



# Kommunikation und Datenhaltung

## 2. Übung zur Datenhaltung

### EER & Relationenmodell



# Agenda

- Kurze Nachbesprechung zum ersten Übungsblatt (Relationale Algebra & SQL)
- Besprechung des zweiten Übungsblatts (EER & Relationenmodell)

# Nachtrag zum 1. Übungsblatt: Aufgabe 3e

- Join mit Relation „Album\_Track“ unnötig.
- Ebenfalls richtig und kürzer:

```
SELECT  a.id, a.name,  
        ROUND((AVG(length) / 1000 / 60), 2) as  
        "Average Track Length"  
FROM    Track t, Artist a  
WHERE   a.id = t.artist  
AND     length != 0 AND length IS NOT NULL  
GROUP  BY a.id, a.name  
HAVING  COUNT(*) > 100  
ORDER  BY "Average Track Length" DESC
```

# Nachtrag zum 1. Übungsblatt: LEFT JOIN

Artist

id	name	type	begindate	enddate
34778	David Hasselhoff	1	1952-07-17	NULL
377499	Macaulay Culkin	1	1980-08-26	NULL

Track

id	name	artist	length
6534544	Looking for Freedom	34778	235000
7605384	Jump in My Car	34778	178000

**Query von Aufgabe 4i) vor Projektion/Selektion auf obigen Datensätzen:**

id	name	type	begindate	enddate	id	name	artist	length
34778	D.H.	1	1952-07-17	NULL	6534544	Looking [...]	34778	235000
34778	D.H.	1	1952-07-17	NULL	7605384	Jump [...]	34778	178000
377499	M.C.	1	1980-08-26	NULL	NULL	NULL	NULL	NULL

# Nachtrag zum 1. Übungsblatt: LEFT JOIN

id	name	type	begindate	enddate	id	name	artist	length
34778	D.H.	1	1952-07-17	NULL	6534544	Looking [...]	34778	235000
34778	D.H.	1	1952-07-17	NULL	7605384	Jump [...]	34778	178000
377499	M.C.	1	1980-08-26	NULL	NULL	NULL	NULL	NULL

```
FROM Artist a LEFT JOIN Track t
ON a.id = t.artist
WHERE [...] AND t.artist IS NULL
```

- Für jedes Tupel aus Artist wird nach Join-Partnern in der Relation Track gesucht ( $a.id = t.artist$ ). Werden Tracks gefunden, enthält das Ergebnis für jeden Track zum Künstler eine Zeile mit dem Verbund.
- Wurde kein Track gefunden, so wird der Künstler dennoch ein mal in das Ergebnis übernommen. Diese Zeile enthält NULL-Werte für die Attribute aus der Relation Track.

# Nachtrag zum 1. Übungsblatt: Aufgabe 4b i)

*Lösung über Join-Anfrage:*

Welchem noch lebenden Künstler ist kein Track zugeordnet? Ein lebender Künstler zeichnet sich durch `begindate` ungleich `NULL` und einem `enddate` gleich `NULL` aus. Geben Sie Namen und „`begindate`“ des Künstlers aus.

```
SELECT a.name, begindate
FROM Artist a LEFT JOIN Track t
ON a.id = t.artist
WHERE begindate IS NOT NULL
AND enddate IS NULL
AND t.artist IS NULL
```



# Zweite Übungsblatt

- Thema: EER und Relationenmodell

# Vorgehensweise bei EER-Modellierung

- Schrittweises Vorgehen
  - 1) Entity-Typen und Attribute identifizieren
  - 2) Beziehungstypen und deren Attribute extrahieren
  - 3) Kardinalitäten festlegen
- Hilfreich:
  - Szenario komplett durchlesen
  - Entitäten und Relationen im Text markieren
- Kontrolle zum Schluss:
  - Alle Entitäten und Relationen modelliert?
  - Jede Kante mit Kardinalität versehen?

# Aufgabe 1: Szenario

- Eine TV-Serie hat einen eindeutigen Namen und mindestens eine Episode an Material.
- Jede Episode hat einen optionalen Namen und eine fortlaufende Nummer, die zur eindeutigen Identifikation dient.
- Eine Episode wird von TV-Sendern ausgestrahlt. Eine Übertragung ist gekennzeichnet durch eine Startzeit, Endzeit und den übertragenden Sender. In einer Übertragung wird immer genau eine Episode gesendet. Eine Episode selbst kann aber beliebig oft übertragen werden.
- Gedreht wird in Studios. Zu jedem Studio ist die Größe in Quadratmeter und seine Position in Längen- und Breitengrad gespeichert. Über Angabe des Längen- und Breitengrads kann jedes Studio von einem anderen unterschieden werden.

# Aufgabe 1: Szenario

- An der Produktion einer Episode sind verschiedene Personen beteiligt. Eine Person ist entweder Schauspieler, Kameramann oder Regisseur. Jede Person hat eine eindeutige ID, einen Namen und ein Alter, welches sich aus dem Geburtsdatum ableiten lässt. Für Schauspieler muss zusätzlich die Körpergröße gespeichert werden. Jede Episode hat einen optionalen Namen und eine fortlaufende Nummer, die zur eindeutigen Identifikation dient.
- Ein Produktionsplan sieht vor, von wann bis wann welche Episode produziert wird, welche Personen beteiligt sind und in welchem Studio gedreht wird. An einer Produktion ist mindestens eine Person beteiligt. Zu einem Zeitpunkt findet eine Produktion immer nur in einem Studio statt. Während einer Produktion können (Teile für) verschiedene Episoden produziert werden, mindestens wird aber Material für eine Episode gedreht.

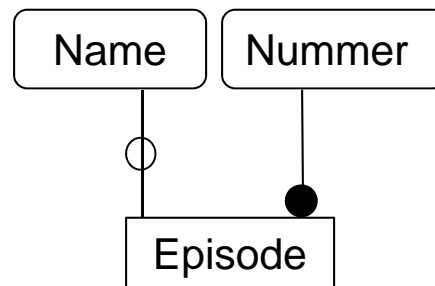
# Schritt 1: Entity-Typen und Attribute

Eine TV-Serie hat einen eindeutigen Namen und mindestens eine Episode an Material.



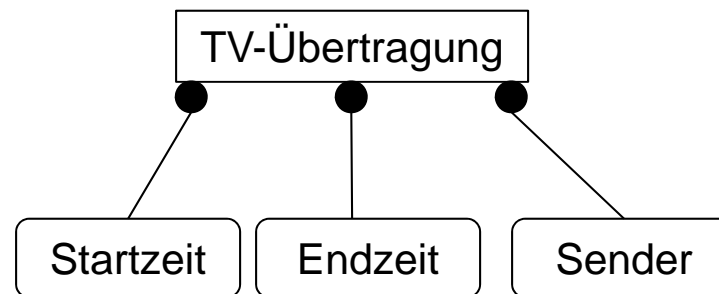
# Schritt 1: Entity-Typen und Attribute

Jede Episode hat einen optionalen Namen und eine fortlaufende Nummer, die zur eindeutigen Identifikation dient.



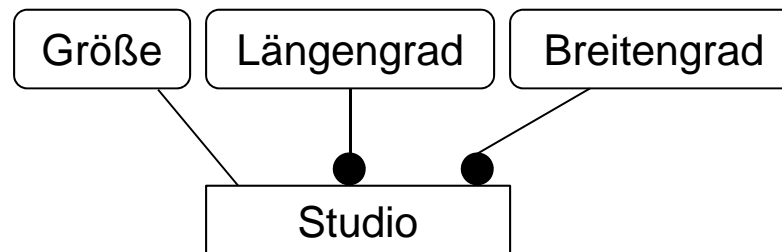
# Schritt 1: Entity-Typen und Attribute

Eine Episode wird von TV-Sendern ausgestrahlt. Eine Übertragung ist gekennzeichnet durch eine Startzeit, Endzeit und den übertragenden Sender. In einer Übertragung wird immer genau eine Episode gesendet. Eine Episode selbst kann aber beliebig oft übertragen werden.



