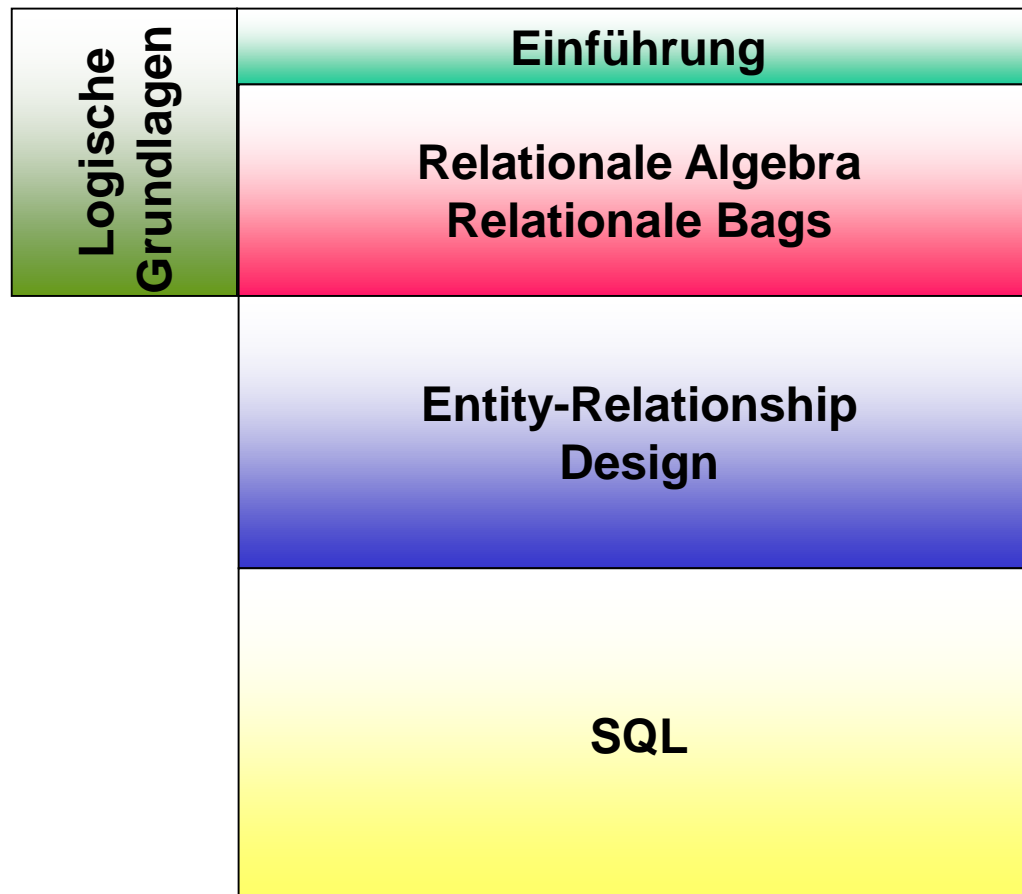


DAB1 – Datenbanken 1

Dr. Daniel Aebi (aebd@zhaw.ch)

Lektion 7: Design mit dem Entity-Relationship Modell (ERM)

Wo stehen wir?

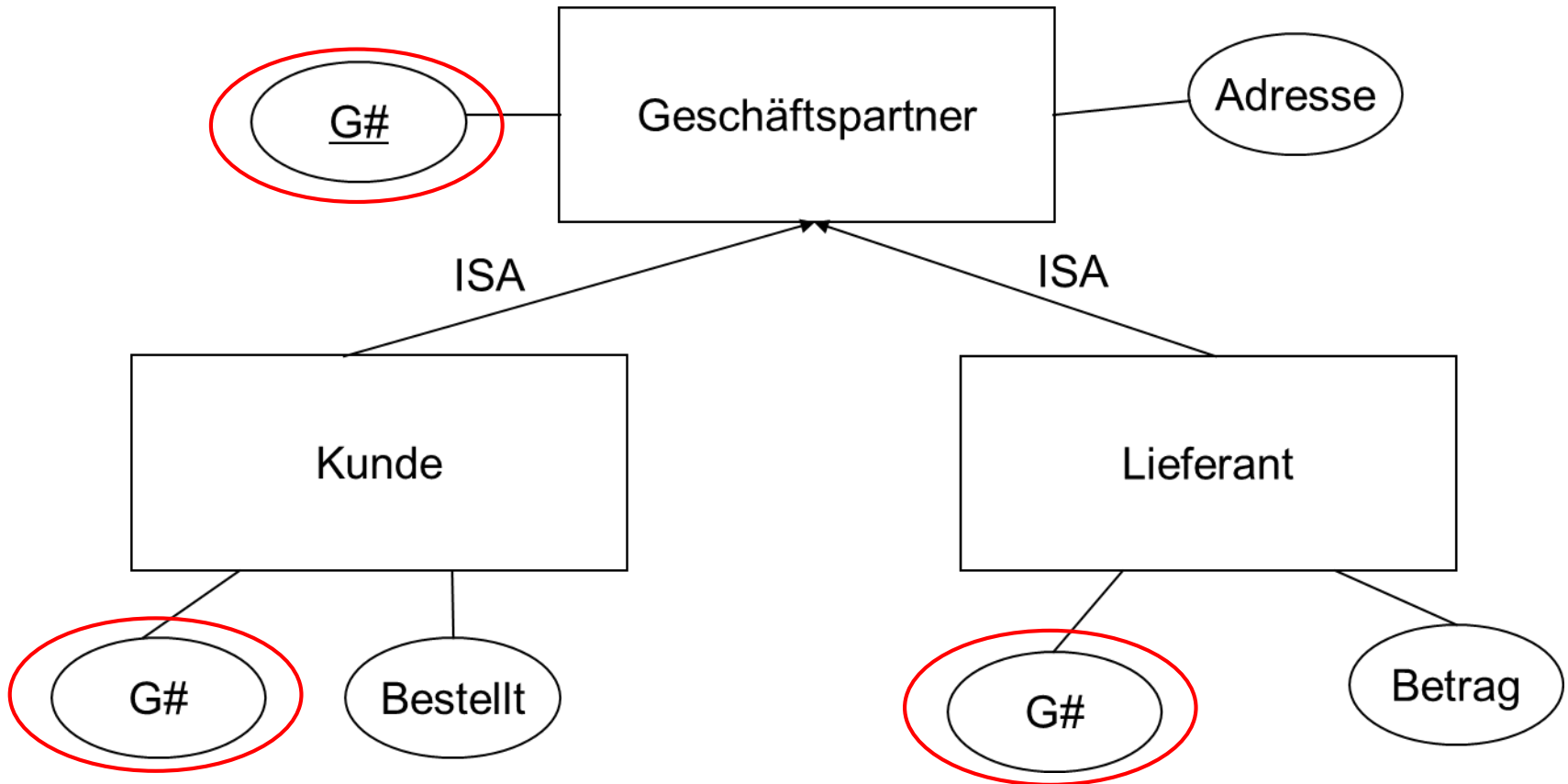


← "You are here"

Diskutiert im Unterricht. Machen Sie Ihre eigenen Notizen.

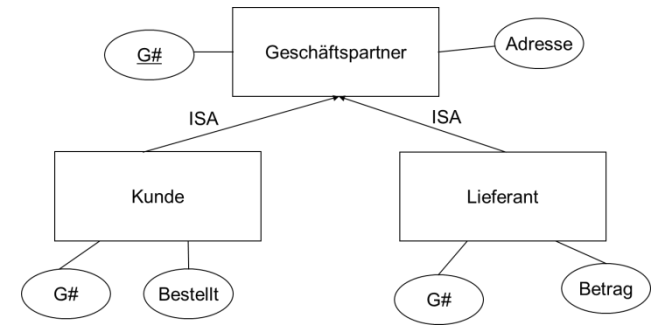
Begriff: ISA-abhängiger Entitätstyp

- Jeder Kunde resp. jeder Lieferant ist auch Geschäftspartner:



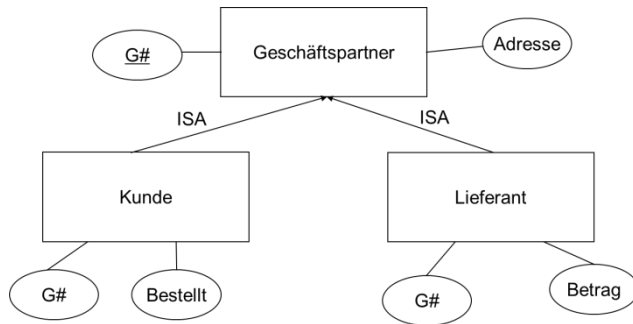
Begriff: ISA-abhängiger Entitätstyp

- Jeder Kunde resp. jeder Lieferant ist auch Geschäftspartner:
- Die Pfeile zeigen eine **existentielle Abhängigkeit** ("is a" = "ist ein")
- Die Pfeile führen zu **Schlüsselbedingungen**
- {G#} ist Schlüssel in Kunde und in Lieferant. Es wird erzwungen, dass jedem Kunden bzw. Lieferanten ein Geschäftspartner entspricht – der Kunde selbst!
- Ein Geschäftspartner kann Kunde **UND/ODER** Lieferant sein



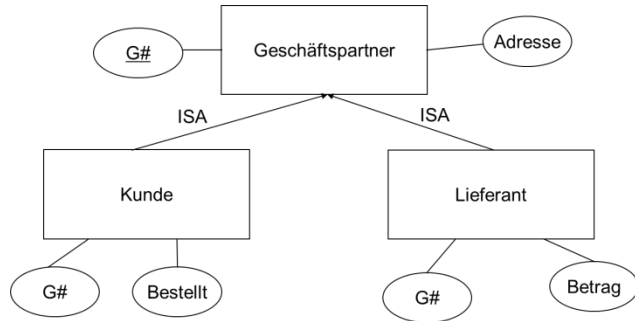
Begriff: ISA-abhängiger Entitätstyp

- «ISA» ist ein Generalisierungs-/Spezialisierungsmuster



- Kunde und Lieferant werden zu Geschäftspartnern *verallgemeinert*
- Kunde und Lieferant sind *Spezialisierungen* von Geschäftspartner
- *Gemeinsame* Attribute werden nur *einmal* benötigt

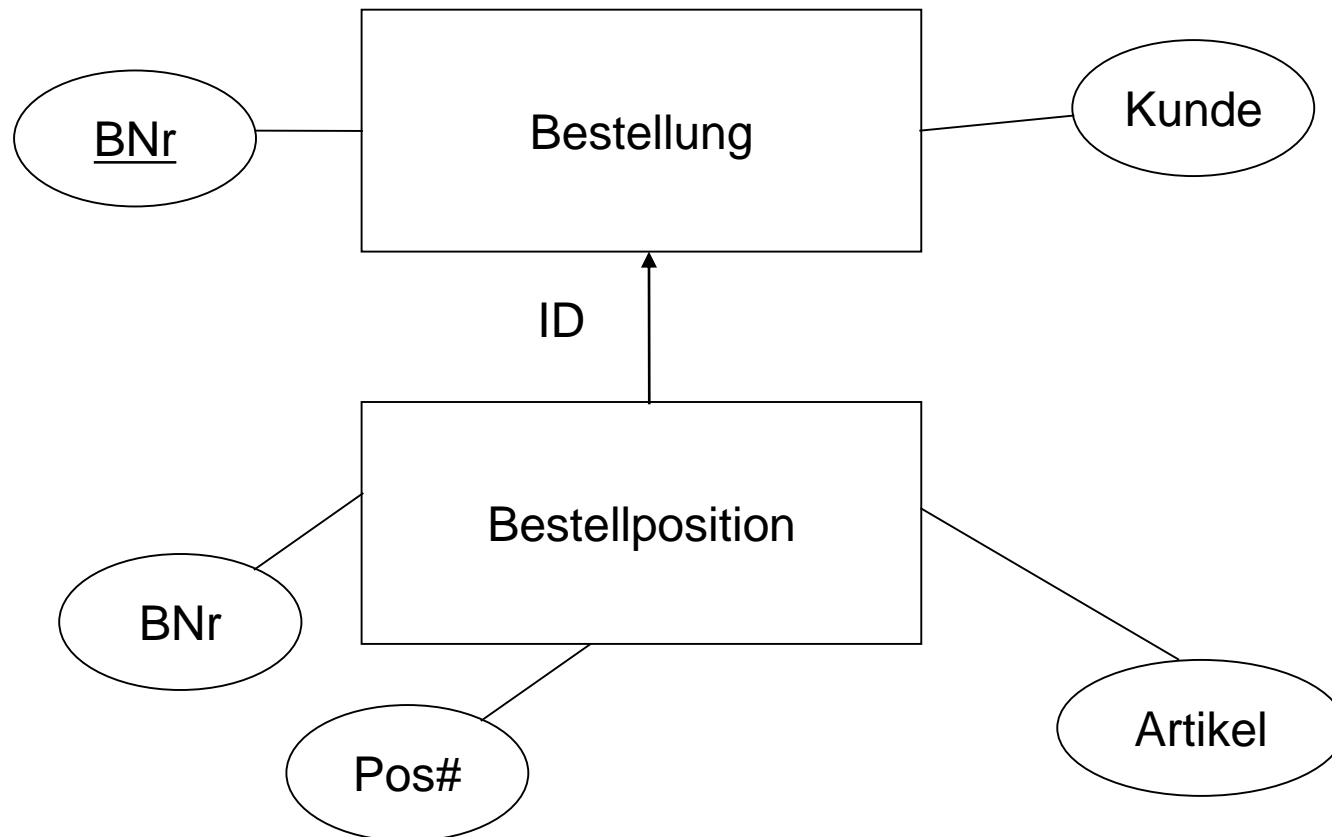
Begriff: ISA-abhängiger Entitätstyp



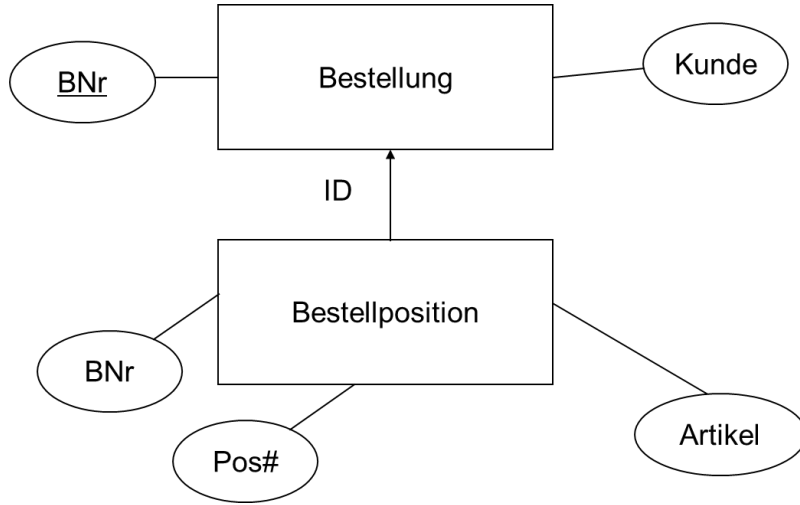
- Eine Generalisierung ist dann interessant wenn:
 - Die einzelnen Spezialisierungen sich deutlich voneinander unterscheiden (Kunde hat andere Attribute als Lieferant oder hängt mit anderen Beziehungstypen zusammen)
 - Gemeinsame Attribute vorhanden sind (im Beispiel das Attribut, "Adresse")

Begriff: ID-abhängiger Entitätstyp

- Bestellungen umfassen verschiedene Bestellpositionen:

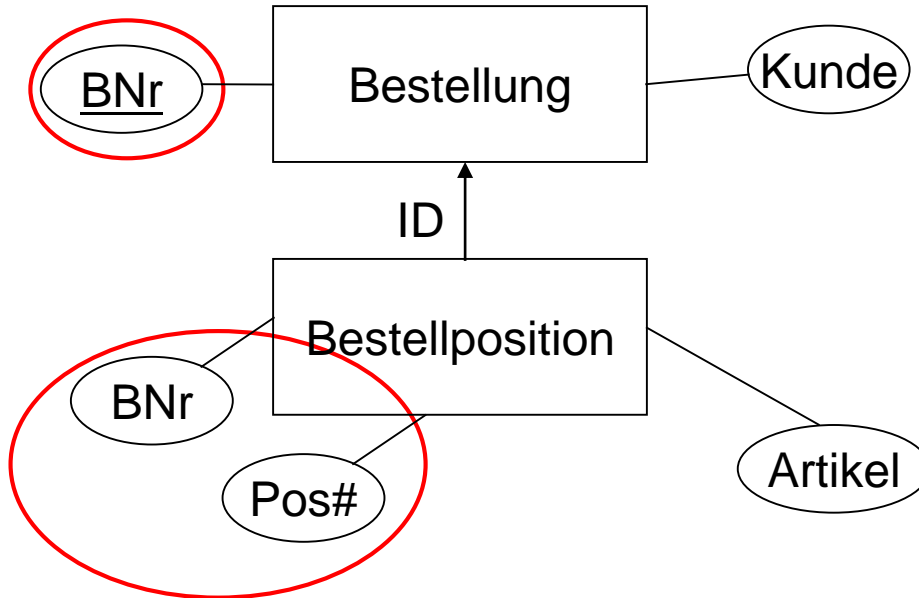


Begriff: ID-abhängiger Entitätstyp



- Der Entitätstyp «Bestellposition» ist **ID-abhängig** vom Entitätstyp «Bestellung».
- Zu einer Bestellung (Primärschlüssel {BNr}) gehört eine **Menge** von Bestellpositionen.

