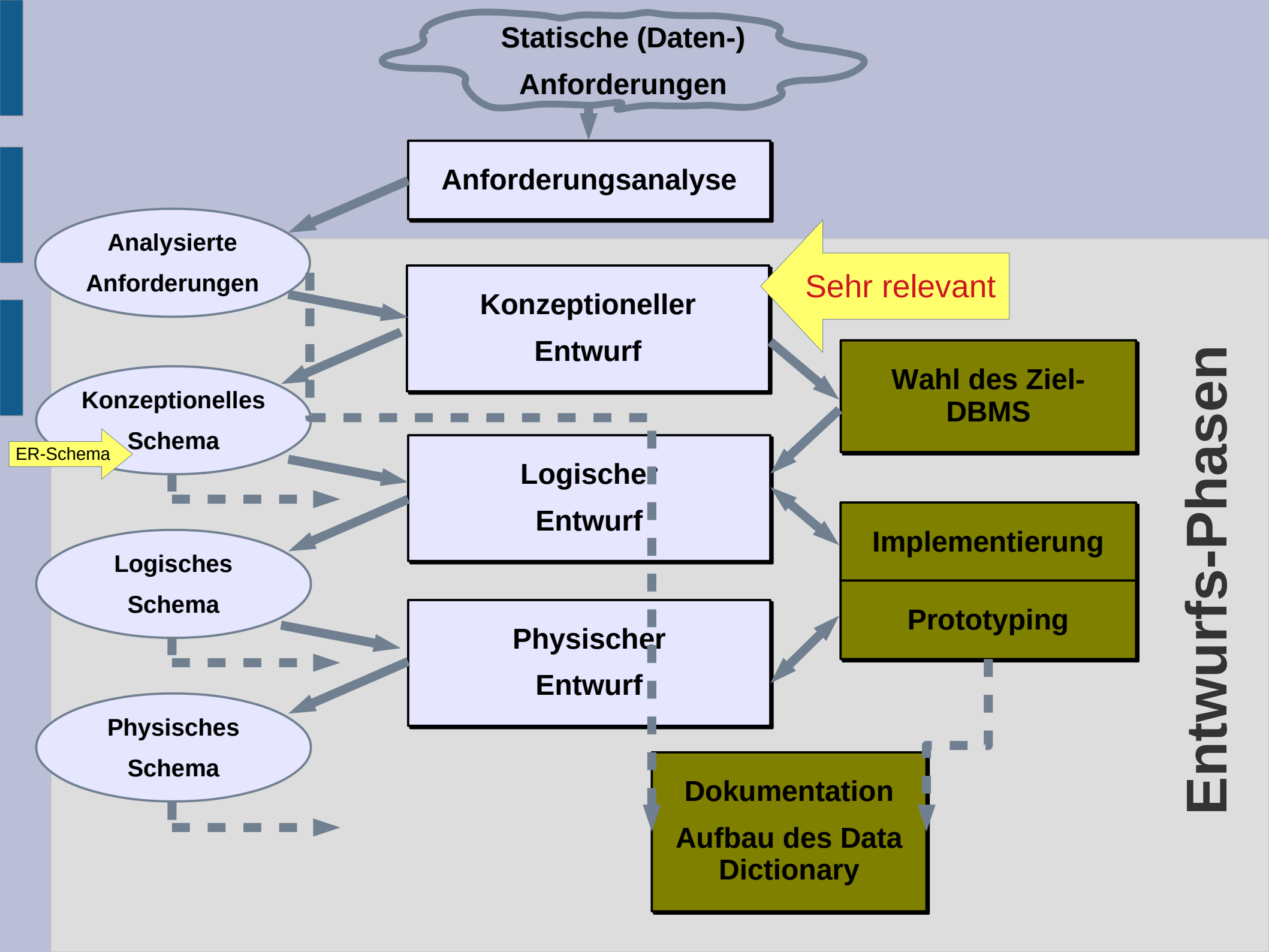

Statische (Daten-) Anforderungen
Anforderungsanalyse
Analysierte Anforderungen
Konzeptioneller Entwurf
Sehr relevant
Konzeptionelles Schema
ER-Schema
Wahl des Ziel-DBMS
Logischer Entwurf
Implementierung
Logisches Schema
Prototyping
Physischer Entwurf
Physisches Schema
Dokumentation Aufbau des Data Dictionary
Entwurfs-Phasen

Entwurfsphasen

Im folgenden der Vollständigkeit halber einige Worte zu allen Entwurfsphasen.

Das hier vorgestellte **Entity-Relationship-Modell** gehört dabei zum konzeptionellen Entwurf.

Anforderungsanalyse

- Informationsanforderungen
  - Welche statischen Informationen wird das DBS benutzen (Daten, Realwelt-Objekte, Typen, Attribute, Wertebereiche, Beziehungen ...)
  - Integritätsbedingungen
- Bearbeitungsanforderungen (Processing Requirements)
  - Datenvolumen
- Typische Aktivitäten während der Anforderungsanalyse
  - Identifikation der wesentlichen Benutzergruppen
  - Sichtung existierender Dokumentation
  - Fragebögen und Interviews mit Kunden

Aufgabe des Designers

- Alle Datenkonstrukte erfassen, die gebraucht werden, z.B. Rechnungsdaten und Materialdaten

- Namenskonflikte, Inkonsistenzen vermeiden / beheben (Preis, Preis zu „Zement.Preis“, „Montage.Preis“ aufspalten)

- Dabei aber vom Ziel - DBS abstrahieren, denn dieses kennt man am Anfang evtl. nicht nicht.

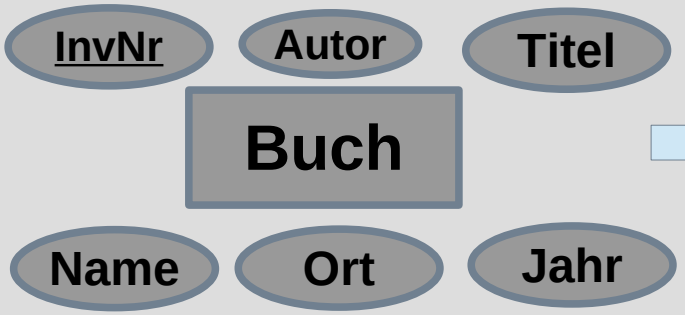
Aufgabe des Designers

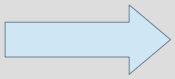
Natürliche Sprache des Pflichtenheftes aus der Analysephase in eine formale Beschreibung transformieren

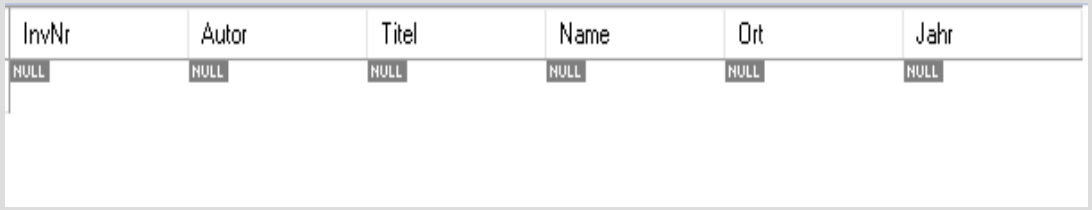
Logischer Entwurf

Übersetzung des abstrakten ER - Schemas in ein konkretes Datenmodell des verwendeten DBS

→ z.B. aus ER in relationales Modell übertragen
Dies wird Inhalt der weiterführenden Vorlesung sein.







Physischer Entwurf

In der Regel schwer, da dies Aufgabe des DBMS ist

(Aufgaben wären hier z.B. effiziente Speicherung auf Datenträgern)

Vorteile des ER-Modells

Unabhängig von einem konkreten DBS

Grundkonstrukte: ***Entity*** *und* ***Relationship***

Entities und Attribute

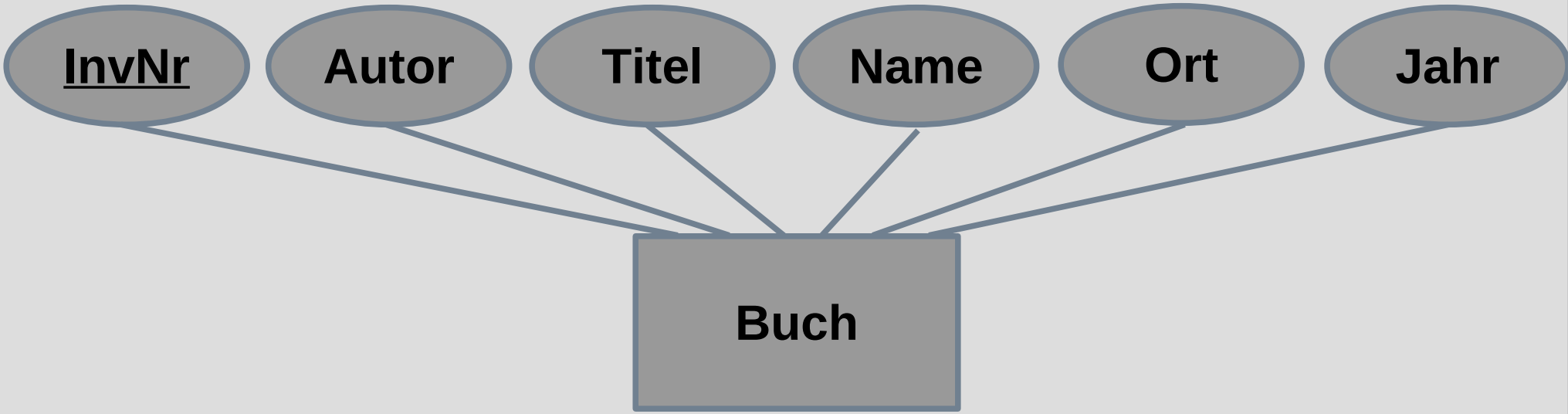
Entities:

- Wohlunterscheidbare Dinge der realen Welt
- Besitzen Eigenschaften, konkrete Ausprägung: Werte
- Beispiele: Personen, Autos, Städte, Firmen

Im Prinzip kann man sich Entities wie Klassen vorstellen:

```
entity Buch {
 InvNr,
 Autor, Titel,
 Name, Ort, Jahr
}
```