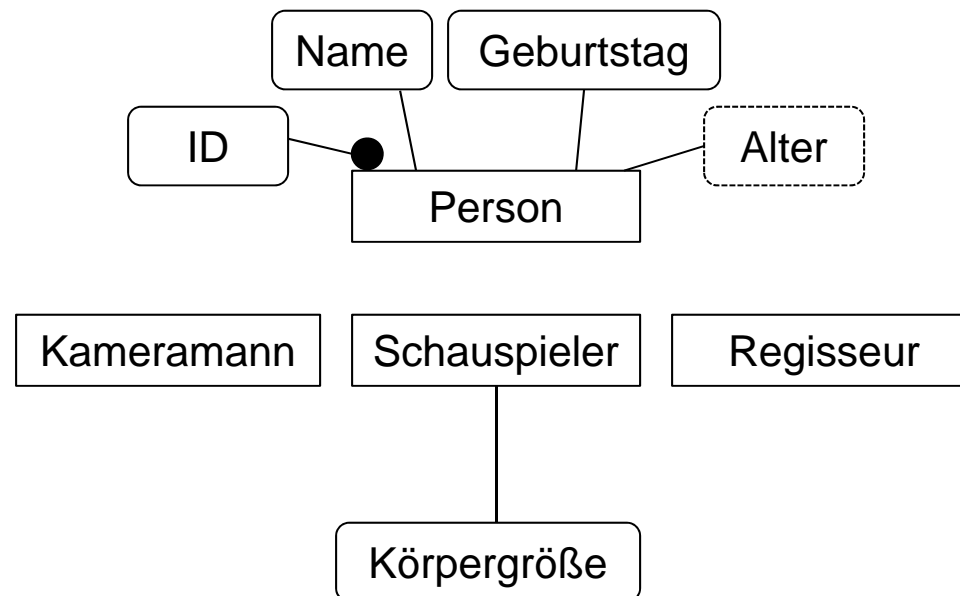
# Schritt 1: Entity-Typen und Attribute

Gedreht wird in Studios. Zu jedem Studio ist die Größe in Quadratmeter und seine Position in Längen- und Breitengrad gespeichert. Über Angabe des Längen- und Breitengrads kann jedes Studio von einem anderen unterschieden werden.



# Schritt 1: Entity-Typen und Attribute

An der Produktion einer Episode sind verschiedene Personen beteiligt. Eine Person ist entweder Schauspieler, Kameramann oder Regisseur. Jede Person hat eine eindeutige ID, einen Namen und ein Alter, welches sich aus dem Geburtsdatum ableiten lässt. Für Schauspieler muss zusätzlich die Körpergröße gespeichert werden.

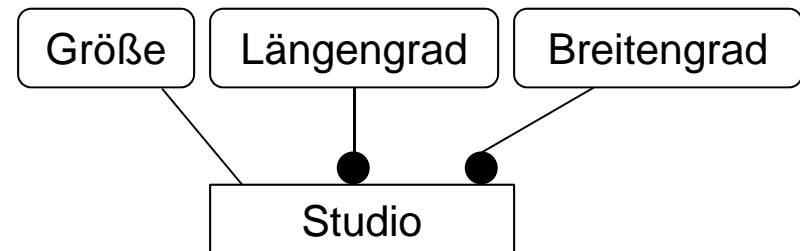
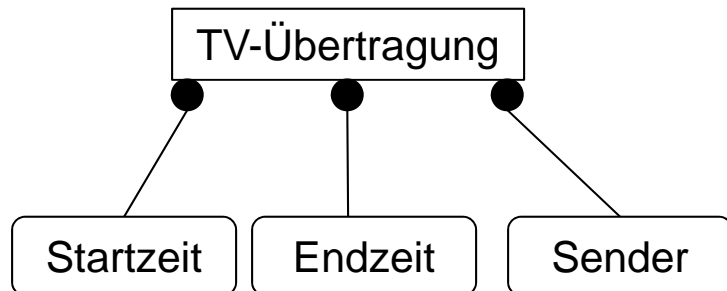
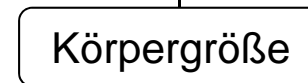
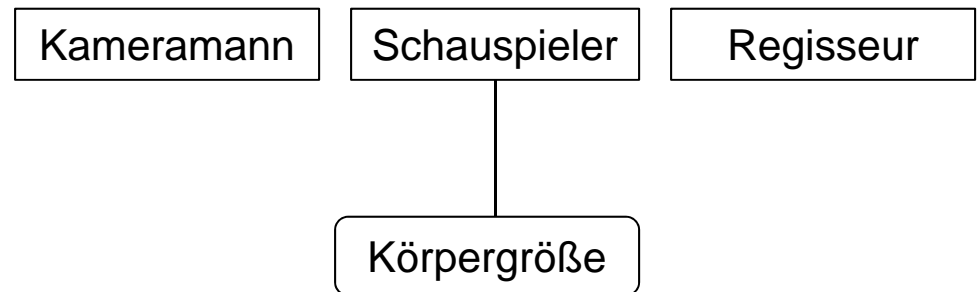
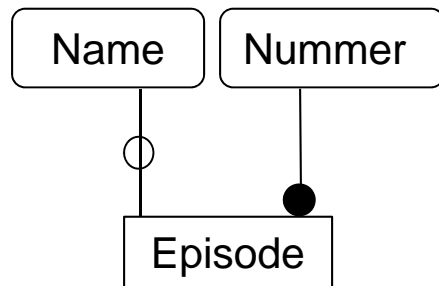
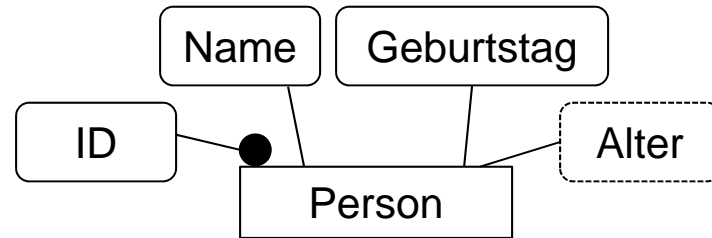
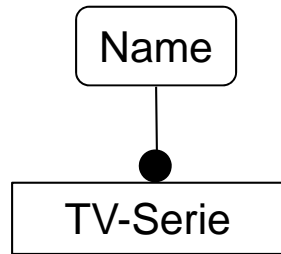




# Schritt 1: Entity-Typen und Attribute

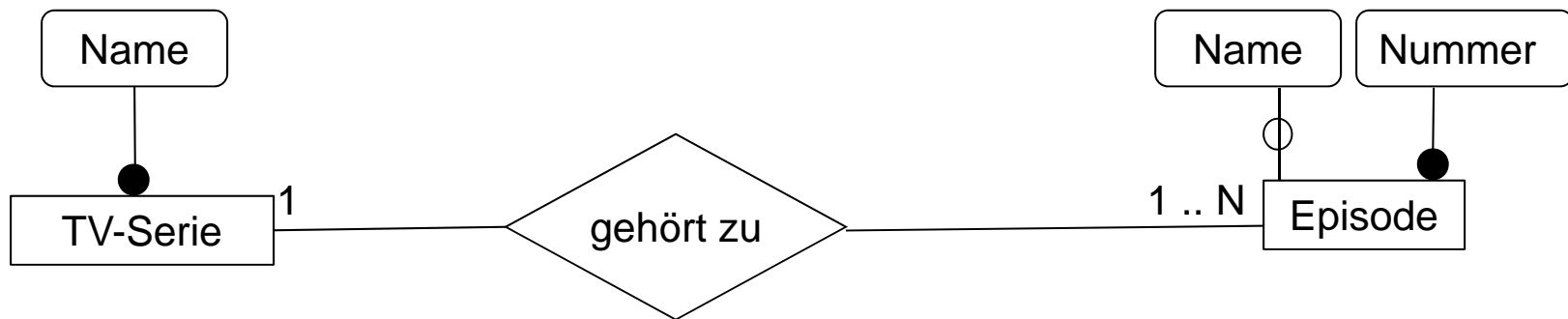
Ein Produktionsplan sieht vor, von wann bis wann welche Episode produziert wird, welche Personen beteiligt sind und in welchem Studio gedreht wird. An einer Produktion ist mindestens eine Person beteiligt. Zu einem Zeitpunkt findet eine Produktion immer nur in einem Studio statt. Während einer Produktion können (Teile für) verschiedene Episoden produziert werden, mindestens wird aber Material für eine Episode gedreht.

