

IMDb (Teil 1)  
Relationales Modell, ER-Modellierung  
VL Big Data Engineering  
(aka Informationssysteme)

Prof. Dr. Jens Dittrich

[bigdata.uni-saarland.de](http://bigdata.uni-saarland.de)

7. Mai 2020

# IMDb (Teil 1)

## Geplante Struktur für jeweils zwei Wochen Vorlesung:

1. Konkrete Anwendung: IMDb
2. Was sind die Datenmanagement und -analyseprobleme dahinter?
3. Grundlagen, um diese Probleme lösen zu können
  - (a) Folien
  - (b) Jupyter/Python/SQL Hands-on
4. Transfer der Grundlagen auf die konkrete Anwendung

# IMDb (Teil 1)

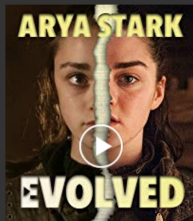
## 1. Konkrete Anwendung: IMDb

- IMDb: Internet Movie Database
- <https://www.IMDb.com>
- Daten zu Film- und Fernsehproduktionen
- seit 1990
- 2018: mehr als 5 Millionen Produktionen und mehr als 9 Millionen Film- und Fernsehschaffenden
- kurze Demo



Shazam! Soars to the  
Top  
Weekend Box Office

Browse trailers »



The Evolution of Arya  
Stark  
From Misfit to Assassin



From 'Holby City' to  
'Killing Eve'  
The Rise of Jodie Comer

## Opening This Week

- + Hellboy - Call of Darkness
- + Mister Link - Ein fellig verrücktes Abenteuer
- + After Passion
- + Little
- + High Life
- + Les filles du soleil
- + Sauvage

Opens Apr.  
10

See more opening this week »

Get Showtimes »

## Now Playing (Box Office)

- + Shazam!  
\$53.5M Showtimes
- + Friedhof der Kuscheltiere  
\$25.0M Showtimes
- + Dumbo  
\$18.2M Showtimes

## Zachary Levi Reveals His 'Shazam!' Suit Struggles





IMDb

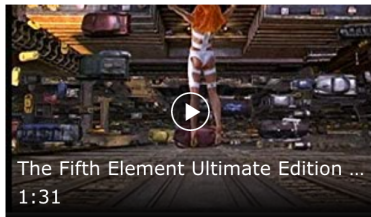
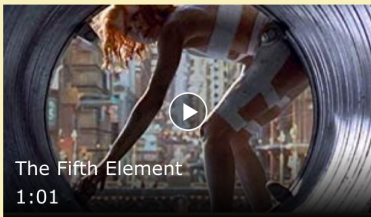
the fifth element

All



### **The Fifth Element** (1997)

Bruce Willis, Milla Jovovich



### **The Fifth Element** (1998)

Action, Adventure, Sci-Fi



### **Elementary: Season 5 - The Fifth Elementary** (2017)

Nelsan Ellis, Jon Michael Hill



### **The Fifth Element** (2002)



### **Beatboxing: The Fifth Element of Hip Hop** (2011)



Shazam! Soars  
Top  
Weekend Box Office

[Browse trailers »](#)

Zachary Le

IMDb

Find Movies, TV shows, Celebrities and more...

All

Movies, TV  
& Showtimes

Celebs, Events  
& Photos

News &  
Community

Watchlist

FULL CAST AND CREW

TRIVIA

USER REVIEWS

IMDbPro

MORE

SHARE

+

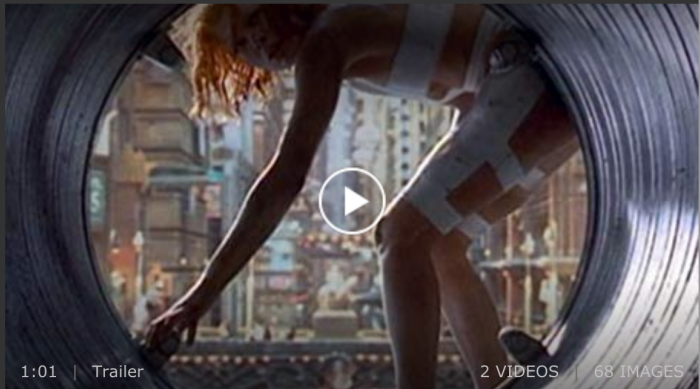
# Das fünfte Element (1997)

★ 7,7<sub>/10</sub>  
399.500

★ Rate  
This

The Fifth Element (*original title*)




12 | 2h 6min | Action, Adventure, Sci-Fi | 28 August 1997 (Germany)



# Cast

[Edit](#)

Cast overview, first billed only:

	Bruce Willis	...	Korben Dallas
	Gary Oldman	...	Zorg
	Ian Holm	...	Cornelius
	Milla Jovovich	...	Leeloo
	Chris Tucker	...	Ruby Rhod
	Luke Perry	...	Billy
	Brion James	...	General Munro
	Tommy 'Tiny' Lister	...	President Lindberg (as Tommy 'Tiny' Lister Jr.)
	Lee Evans	...	Fog

# IMDb (Teil 1)

2. Was sind die Datenmanagement und -analyseprobleme dahinter?

## Frage 1

Wie werden in IMDb die Daten zu Filmen, Schauspielern, Regisseuren usw. modelliert und abgelegt?