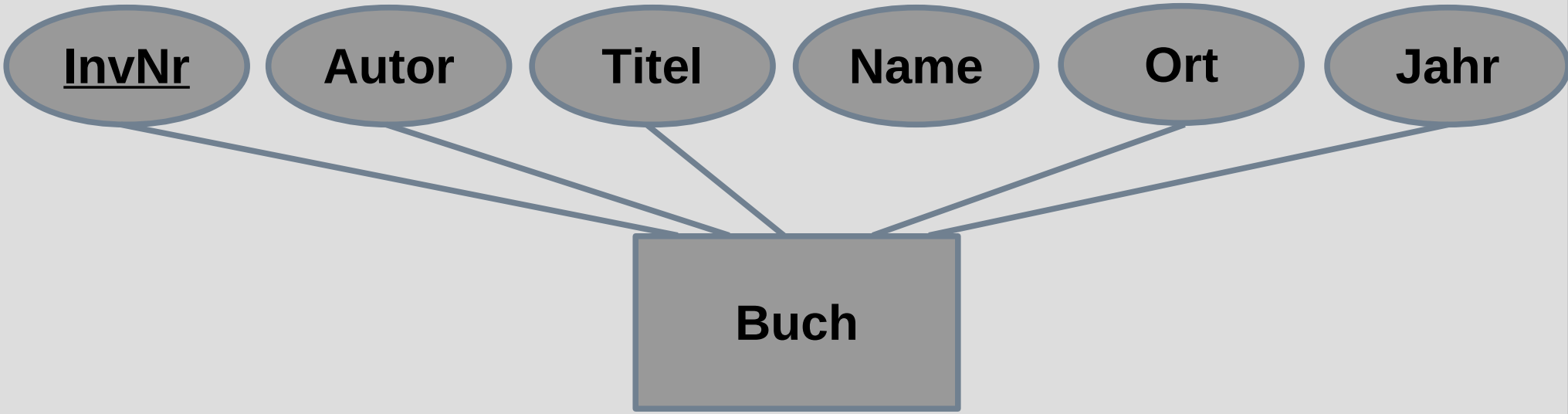
Entities
Diagramm für Beispiel Buch



Graphische Notation

Entity - Bezeichner in Rechteck
Attribute in Ellipsen

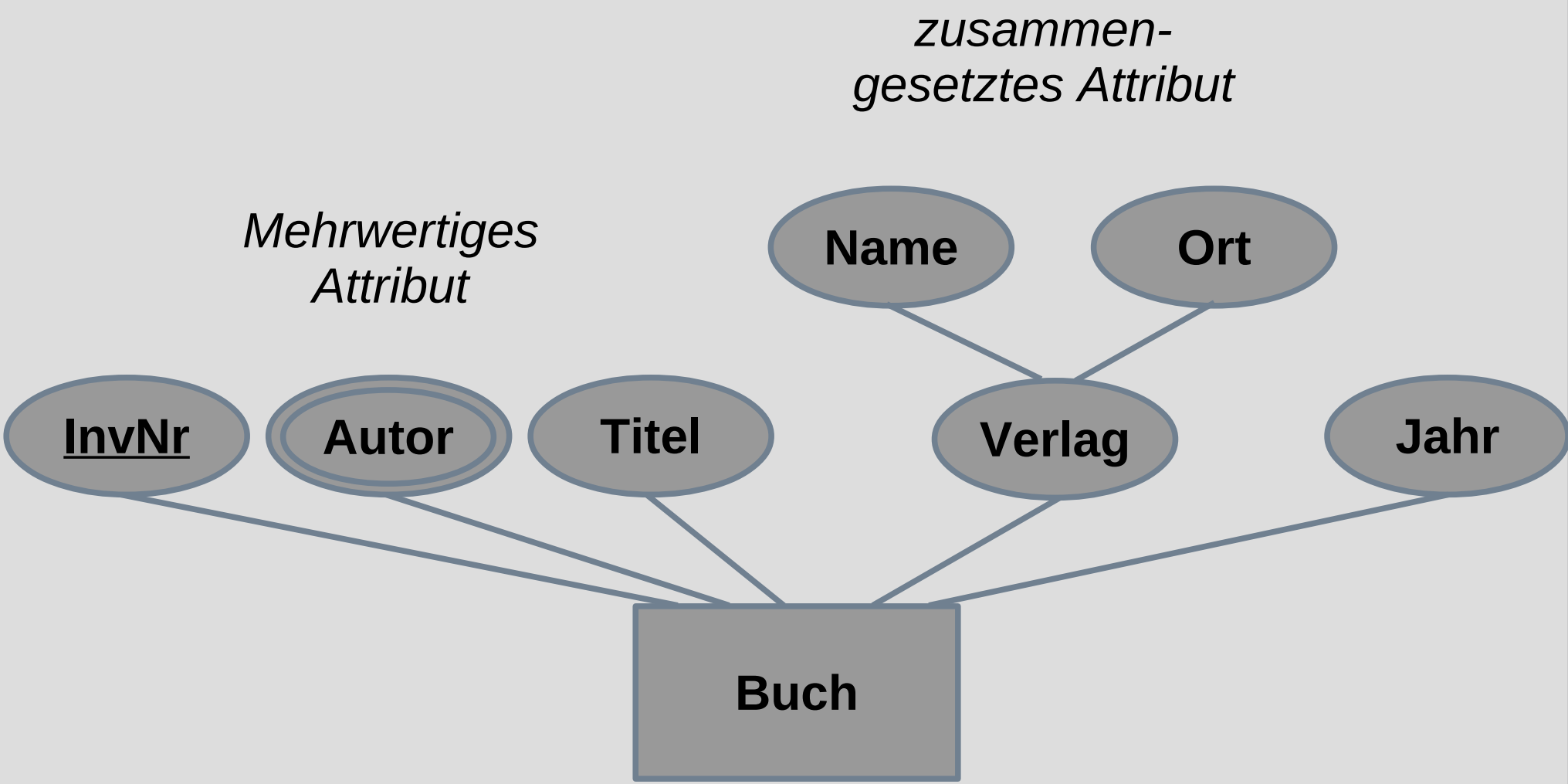
Entities
Diagramm für Beispiel Buch



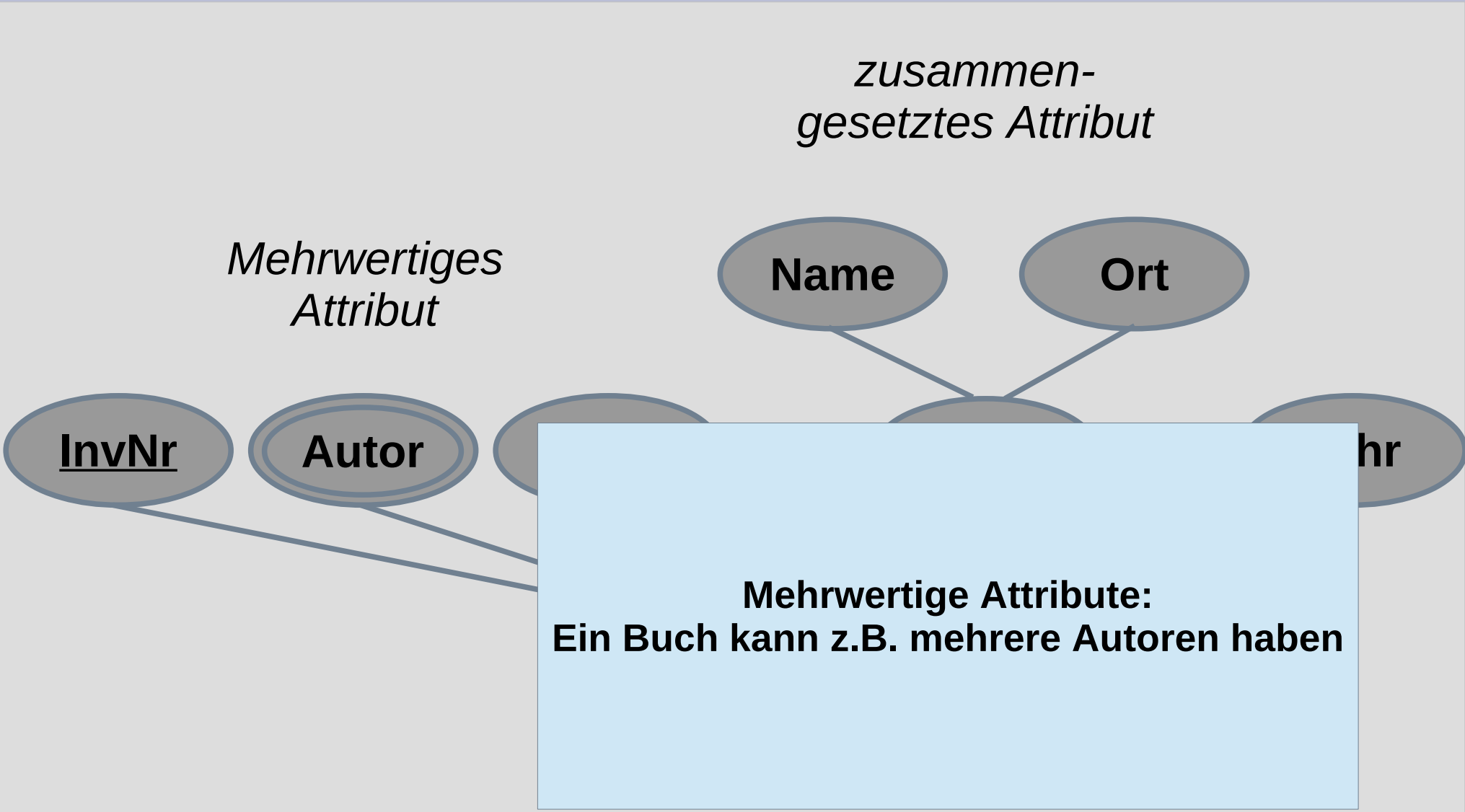
Wichtig:

Attribute von Entities sind zeitinvariant

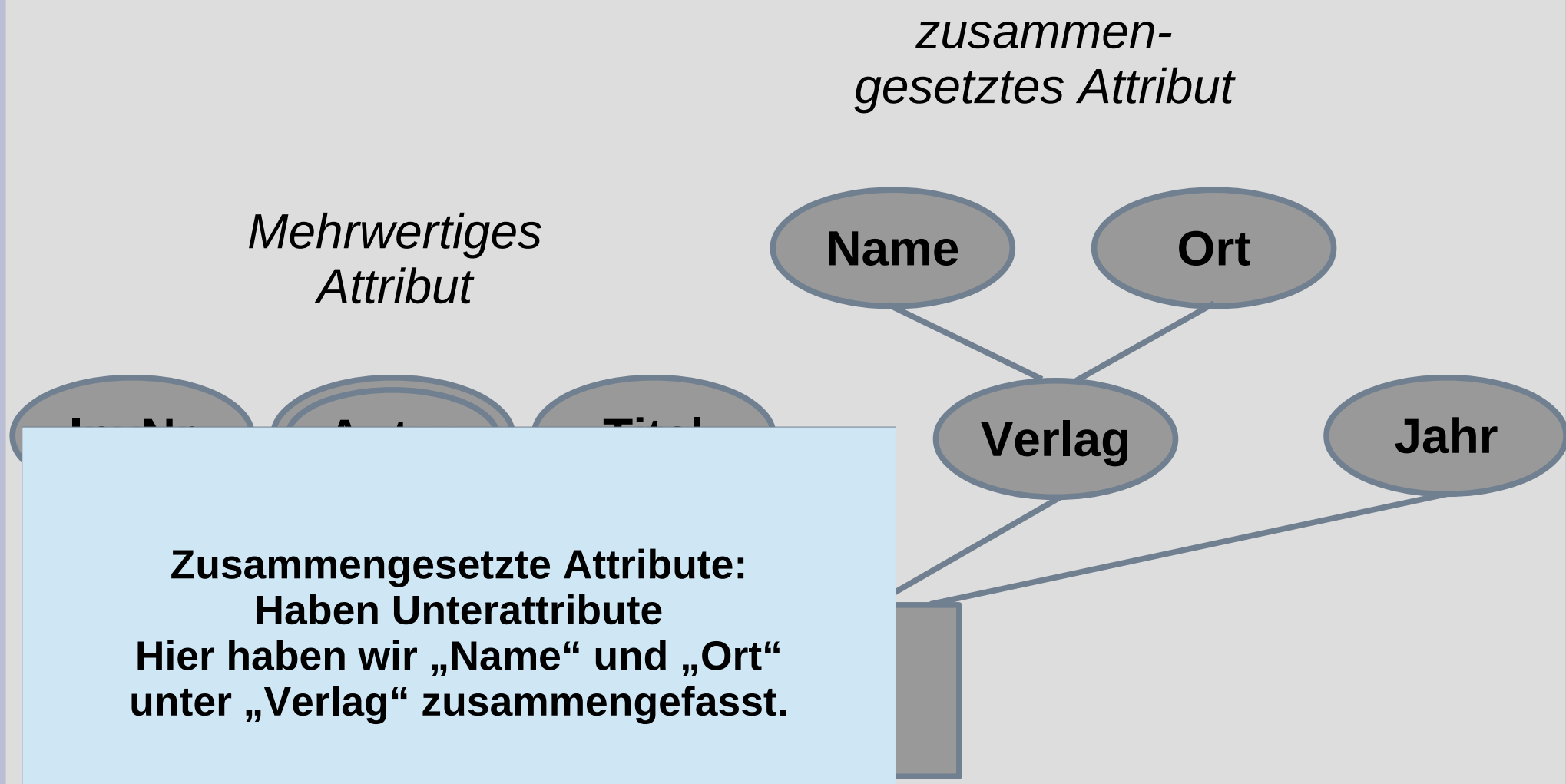
Entities
Diagramm für Beispiel Buch



Entities
Diagramm für Beispiel Buch

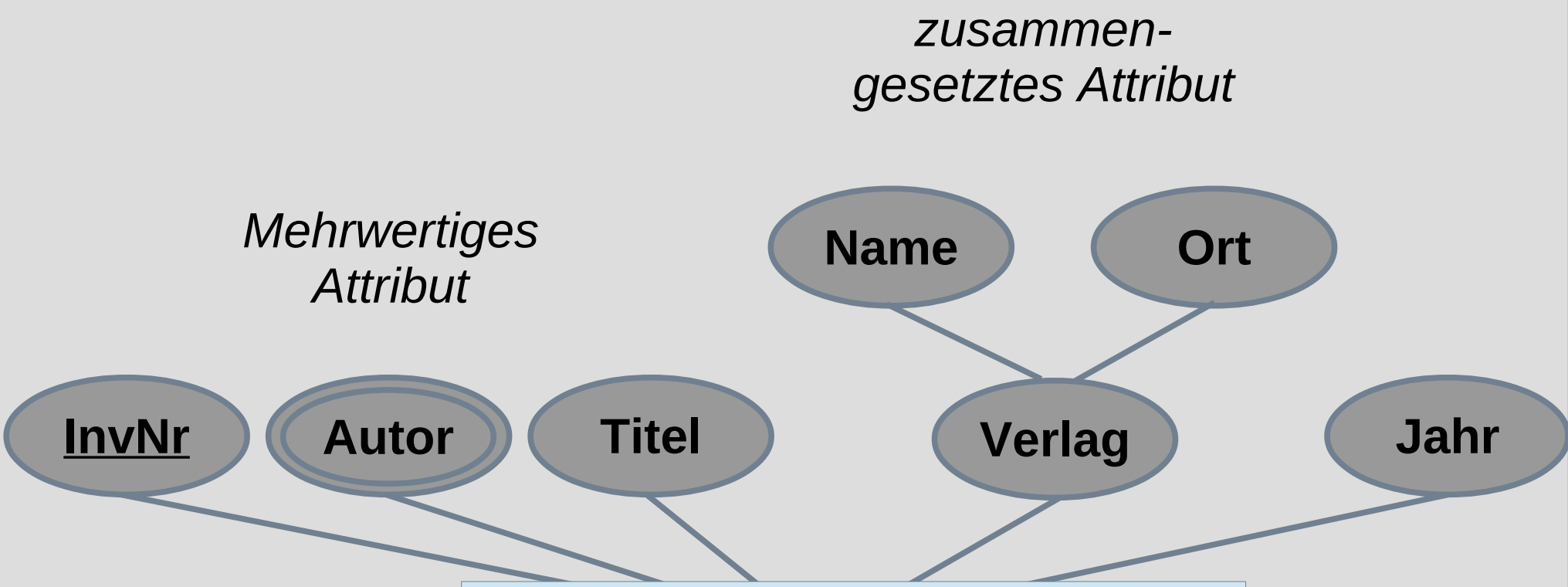


Entities
Diagramm für Beispiel Buch



Zusammengesetzte Attribute:
Haben Unterattribute
Hier haben wir „Name“ und „Ort“
unter „Verlag“ zusammengefasst.

Entities
Diagramm für Beispiel Buch



Wie man mehrwertige Attribute und zusammengesetzte Attribute am Ende in Tabellen abbildet, ist Aufgabe des logischen Entwurfs, im ER-Schema ist das noch abstrakt.

Entity-Deklaration nach Lehrbuch

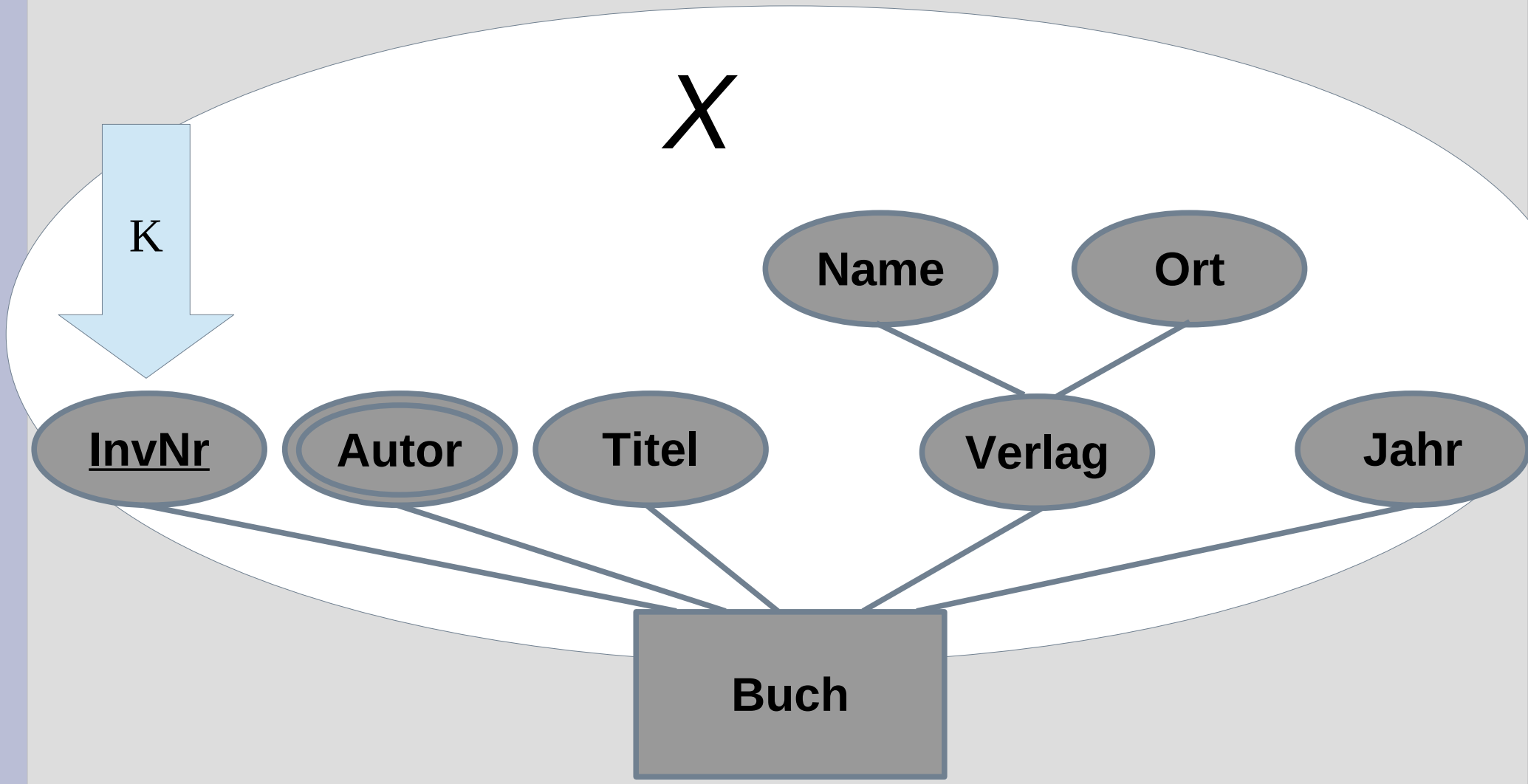
Eine Entity-Deklaration hat die Form ***E* = (*X*, *K*)**.