# Schritt 1: Entity-Typen und Attribute



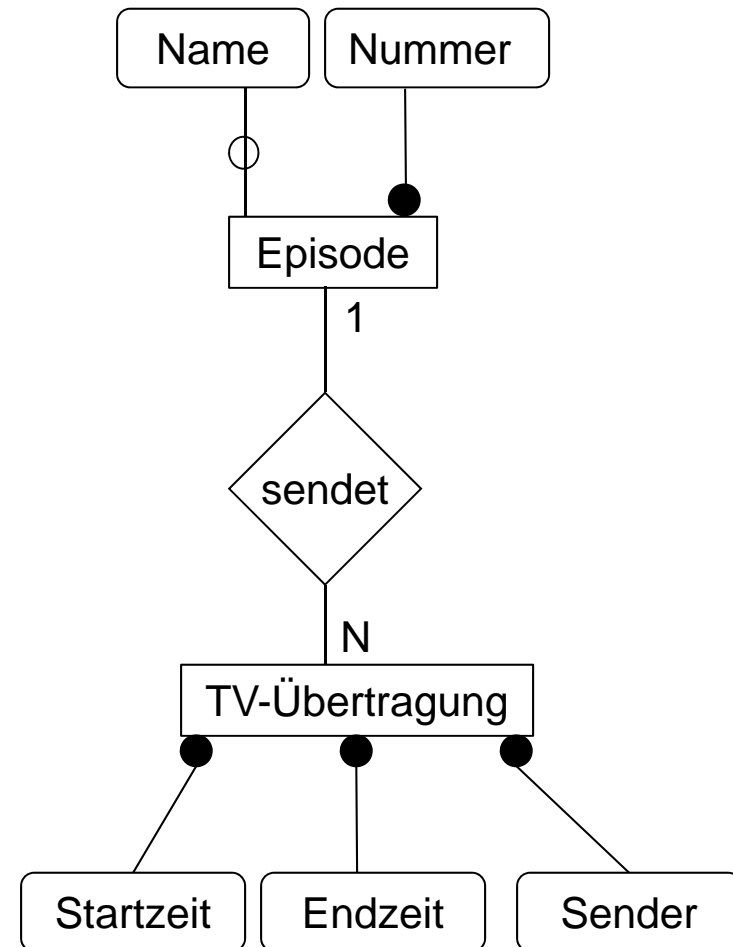
## Schritte 2 und 3: Beziehungstypen

Eine TV-Serie hat einen eindeutigen Namen und mindestens eine Episode an Material.



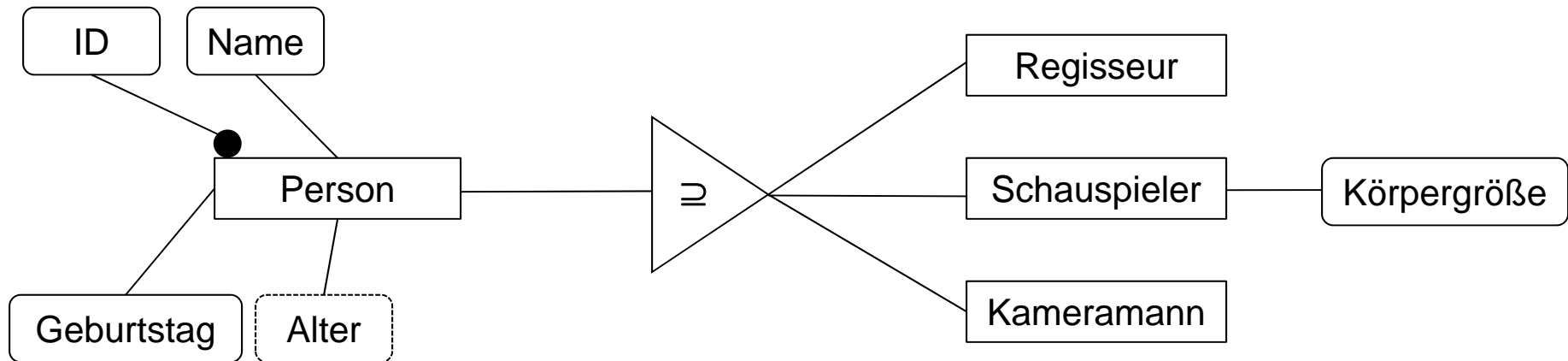
## Schritte 2 und 3: Beziehungstypen

Eine Episode wird von TV-Sendern ausgestrahlt. Eine Übertragung ist gekennzeichnet durch eine Startzeit, Endzeit und den übertragenden Sender. In einer Übertragung wird immer genau eine Episode gesendet. Eine Episode selbst kann aber beliebig oft übertragen werden.



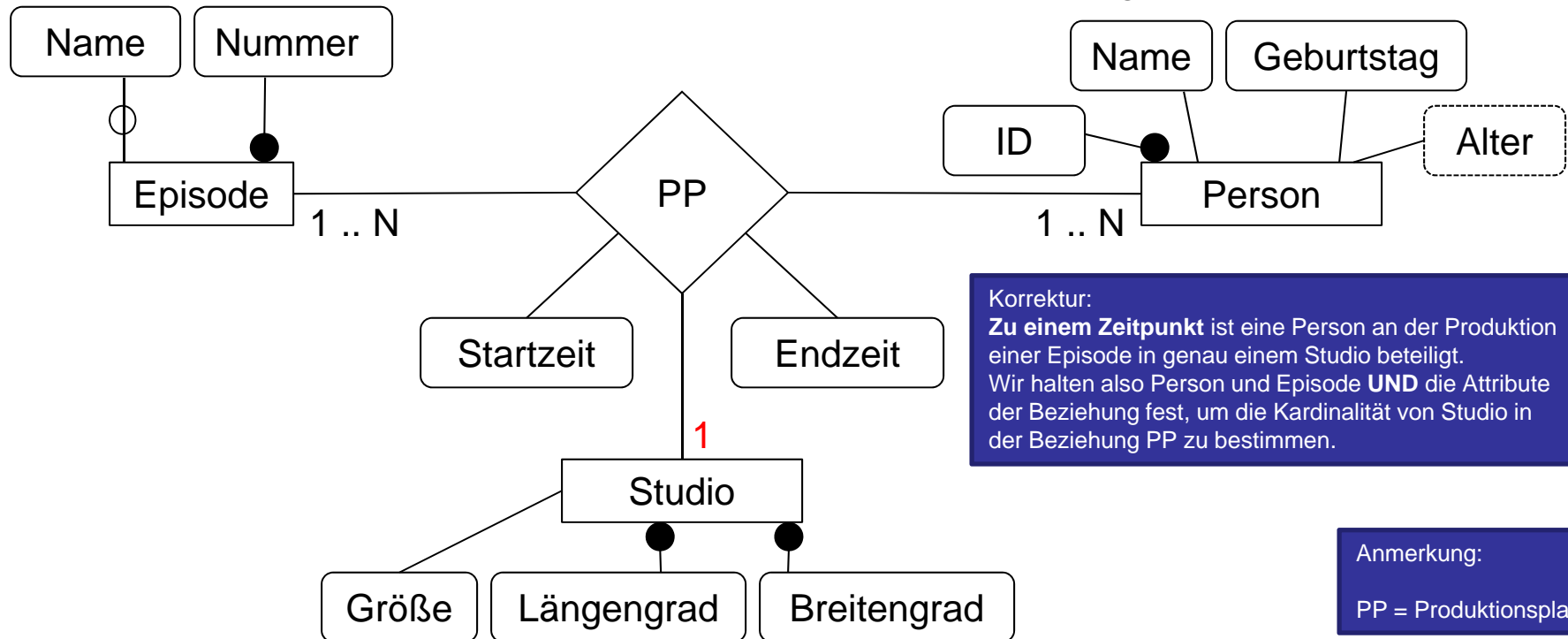
## Schritte 2 und 3: Beziehungstypen

An der Produktion einer Episode sind verschiedene Personen beteiligt. Eine Person ist entweder Schauspieler, Kameramann oder Regisseur. Jede Person hat eine eindeutige ID, einen Namen und ein Alter, welches sich aus dem Geburtsdatum ableiten lässt. Für Schauspieler muss zusätzlich die Körpergröße gespeichert werden. Jede Episode hat einen optionalen Namen und eine fortlaufende Nummer, die zur eindeutigen Identifikation dient.