Begriff: ID-abhängiger Entitätstyp



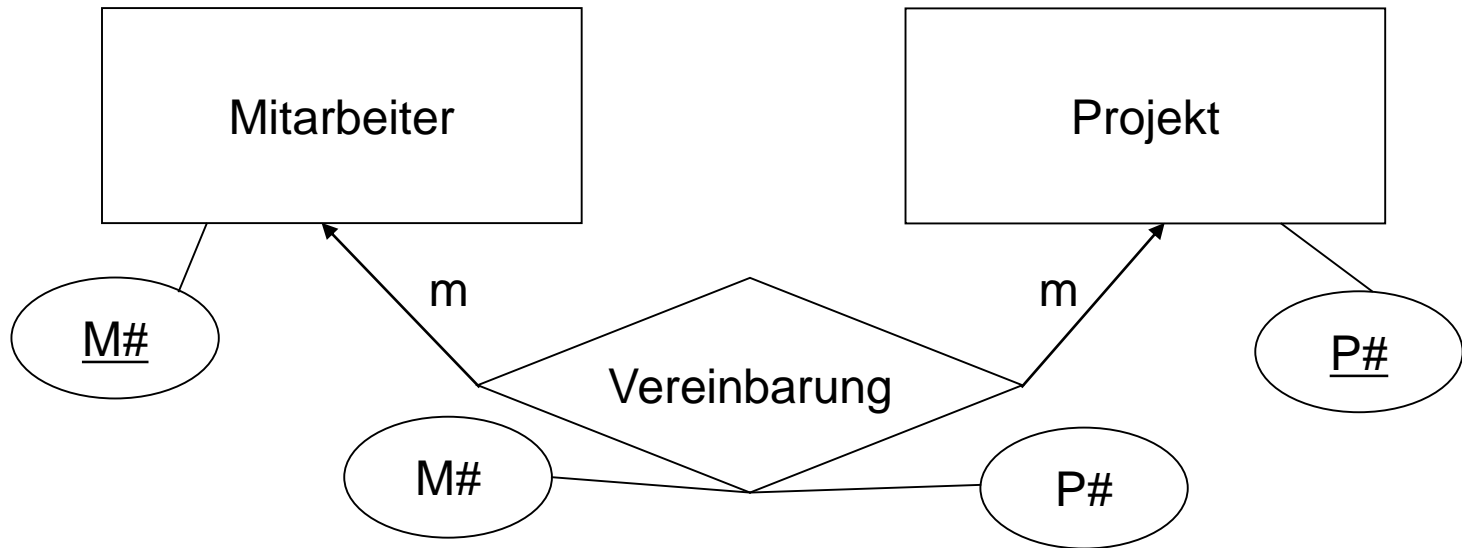
- «Innerhalb» einer Bestellung brauchen wir ein **weiteres Attribut**, z.B. «Pos#», um die einzelnen Bestellpositionen zu **unterscheiden**.
- Es bildet also z.B. {BNr, Pos#} einen Schlüssel.

ISA- versus ID-abhängiger Entitätstyp

- ISA-abhängig:
 - **Genauere Spezifikation** (Spezialisierung)
 - Ist Entitätstyp F von Entitätstyp E ISA-abhängig, so gilt: Ist M die Menge der Primärschlüsselattribute in E, so muss M in F ein Schlüssel sein.
- ID-abhängig:
 - **Hierarchie**, hängt an "Oberklasse". Ist nur innerhalb der Hierarchie definiert
 - Ist Entitätstyp F von Entitätstyp E ID-abhängig, so gilt: Ist M die Menge der Primärschlüsselattribute in E, so muss $M \cup N$ ein Schlüssel in F sein, wobei N eine zu M elementfremde Menge von Attributen von F ist (min. 1 Element)

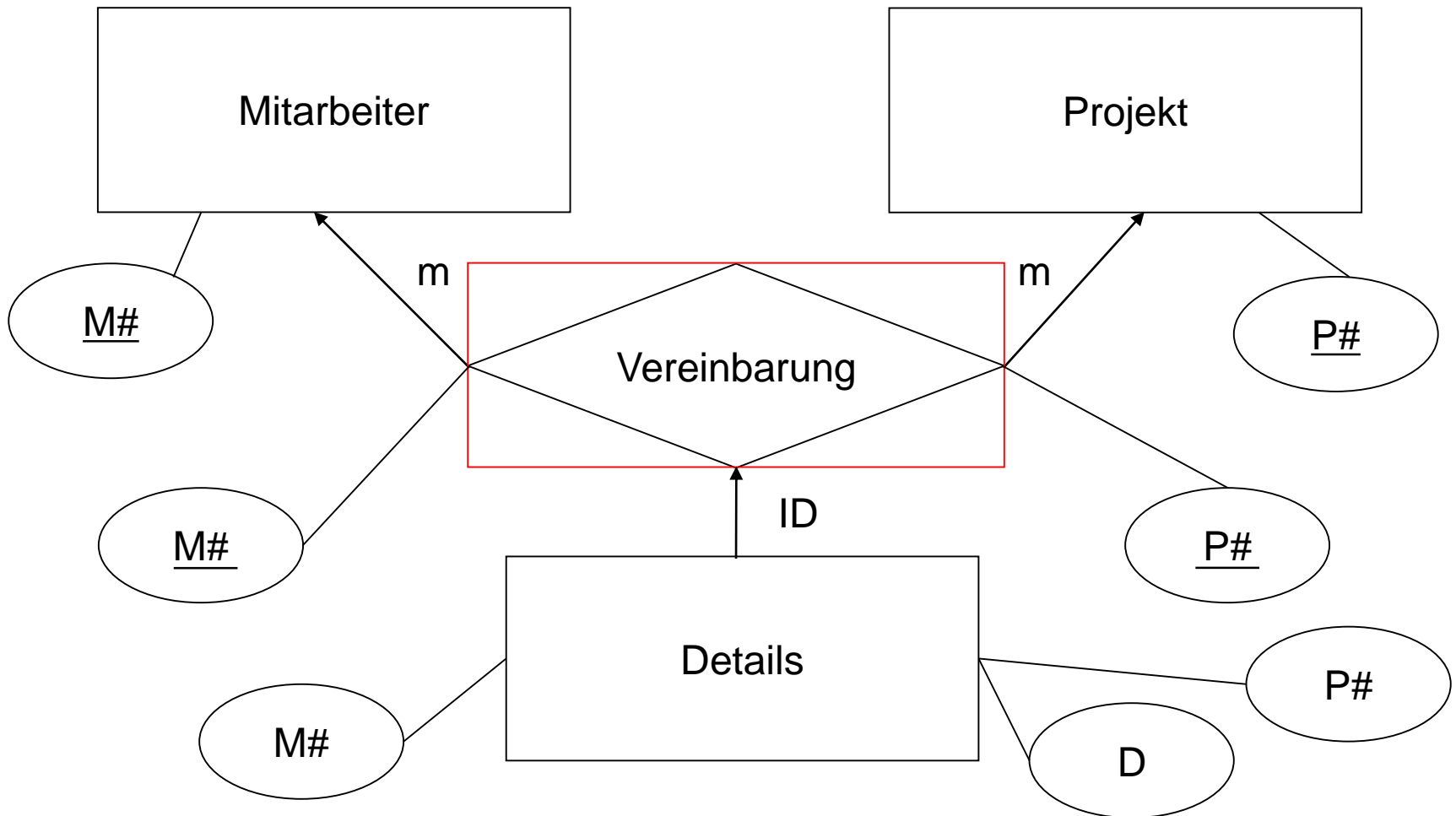
Zusammengesetzter Entitätstyp

- Wir behandeln noch einen letzten Sonderfall, der auftritt, wenn wir an Beziehungstypen Entitätstypen **anhängen** wollen.
- Ausgangslage:



- Wir wollen unterschiedlich viele (auch keine) Details zu jeder Vereinbarung festhalten können

Zusammengesetzter Entitätstyp



Zusammengesetzter Entitätstyp

- Wir hängen einen ID-abhängigen Entitätstyp "Details" an, der von Vereinbarung zu Vereinbarung unterschiedlich viele Details aufnehmen soll.
- Der Beziehungstyp wird "umgewandelt" in einen **zusammengesetzten** Entitätstypen (zeichnerisch: wir zeichnen ein Rechteck um den Beziehungstypen).
- Es sei $\{M\#,P\# \}$ Primärschlüssel in Vereinbarung.
- Es ist mindestens ein weiteres Attribut nötig. So kann z.B. $\{M\#,P\#,D\}$ ein **Schlüssel** sein (wegen ID-Abhängigkeit).

Lernziele Lektion 7

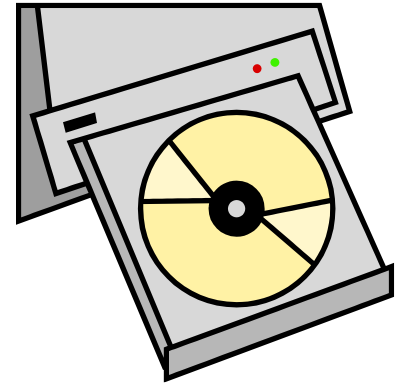
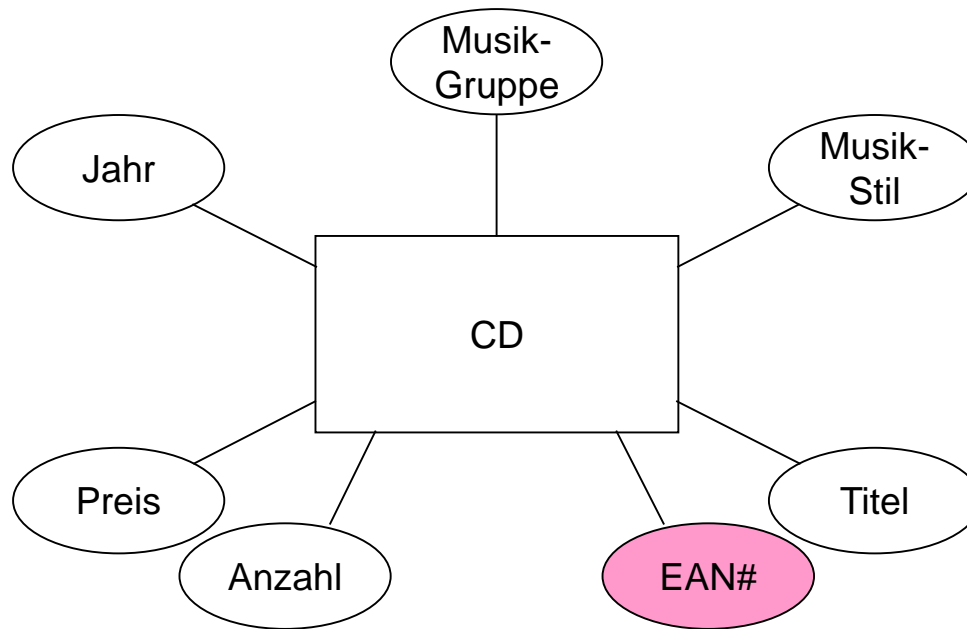
- Notation anwenden an einem zusammenhängenden Beispiel
- Nehmen Sie ein Blatt Papier und zeichnen Sie mit (zumindest am Anfang)

Ein komplexeres Beispiel

- Es soll ein Design für eine Datenbank für einen CD-Shop erstellt werden.
- Der Shop verkauft CDs. Eine CD ist dabei ein Titel, nicht eine einzelne Scheibe.
- Attribute:
 - Jede CD ist durch eine eindeutige Artikelnummer ("EAN#") identifiziert
 - Jede CD hat Informationen über die betreffende Musikgruppe, den Titel, den Musikstil und das Erscheinungsjahr zugeordnet.
 - Jede CD hat Informationen über den aktuellen Preis und die Anzahl Exemplare an Lager zugeordnet

Entitätstyp "CD"

- Entitätstyp "CD":

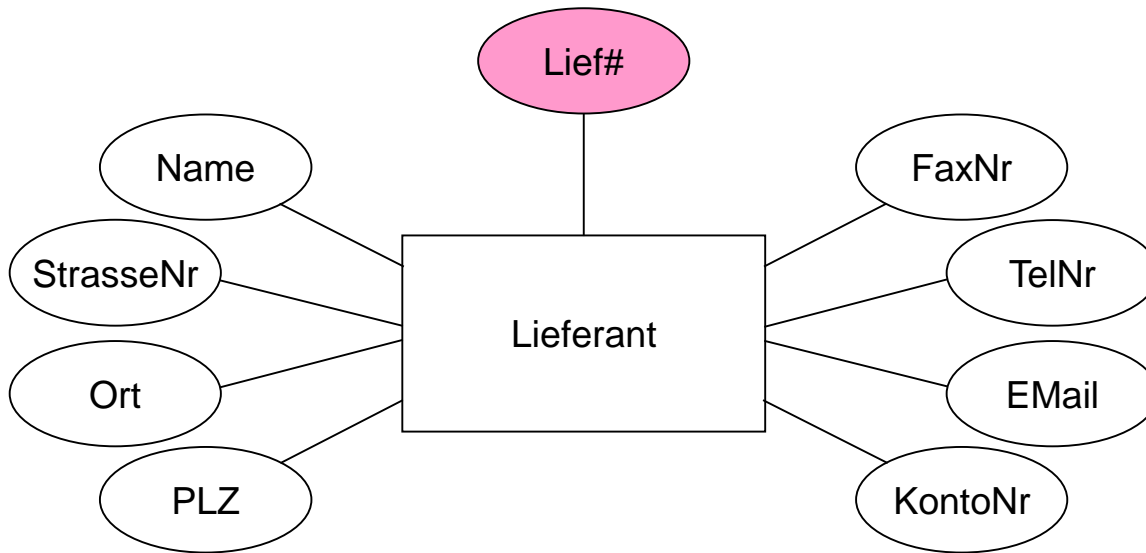


Entitätstyp "Lieferant"

- Wir betrachten nun Lieferanten von CDs
- Ein Lieferant ist eine Firma, keine Person
- Attribute:
 - Es wird eine **"künstliche"**, eindeutige Lieferantenummer, ein (Firmen-)name, eine Strasse, ein Ort, eine PLZ, sowie Telefon-/Faxnummer, Email und Kontonummer verwaltet.
- Bemerkung: Solche identifizierenden "künstlichen" Attribute werden Surrogatschlüssel ("künstliche Schlüssel") genannt.