Sie besteht aus einem Format *X* und einem **Primärschlüssel *K***, welcher aus (einwertigen) Elementen von *X* zusammengesetzt ist.

Die Elemente eines Formates *X* werden dabei wie folgt notiert:

- Einwertige Attribute: *A*
- Mehrwertige Attribute: {*A*}
- Zusammengesetzte Attribute: $A(B_1, \ldots, B_k)$

Entity-Diagramm für Beispiel Buch



Beispiel für Entity-Typ *Buch*

Wichtig für spätere Füllung der Tabellen

Zu einem Zeitpunkt *t* könnte folgendes vorliegen:

$$Buch^t = \{b_1, b_2, b_3\}$$

Im ER-Modell definiert man b_1 als Schlüssel. Das hat später beim Füllen von Tabellen Konsquenzen. Das hier geht:

b_1 = (123, {'Vossen', 'Witt'}, 'DB2 Handbuch', ('Addison-Wesley', 'Bonn'), 1990)
b_2 = (125, {'Vossen', 'Witt'}, 'SQL/DS Handbuch', ('Addison-Wesley', 'Bonn'), 1988)
b_3 = (130, {'Witt'}, 'OO Programmierung', ('Oldenbourg', 'München'), 1992)

Nicht vorkommen darf dann aber:
b_4 = (123, {'Vossen'}, 'Transaktionsverarbeitung', ('Hüthig', 'Heidelberg'), 1990)

(Dies würde den Schlüssel InvNr verletzen)

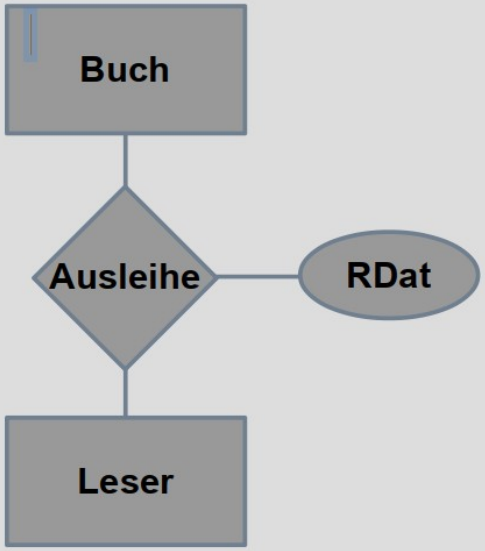
Zusammenfassung Entities

- Entities
  - Entity-Typen: Zusammenstellung von Attributen

- Entity-Deklarationen sind vollständig graphisch darstellbar
  - Entity-Deklaration als Rechteck mit Namen
  - Attribute als Kreise oder Ellipsen
  - (Primär-) Schlüssel unterstrichen
  - Zusammengesetzte Attribute als Baum
  - Mehrwertige Attribute in Doppelellipsen eingeschlossen

Relationships
Nach Lehrbuch

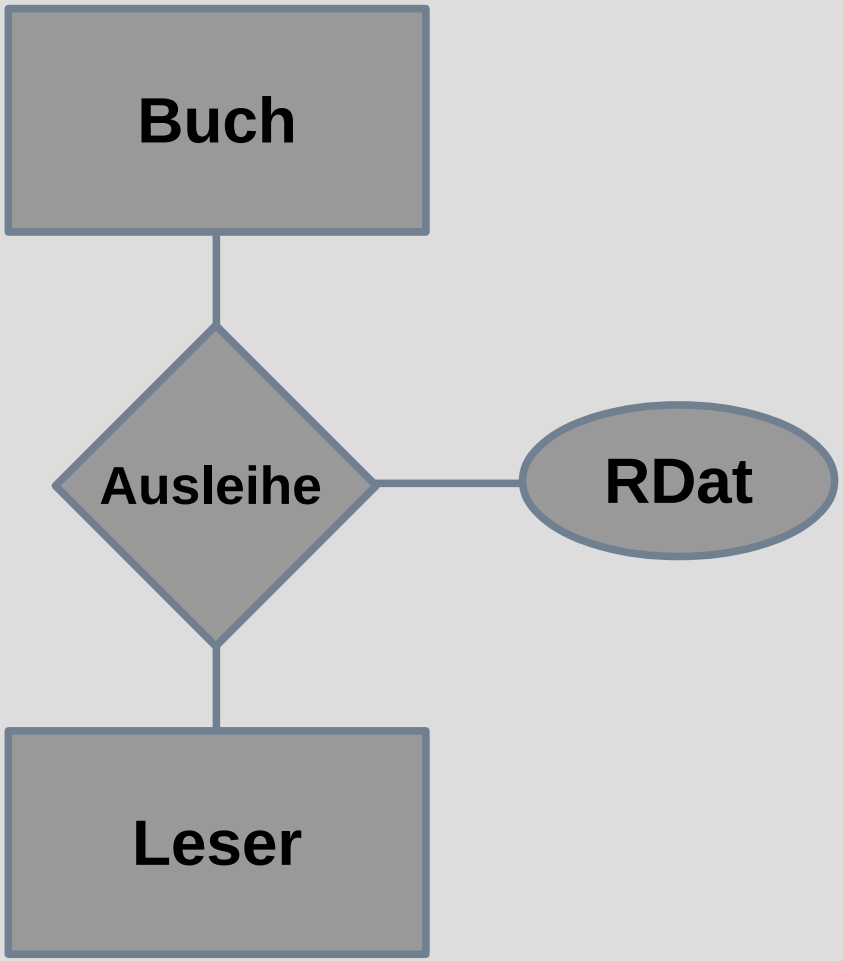
Eine Relationship-Deklaration hat die Form ***R* = (Ent, *Y*)**.
Dabei ist *R* der Name der Deklaration, Ent bezeichnet die Menge der Namen der Entity-Deklarationen, zwischen denen eine Beziehung definiert werden soll, und *Y* ist eine (möglicherweise leere) Menge von Attributen (der Beziehung).



R = „Ausleihe“

Ent = {Buch, Leser}
Y = {Rdat}

Realtionships
Bsp. Ausleihe



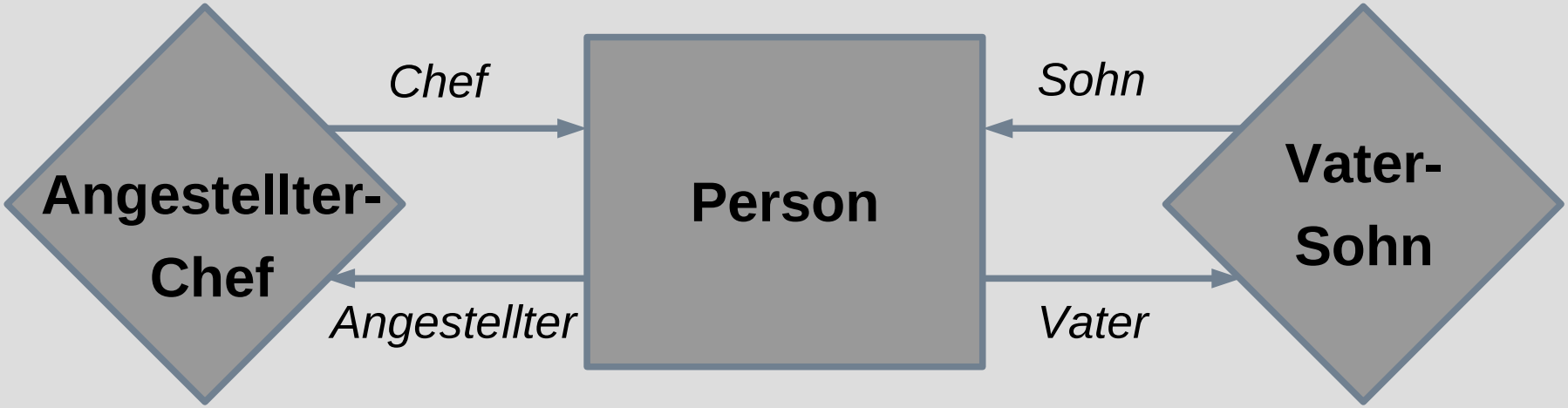
Relationship-Deklaration:

Raute mit dem Namen der Deklaration

Entities (hier Buch und Leser) als Rechtecke und mit Kanten mit der Raute verbunden

Eventuelle Attribute (hier RDat) in Ellipsen und mit Kanten mit der Raute verbunden