# Schritte 2 und 3: Beziehungstypen

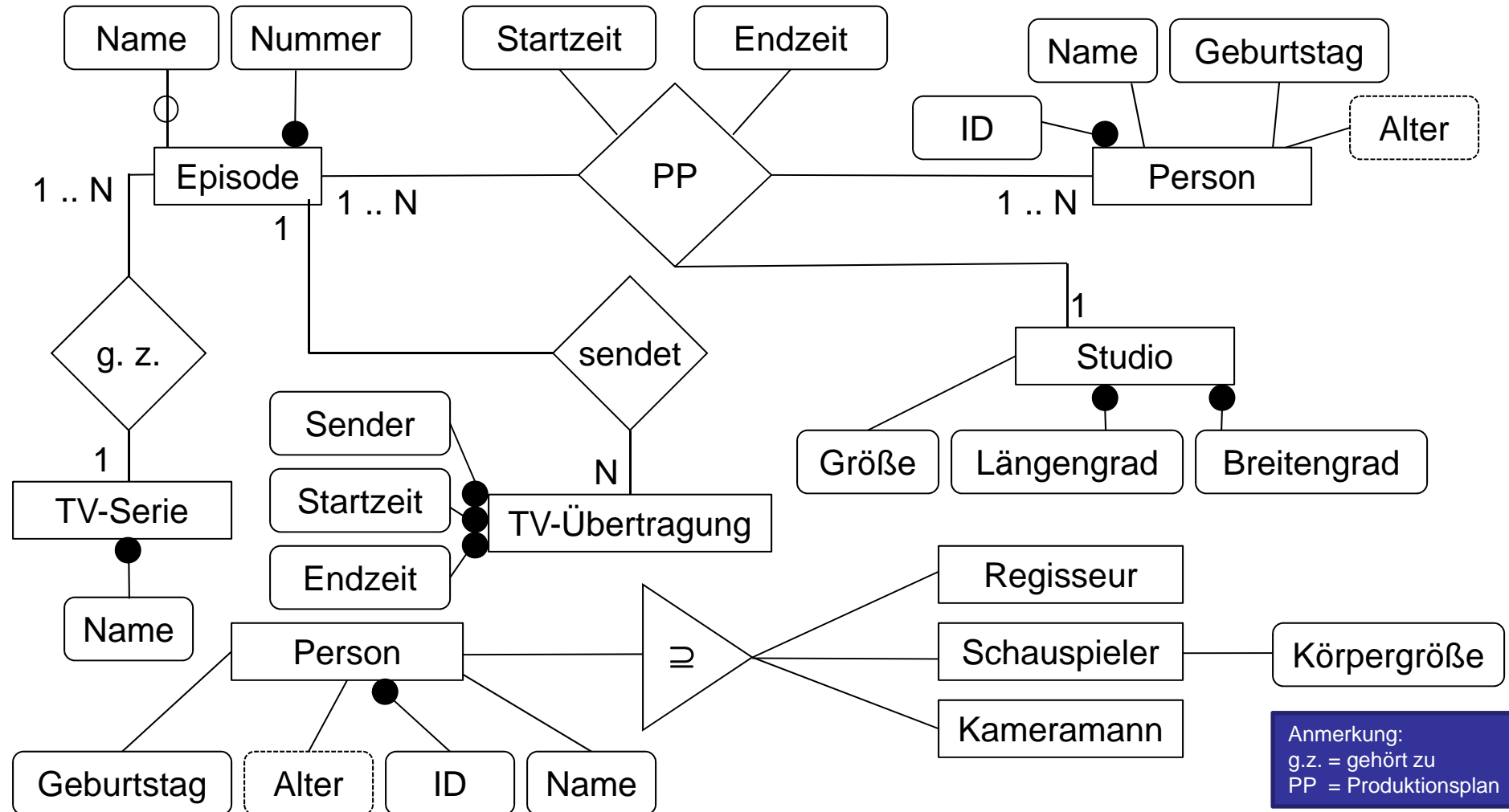
Ein Produktionsplan sieht vor, von wann bis wann welche Episode produziert wird, welche Personen beteiligt sind und in welchem Studio gedreht wird. An einer Produktion ist mindestens eine Person beteiligt. Zu einem Zeitpunkt findet eine Produktion immer nur in einem Studio statt. Während einer Produktion können (Teile für) verschiedene Episoden produziert werden, mindestens wird aber Material für eine Episode gedreht.



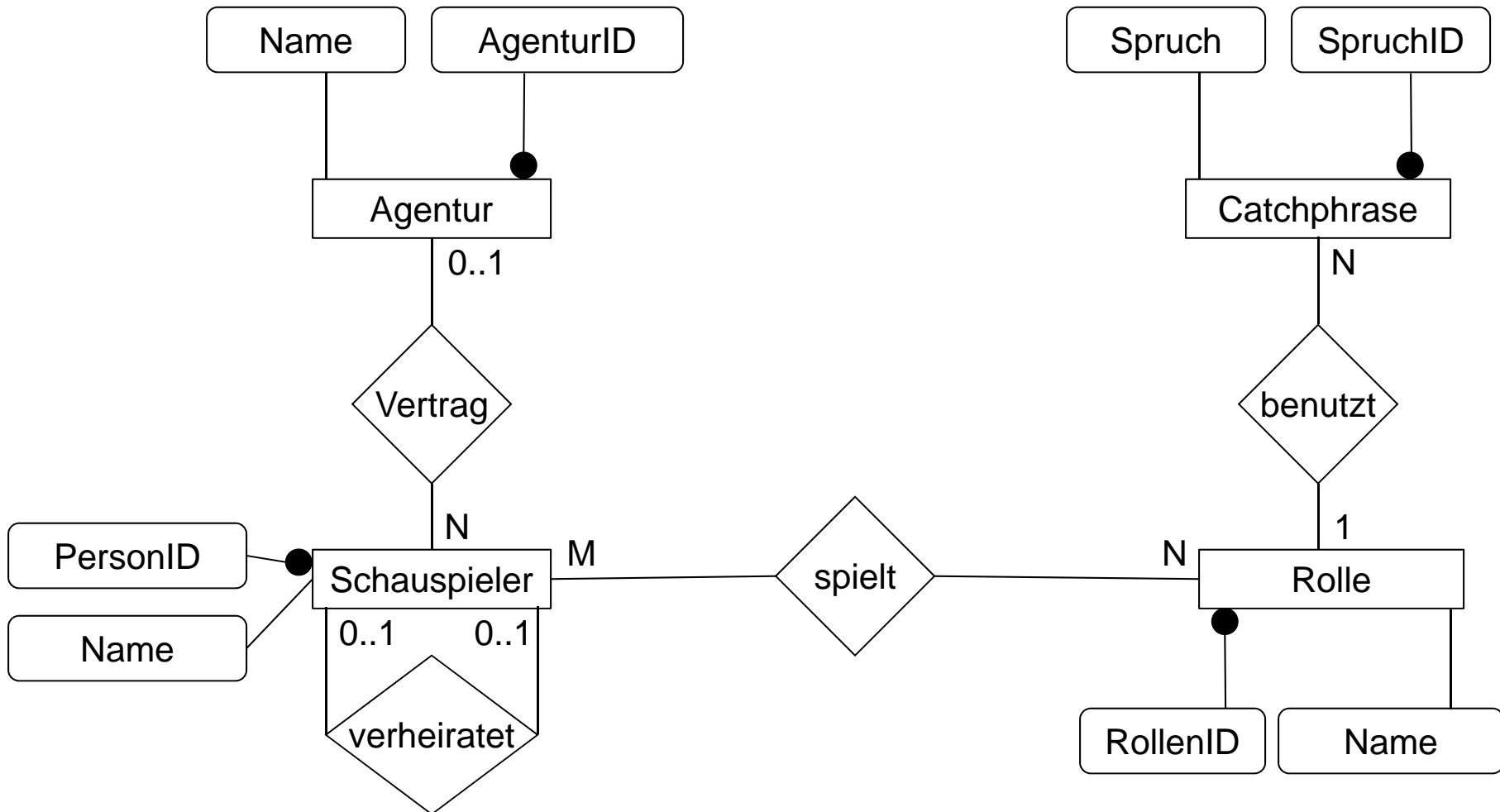
Korrektur:  
Zu **einem Zeitpunkt** ist eine Person an der Produktion einer Episode in genau einem Studio beteiligt. Wir halten also Person und Episode **UND** die Attribute der Beziehung fest, um die Kardinalität von Studio in der Beziehung PP zu bestimmen.

