

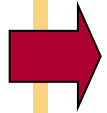
Datenbanksysteme I  
Relationaler Datenbankentwurf

14.4.2014

Dr. Gjergji Kasneci (Folien von Prof. Dr. Felix Naumann)

# Überblick

2



- Motivation und Einbettung
- Wiederholung: ER-Diagramme
- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Konvertierung von Spezialisierung
- Funktionale Abhängigkeiten (FDs)
- Ableitungsregeln für FDs
- Normalformen



# Motivation

3

- Datenbankdesign als Skizze mittels ER-Modellierung
  - Welche Daten? Welche Beziehungen?
  - Nicht wie Daten erzeugt oder verändert werden.
- Skizzen heißen „Entity-Relationship-Diagramme“
  - ER-Diagramme
- In 1 Woche: Überführung von ER-Diagrammen in das relationale Modell
  - Relationen, Attribute, Integritätsbedingungen
- In 2 Wochen: Überführung von Relationenschemata in SQL Ausdrücke
  - **CREATE TABLE ...**

# Entwurfsaufgabe

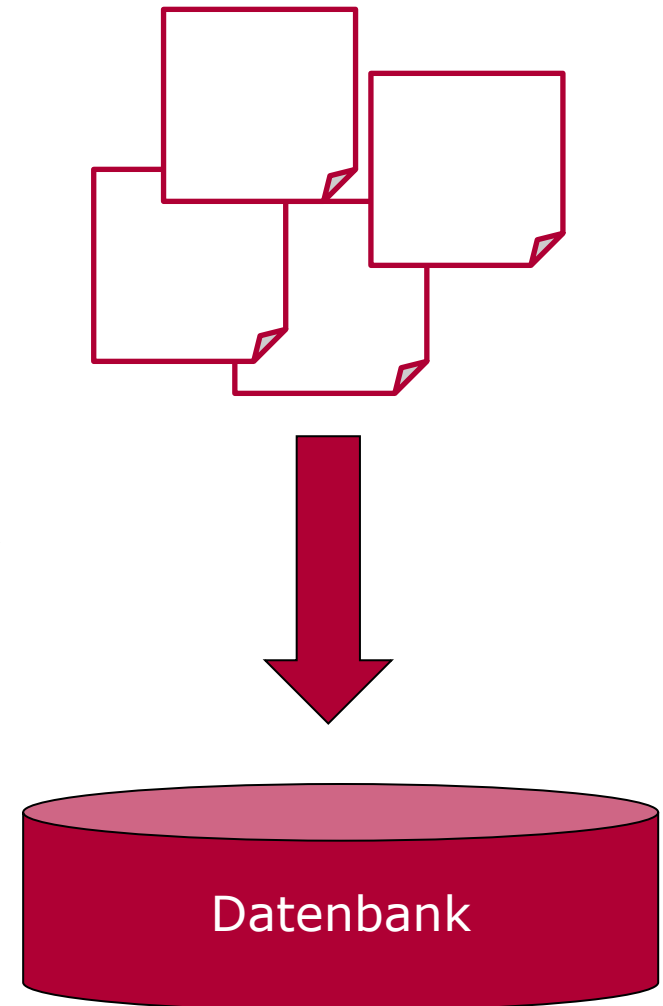
4

- Datenhaltung für **mehrere** Anwendungssysteme und **mehrere** Jahre
- daher: besondere Bedeutung
- Anforderungen an Entwurf
  - Anwendungsdaten jeder Anwendung sollen aus Daten der Datenbank ableitbar sein.
    - ◇ Möglichst effizient
  - Nur „vernünftige“ (wirklich benötigte) Daten sollen gespeichert werden.
  - Nicht-redundante Speicherung

# Entwurfsprozess

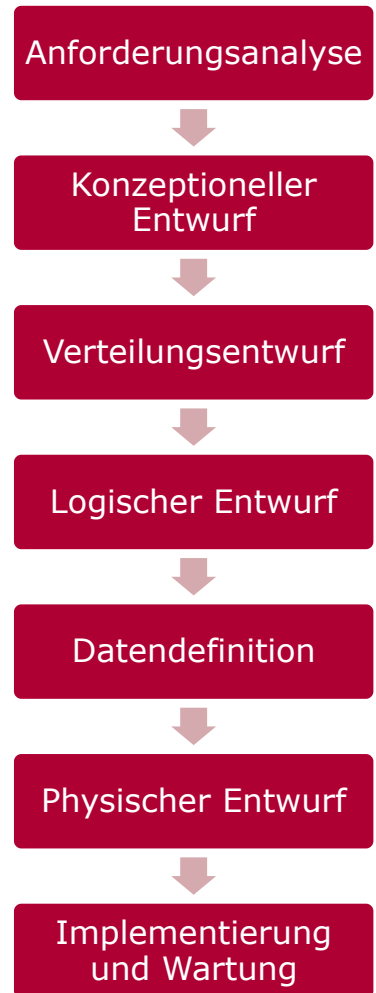
5

- Abfolge von Entwurfsdokumenten
  - Von abstrakter Beschreibung
  - Bis tatsächlichen Realisierung in einem DBMS
  - Verschiedene Beschreibungsformalismen
    - ◇ ER, Relationenmodell, SQL DDL, usw.
  