## Frage 2

Wie werden in IMDb Verknüpfungen dieser Daten modelliert und abgelegt?

nächste Woche:

## Frage 3

Wie stellen wir Anfragen an diese Daten?

...

## 3. Grundlagen, um diese Probleme lösen zu können

(a) Folien

(b) Jupyter/Python/SQL Hands-on

- ER-Modellierung
- Das relationale Modell

# Wichtigste Lernziele

## Entity-Relationship-Modellierung

ER, Entitätstypen, Beziehungstypen (auch mehrstellig), Domäne, Attribut, Schlüssel, Funktionalitäten (Chen-Notation), Funktionale Bestimmtheit, N:M, 1:N, 1:1, N:M:P, Generalisierung

## Relationales Modell

Relation, Tupel, Relationenschema, Umsetzung von ER nach RM, Schlüssel, Fremdschlüssel

# IMDb Datengrundlage [<https://www.IMDb.com/interfaces/>]

## IMDb Datasets

Subsets of IMDb data are available for access to customers for personal and non-commercial use. You can hold local copies of this data, and it is subject to our terms and conditions. Please refer to the [Non-Commercial Licensing](#) and [copyright/license](#) and verify compliance.

### Data Location

The dataset files can be accessed and downloaded from <https://datasets.imdbws.com/>. The data is refreshed daily.

### IMDb Dataset Details

Each dataset is contained in a gzipped, tab-separated-values (TSV) formatted file in the UTF-8 character set. The first line in each file contains headers that describe what is in each column. A '\N' is used to denote that a particular field is missing or null for that title/name. The available datasets are as follows:

**title.akas.tsv.gz** - Contains the following information for titles:

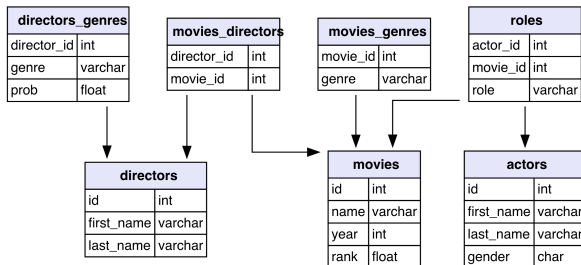
- titleId (string) - a tconst, an alphanumeric unique identifier of the title
- ordering (integer) - a number to uniquely identify rows for a given titleId
- title (string) - the localized title
- region (string) - the region for this version of the title
- language (string) - the language of the title
- types (array) - Enumerated set of attributes for this alternative title. One or more of the following: "alternative", "dvd", "festival", "tv", "video", "working", "original", "imdbDisplay". New values may be added in the future without warning
- attributes (array) - Additional terms to describe this alternative title, not enumerated
- isOriginalTitle (boolean) - 0: not original title; 1: original title

**title.basics.tsv.gz** - Contains the following information for titles:

- tconst (string) - alphanumeric unique identifier of the title
- titleType (string) - the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
- primaryTitle (string) - the more popular title / the title used by the filmmakers on

# IMDb Daten vereinfacht

[<https://relational.fit.cvut.cz/dataset/IMDb>]



- 7 Tabellen; Kopfzeile: Tabellenname, linke Spalte: Attributname, rechte Spalte Attributtyp
- T1 → T2: Attribut aus T1 referenziert Attribute in T2

**Beispiel:** directors\_genres → directors

meint:

directors\_genres.director\_id → directors.id



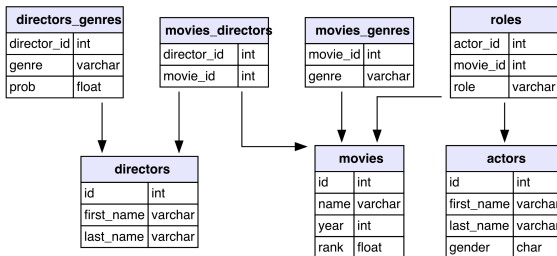
# Semantik der Tabellen

directors_genres:	zugeordnete Genre des Regisseurs
directors:	Regisseure
movies_directors:	Filme, die ein Regisseur gemacht hat
movies:	Filme
movies_genres:	zugeordnete Genre des Films
actors:	Schauspieler
roles:	Schauspieler, die in einem Film mitgespielt haben und ihre Rollen

# Ausgangssituation

## Achtung

Dies ist kein  
korrektes  
ER-Diagramm!