Anmerkung:  
PP = Produktionsplan

# Aufgabe 1: Gesamtlösung



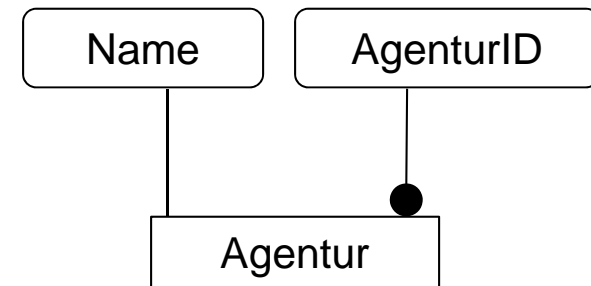
# Aufgabe 2



## Aufgabe 2a)

Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

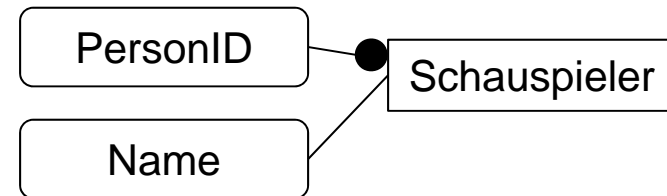
Agentur (AgenturID, Name)



## Aufgabe 2a)

Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

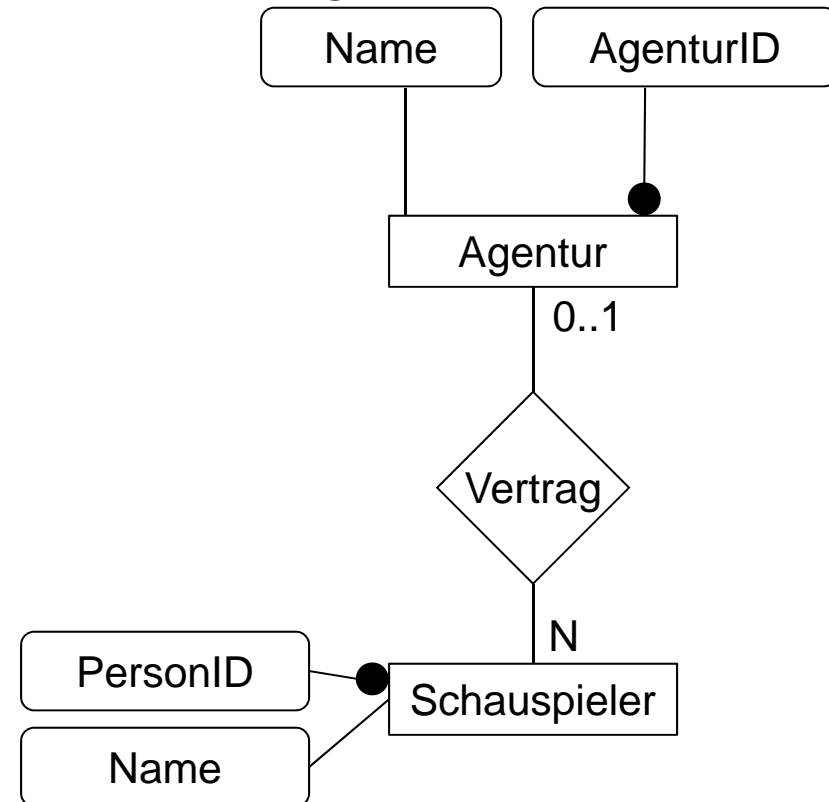
Schauspieler (PersonID, Name)



## Aufgabe 2a)

Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

Vertrag (PersonID, AgenturID)



## Aufgabe 2a)

Vertrag (PersonID, AgenturID)

Richtig! 😊

PersonID	AgenturID
1	100
2	100
1	200

Vertrag (PersonID, AgenturID)

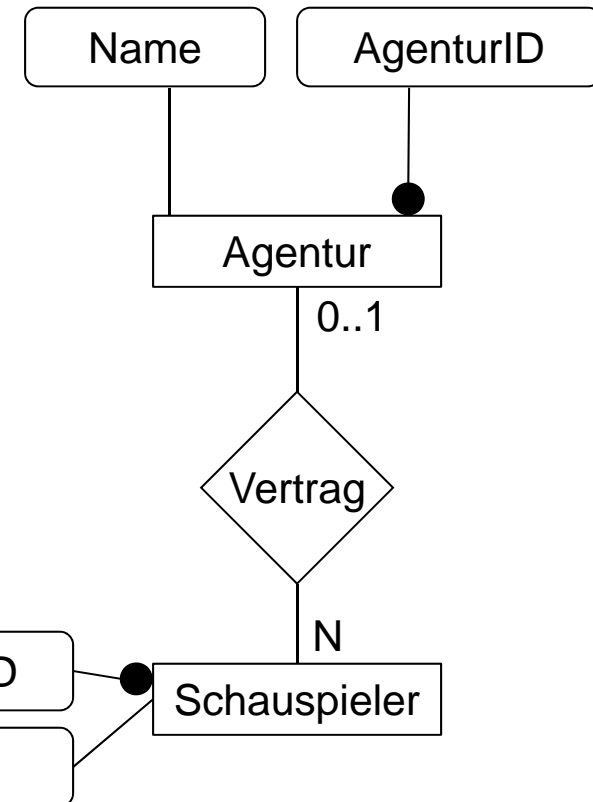
Falsch!

PersonID	AgenturID
1	100
1	200

Vertrag (PersonID, AgenturID)

Falsch!

PersonID	AgenturID
1	100
1	200



# Abbildung ER-Schema nach RDM

$E_1, E_2$ : an Beziehung beteiligte Entity-Typen,

$P_1, P_2$ : deren Primärschlüssel,

1:n-Beziehung:  $E_2$  ist n-Seite,

IST-Beziehung:  $E_1$  ist speziellerer Entity-Typ.

ER-Konzept	wird abgebildet auf relationales Konzept
Entity-Typ $E_i$	Relationenschema $R_i$
Attribute von $E_i$	Attribute von $R_i$
Primärschlüssel $P_i$	Primärschlüssel $P_i$
Beziehungstyp	Relationenschema
dessen Attribute	Attribute: $P_1, P_2$
1:n	weitere Attribute
1:1	$P_2$ wird Primärschlüssel der Beziehung
m:n	$P_1$ und $P_2$
	werden Schlüssel der Beziehung
	$P_1 \cup P_2$
	wird Primärschlüssel der Beziehung
IST-Beziehung	$R_1$ erhält zusätzlichen Schlüssel $P_2$

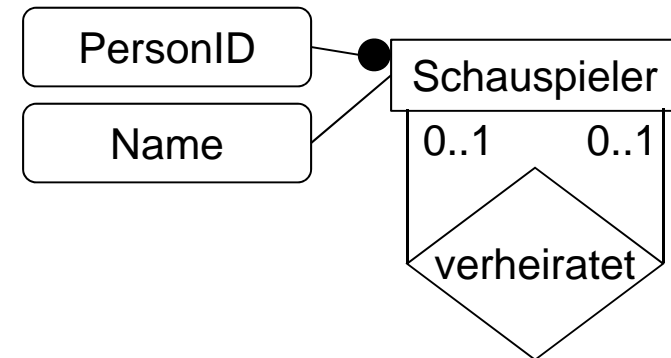
## Aufgabe 2a)

Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

Hier Angabe von K wichtig, da es die Information enthält, dass beide Attribute Kandidaten für Schlüssel sind und daher eindeutig sein müssen. Eines der Attribute wird zum Primärschlüssel gewählt (in diesem Fall PersonA\_ID), das andere Attribut ist aber immer noch Schlüssel(kandidat) und muss eindeutig sein.