Entitätstyp "Lieferant"

- Entitätstyp "Lieferant":

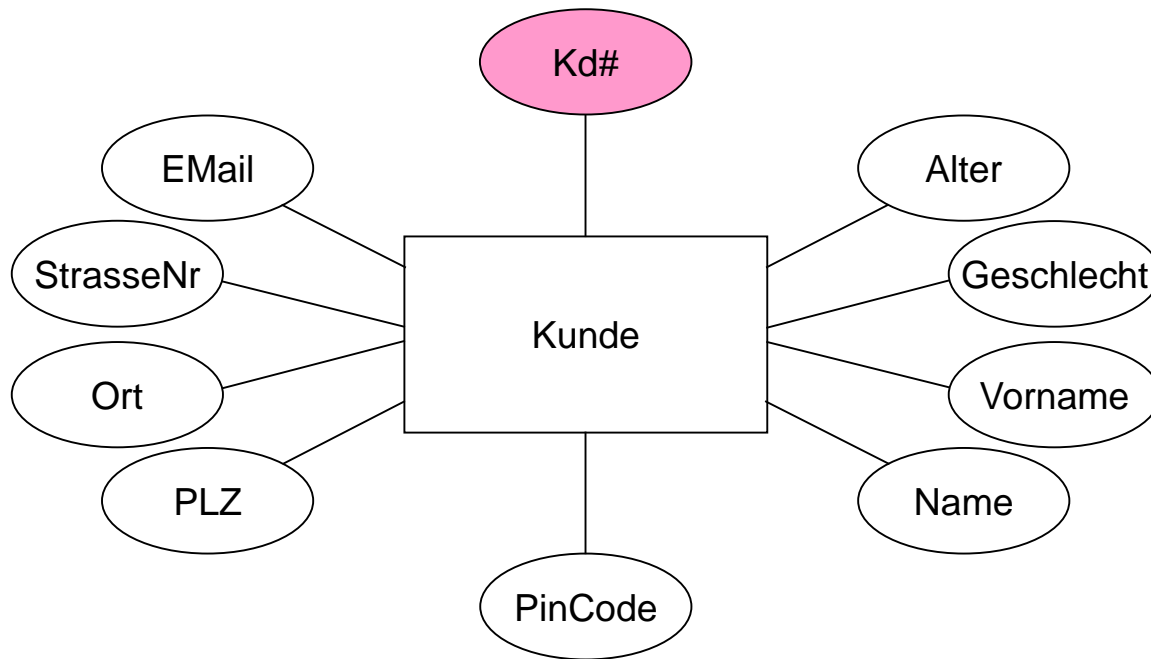


Entitätstyp "Kunde"

- Unser Shop hat hoffentlich auch Kunden
- Attribute:
 1. Jedem Kunden/jeder Kundin wird eine eindeutige, "künstliche" Kundennummer zugeordnet
 2. Es werden Name, Vorname, Geschlecht und Alter der Person verzeichnet
 3. Adressdaten werden ebenfalls erfasst: Strasse, Ort, PLZ, sowie Email
 4. Kunden haben einen Pin-Code für Online-Bestellungen

Entitätstyp "Kunde"

- Entitätstyp "Kunde":



Entitätstyp "Mitarbeiter"

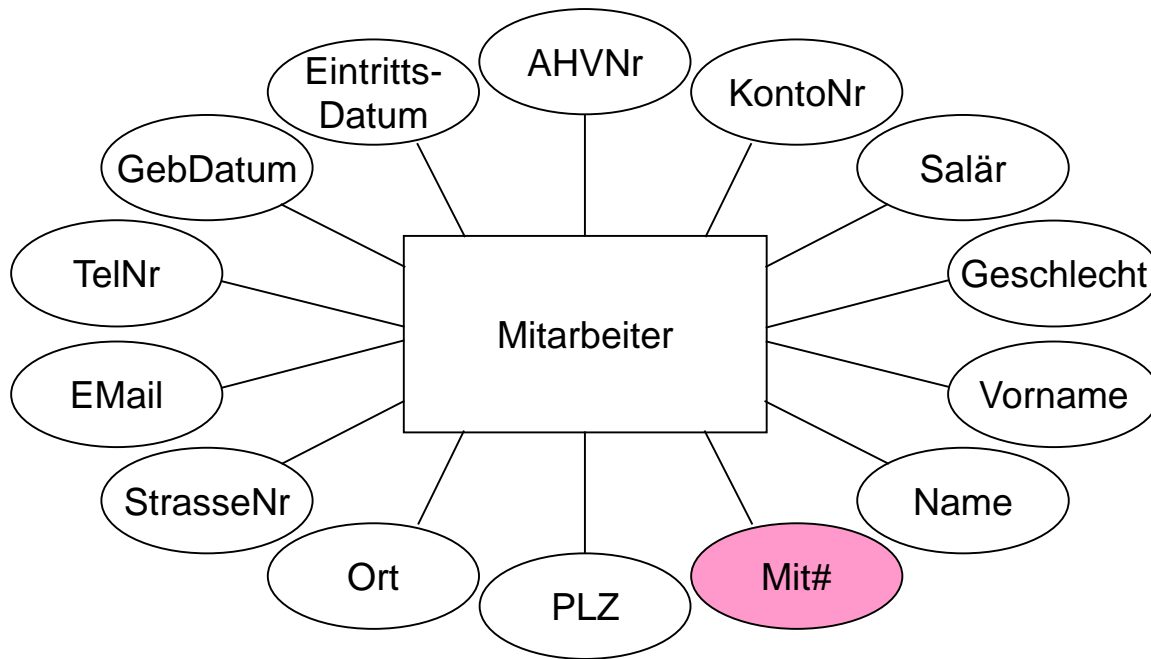
Kein Shop kann ohne Mitarbeiter auskommen

Attribute:

1. Jedem Mitarbeiter/jeder Mitarbeiterin wird eine eindeutige, "künstliche" Mitarbeiternummer zugeordnet.
2. Es werden Name, Vorname, und Geschlecht der Person verzeichnet
3. Adressdaten werden ebenfalls erfasst: Strasse, Ort, PLZ, Email, sowie Telefonnummer
4. Verschiedene weitere Daten helfen bei der Personaladministration: AHV-Nummer, Konto-Nummer, Salär, Eintrittsdatum, Geburtsdatum

Entitätstyp "Mitarbeiter"

- Entitätstyp "Mitarbeiter":



Entitätstyp "Mitarbeiter"

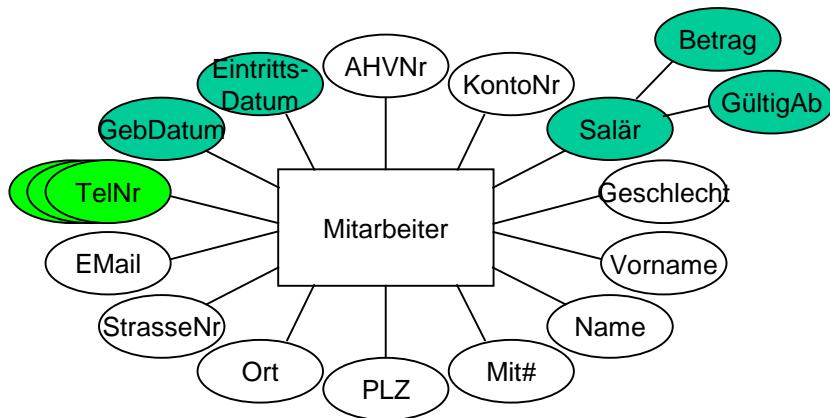
- Ganz sind wir noch nicht zufrieden mit unserem Entitätstyp "Mitarbeiter":
 1. Wir wollen für das **Salär** vermerken, ab wann es gültig ist
 2. Wir wollen **mehrere Telefonnummern** für einen Mitarbeiter verwalten
 3. Wir wollen die **Geburtstage** und **Eintrittstage** unabhängig vom Jahr abfragen können (Zwischenfrage: was ermöglicht uns das?)

Entitätstyp "Mitarbeiter" (verfeinert)

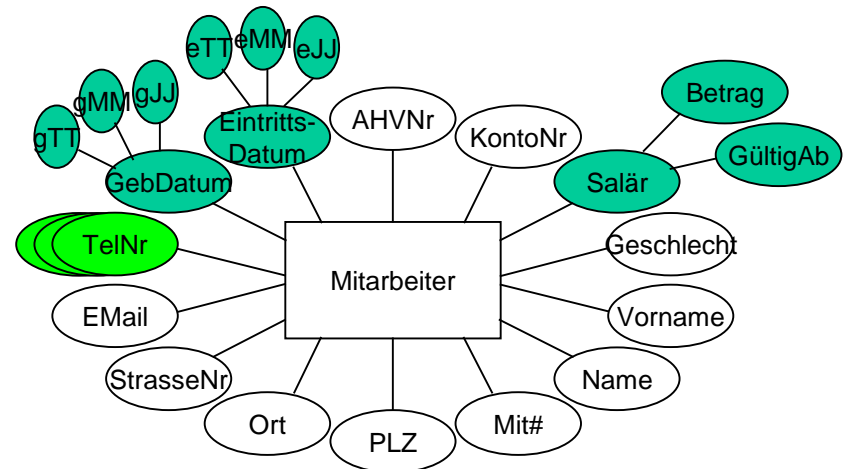
"Lösung": **zusammengesetzte Attribute** resp. **mehrwertige Attribute**

Die meisten Datenbanken haben Datumstyp, aber oft proprietär. Dies wäre hier eine Alternative

Mit Datumstyp

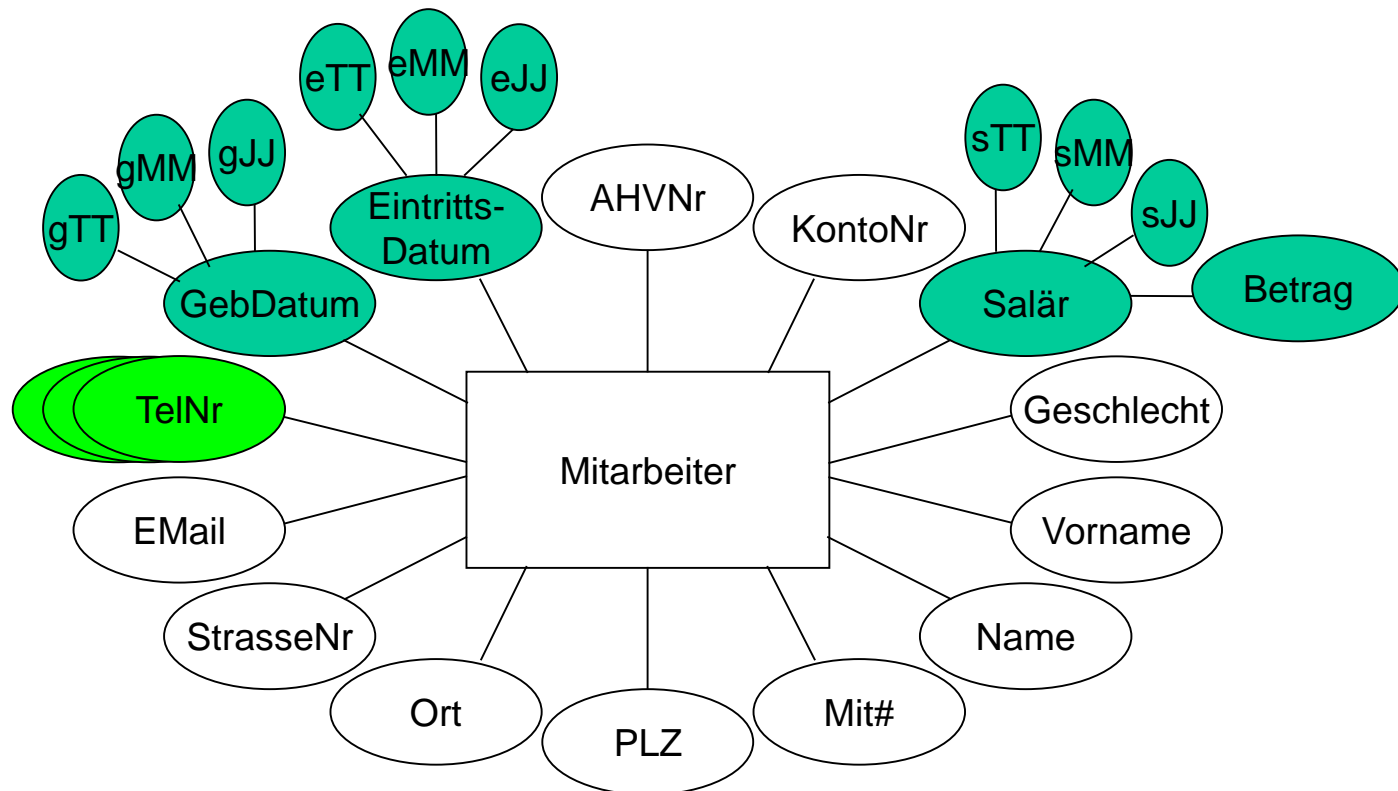


Ohne Datumstyp



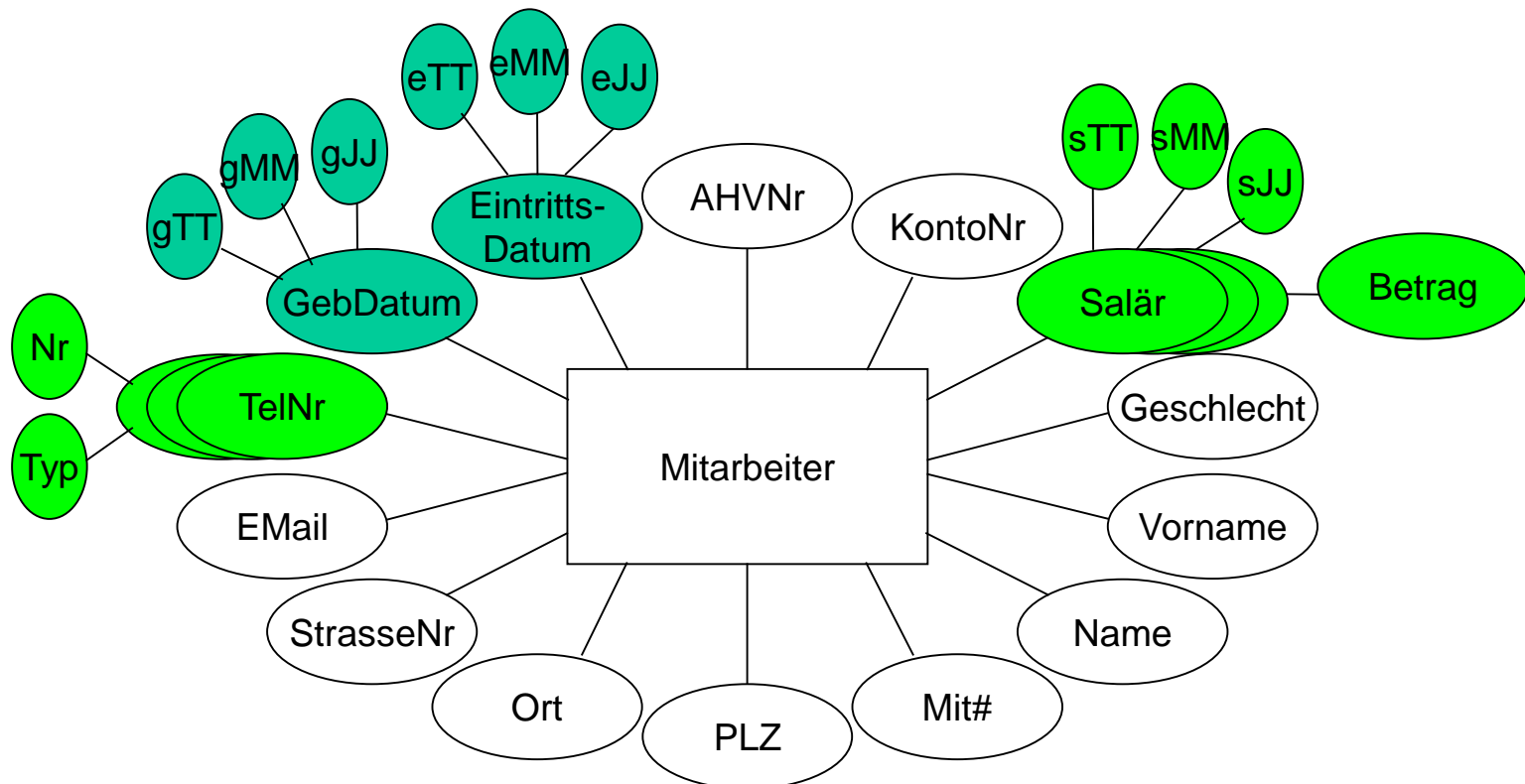
Entitätstyp "Mitarbeiter" (verfeinert)

- Wir können jetzt Mitarbeiter verfeinern. Der Gültigkeitswert beim Salär ist ein Datum und wird zerlegt.