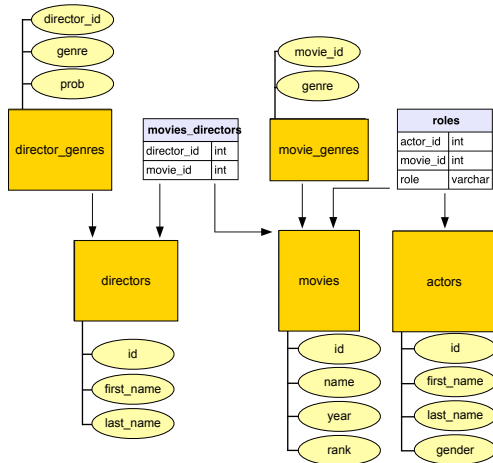


- Tabellenmodell wie auf der Webseite vorgefunden:  
<https://relational.fit.cvut.cz/dataset/IMDb>
- Das zugrundeliegende Entity-Relationship Datenmodell (ER-Modell) werden wir hieraus im folgenden rekonstruieren (reverse engineeren).

# Entitätstypen: Intuition

## Achtung

Dies ist kein  
korrektes  
ER-Diagramm!



- Entitätstypen: beschreiben jeweils eine Klasse von Entitäten, die semantisch dasselbe Konzept repräsentieren.
- Notation: Rechteck (Farbe spielt keine Rolle)

# Entitätstypen

## Entitätstypen

Ein Entitätstyp beschreibt und modelliert eine Menge von Entitäten. Jeder Entitätstyp hat einen eindeutigen Namen (am besten im Plural). Ein Entitätstyp kann mit beliebig vielen Beziehungstypen verbunden werden, aber **niemals** direkt mit anderen Entitätstypen.

### Beispiele:

- Regisseure (directors)
- Filme (movies)
- Schauspieler (actors)
- Der Entitätstyp „Regisseure“ enthält die Entität „Quentin Tarantino“.

### Vergleiche:

Klasse und Objekt (in objektorientierter Modellierung)

**vs**

Entitätstyp und Entität (in Entity-Relationship Modellierung)

# Domäne und Attribute

## Domäne (Wertebereich)

Eine Domäne ist eine Menge **atomarer** Werte. Diese Werte dürfen nicht strukturiert (d.h. weiter aufteilbar sein). Domänen werden notiert als  $D$ , z.B. integer, float, String, etc.

## Attribut

Ein Attribut beschreibt atomar einen Aspekt eines Entitäts- oder Beziehungstypen; die Domäne des Attributs wird in ER üblicherweise nicht spezifiziert, kann aber zusätzlich an das Attribut drangeschrieben werden. Ein Attribut gehört zu genau einem Entitäts- oder Beziehungstypen. Für einen Entitätstypen  $E_i$  werden seine Attribute benannt mit  $A_{i,1}, \dots, A_{i,k_i}$ . Für einen Beziehungstypen  $B$  werden seine Attribute benannt mit  $A_1^B, \dots, A_{k_B}^B$ .

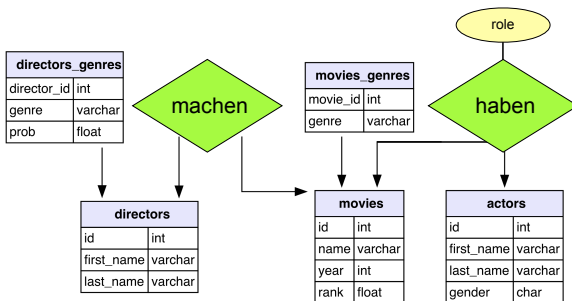
## Beispiele:

- jeder Regisseur (director) hat die Attribute *Vorname* und *Nachname*
- jeder Film (movie) hat das Attribut *Erscheinungsjahr*

# Beziehungstypen: Intuition

## Achtung

Dies ist kein  
korrektes  
ER-Diagramm!



- Beziehungstypen: drücken aus, dass zwei Entitätstypen miteinander in Verbindung (in Beziehung) stehen
- Notation: Raute (Farbe spielt keine Rolle)
- Achtung: Leserichtung nicht immer eindeutig aber häufig gilt die Konvention „von links nach rechts“

**Beispiele:** Regisseure machen Filme, Filme haben Schauspieler

# Beziehungstypen

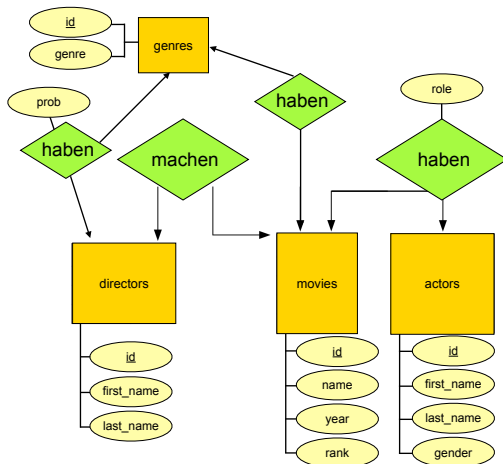
## Beziehungstypen

Ein Beziehungstyp beschreibt eine Menge von Beziehungen. Jeder Beziehungstyp hat einen eindeutigen Namen (am besten im Plural). Ein Beziehungstyp kann mit beliebig vielen Entitätstypen verbunden werden aber **niemals** direkt mit anderen Beziehungstypen. Zusätzlich kann ein Beziehungstyp eigene Attribute  $A_1^B, \dots, A_{k_B}^B$  haben.