- In jedem Schritt
  - Informationserhaltung
  - Konsistenzerhaltung



6

1. Anforderungsanalyse
  - Informale Beschreibung
2. Konzeptioneller Entwurf
  - Entity-Relationship-Diagramm
3. Verteilungsentwurf
  - Partitionierung
4. Logischer Entwurf
  - Relationenschemata
5. Datendefinition
  - SQL DDL
6. Physischer Entwurf
  - Parameter und Indizes
7. Implementierung und Wartung
  - Installationen, Backups, etc.



# Anforderungsanalyse

7

- Vorgehensweise
  - Sammlung des Informationsbedarfs in den Fachabteilungen
- Ergebnis
  - Informale Beschreibung des Fachproblems
    - ◇ Texte, tabellarische Aufstellungen, Formblätter, ...
  - Trennen der Information über Daten (Datenanalyse) von den Information über Funktionen (Funktionsanalyse)
- „Klassischer“ DB-Entwurf
  - Nur Datenanalyse und Folgeschritte
- Funktionsentwurf
  - Siehe Methoden des Software Engineering



# Konzeptioneller Entwurf

8

- Erste formale Beschreibung des Fachproblems
  - Diskurswelt (*Universe of Discourse*)
- Sprachmittel: semantisches Datenmodell
  - ER-Modell (Entity-Relationship-Modell)
- Vorgehensweise
  - Modellierung von Sichten z.B. für verschiedene Fachabteilungen
  - Analyse der vorliegenden Sichten in Bezug auf Konflikte
    - ◇ Namenskonflikte (Synonyme, Homonyme)
    - ◇ Typkonflikte
    - ◇ Bedingungskonflikte
    - ◇ Strukturkonflikte
  - Integration der Sichten in ein Gesamtschema
- Ergebnis: konzeptionelles Gesamtschema, z.B.(E)ER-Diagramm