Spezielle Relationships
Rekursive Beziehung



Eine Entity-Deklaration nimmt **mehrfach** an einem Relationship teil

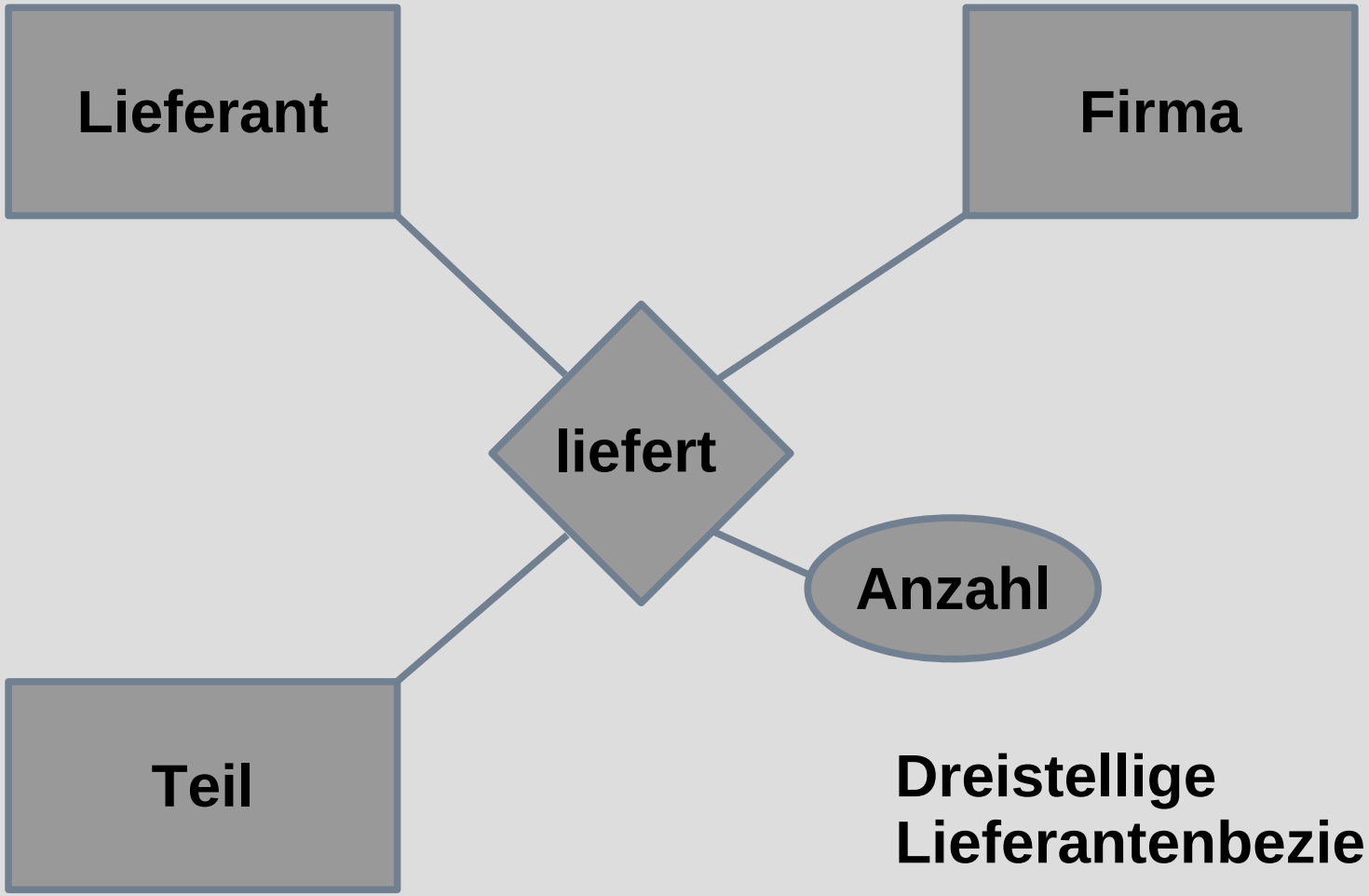
→ das ist laut Definition zulässig

Eigenschaften von Relationships

- Stelligkeit (grad)

- 1:n und m:n - Beziehung

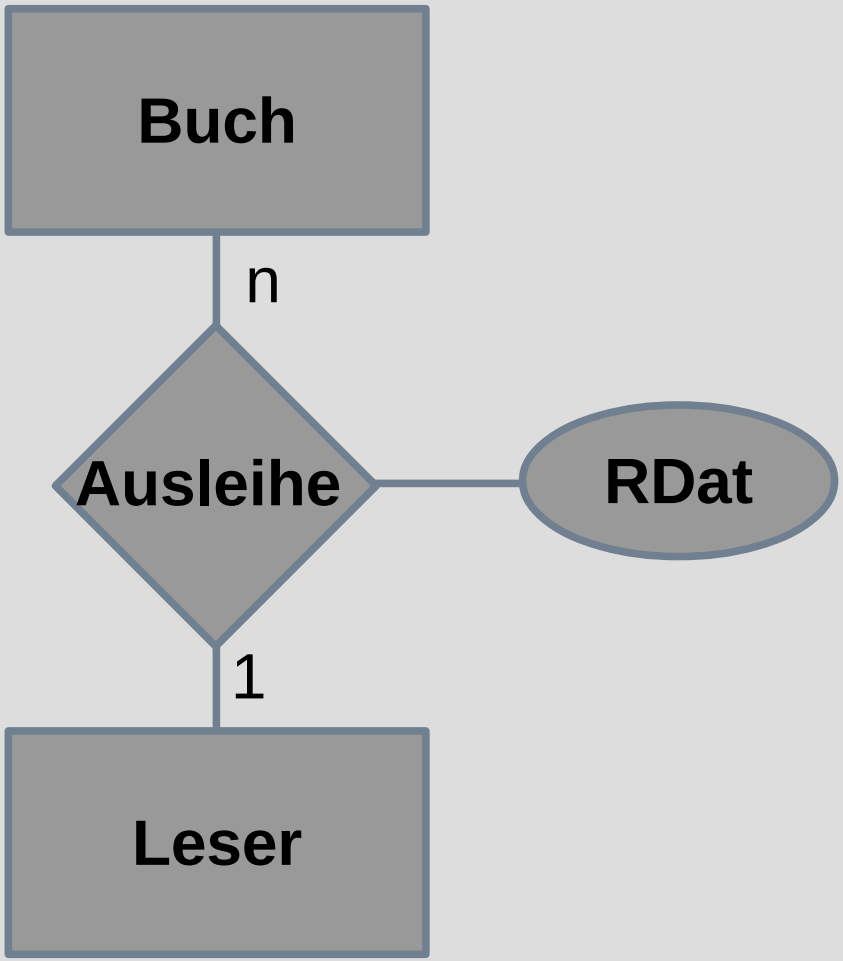
Stelligkeit einer Relationship-Deklaration



Dreistellige Lieferantenbeziehung

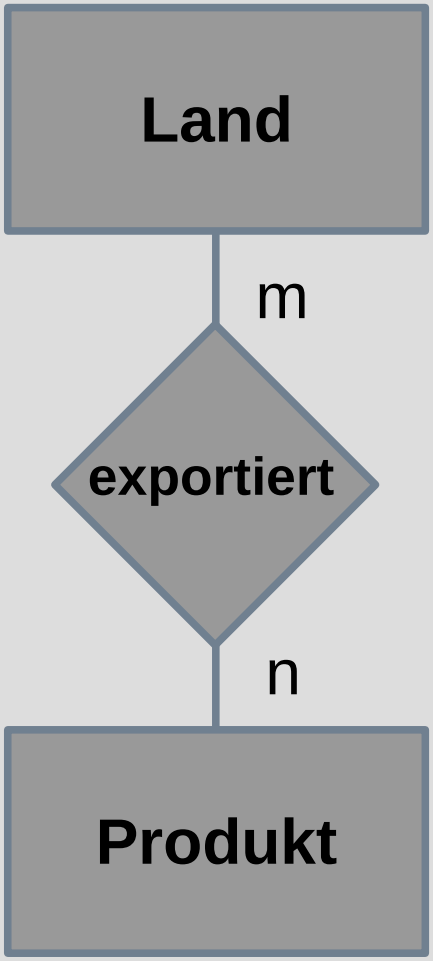
grad(*R*)

1:n Beziehungen
(many - one)



Ein Leser kann n Bücher ausleihen.

m:n – Beziehung
(many - many)