### Beispiele:

- machen
- spielen in (haben)

# Schlüssel: Intuition



- Schlüssel(-attribute) identifizieren die Entitäten eines Entitätstypen eindeutig
- Notation: unterstreichen der Schlüsselattribute



# Schlüssel

## Schlüssel

Eine Teilmenge der (oder alle) Attribute eines Entitätstyps kann durch Unterstreichung gekennzeichnet werden. Diese Teilmenge, auch Schlüsselattribute genannt, muss jede Entität des Entitätstypen anhand dieser Attribute eindeutig identifizieren können.

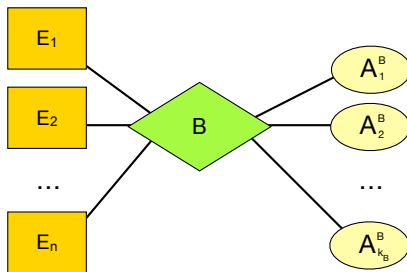
## Minimalität des Schlüssels

1. Falls ich die Teilmenge verkleinern kann und trotzdem damit alle Entitäten identifizieren kann: dann bildet die aktuelle Teilmenge **keinen** Schlüssel. D.h. ich verkleinere die Teilmenge.
2. Falls die Teilmenge nicht alle Entitäten unterscheiden kann: dann bildet die Teilmenge ebenfalls **keinen** Schlüssel. D.h. ich vergrößere die Teilmenge auf geeignete Weise.

# Zusammenspiel von Beziehungs- und Entitätstypen

## Zusammenspiel von Beziehungs- und Entitätstypen

Angenommen wir haben einen Beziehungstyp  $B$  mit  $n$  teilnehmenden Entitätstypen  $E_1, \dots, E_n$  sowie  $k_B$  eigenen Attributen  $A_1^B, \dots, A_{k_B}^B$ . Jedem Attribut sei eine Domäne zugeordnet:  $A_1^B \in D_1^B, \dots, A_{k_B}^B \in D_{k_B}^B$ . Ebenso interpretieren wir die Entitätstypen  $E_1, \dots, E_n$  als Domänen. Dann lässt sich jede Beziehung aus  $B$  als ein  $(n + k_B)$ -stelliges Tupel  $\{e_1, \dots, e_n, a_1^B, \dots, a_{k_B}^B\} \in E_1 \times \dots \times E_n \times D_1^B \times \dots \times D_{k_B}^B$  darstellen.



## Beispiel: Vierstelliger Beziehungstyp

Beziehungstyp  $B$  mit vier beteiligten Entitätstypen  $E_1, \dots, E_4$  und jeweils 5 Beziehungen.  $B$  hat ein eigenes Attribut  $A_1^B$ .

### Beispiel:

$B$				
$E_1$	$E_2$	$E_3$	$E_4$	$A_1^B$
4	8	1	9	c
6	4	1	3	d
8	6	2	9	j
1	8	<b>1</b>	9	c
1	8	<b>7</b>	9	d

# Funktionale Bestimmtheit

## Funktionale Bestimmtheit

Angenommen, wir haben einen Beziehungstyp  $B = E_1 \times \dots \times E_n$ . Weiterhin stehe an Entitätstyp  $E_i, 1 \leq i \leq n$  eine „1“. Dann muss für jedes Paar von Beziehungen  $b_1 = \{e_1, \dots, e_n\} \in B$  und  $b_2 = \{e'_1, \dots, e'_n\} \in B$  gelten: falls  $(\forall_{1 \leq k \neq i \leq n} e_k = e'_k) \Rightarrow e_i = e'_i$ . Da  $B$  eine Menge ist folgt  $b_1 = b_2$ .

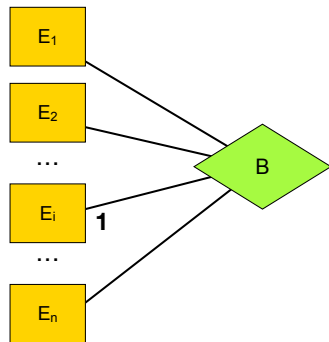
Mit anderen Worten:

falls in beiden Beziehungen  $b_1$  und  $b_2$  alle Entitäten der Entitätstypen

$E_1, \dots, E_{i-1}, E_{i+1}, \dots, E_n$ , d.h. alle außer  $E_i$  gleich sind, dann impliziert dies, dass  $e_i$  und  $e'_i$  dieselbe Entität referenzieren müssen.

## Leseweise

„ $E_1, \dots, E_{i-1}, E_{i+1}, \dots, E_n$  bestimmen  $E_i$ “  
oder einfach: „ $E_i$  wird funktional bestimmt.“



## Beispiel: Vierstelliger Beziehungstyp mit Funktionalitäten

Beziehungstyp  $B$  mit vier beteiligten Entitätstypen  $E_1, \dots, E_4$  und jeweils 5 Beziehungen.  $B$  hat keine eigenen Attribute.

**Beispiel:** ( $i = 3$ )

$B$			
$E_1$	$E_2$	$E_3$	$E_4$
4	8	1	9
6	4	1	3
8	6	2	9
1	8	<b>1</b>	9
1	8	<b>7</b>	9

Nehmen wir an, dass an  $E_3$  im ER eine 1 steht,  $E_3$  wird also laut ER funktional bestimmt.