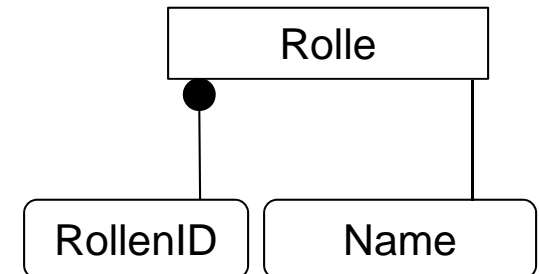
verheiratet (PersonA\_ID, PersonB\_ID)

$K = \{\{PersonA\_ID\}, \{PersonB\_ID\}\}$



## Aufgabe 2a)

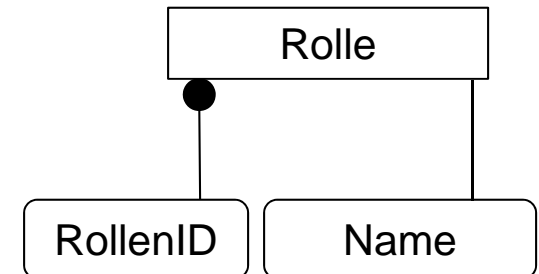
Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.



## Aufgabe 2a)

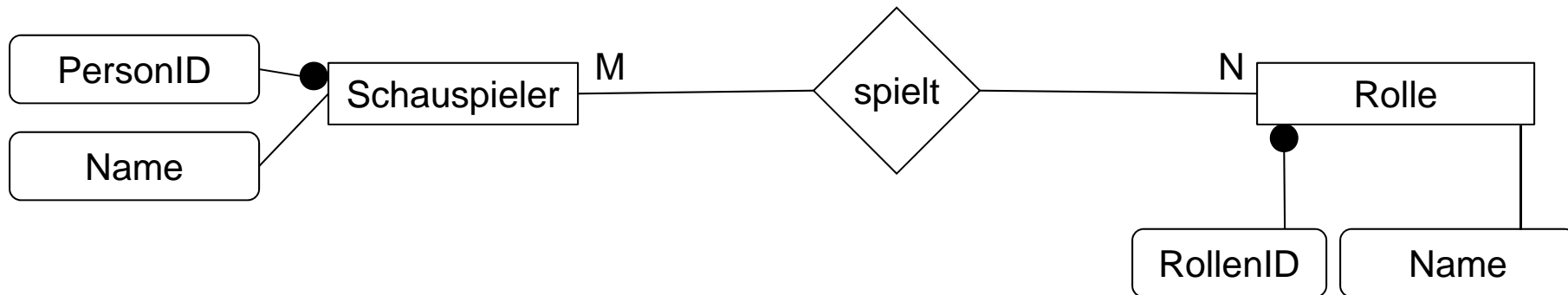
Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

Rolle (RollenID, Name)



## Aufgabe 2a)

Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

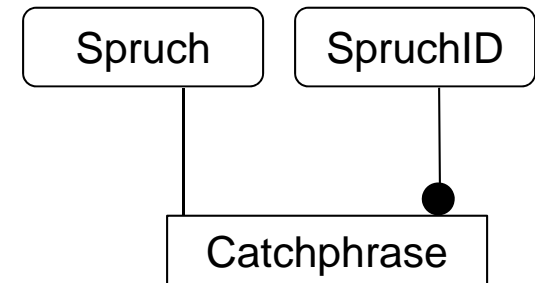


spielt (PersonID, RollenID)

## Aufgabe 2a)

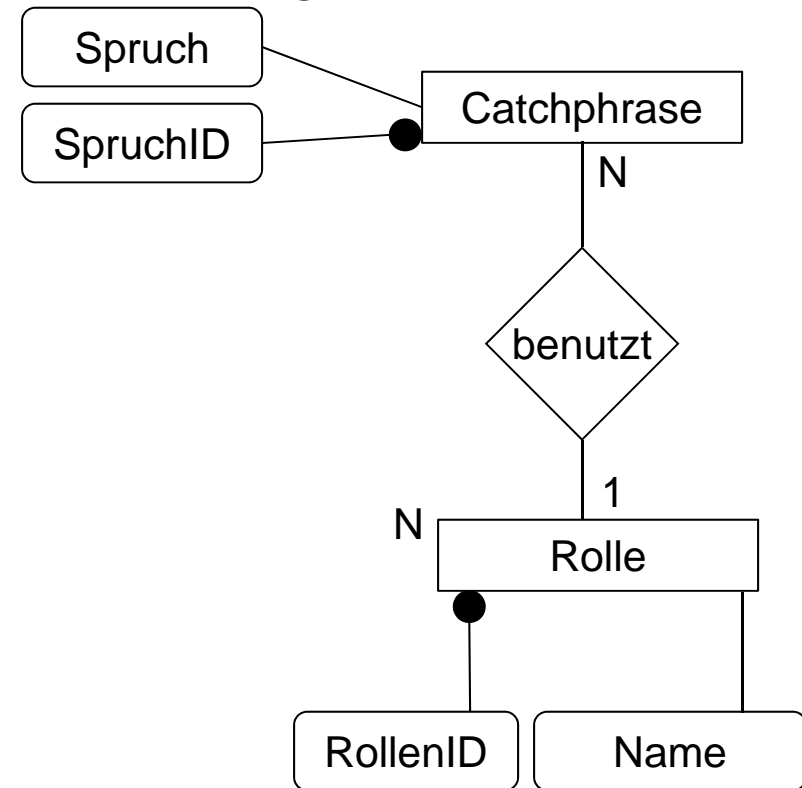
Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

Catchphrase (SpruchID, Spruch)



## Aufgabe 2a)

Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

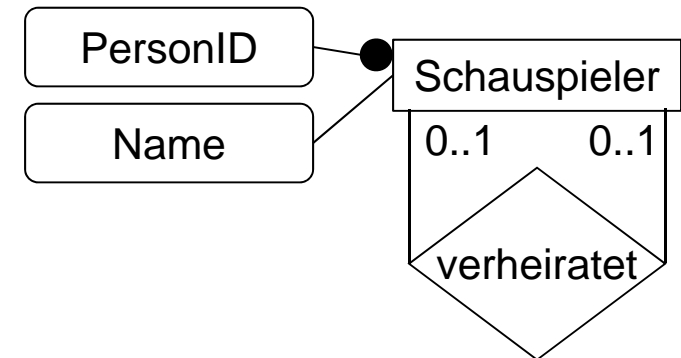


benutzt (SpruchID, RollenID)

## Aufgabe 2b)

Welcher Beziehungstyp lässt sich nicht kapazitätserhaltend in das Relationenmodell überführen? Durch welche Maßnahme könnte eine Kapazitätserhaltung durch ein DBMS gewährleistet werden?