Entitätstyp "Mitarbeiter" (verfeinert)

- Wir möchten nun noch die **Salärgeschichte** festhalten. Wir benötigen ein "mehrwertiges, zusammengesetztes" Attribut.
- Wir wollen den jeweiligen **Typ** der Telefonnummer vermerken

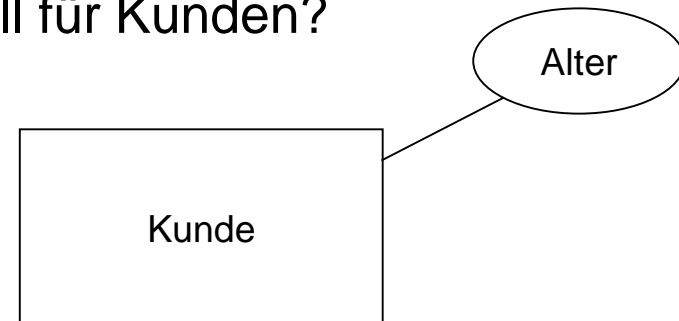


Die Krux mit dem Alter

- Wir haben für Mitarbeiter die Datumsangaben "zerlegt".
- Manche Datenbanken haben **Datumsfunktionen** aber oft sind diese oder das **Datumsformat proprietär**
- Indem wir zerlegen, lagern wir diese Problematik in ein evtl. **Auswertungsprogramm** aus
- Wir können z.B. auch Geburtstaglisten generieren etc.



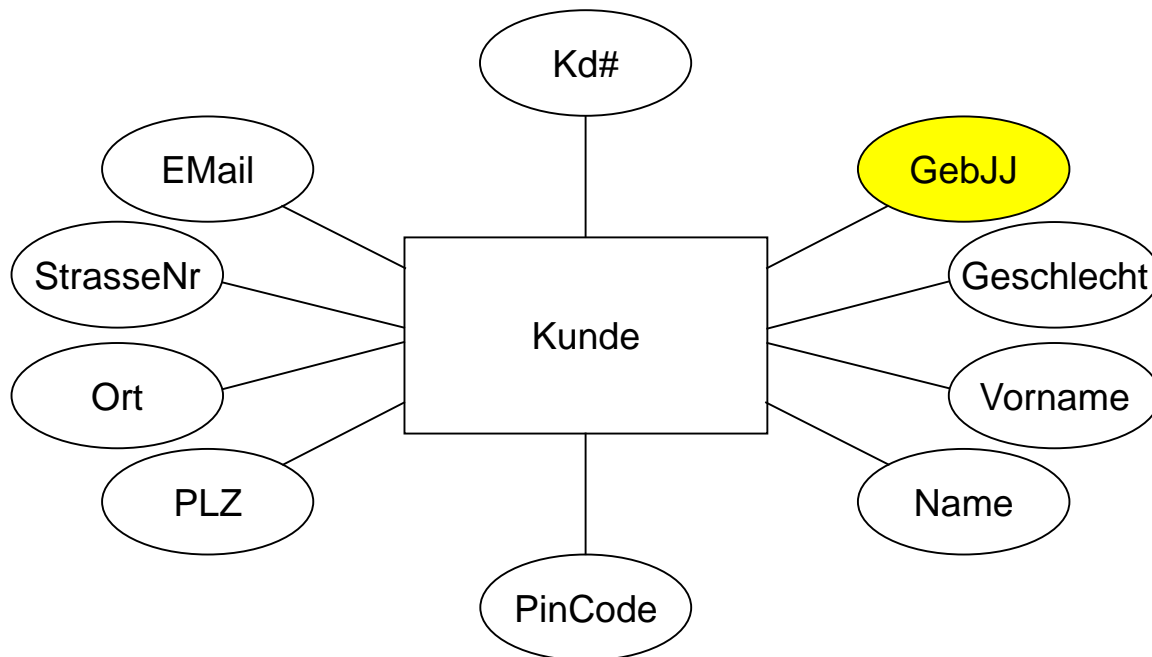
? Da wir gerade Datumsüberlegungen anstellen, die folgende Frage: ist die Speicherung eines "Alters" sinnvoll für Kunden?



Entitätstyp "Kunde" (verfeinert)

Entitätstyp "Kunde" (verfeinert):

Das Geburtsjahr ist **langlebige** Information!

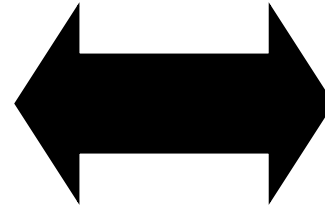
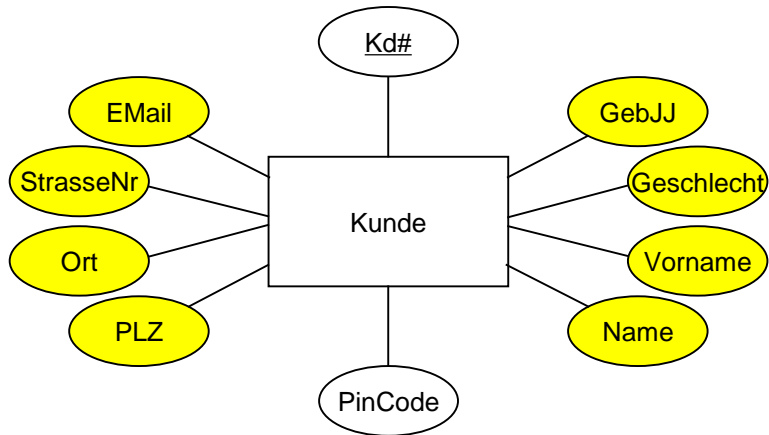


Vergleich Kunde-Mitarbeiter

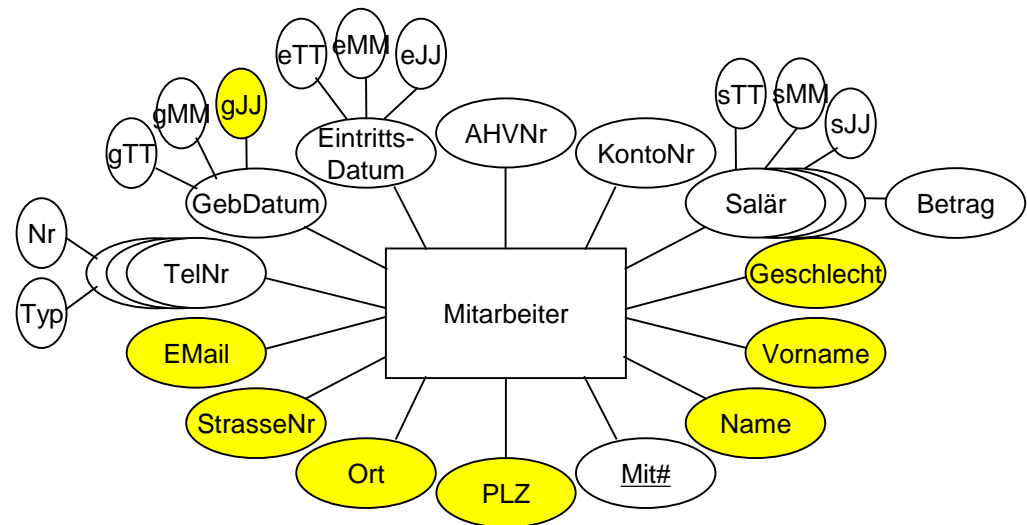
? Was lässt sich zum Vergleich der beiden Entitätstypen "Kunde" und "Mitarbeiter" sagen?

Kunden und Mitarbeiter – ein Vergleich

- Wir wollen die beiden Entitätstypen Kunden und Mitarbeiter **vergleichen**

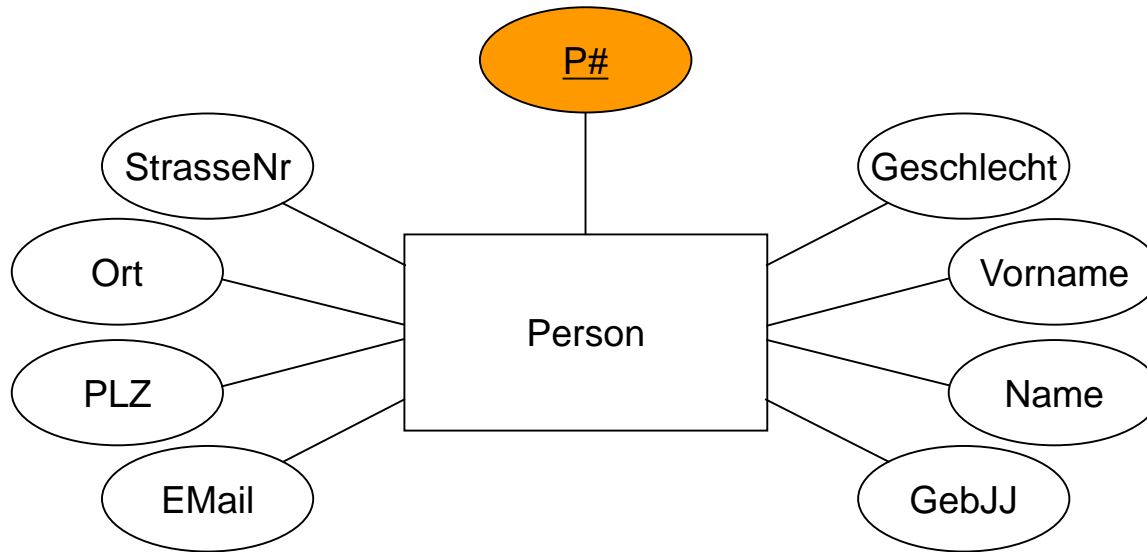
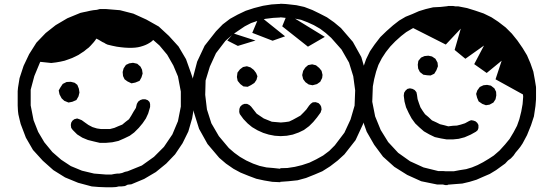


- Es bestehen etliche **Gemeinsamkeiten**



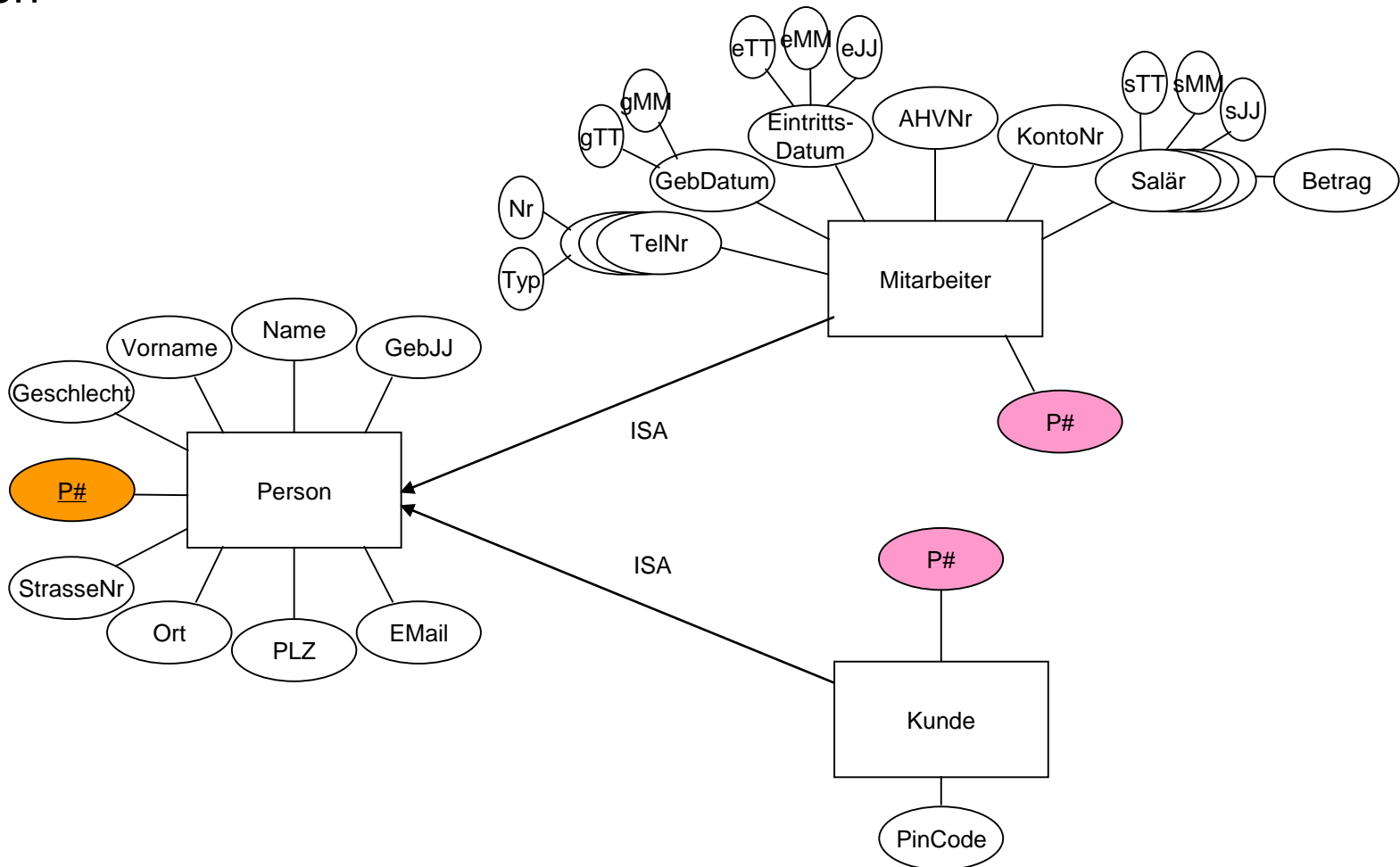
Entitätstyp "Person"

- Wir **fassen** die Gemeinsamkeiten in einem neuen Entitätstyp **zusammen** (Person):



Generalisierung

- Die Entitätstypen "Kunde" und "Mitarbeiter" werden "Subklassen" von Person



Konsequenzen

Konsequenzen:

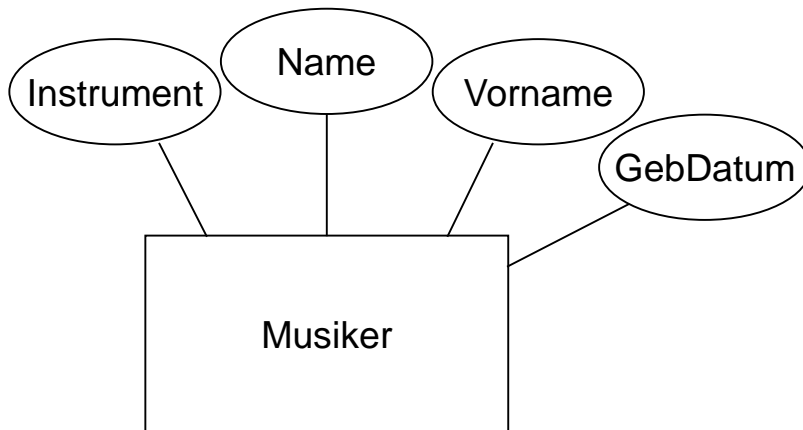
1. Kunden und Mitarbeiter werden als **Spezialfälle** von Personen behandelt.
2. Die **gemeinsamen Attribute** sind nun der **Superklasse** "Person" zugeordnet, und in den Subklassen nicht mehr aufgeführt
3. Die Schlüssel der Superklasse werden "**geerbt**".