Die Daten stehen im Widerspruch dazu:  $\{1, 8, 1, 9\}$  und  $\{1, 8, 7, 9\}$  haben für alle Attribute außer  $E_3$  denselben Wert.

**Beispiel:** ( $i = 2$ )

$B$			
$E_1$	$E_2$	$E_3$	$E_4$
4	8	1	9
6	4	1	3
8	6	2	9
1	8	1	9
1	7	2	9

Nehmen wir an, dass an  $E_2$  im ER eine 1 steht,  $E_2$  wird also laut ER funktional bestimmt.

Die Daten stehen nicht im Widerspruch dazu.

# Funktionale Bestimmtheit und Chen-Notation im ER-Diagramm

## Funktionalität (auch Chen-Notation genannt)

Funktionalitäten werden zwischen Entitäts- und Beziehungstyp notiert. Die Leserichtung dieser Notation ist zum Entitätstypen hin. D.h. diese Notation macht eine Aussage darüber, ob andere an diesem Beziehungstyp teilnehmenden Entitätstypen diesen Entitätstyp funktional bestimmen.

## Achtung

Es gibt ganz unterschiedliche Notationen für Funktionale Bestimmtheit, eine wichtige Alternative, die wir im Rahmen dieser VL nicht verwenden werden, ist die (min,max)-Notation, siehe

<https://youtu.be/oxcaqMQdGyw>.

# Funktionale Bestimmtheit und Unbestimmtheit im ER-Diagramm

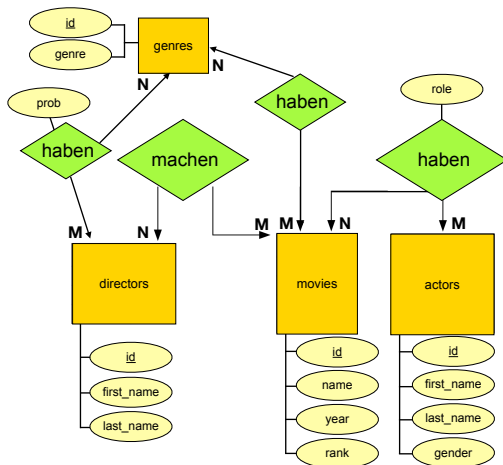
## Funktionale Unbestimmtheit im ER-Diagramm

Ein „N“ oder „M“ bedeutet: Die Entitäten dieses Entitätstypen werden durch die Kombination der anderen Entitätstypen **nicht funktional bestimmt**.

## Funktionale Bestimmtheit im ER-Diagramm

Eine „1“ bedeutet: Die Entitäten dieses Entitätstypen werden durch die Kombination der anderen Entitätstypen **funktional bestimmt**.

# Funktionalitäten



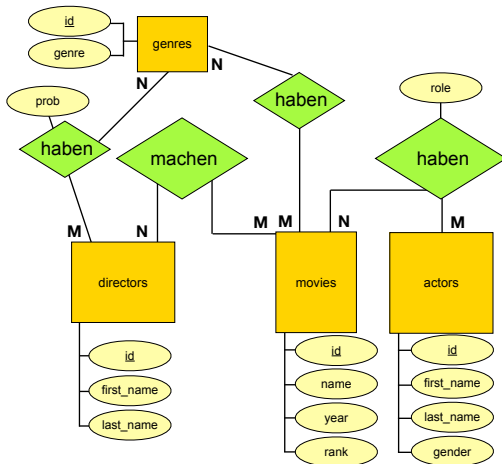
## Achtung

Dies ist kein korrektes ER-Diagramm!

Funktionalitäten zeigen an, ob die Entitäten eines Entitätstyps von anderen Entitätstypen funktional bestimmt werden. Hier stehen nur Ns und Ms, d.h. hier wird nichts funktional bestimmt.

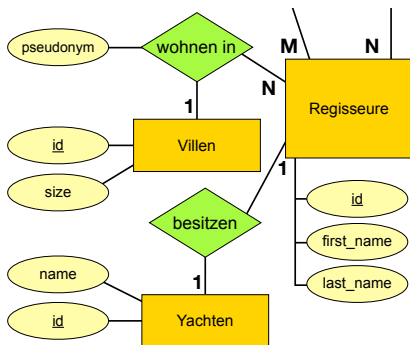


# Funktionalitäten (Notation ohne Pfeile)



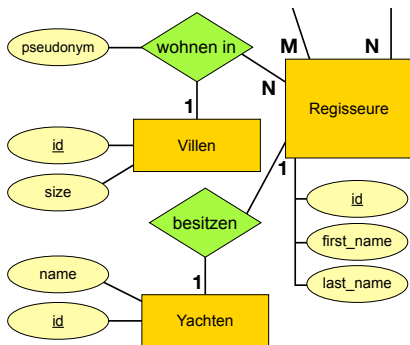
Die Pfeile lassen wir im Folgenden weg, da sie sonst zu Verwirrung mit anderen Notationselementen führen können.