1:1-Beziehung => Entweder PersonA\_ID oder PersonB\_ID wird Primärschlüssel der Relation „verheiratet“:



verheiratet (PersonA\_ID, PersonB\_ID)

PersonA_ID	PersonB_ID
1	2
2	3

verheiratet (PersonA\_ID, PersonB\_ID)

PersonA_ID	PersonB_ID
1	2
2	3

## Aufgabe 2b)

Lösung:

Zusätzliche Integritätsbedingungen:

```
CHECK(  
  NOT EXISTS(  
    (SELECT PersonA_ID  
     FROM verheiratet)  
    INTERSECT  
    (SELECT PersonB_ID  
     FROM verheiratet)  
  )  
)
```

```
PRIMARY KEY PersonA_ID,  
UNIQUE PersonB_ID
```

verhindert

PersonA_ID	PersonB_ID
1	2
2	3

verhindert

PersonA_ID	PersonB_ID
2	1
3	1



## Aufgabe 2c)

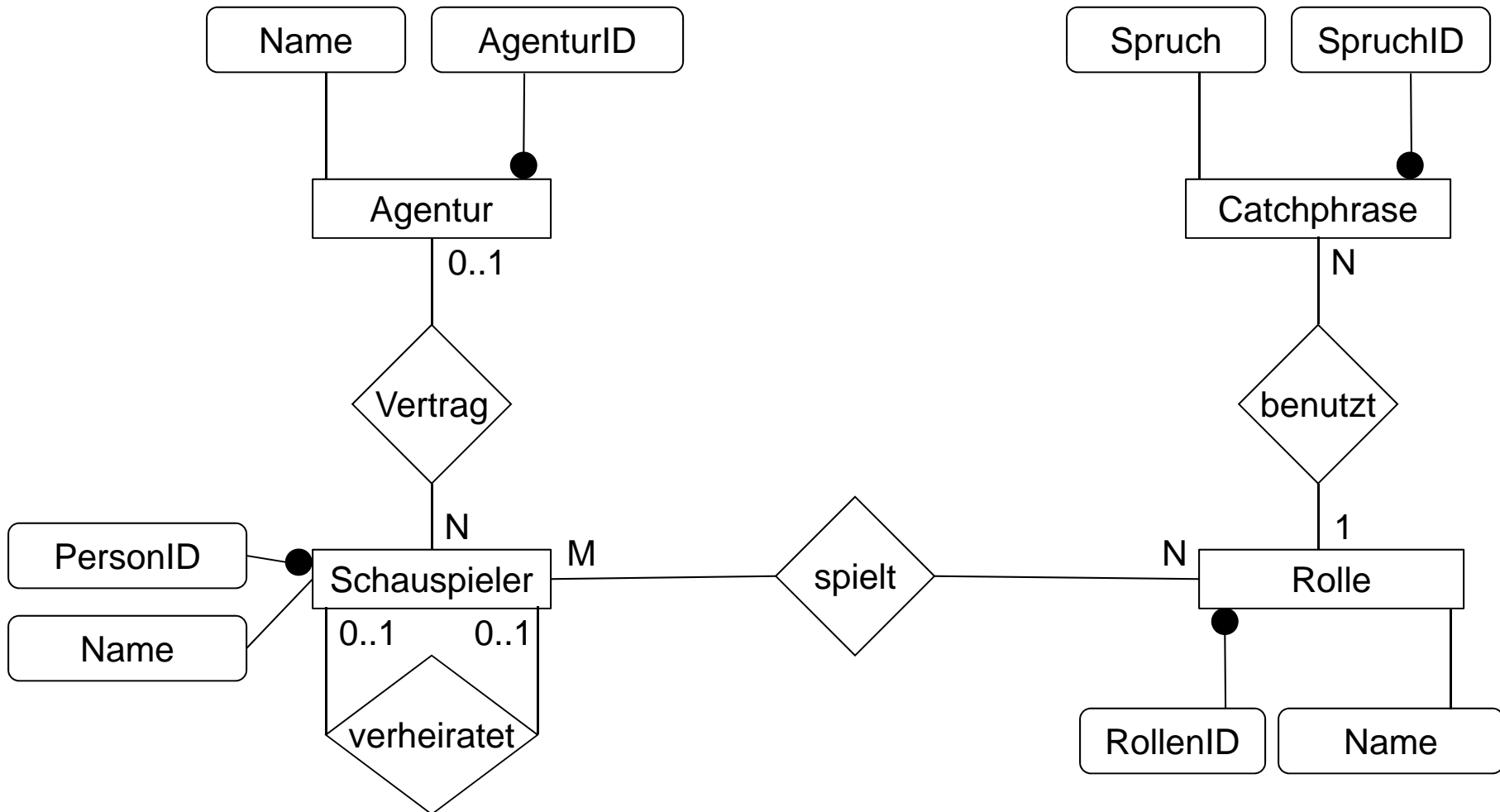
Minimieren Sie die Anzahl der Relationen durch geeignete Verschmelzung von Relationenschemata. Welches Ziel wird damit im Allgemeinen verfolgt?

# Verschmelzen von Relationenschemata

Bei zwingenden Beziehungen:

- 1:n-Beziehung: Das Entity-Relationenschema der n-Seite kann in das Relationenschema der Beziehung integriert werden.
- 1:1-Beziehung: Beide Entity-Relationenschemata können in das Relationenschema der Beziehung integriert werden.

## Aufgabe 2

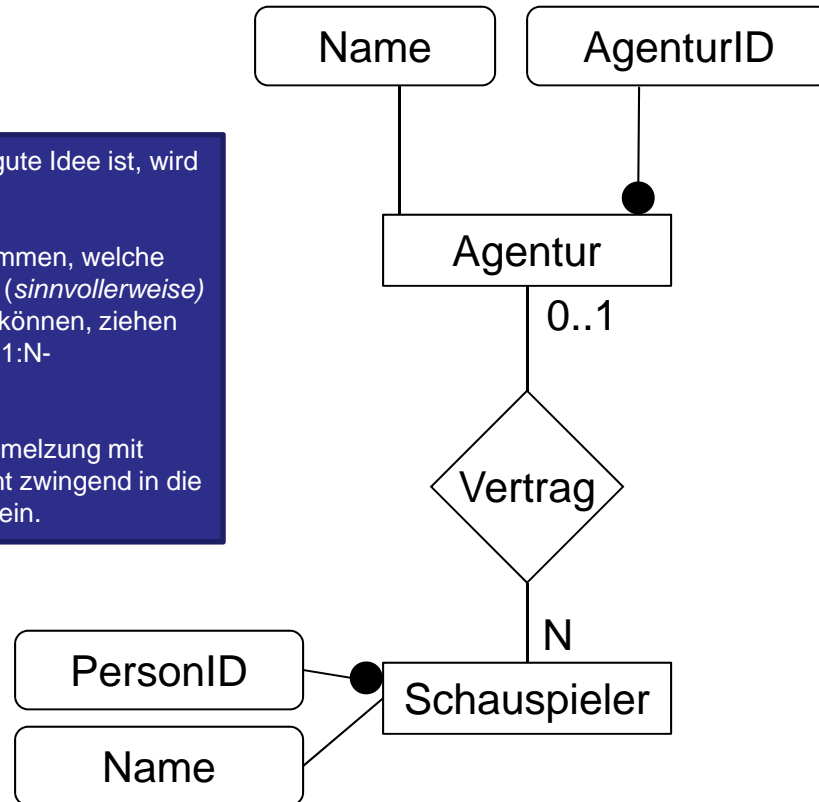


## Aufgabe 2c: Verschmelzung

Warum das möglicherweise keine gute Idee ist, wird in Aufgabe 2d) behandelt.

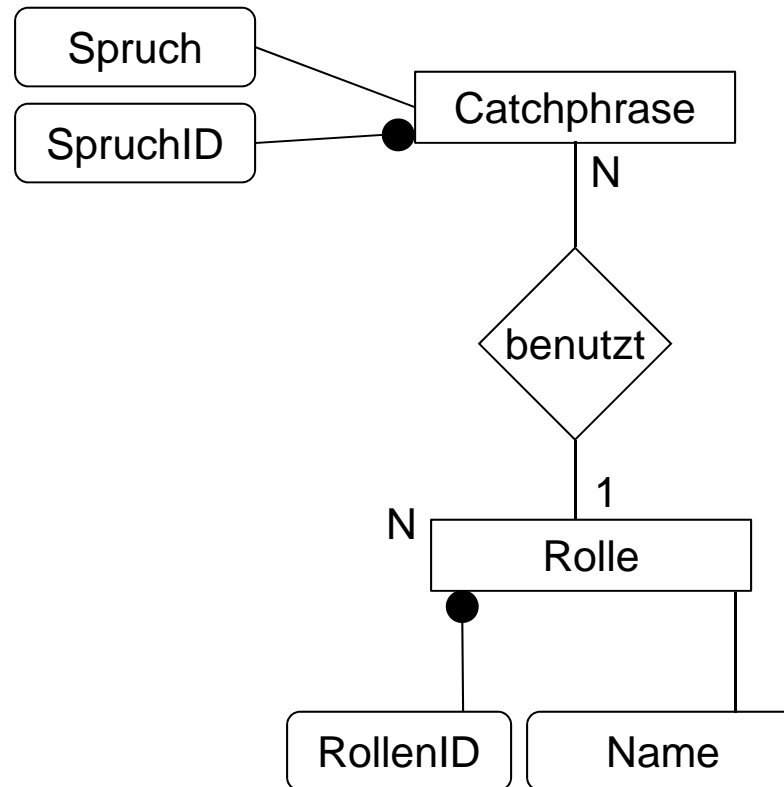
Sollte in der Klausur eine Frage kommen, welche Entity-Typen und Beziehungstypen (*sinnvollerweise*) miteinander verschmolzen werden können, ziehen Sie bitte nur 1:1- Beziehungen und 1:N- Beziehungen in Betracht.