Weitere Details

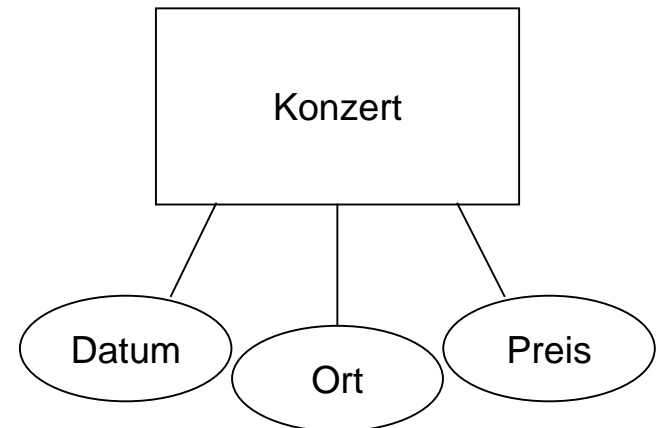
? Es entstehen Begehrlichkeiten, weitere Details im Umfeld der CDs festzuhalten: z.B. Informationen über Musiker und Konzerte. Ihr Kommentar?

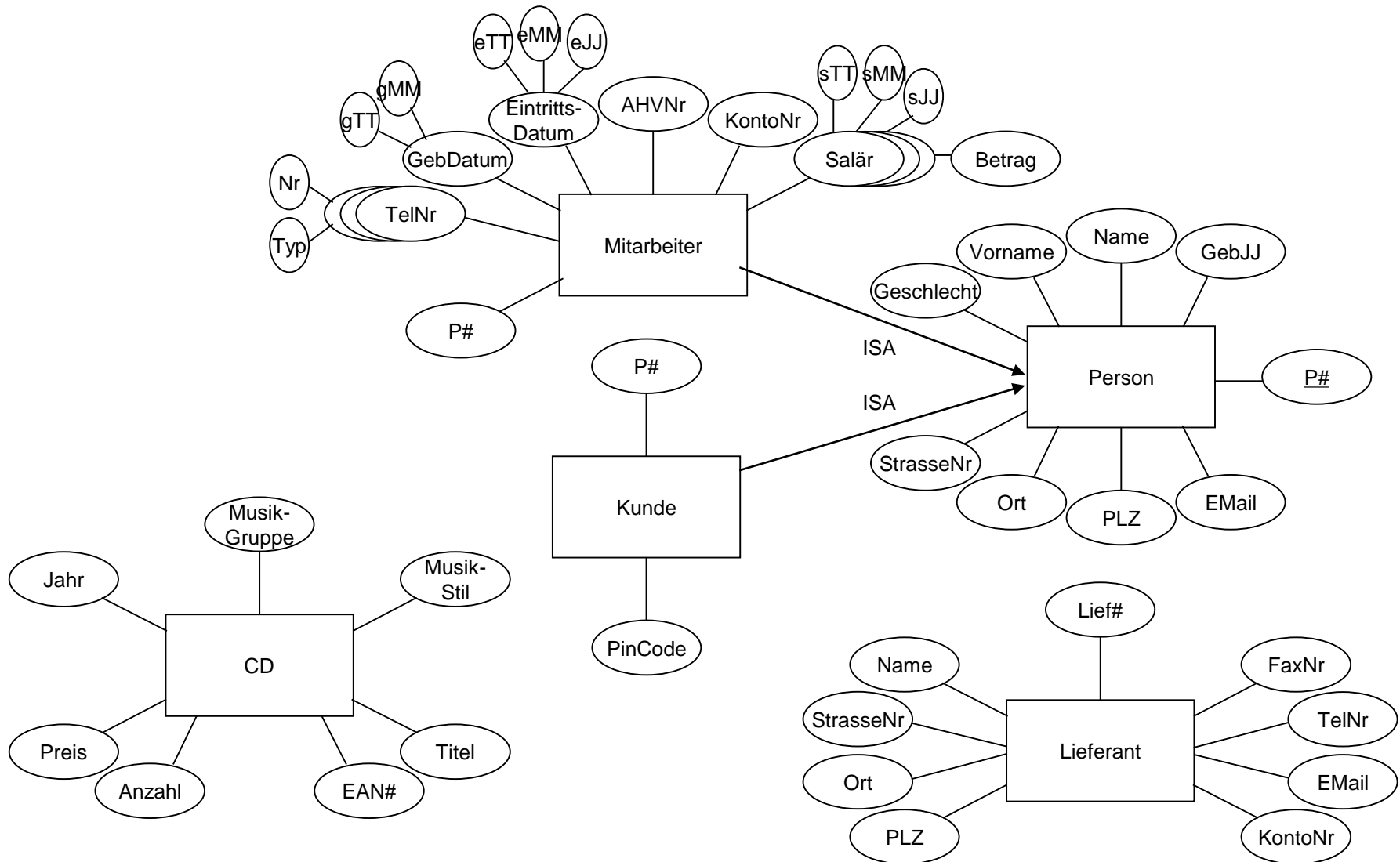
Systemgrenzen

- Systemgrenzen:
- Wir könnten zu den Musikgruppen **weitere Details** festhalten: Künstler, ihre Konzerte etc.



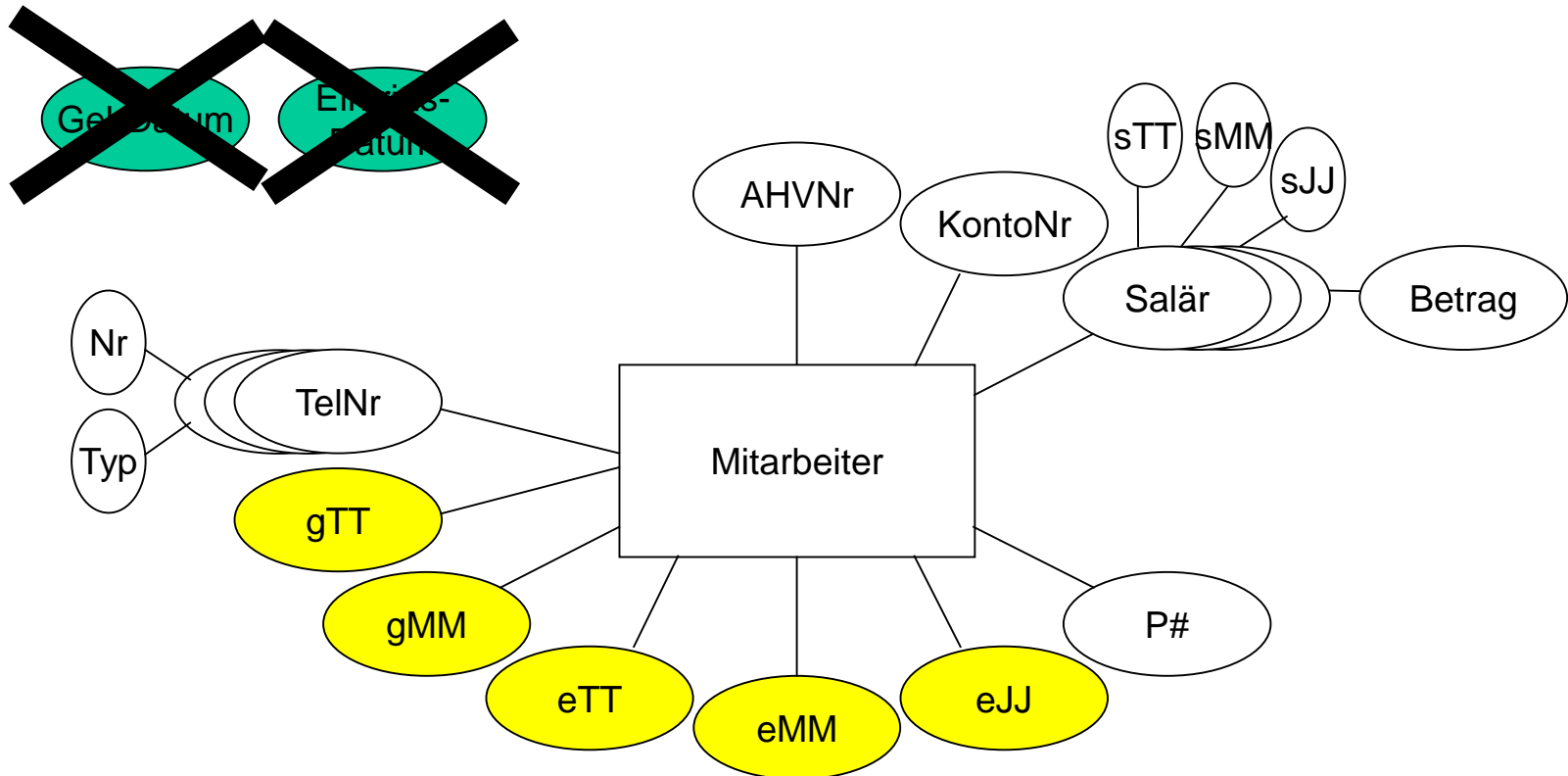
- Wir **benötigen** diese Angaben **nicht** für den Shop. Wir grenzen das "System" ab.
- Wir bleiben beim Attribut "Musikgruppe"





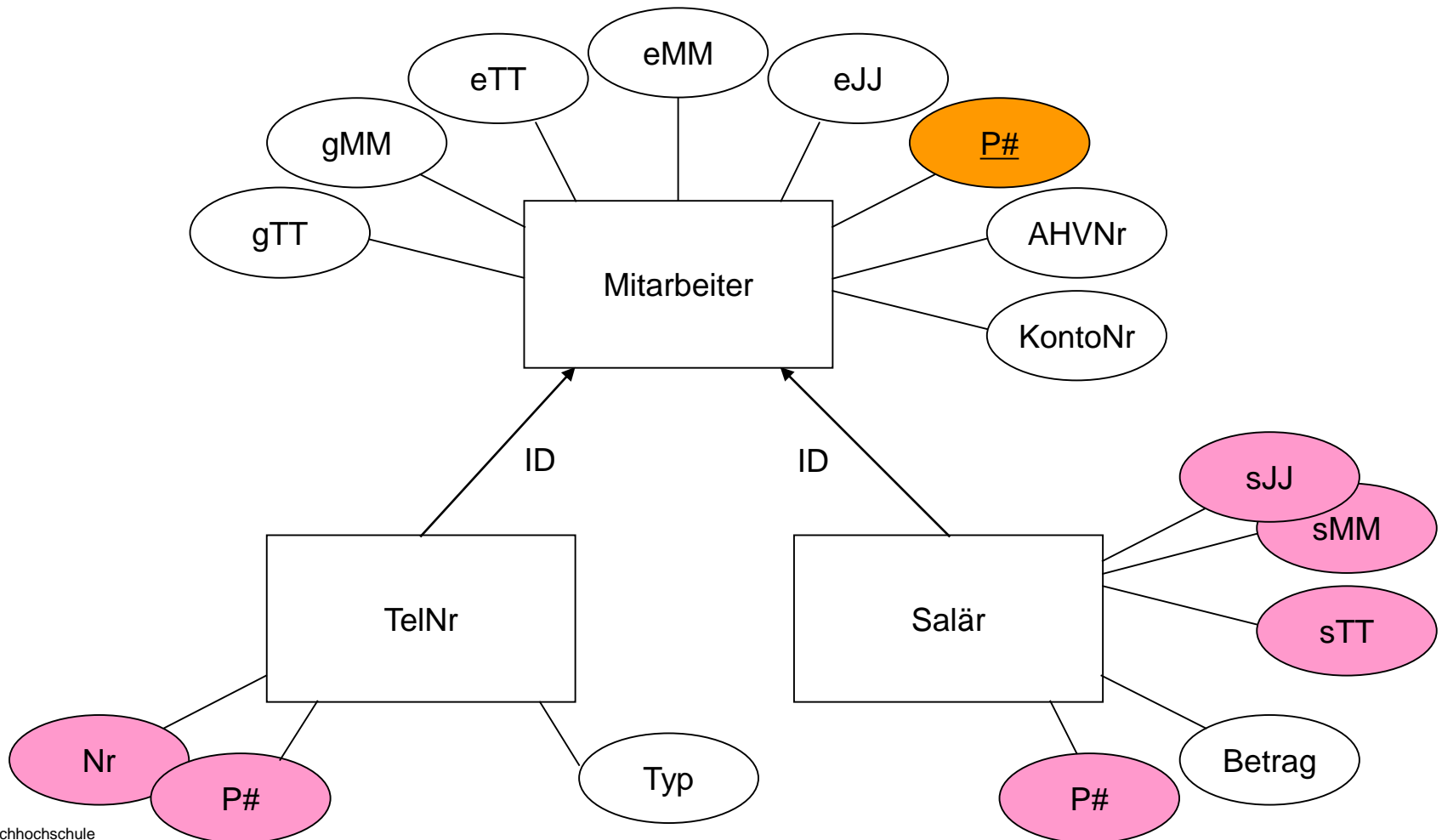
Umwandeln einf. zusammenges. Attr.