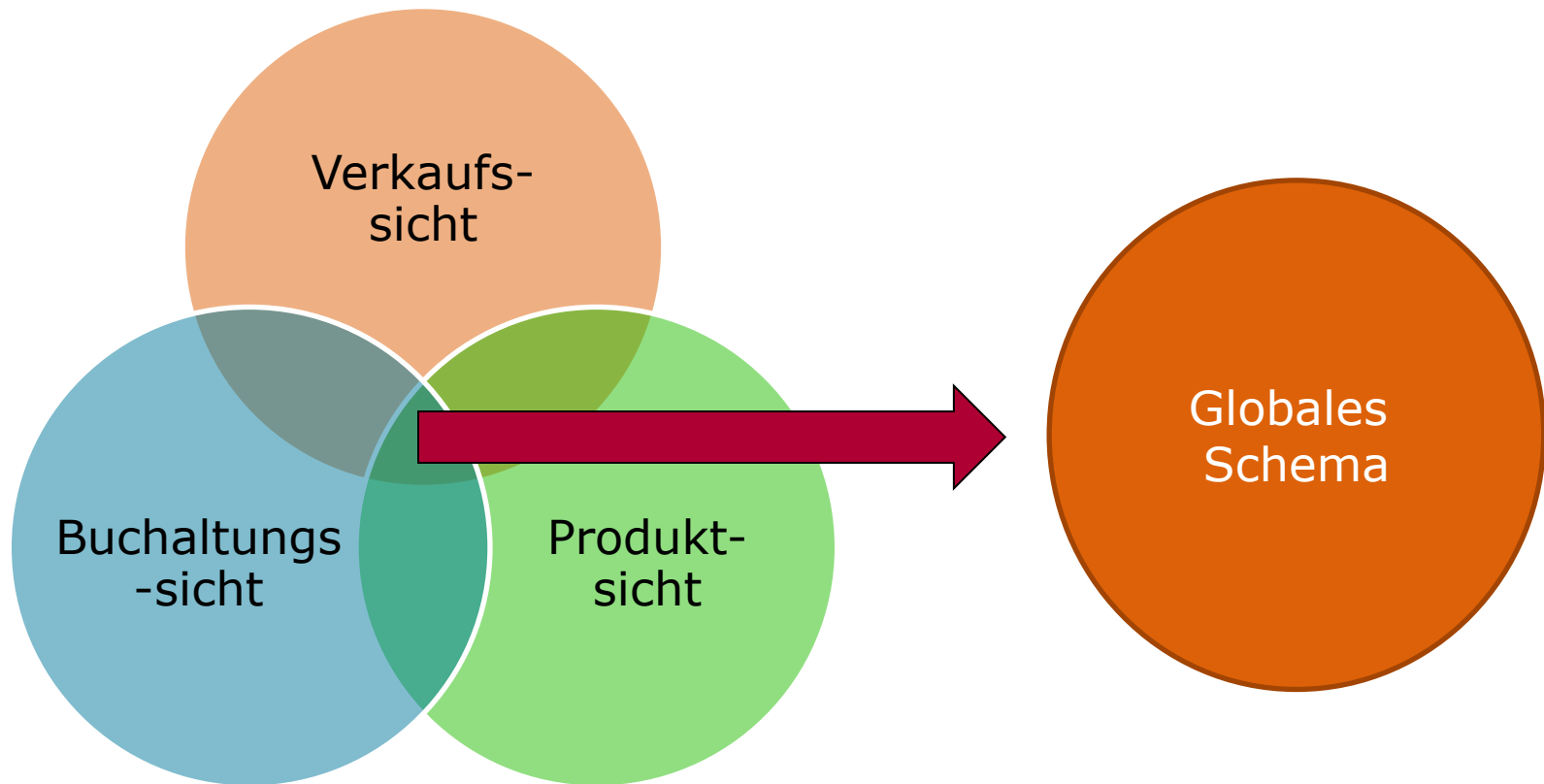




# Sichtenintegration

9

- Analyse der vorliegenden Sichten in Bezug auf Konflikte
- Integration der Sichten in ein Gesamtschema



# Integrationskonflikte

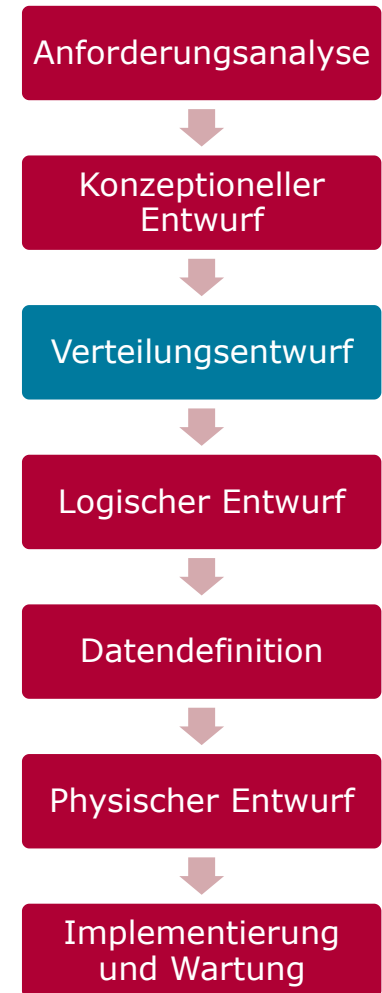
10

- Namenskonflikte
  - Homonyme: Schloss, Kunde
  - Synonyme: Auto, KFZ, Fahrzeug
- Typkonflikte
  - Verschiedene Strukturen für das gleiche Element
- Wertebereichskonflikte
  - Verschiedene Wertebereiche für ein Element
- Bedingungskonflikte
  - z.B. verschiedene Schlüssel für ein Element
- Strukturkonflikte
  - Gleicher Sachverhalt durch unterschiedliche Konstrukte ausgedrückt

# Verteilungsentwurf (Partitionierung)

11

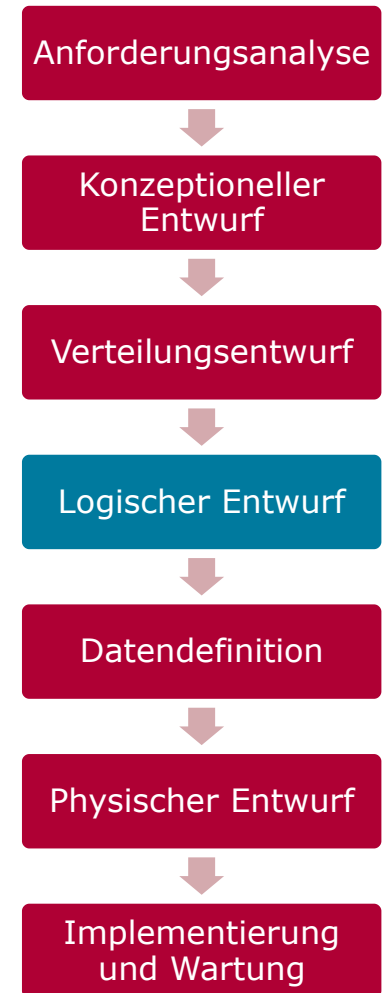
- Sollen Daten auf mehreren Rechnern verteilt vorliegen, muss Art und Weise der verteilten Speicherung festgelegt werden.
- z.B. bei einer Relation
  - `KUNDE (KNr, Name, Adresse, PLZ, Konto)`
  - Horizontale Partitionierung
    - ◇ `KUNDE_1 (KNr, Name, Adresse, PLZ, Konto)`  
`where PLZ < 50.000`
    - ◇ `KUNDE_2 (KNr, Name, Adresse, PLZ, Konto)`  
`where PLZ >= 50.000`
  - Vertikale Partitionierung (Verbindung über KNr Attribut)
    - ◇ `KUNDE_Adr (KNr, Name, Adresse, PLZ)`
    - ◇ `KUNDE_Konto (KNr, Konto)`
- Nicht in diesem Kurs