# 1:N-Beziehung



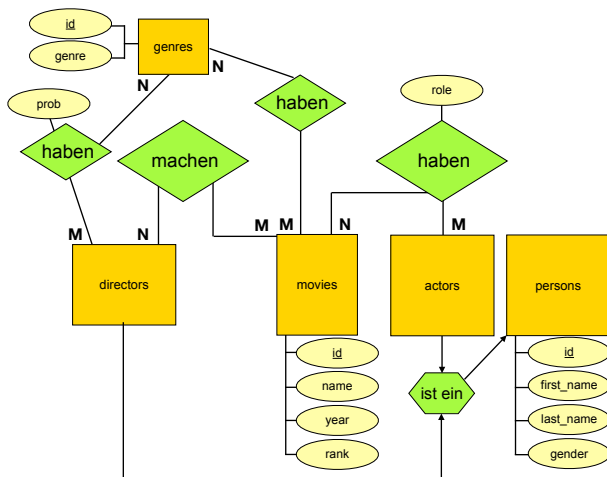
- Regisseure bestimmen Villen: wenn ich eine Entität von Regisseure habe, weiß ich, ob und falls ja in welcher Villa er/sie wohnt
- Villen bestimmt **nicht** Regisseure: wenn ich eine Entität von Villen habe, weiß ich nicht, welcher Regisseur dort wohnt.
- D.h. in derselben Villa können mehrere Regisseure wohnen
- ein Regisseur wohnt in **maximal** einer Villa
- Villen dürfen leerstehen und Regisseure dürfen obdachlos sein!

# 1:1-Beziehung








- Regisseure bestimmen Yachten: wenn ich eine Entität von Regisseure habe, weiß ich, ob und falls ja welche Yacht er/sie besitzt
- Yachten bestimmen Regisseure: wenn ich eine Entität von Yachten habe, weiß ich, ob und falls ja welchem Regisseur sie gehört
- D.h. dieselbe Yacht gehört **maximal** einem Regisseure
- ein Regisseur besitzt **maximal** eine Yacht
- Yachten dürfen regisseurlos und Regisseure dürfen yachtlos sein!

# Generalisierung



- Ähnlich wie in der objektorientierten Modellierung sollten überlappende Konzepte nicht unabhängig voneinander modelliert werden. Dadurch werden Redundanzen vermieden.
- Unterklassen erben alle Attribute ihrer Oberklasse.

# Übersicht über die Entity Relationship Modellierungselemente

Symbol	Bedeutung
	Entitätstyp
	Beziehungstyp
1, N oder M	Funktionalität (Chen-Notation)
	Attribut
	Schlüssel(-attribut)
	Vererbung

Die Farben der Symbole spielen keine Rolle.

# Das Relationale Modell<sup>1</sup>

## Relation

Eine Relation ist die Teilmenge des Kreuzprodukts über  $n$  Domänen. D.h. eine Relation  $R$  ist definiert als  $R \subseteq D_1 \times \dots \times D_n$ .

$R$  heißt **Ausprägung** oder **Instanz**.

## Tupel

Tupel heißt jedes Element  $t = (a_1, \dots, a_n) \in R, a_{1 \leq i \leq n} \in D_i$ . Die einzelnen  $a_i$  heißen **Attributwerte** oder **Attribute**.

# Relationenschema

## Relationenschema

Ein Relationenschema spezifiziert Domänen **und** Attributnamen einer Relation. D.h. im Unterschied zur Definition der Relation spezifizieren wir nicht nur die Domänen sondern zusätzlich die Attributnamen. Ein Relationenschema wird notiert als **Sequenz**  $[R] : \{[A_1 : D_1, \dots, A_n : D_n]\}$ . Die **Schlüsselattribute** werden durch Unterstreichen gekennzeichnet.

### Beispiele:

[movies] : {[ id:int, name:varchar, year:int , rank:float ]}

[machen] : {[ director\_id:int, movie\_id:int ]}

## Reihenfolge von Attributen und Tupeln

In welcher Reihenfolge wir die Attribute eines Relationenschemas definieren **spielt keine Rolle**. Ebenso spielt die „Reihenfolge“ der Tupel in der Relation keine Rolle: in einer Menge ist keine Reihenfolge definiert. Die Reihenfolge von Attributen und Tupeln ist **nicht** informationstragend.

# Umsetzung von ER ins Relationale Modell (1/2)

## Umsetzung von Entitätstypen

Jeder Entitätstyp  $E_i$  mit Attributen  $A_{i,1}, \dots, A_{i,k_i}$  wird folgendermaßen umgesetzt. Seien  $\text{keyset}(E_i) \subseteq \{A_{i,1}, \dots, A_{i,k_i}\}$  die Schlüsselattribute von  $E_i$  und das Komplement  $\text{nonkeyset}(E_i) = \{A_{i,1}, \dots, A_{i,k_i}\} \setminus \text{keyset}(E_i)$  die Nichtschlüsselattribute von  $E_i$ . Dann wird  $E_i$  umgesetzt in ein Relationenschema

$$[E_i] : \{ [ \underbrace{\text{keyset}(E_i)}_{\substack{\text{Schlüsselattribute von } E_i \\ \text{Schlüssel von } [E_i]}}, \underbrace{\text{nonkeyset}(E_i)}_{\text{Nichtschlüsselattribute von } E_i} ] \}.$$

Hierbei sind die Domänen  $D_1, \dots, D_n$  geeignet zu wählen.

