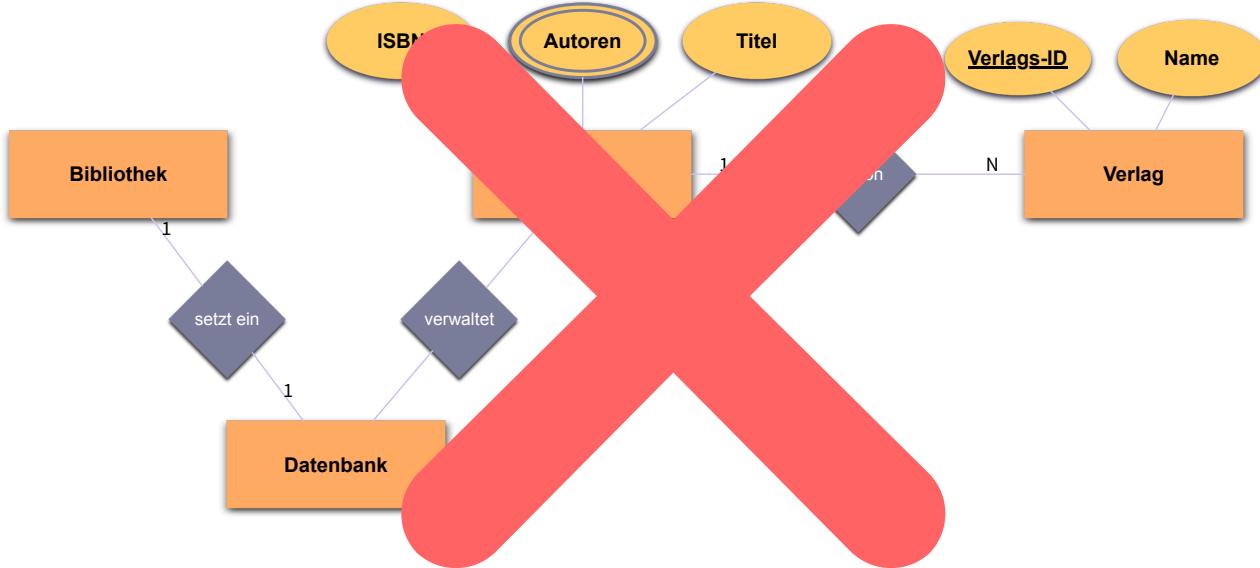
# Beispielaufgabe

Eine Bibliothek möchte eine Datenbank einsetzen, um **Bücher** zu speichern. Jedes Buch hat eine eindeutige ISBN, mehrere Autoren, und einen **Titel**. Bücher sind von **einem Verlag**. Von **einem Verlag** kann es mehrere Bücher geben. Jeder **Verlag** wird durch eine **Verlags-ID** identifiziert und hat einen **Namen** sowie einen **Ort**.



36

# Falsche Lösung



In diesem ER-Diagramm fehlen Kardinalitäten an einer Beziehung, an einer anderen Beziehung sind sie falsch herum. Dem Entitätstyp "Bücher" fehlt der Primärschlüssel. Die Entitätstypen "Bibliothek" und "Datenbank" sind völlig falsch. Wir speichern viele Bücher und viele Verlage, aber wir speichern keine Bibliotheken und Datenbanken! Man erkennt überflüssige Entitätstypen auch daran, dass einem dazu keine Attribute einfallen.