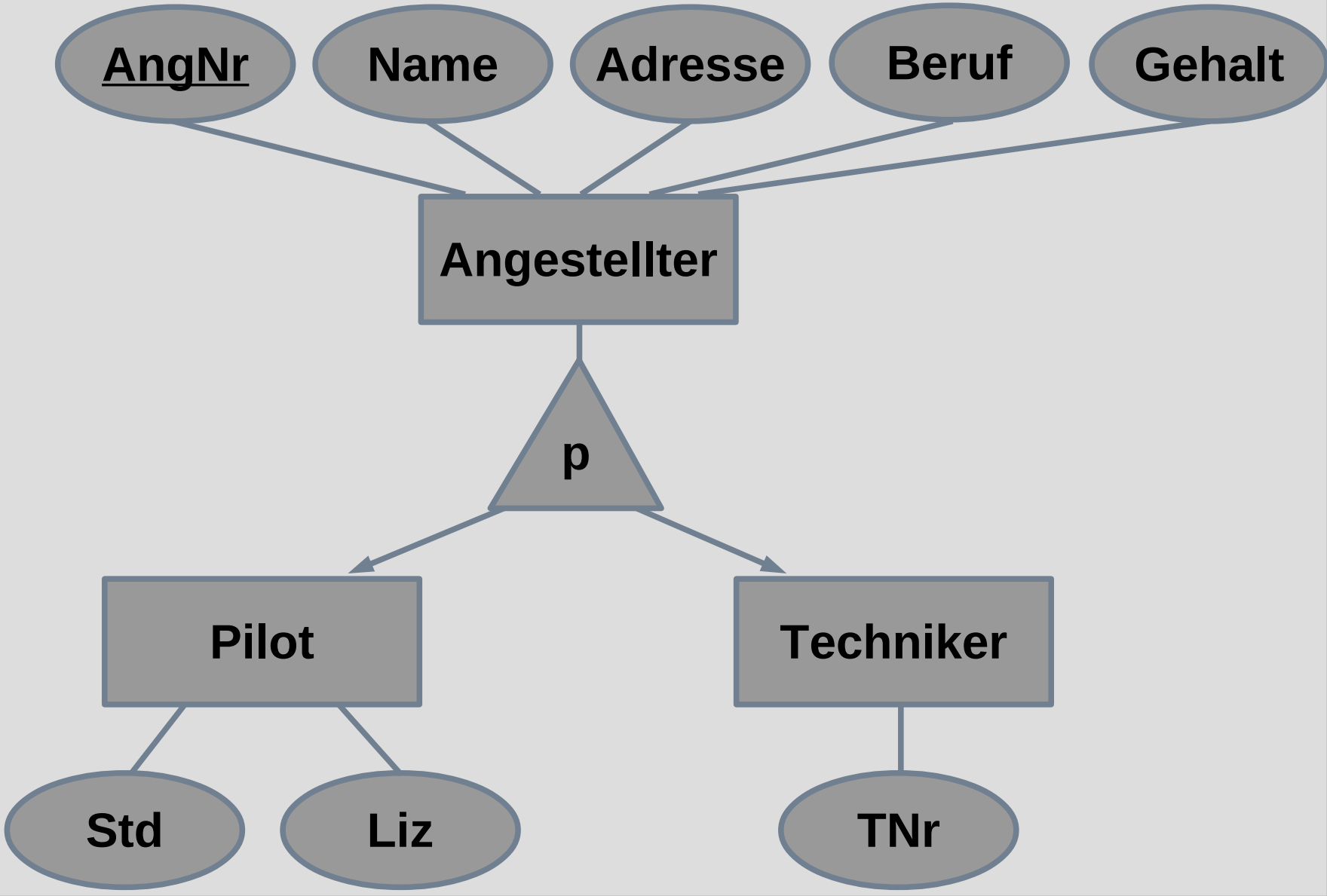
Keine Restriktionen an die

Entity-Paare eines Relationship-Sets

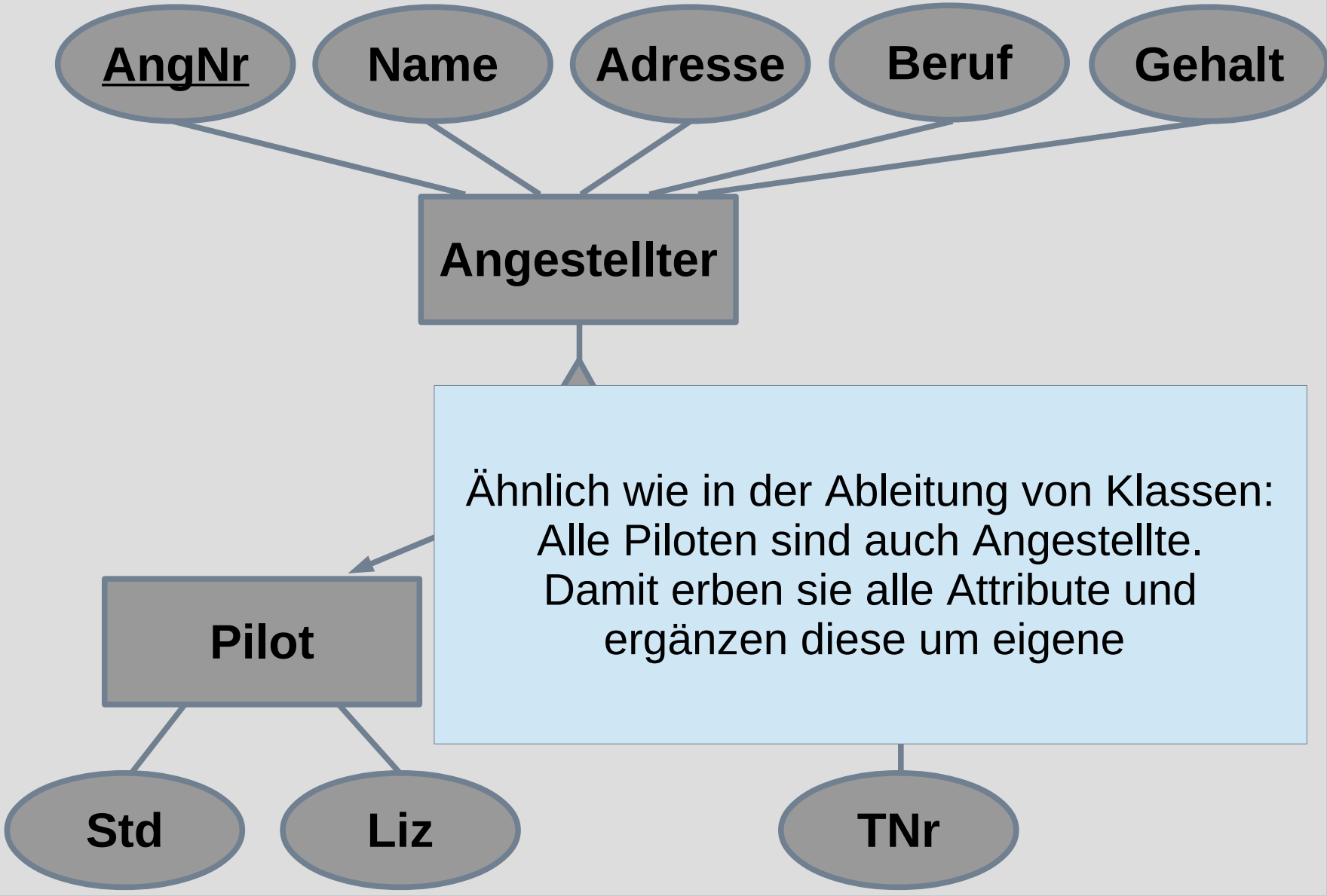
ER - Modell

Spezielle Beziehungen

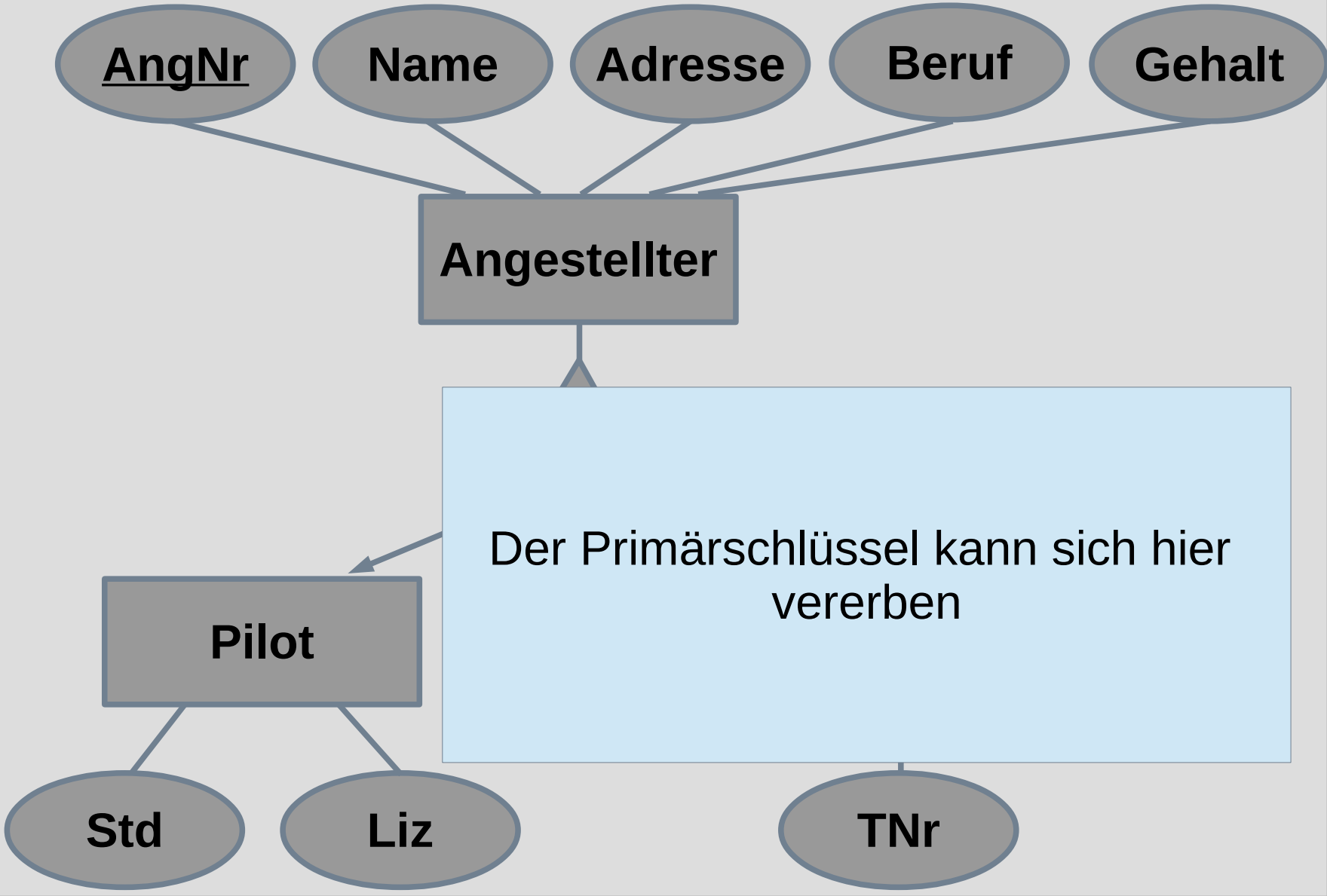
IS-A Beziehung



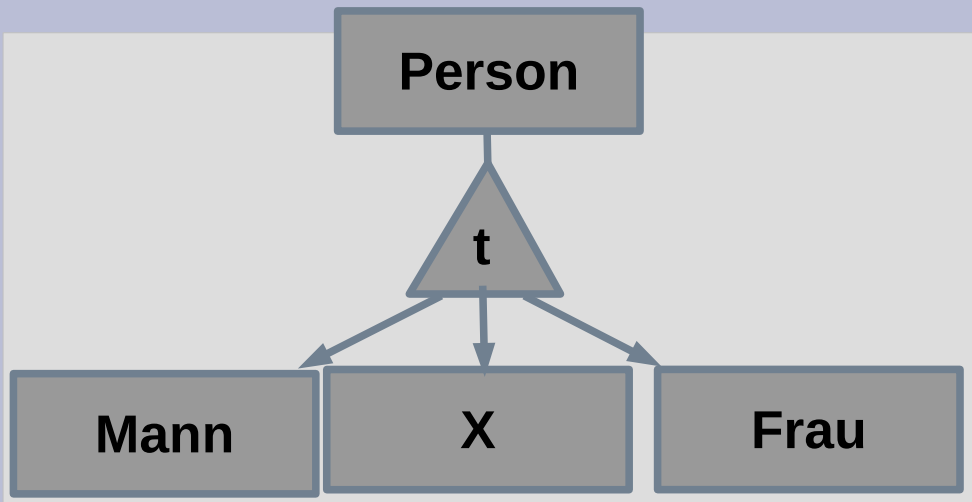
IS-A Beziehung



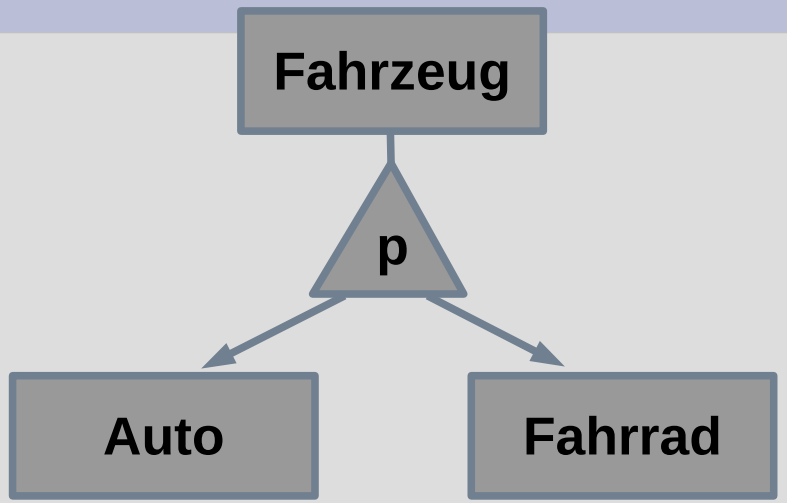
IS-A Beziehung



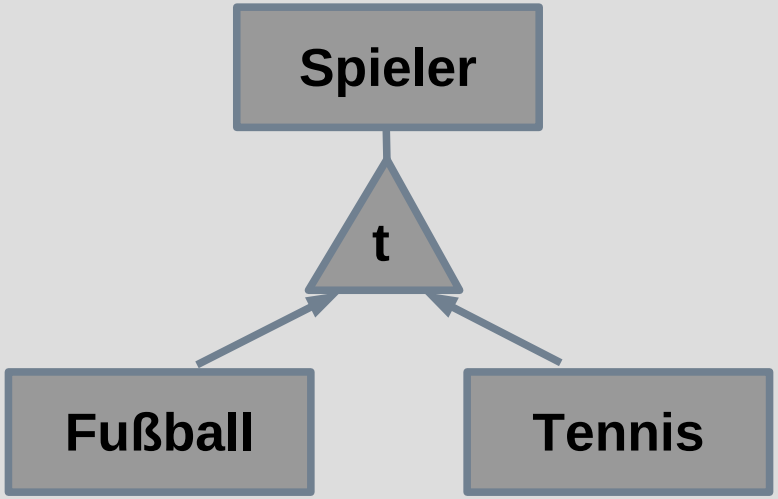
Spezialisierung
Is-A und Part-Of



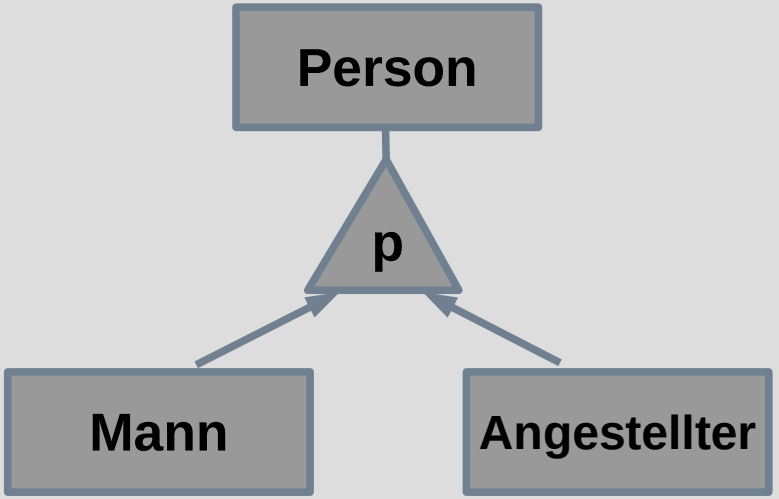
total, disjunkt



partiell, disjunkt

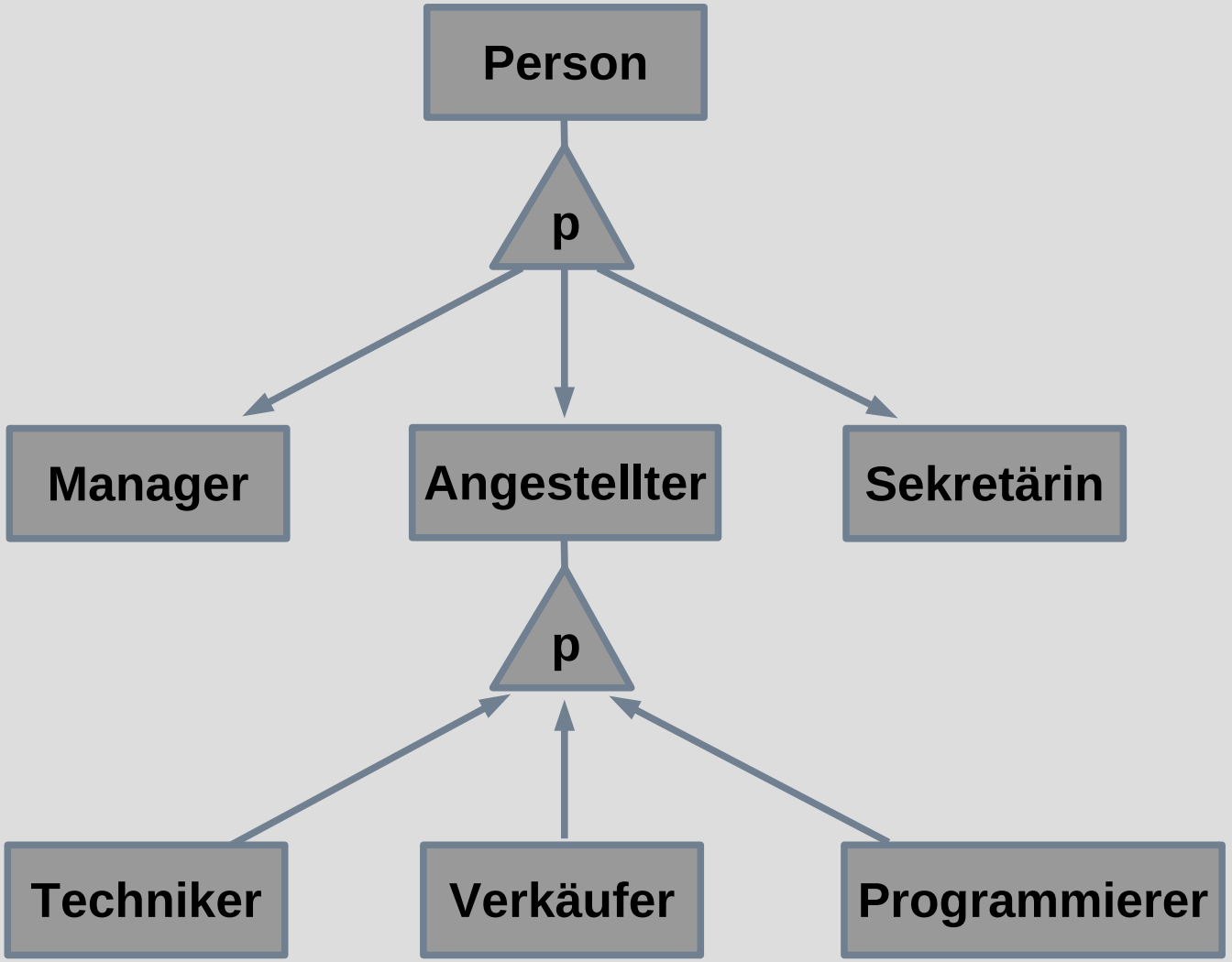


total, nicht disjunkt



partiell, nicht disjunkt