### Beispiel:

Entitätstyp Schauspieler (actors):

$[\text{actors}] : \{ [ \text{id:int}, \text{first\_name:varchar}, \text{last\_name:varchar}, \text{gender:char} ] \}$





# Umsetzung von ER ins Relationale Modell (2/2)

## Umsetzung von Beziehungstypen

Jeder Beziehungstyp  $B$  mit  $n$  teilnehmenden Entitätstypen  $E_1, \dots, E_n$  sowie  $k_B$  eigenen Attributen  $A_1^B, \dots, A_{k_B}^B$  kann ganz allgemein folgendermaßen umgesetzt werden: Seien  $\text{keyset}(E_i) \subseteq \{A_{i,1}, \dots, A_{i,k_i}\}$  die Schlüsselattribute von  $E_i$ . Dann ist das Relationenschema  $[B]$  ganz allgemein gegeben durch:

$$[B] : \left\{ \underbrace{\left[ \underbrace{\text{keyset}(E_1)}_{\text{Schlüsselattribute von } E_1}, \dots, \underbrace{\text{keyset}(E_n)}_{\text{Schlüsselattribute von } E_n} \right]}_{SO=\text{Schlüsselobermenge von } [B]}, \underbrace{A_1^B : D_1^B, \dots, A_{k_B}^B : D_{k_B}^B}_{\text{Attribute von } B} \right\}$$

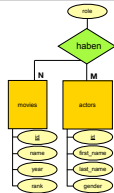
### Beispiel:

„Filme **haben** Schauspieler“:

$[\text{haben}] : \{ [ \underline{\text{movie\_id:int}}, \text{actors\_id:int}, \text{role:str} ] \}$

**oder alternative Modellierung:**

$[\text{haben}] : \{ [ \underline{\text{movie\_id:int}}, \text{actors\_id:int}, \text{role:str} ] \}$





## Dr. Strangelove oder: Wie ich lernte, die Bombe zu lieben (1964)

Edit

## Full Cast &amp; Crew

[See agents for this cast & crew on IMDbPro](#)

## Directed by

Stanley Kubrick

## Writing Credits

Stanley Kubrick

Terry Southern

Peter George

Peter George

## Achtung

Peter Sellers hatte in Dr. Strangelove **drei** verschiedene Rollen!  
Deswegen brauchen wir hier die alternative Modellierung:

[haben] : {[ movie\_id:int, actors\_id:int, role:str ]}

Da dies natürlich prinzipiell in jedem Film vorkommen kann, muss die Relation 'haben' so modelliert werden.

## Cast (in credits order) verified as complete



Peter Sellers

Group Capt. Lionel Mandrake / President Merkin Muffley / Dr. Strangelove



George C. Scott

Gen. 'Buck' Turgidson



Sterling Hayden

Brig. Gen. Jack D. Ripper



Keenan Wynn

Col. 'Bat' Guano



Slim Pickens

Maj. 'King' Kong

## Dr. Seltsam oder: Wie ich lernte, die Bombe zu lieben

## Details

[Full Cast and Crew](#)[Release Dates](#)

## Related lists from IMDb users



## 1960s

a list of 29 titles  
created 29 Oct 2017

## must see

a list of 37 titles  
created 1 day agoThe 25 Best War  
Movies of All Timea list of 25 titles  
created 29 Dec 2016

# Vereinfachte Umsetzung von 1:N-Beziehungstypen

## Umsetzung von 1:N-Beziehungstypen und Fremdschlüssel

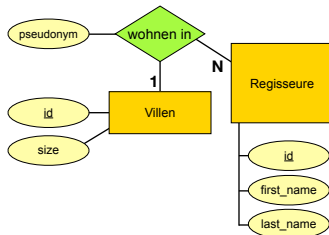
Jeder binäre Beziehungstyp  $B$  mit den teilnehmenden Entitätstypen  $E_1, E_2$ , der Funktionalität 1:N, sowie  $k_B$  eigenen Attributen  $A_1^B, \dots, A_{k_B}^B$  kann ganz allgemein umgesetzt werden, ohne eine eigene Relation für  $B$  zu erstellen. Die Relation des **bestimmenden** Entitätstypen  $E_2$  wird erweitert um einen **Fremdschlüssel** auf  $E_1$  und die Attribute  $A_1^B, \dots, A_{k_B}^B$ .

### Beispiel:

„Regisseure **wohnen in** Villen“:

[directors] : {[ id:int, first\_name:varchar,  
last\_name:varchar, villa\_id:(Villen→id:int),  
pseudonym: str ]}

In dem Beispiel ist villa\_id:(Villen→id:int) ein Fremdschlüssel auf den Schlüssel id der Relation Villen.