- Oberbegriff (=Platzhalter) **verschwindet**. Details werden zu **normalen Attributen**



Umwandeln komplexer Attribute

Komplexe Attribute führen zu ID-abhängigen Entitätstypen

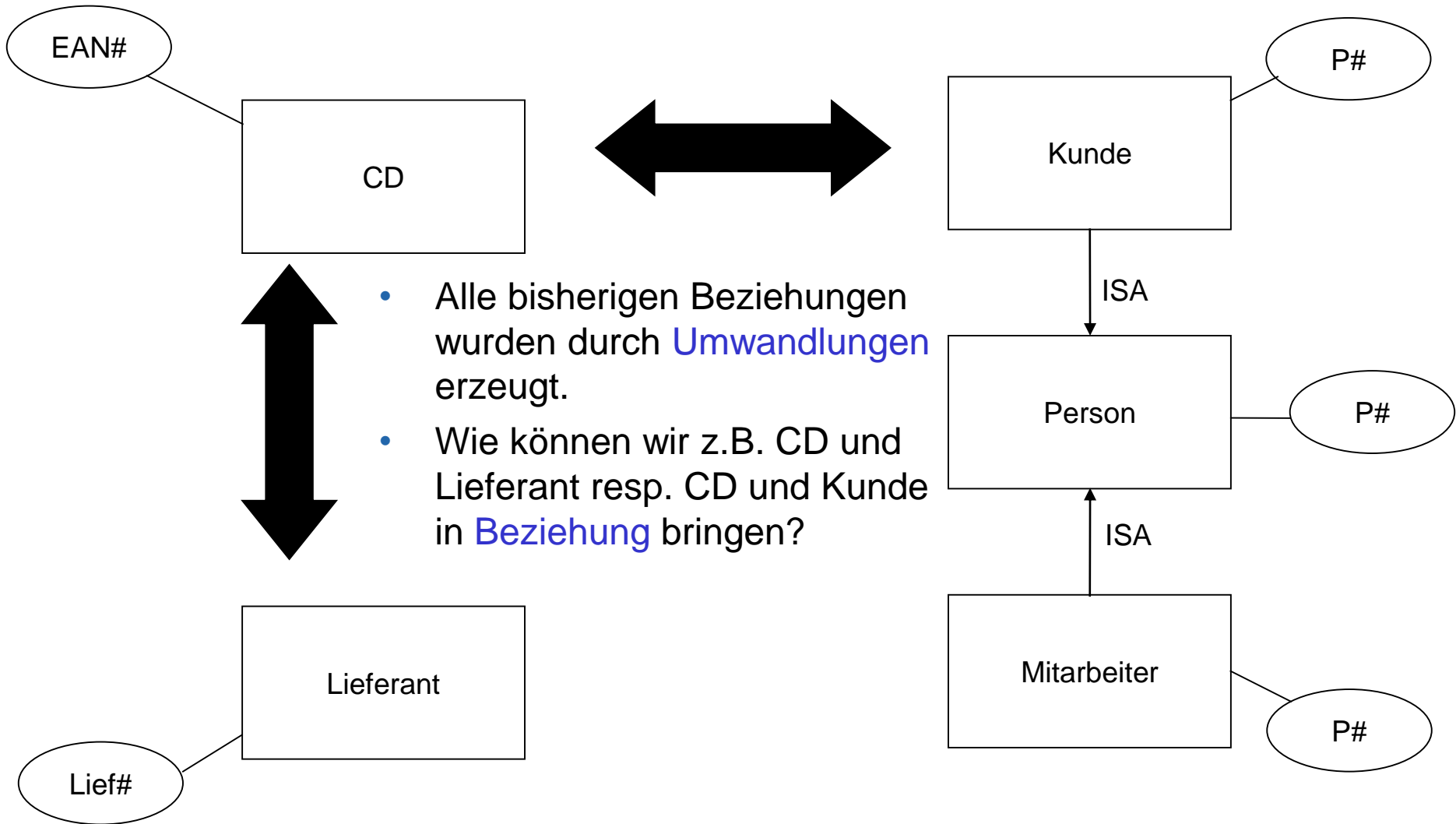


Konsequenzen

Komplexe Attribute führen zu **ID-abhängigen** Entitätstypen

1. Der Entitätstyp "Mitarbeiter" bekommt einen **Primärschlüssel**
2. Es sind jeweils mehrere Entitäten "TelNr" und "Salär" pro Mitarbeiter möglich. Daher reicht {P#} nicht als Schlüssel. Wir brauchen jeweils ein/mehrere **differenzierende(s)** Attribut(e)
3. Schlüssel in TelNr: {P#, TelNr}
4. Schlüssel in Salär: {P#, sTT, sMM, sJJ}
5. Ein Mitarbeiter kann auch **kein** Telefon haben.
6. Eine Telefonnummer kann aber nicht ohne Mitarbeiter "leben", d.h., TelNr ist **existenzabhängig** von Mitarbeiter

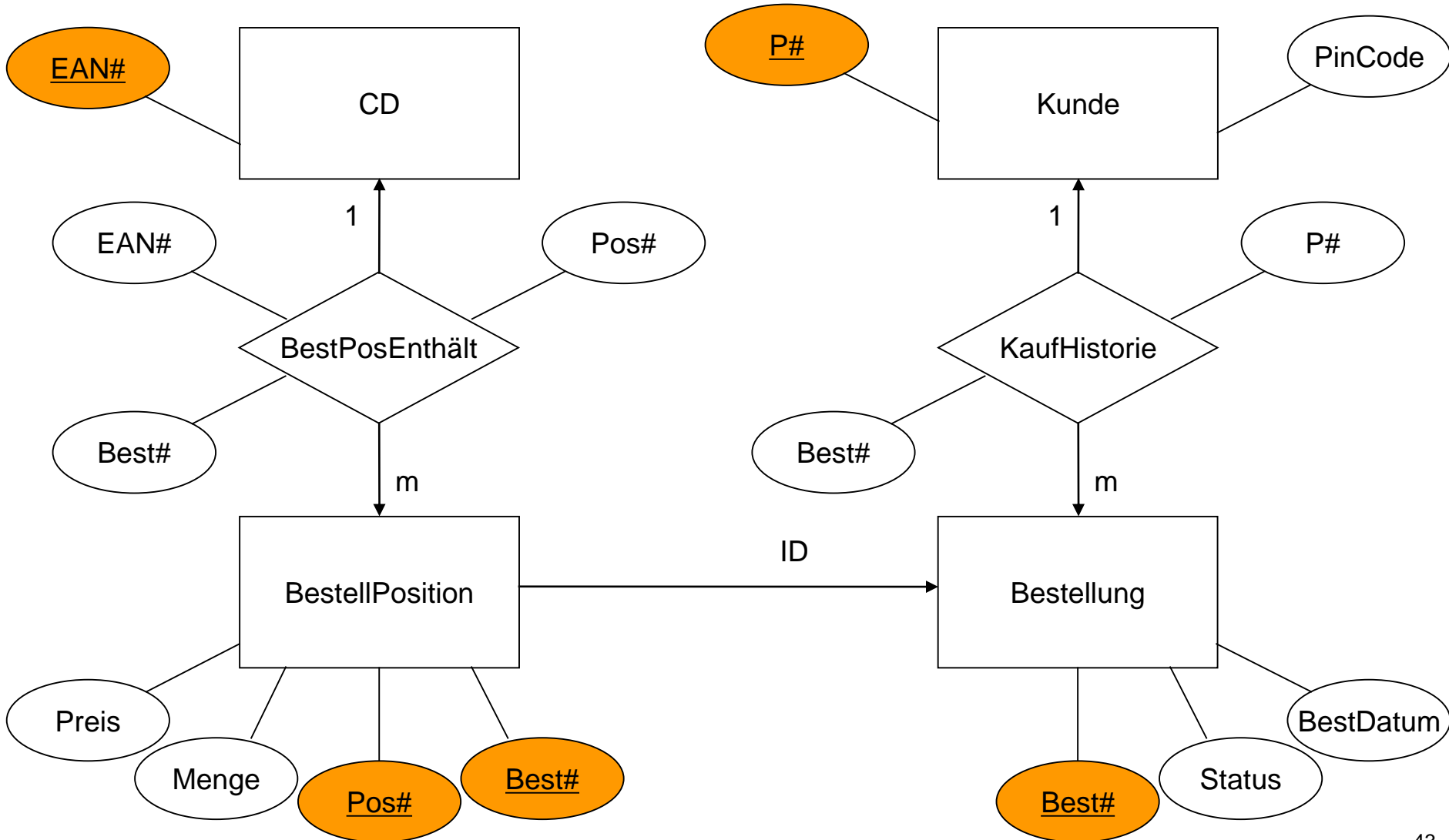
Beziehungen



Beziehungen

- Wir modellieren folgende **Beziehungen**
- Kunde kauft CD
- Lieferant hat CD geliefert
- Lieferant hat CD im Sortiment (aber vl. (noch) nicht geliefert)
- Weitere Überlegungen zum Thema "Kunde kauft CD":
 - wann
 - welche
 - wieviele?
 - (Achtung: CD ist Titel, nicht Scheibe!)
- Ein Kauf wird zur "Bestellung" (Warenkorb) mit mehreren Bestellpositionen (mehrere CD's im Warenkorb)

Beziehungen

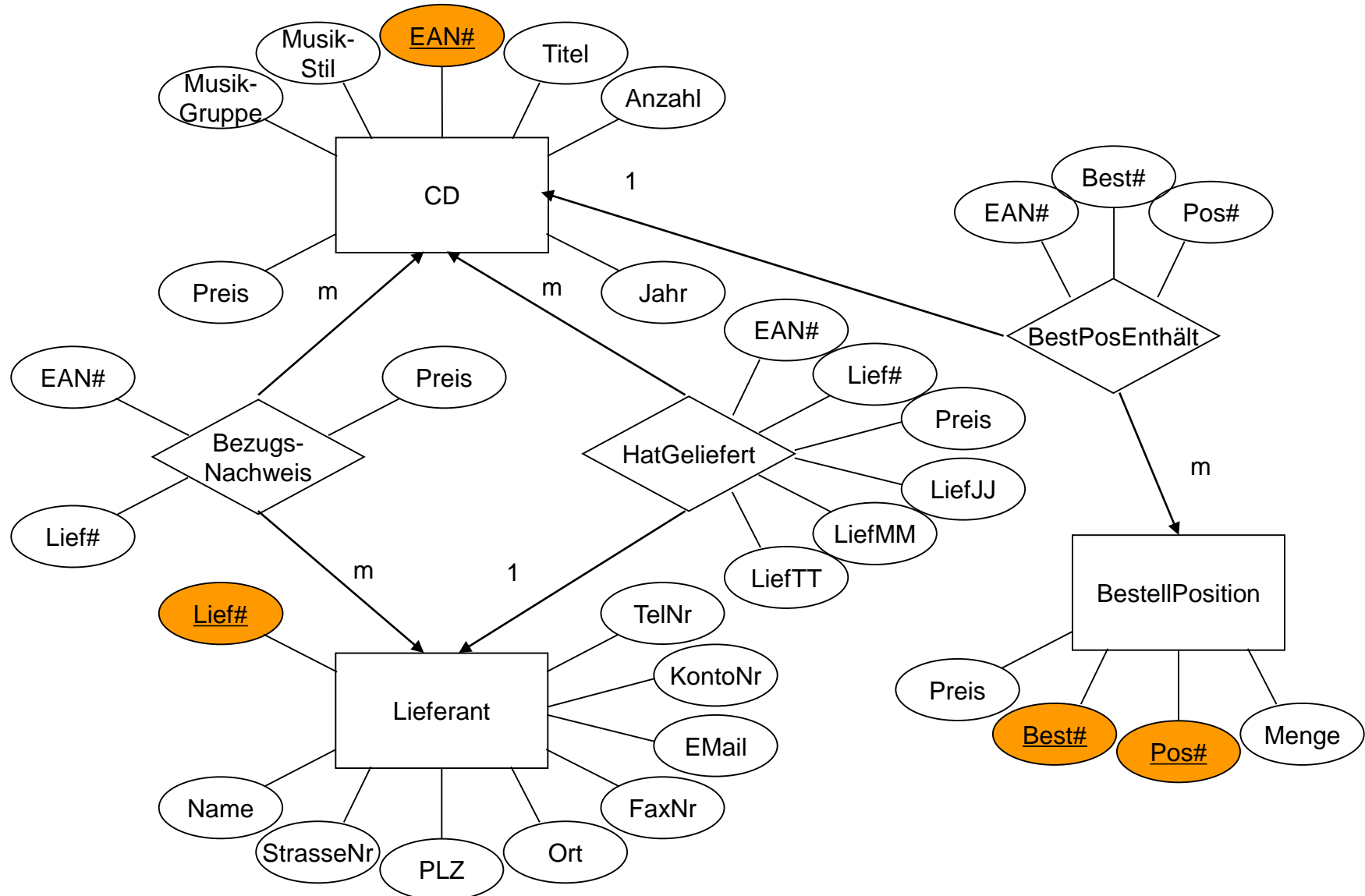


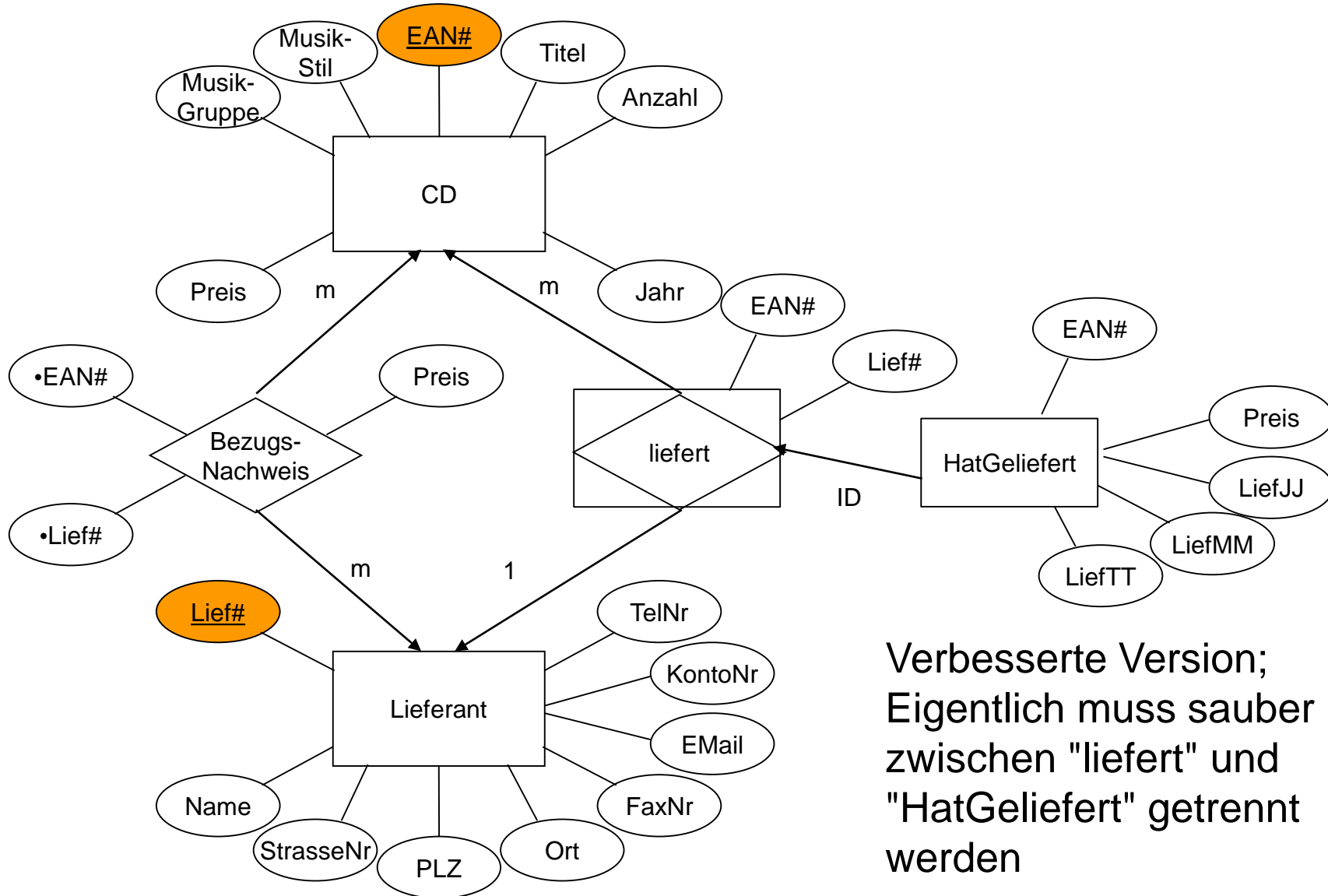
Konsequenzen

- Wir führen zwei **neue Entitätstypen** "Bestellung" und "Bestellposition" ein
 - Wir führen **zwei Beziehungstypen** "BestPosEnthält" und "KaufHistorie" ein
1. Beziehungstypen werden dann verwendet, wenn eine Beziehung dargestellt werden soll, aber **keine existentielle Abhängigkeit** zwischen den Entitätstypen besteht.
 2. Pro Bestellposition gibt es höchstens eine CD. Eine CD kann in mehreren Bestellpositionen vorkommen
 3. Es sind mehrere Bestellpositionen pro Bestellung möglich (ID-Beziehung: Position ohne Bestellung ist unbestimmt)
 4. Beziehungstyp "KaufHistorie" könnte auch als ID-Beziehungstyp dargestellt werden (Bestellung muss Kunden gehören)