Spezialisierungshierarchie



Zusammenfassung ER-Modell

- Entity-Deklarationen mit
  - Namen
  - einwertigen, mehrwertigen und zusammengesetzten Attributen und deren Wertebereichen
  - Primärschlüssel

- Relationship-Deklarationen mit
  - Namen
  - beteiligten Entity-Deklarationen
  - ggf. eigenen Attributen
  - Komplexitätsfestlegung

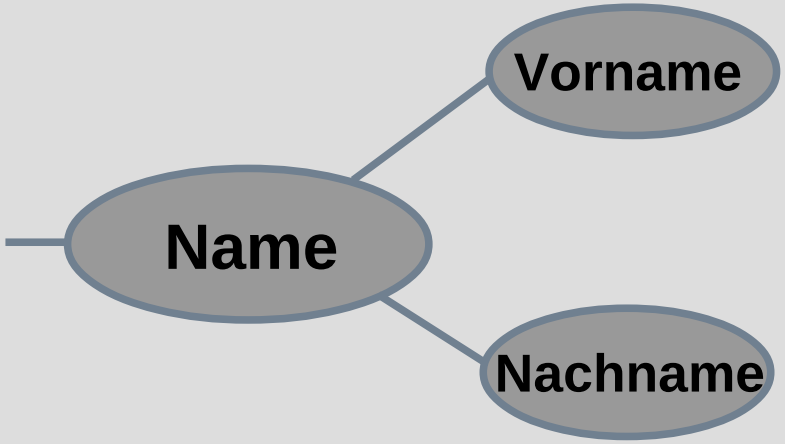
Zusammenfassung ER-Modell

- IS-A Beziehungen als Spezialisierungen von Entity-Deklarationen mit Typ-Festlegung
  - partiell / total
  - disjunkt / nicht disjunkt

Grafische Notation lokaler ER-Konstrukte

Entität (Entity)