# Vereinfachte Umsetzung von 1:1-Beziehungstypen

## Umsetzung von 1:1-Beziehungstypen

Dies ist ein Spezialfall der Umsetzung von 1:N-Beziehungstypen. Es wird entweder der Entitätstypen  $E_2$  erweitert um einen **Fremdschlüssel** auf  $E_1$  und die Attribute  $A_1^B, \dots, A_{k_B}^B$  **oder**  $E_1$  um einen **Fremdschlüssel** auf  $E_2$  und die Attribute  $A_1^B, \dots, A_{k_B}^B$ .

**Beispiel:**

**Entweder:**

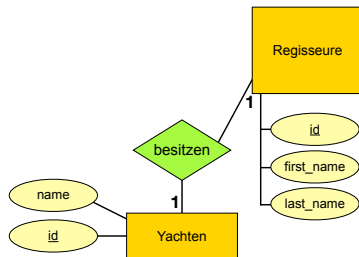
„Regisseure **besitzen** Yachten“:

[directors] : {[ id:int, first\_name:varchar,  
last\_name:varchar, yacht\_id:(Yachten→id:int)

**oder:**

[Yachten] : {[ id:int, name:varchar,  
directors\_id:(Directors→id:int) ]}

**Aber nicht beides!**



# Das Relationale Modell in Python (relation.py von uns)

## The Relational Model in Python

Copyright Jens Dittrich & Marcel Maltry, [Big Data Analytics Group](#), CC-BY-SA

```
In [1]: from ra.csv_utils import load_csv
        from ra.relation import Relation

In [2]: foo = Relation('foo', [('id', int), ('name', str)])
        foo.add_tuple( (2, 'Hello') )
        foo.add_tuple( (7, 'World') )
        foo.add_tuple( (1, '!') )

        foo.print_table()
```

```
---
foo
-----
id    name
-----
7      World
2      Hello
1       !
```

**Again:** Neither the order of rows nor the order of columns carry any meaning in a relation!

- <https://github.com/BigDataAnalyticsGroup/bigdataengineering/blob/master/Relational%20Model.ipynb>
- Anmerkung: wir benutzen die Originalversion der Daten von <https://relational.fit.cvut.cz/dataset/IMDb> und nicht unsere verbesserte Version des Modells.

# IMDb (Teil 1)

## 4. Transfer der Grundlagen auf die konkrete Anwendung

siehe diverse Beispiele zu IMDb oben, zusammengefasst:

### Frage 1

Wie werden in IMDb die Daten zu Filmen, Schauspielern, Regisseuren usw. modelliert und abgelegt?

Modellierung: Entity-Relationship-Modell, relationales Modell

Datenablage: Dies ist ein Freiheitsgrad, über den wir im Moment keine Entscheidung treffen müssen: jede Form der Datenablage, die Konform ist mit dem Relationalen Modell, kann gewählt werden!

Diese Eigenschaft nennt man **Physische Datenunabhängigkeit**, d.h. als Nutzer müssen wir nichts darüber wissen, wie die Daten physisch abgelegt und repräsentiert werden. Mit anderen Worten: uns kann egal sein, wie das relationale Modell implementiert wird<sup>2</sup>.

# IMDb (Teil 1)

## Frage 2

Wie werden in IMDb Verknüpfungen dieser Daten modelliert und abgelegt?

durch Fremdschlüsselbeziehungen im relationalen Modell  
wie auch immer diese physisch... (siehe oben)


nächste Woche:

## Frage 3

Wie stellen wir Anfragen an diese Daten?

...

# Weiterführendes Material



Entity Relationship-Modellierung









10 Videos • 59.797 Aufrufe • Zuletzt am 27.01.2014 aktualisiert

Öffentlich

Grundlagen der Entity-Relationship-Modellierung. ER

Prof. Dr. Jens Dittrich

SORTIEREN NACH

-  13.07 Übersicht über die Modellierungsschritte: von der Realität zum Datenbankschema  
Prof. Dr. Jens Dittrich
-  13.08 Entity Relationship Modellierung: Grundlagen, Funktionalitäten, Rollen, Rekursion  
Prof. Dr. Jens Dittrich
-  13.09 Entity Relationship Modellierung II: Chen versus min/max, mehrstellige Beziehungstypen  
Prof. Dr. Jens Dittrich
-  13.10 Entity Relationship Modellierung III: schwache Entitätstypen, N:M, Generalisierung, Teil-von  
Prof. Dr. Jens Dittrich
-  13.11 Entity Relationship Modellierung IV: Sichtenkonsolidierung  
Prof. Dr. Jens Dittrich
-  13.12 Das relationale Modell: Relationen, Domänen, Relationenschema, Schlüssel  
Prof. Dr. Jens Dittrich
-  13.13 Umsetzung ER nach Relationalem Modell: Grundlagen, binäre und mehrstellige Beziehungstypen  
Prof. Dr. Jens Dittrich
-  13.14 Verfeinerung des Relationalen Modells: Zusammenfassen von Relationen  
Prof. Dr. Jens Dittrich

Youtube Videos von Prof. Dittrich zu ER-Modellierung  
sowie Kapitel in Kemper&Eickler