Beziehungen

- Wir modellieren nun die **Beziehung** zwischen CD und Lieferant
- Wie bereits ausgeführt, haben wir zwei **Grundüberlegungen**:
 1. Lieferant hat CD geliefert
 2. Lieferant hat CD im Sortiment (aber vielleicht (noch) nicht geliefert)





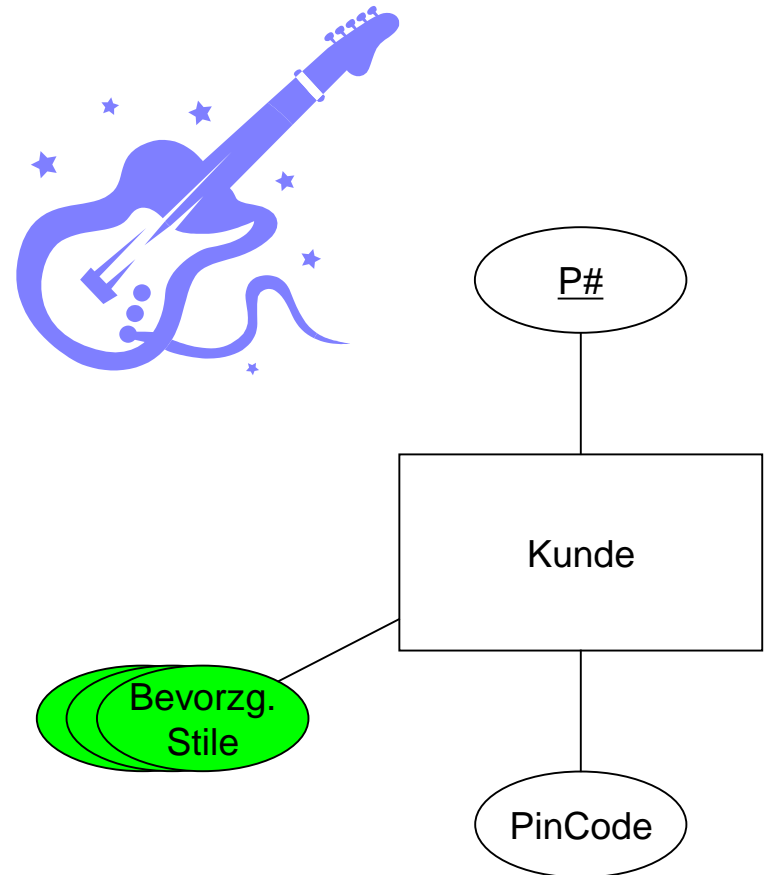
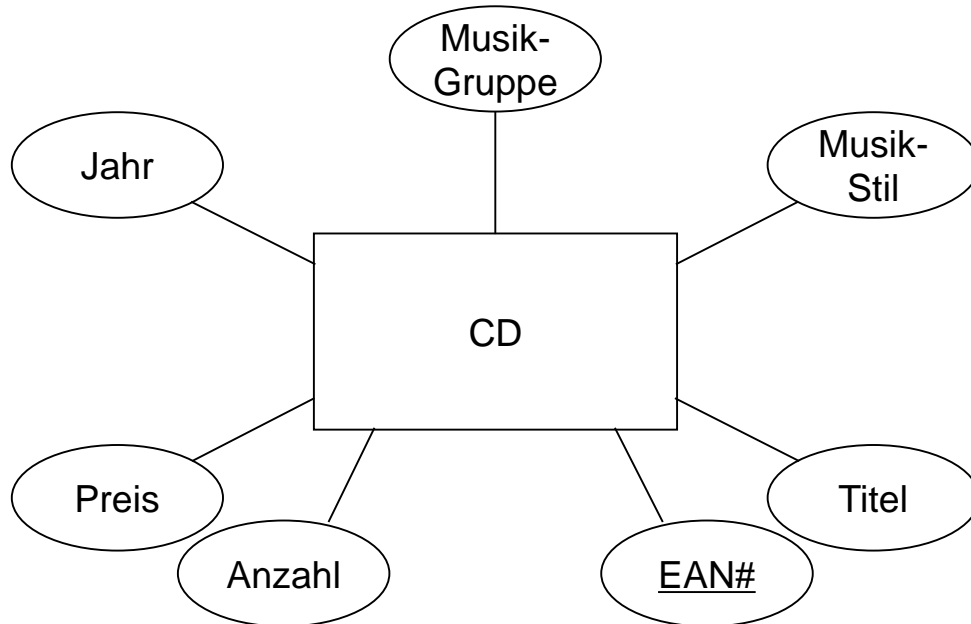
Verbesserte Version;
Eigentlich muss sauber
zwischen "liefert" und
"HatGeliefert" getrennt
werden

Konsequenzen

- Wir führen **zwei Beziehungstypen** ein, "BezugsNachweis" und "HatGeliefert"
- Die beiden Beziehungstypen bilden **verschiedene Aussagen** ab und enthalten nicht dieselben Daten!
- Attribut "Preis" in beiden Beziehungstypen: vielleicht wurde zu einem anderen als dem Katalogpreis geliefert

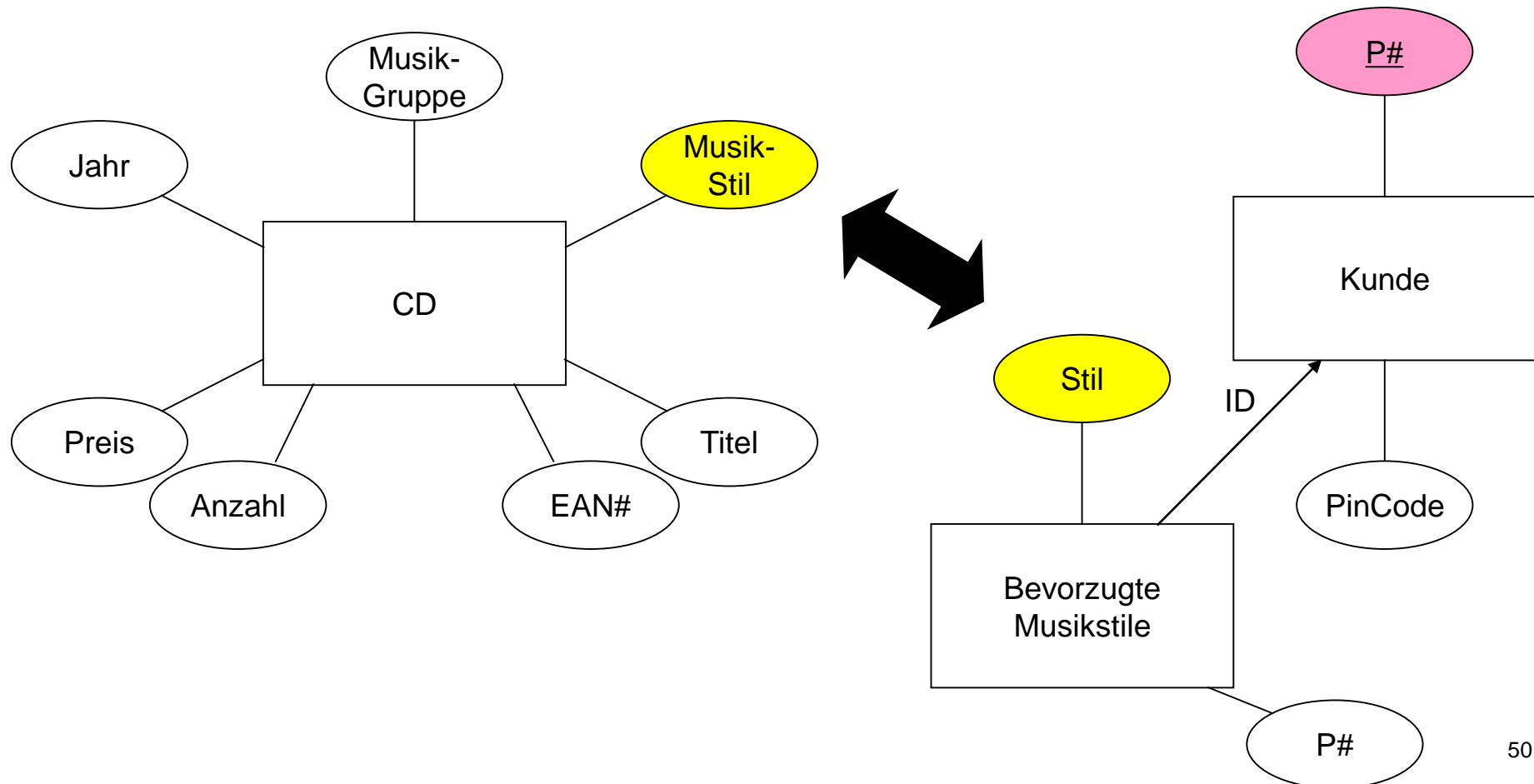
Kundenpräferenzen

- Wir wollen festhalten können, dass ein Kunde **verschiedene Musikstile** bevorzugt.

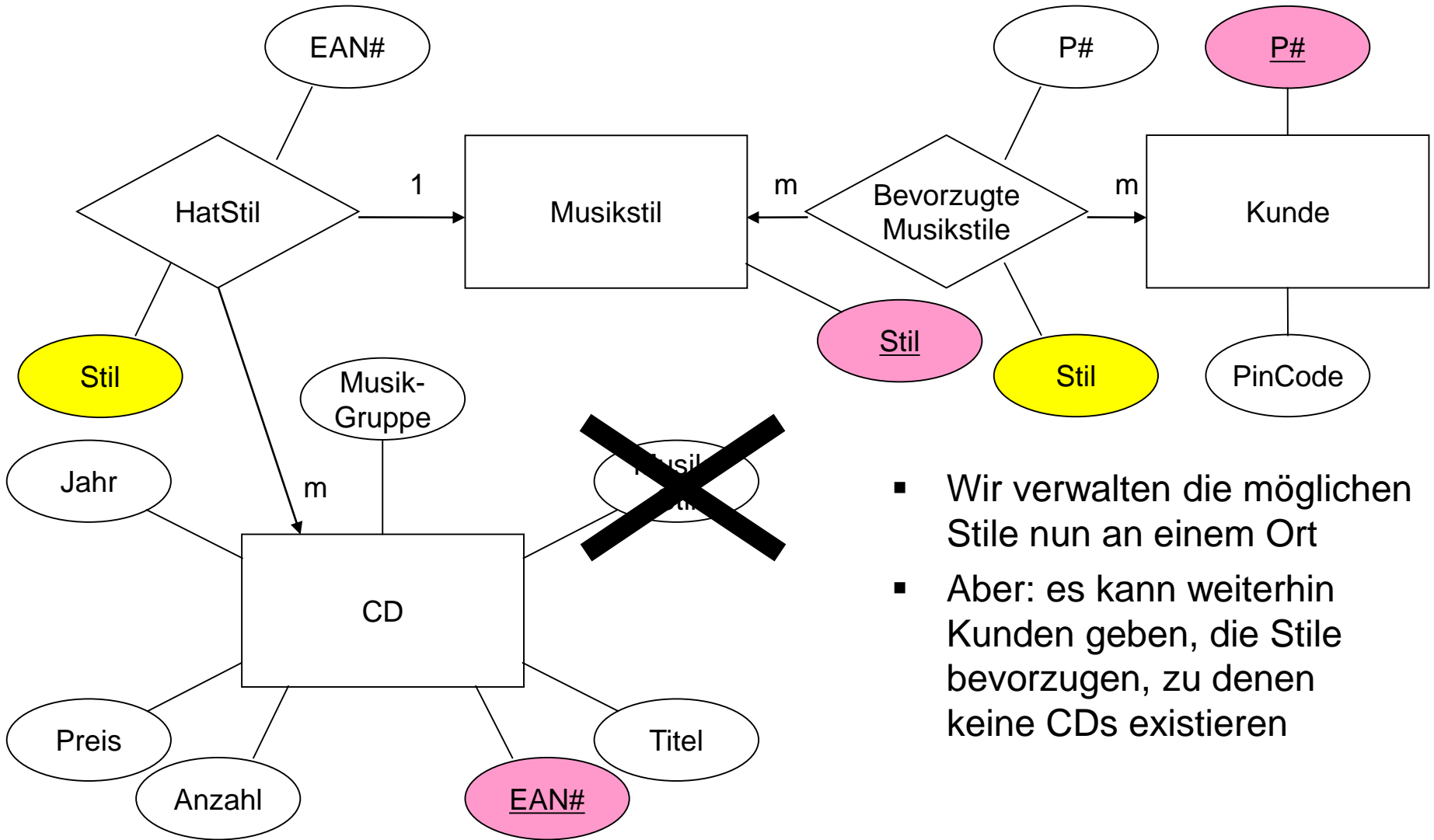


Kundenpräferenzen (2.Versuch)

- Mit dieser Lösung ist **nicht sichergestellt**, dass die bevorzugten Stile auch existierenden Stilen entsprechen



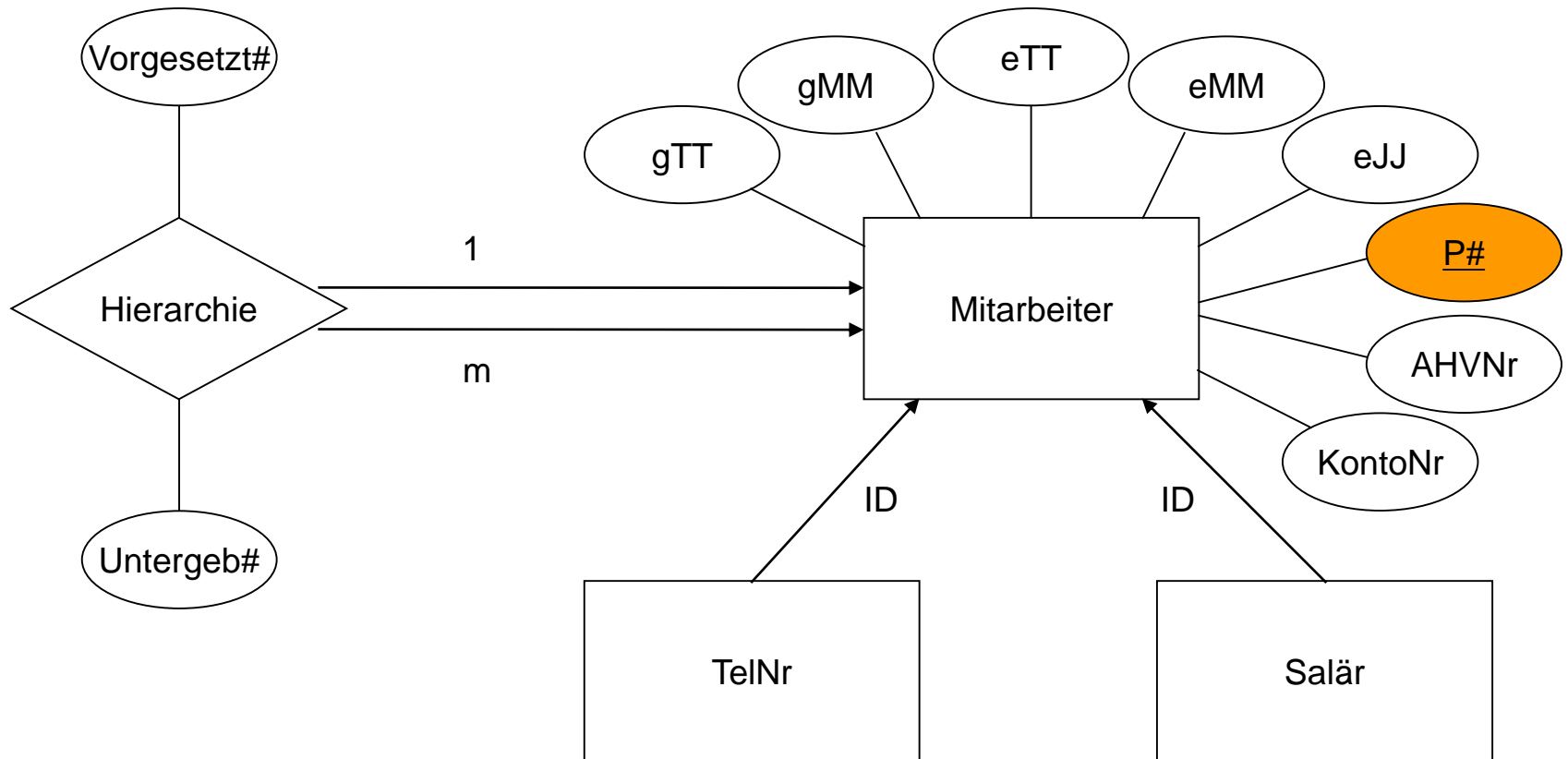
Kundenpräferenzen (2.Versuch)