Attribut

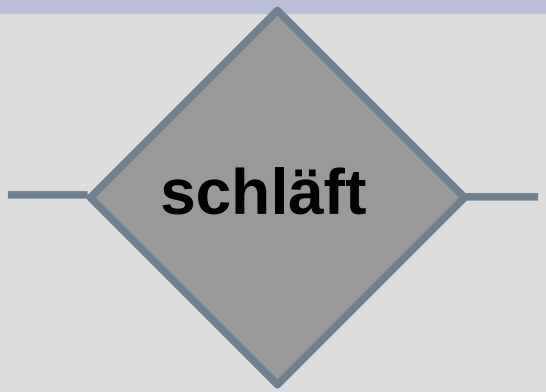


zusammengesetztes Attribut

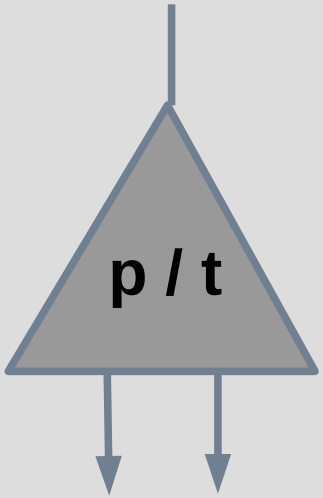
Schlüsselattribut

mehrwertiges Attribut

Grafische Notation lokaler ER-Konstrukte (2)



Relationship

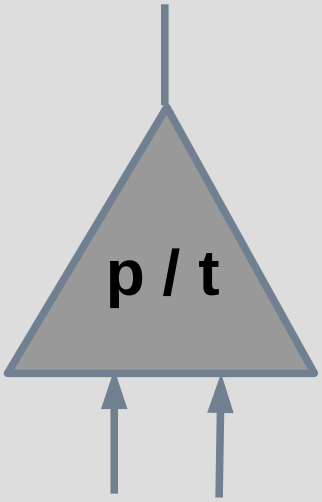


disjunkt

IS-A Beziehung

p – partiell

t - total



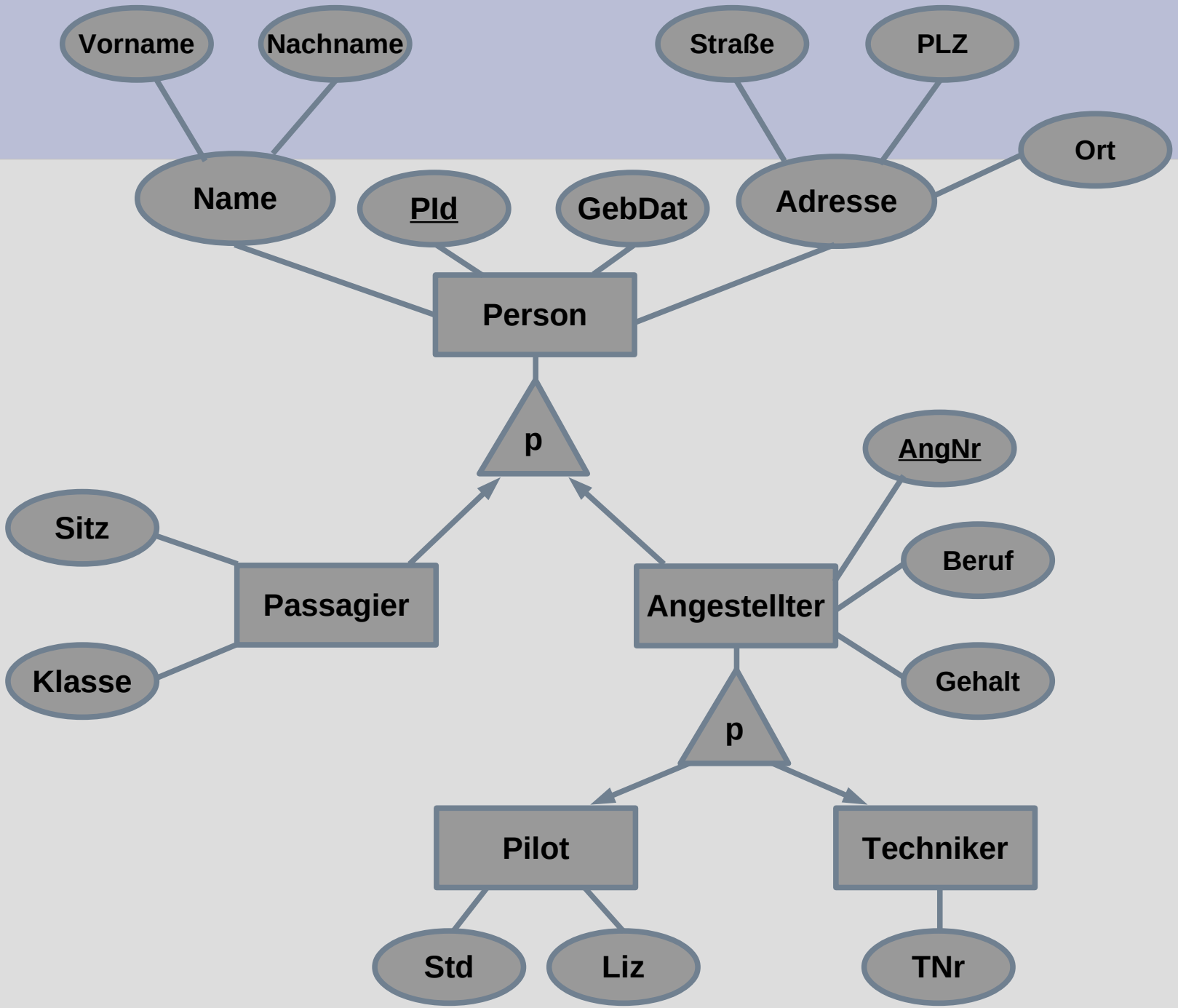
nichtdisjunkt

Beispiel Fluggesellschaft

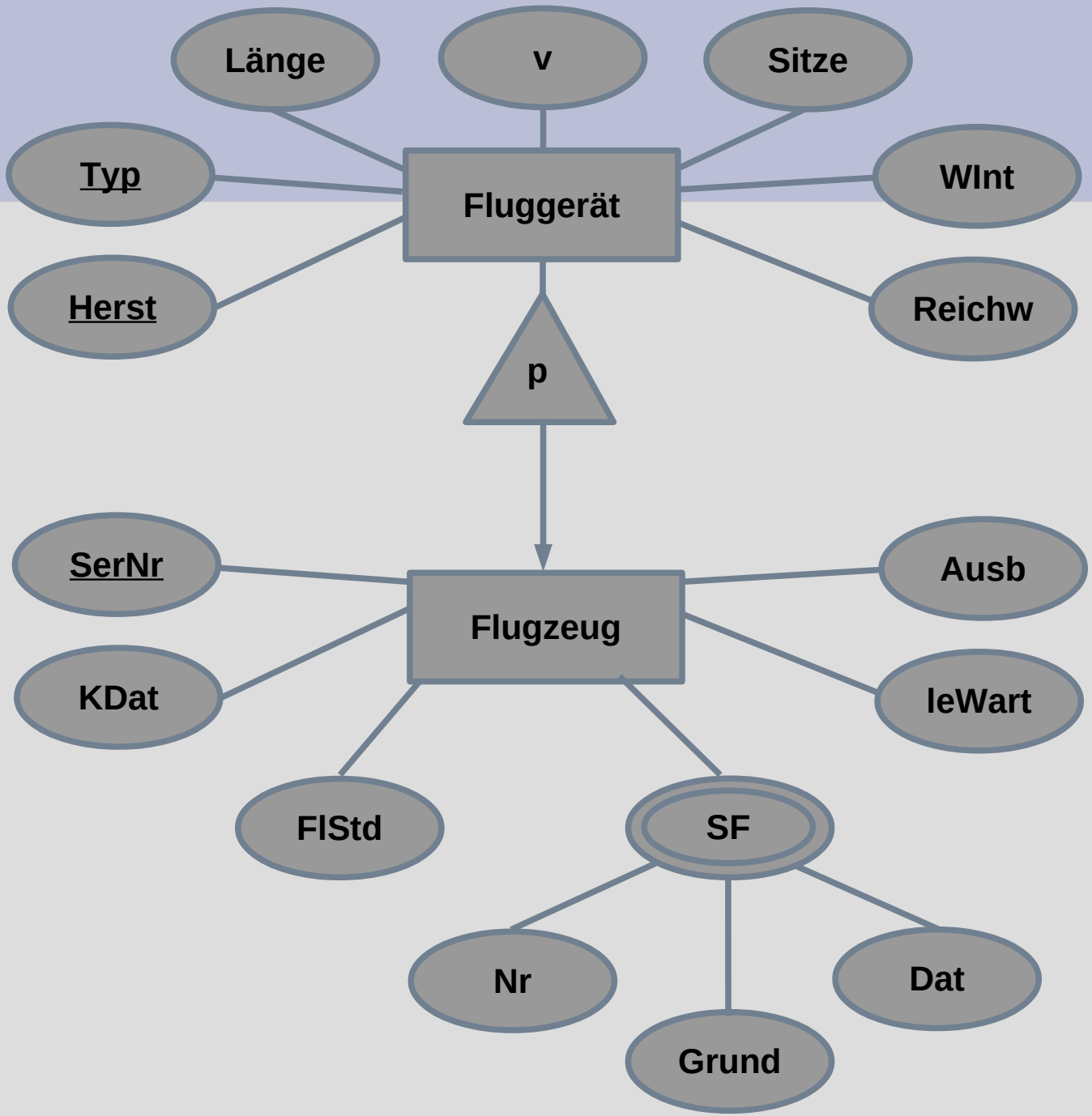
Folgende Daten sind zu speichern (Anforderungen):

- Personen
  - Passagiere
  - Angestellte
- Flugzeuge
- Flüge

Fluggesellschaft



Fluggesellschaft (2)



Qualitätsmerkmale ER-Diagramm

- Vollständigkeit
  - Schwierig zu prüfen (nur Vergleich Anforderungsanalyse mit ER Diagramm ist möglich)
- Korrektheit
  - Syntaktische Korrektheit
  - Semantisch korrekt
- Minimalität
- Lesbarkeit
  - Übersichtliche Anordnung / Größen der Symbole
  - Spezialisierunghierachien beginnen mit dem allgemeinsten Typ
  - Symmetrien werden betont
  - Diagramm möglichst kreuzungfrei
- Modifizierbarkeit