Welche Nachteile aber eine Verschmelzung mit einer Entität mit sich bringt, die nicht zwingend in die Relation eingeht, sollte Ihnen klar sein.

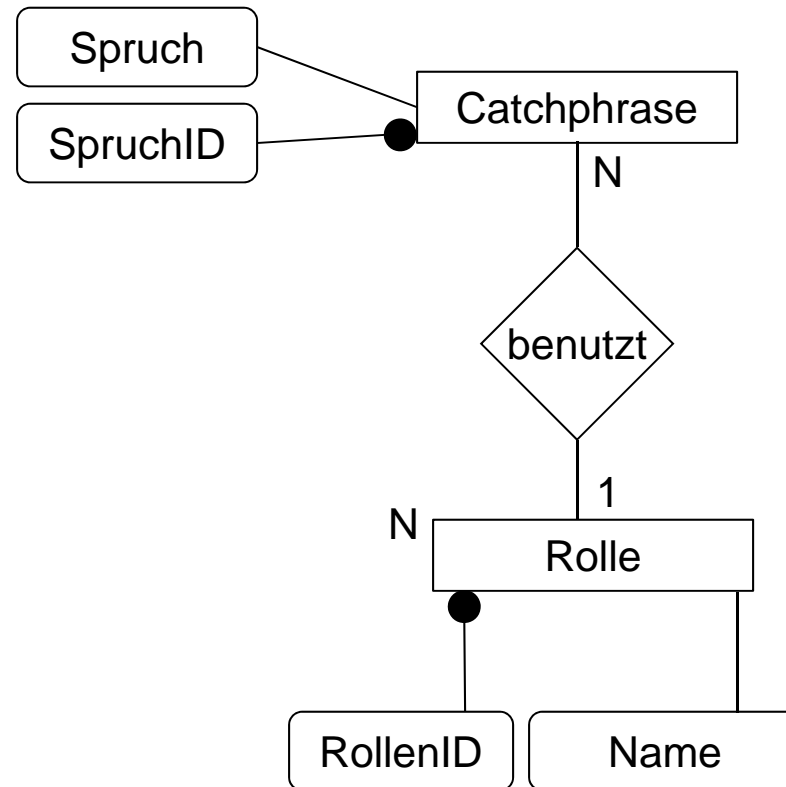


Schauspieler (PersonID, Schauspielername, AgenturID, Agenturname)

## Aufgabe 2c: Verschmelzung

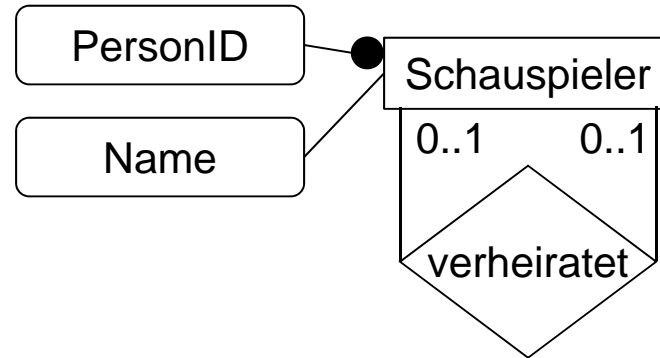


## Aufgabe 2c: Verschmelzung



Catchphrase (SpruchID, Spruch, RollenID, Name)

## Aufgabe 2c: Verschmelzung



Warum das möglicherweise keine gute Idee ist, wird in Aufgabe 2d) behandelt.

Sollte in der Klausur eine Frage kommen, welche Entity-Typen und Beziehungstypen (*sinnvollerweise*) miteinander verschmolzen werden können, ziehen Sie bitte nur 1:1- Beziehungen und 1:N- Beziehungen in Betracht.

Welche Nachteile aber eine Verschmelzung mit einer Entität mit sich bringt, die nicht zwingend in die Relation eingeht, sollte Ihnen klar sein.

Schauspieler (PersonA\_ID, PersonA\_Name, PersonB\_ID, PersonB\_Name)

$K = \{\{PersonA\_ID\}, \{PersonB\_ID\}\}$

## Aufgabe 2c: Verschmelzung

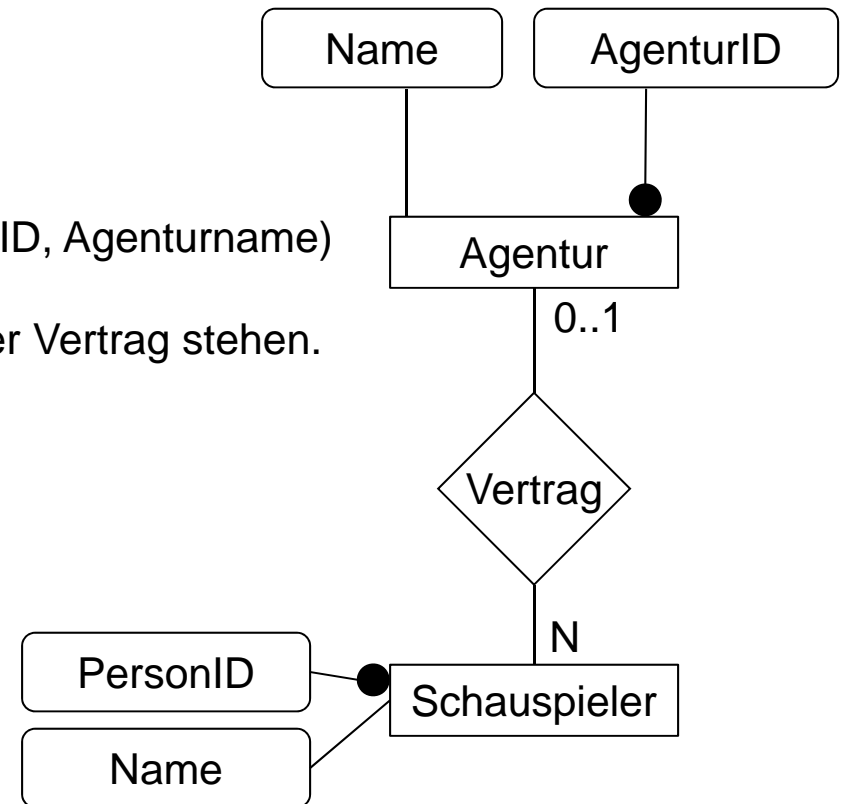
- Warum Relationen verschmelzen?
  - Anzahl der Relationen reduzieren
  - Dadurch weniger Joins nötig
  - Performance-Gewinn
- Bringt aber manche Nachteile mit sich

## Aufgabe 2d: Nachteile Verschmelzung

Wann macht eine Verschmelzung keinen Sinn? Geben Sie dazu ein Beispiel.

Schauspieler (PersonID, Schauspielername, AgenturID, Agenturname)

Nicht jeder Schauspieler muss bei einer Agentur unter Vertrag stehen.  
Dann viele NULL-Werte!



## Aufgabe 2d: Nachteile Verschmelzung

PersonID	Schauspielername	AgenturID	Agenturname
1	Al Pacino	1	Some Agency
2	John Travolta	NULL	NULL
3	Joaquin Phoenix	NULL	NULL
4	Jessica Alba	1	Some Agency
5	Julia Roberts	NULL	NULL

- Jeder NULL-Wert kostet Speicherplatz
- Kenntnis über Datenbestand nötig, für sinnvolle Entscheidung, ob Verschmelzung oder nicht
- Redundanzen im Datenbestand (Agenturname)
- Weniger schöne Modellierung: Person und Agentur verschiedene Dinge, nun aber eine Entität