- Wir verwalten die möglichen Stile nun an einem Ort
- Aber: es kann weiterhin Kunden geben, die Stile bevorzugen, zu denen keine CDs existieren

Noch ein Detail

- Wir wollen die **Geschäftshierarchie** modellieren

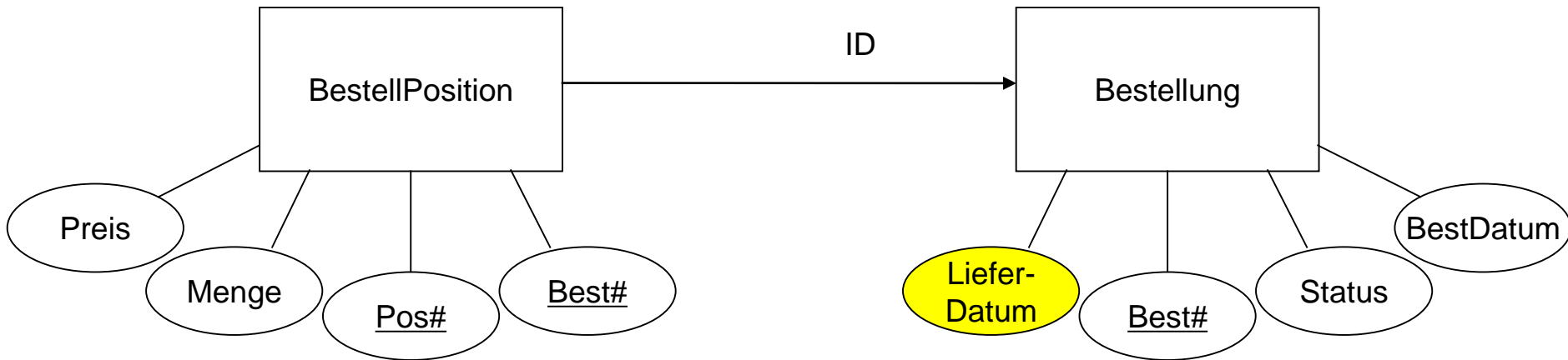
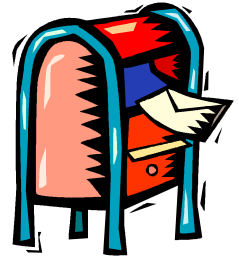


Konsequenzen

- Wir führen einen Beziehungstypen "Hierarchie" ein
- Beide Fremdschlüssel referenzieren denselben Primärschlüssel P# in Mitarbeiter
- Die Bezeichnungen sollten so gewählt werden, dass die **Semantik verständlich** ist
- Jedem Mitarbeiter ist höchstens ein Vorgesetzter zugeordnet
- Jeder Vorgesetzte hat beliebig viel Untergebene

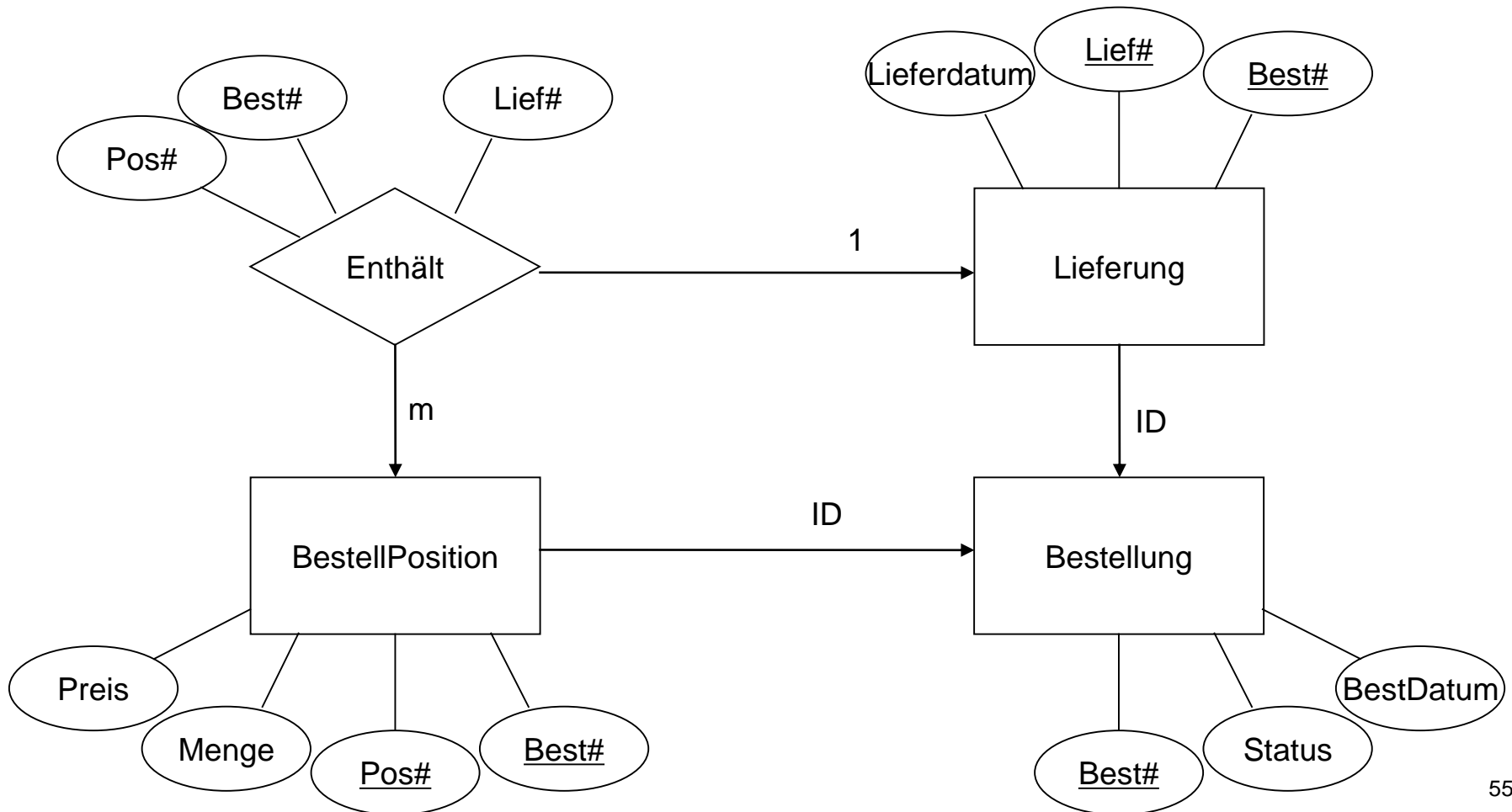
"You've got mail"

- Über kurz oder lang sollten wir die Bestellung auch liefern...
- Wenn wir alles in einer Lieferung ausliefern:

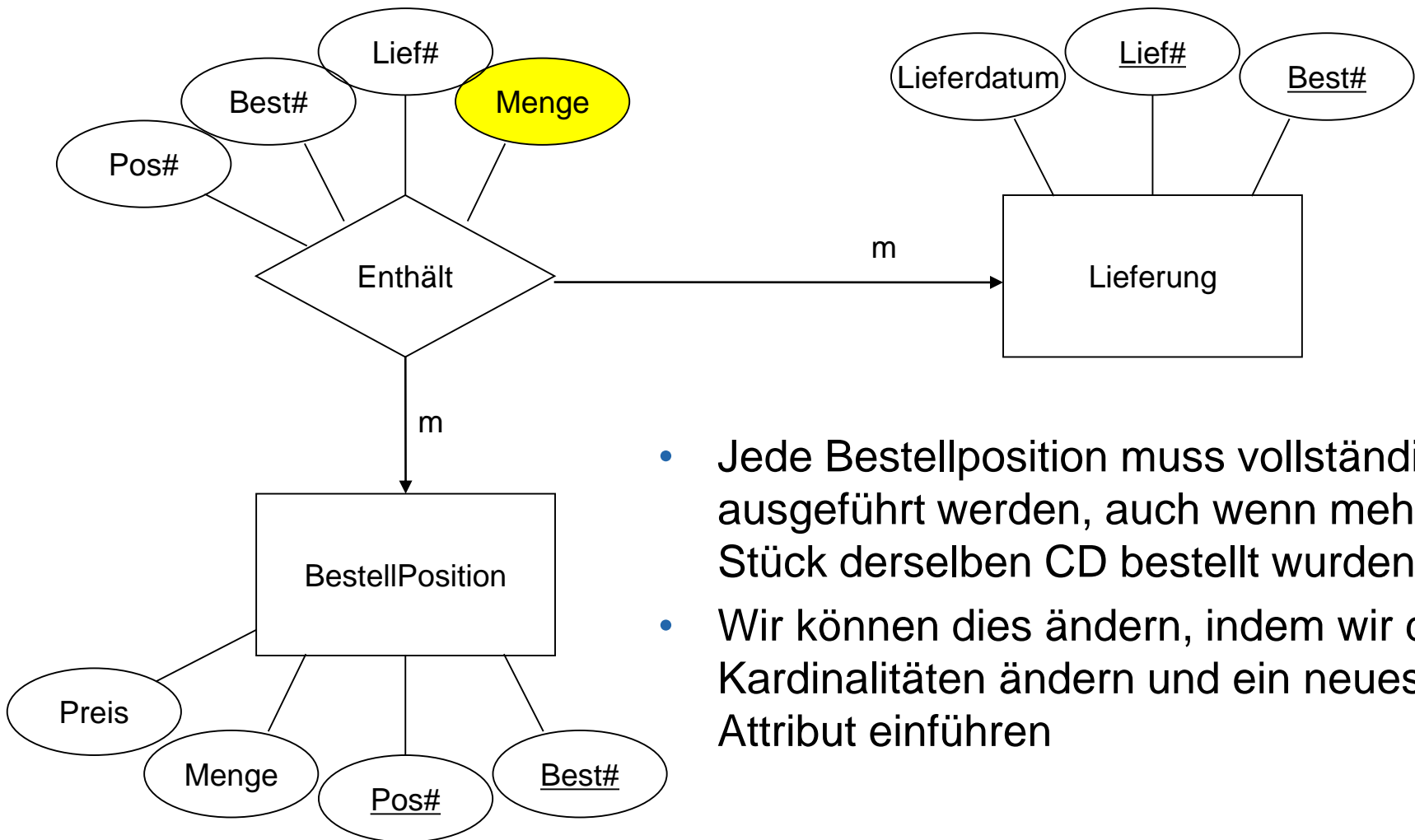


Mehr Flexibilität

- Wenn wir Teillieferungen erlauben wollen:

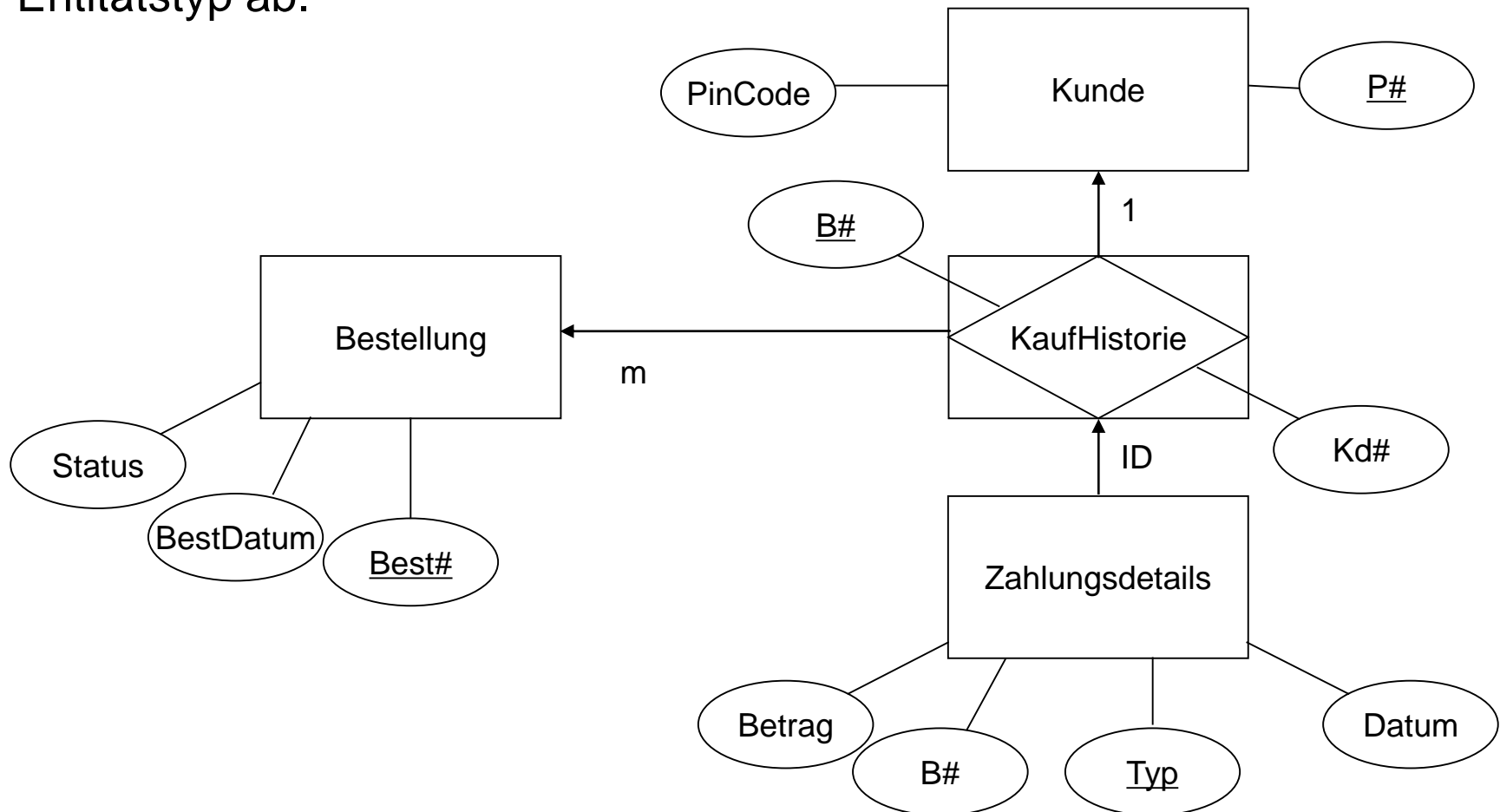


Konsequenzen



- Jede Bestellposition muss vollständig ausgeführt werden, auch wenn mehrere Stück derselben CD bestellt wurden
- Wir können dies ändern, indem wir die Kardinalitäten ändern und ein neues Attribut einführen

- **Umrandeter Rhombus:** von diesem Beziehungstyp hängt ein weiterer Entitätstyp ab.



Und weiter...

Das nächste Mal: ERM-Regeln, Muster, Normalisierung