# Logischer Entwurf

12

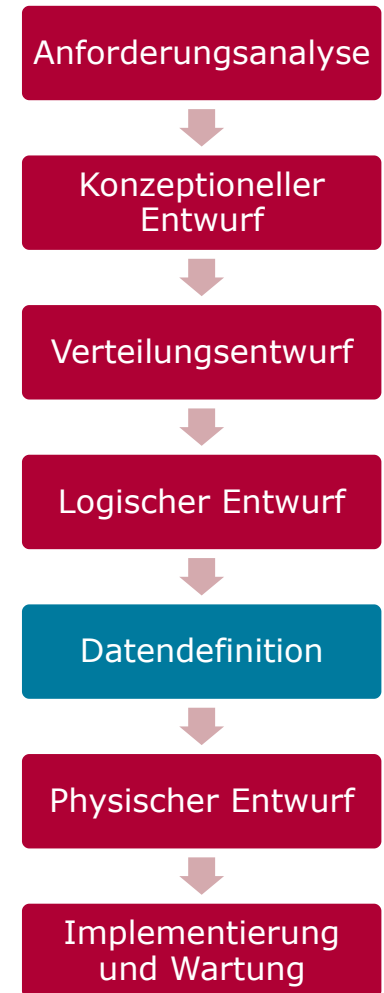
- Sprachmittel: Datenmodell des ausgewählten DBMS
  - z.B. DB2, Oracle, ... => relationales Modell
  - Tamino => XML
- Vorgehensweise
  - (Automatische) Transformation des konzeptionellen Schemas
    - ◇ z.B. ER in relationales Modell
  - Verbesserung des relationalen Schemas anhand von Gütekriterien
    - ◇ Normalisierung, Redundanzvermeidung, ...
- Ergebnis: logisches Schema, z.B. Sammlung von Relationenschemata



# Datendefinition

13

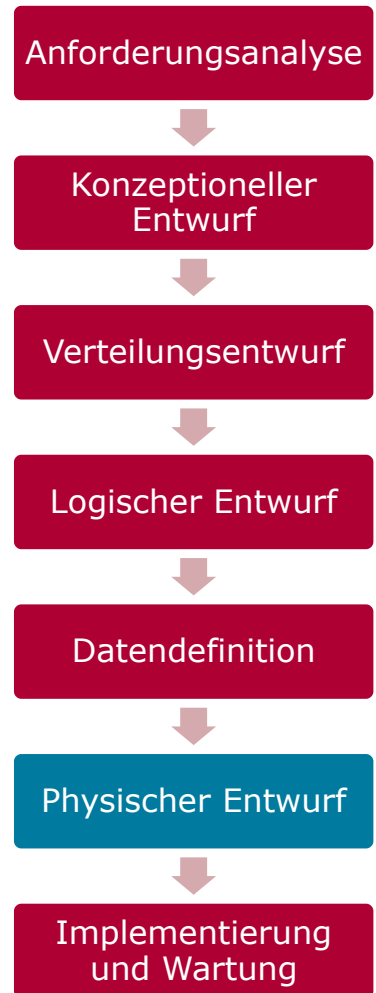
- Umsetzung des logischen Schemas in ein konkretes Schema
- Sprachmittel:
  - DDL (*data definition language*) und DML (*data manipulation language*) eines DBMS
    - ◇ z.B. Oracle, DB2, SQL Server, ...
  - Datenbankdeklaration in der DDL des DBMS
    - ◇ **CREATE TABLE ...**
  - Realisierung der Integritätssicherung
    - ◇ Schlüssel, Fremdschlüssel, Nebenbedingungen, Datentypen
  - Definition von Benutzersichten
    - ◇ **CREATE VIEW ...**



# Physischer Entwurf

14

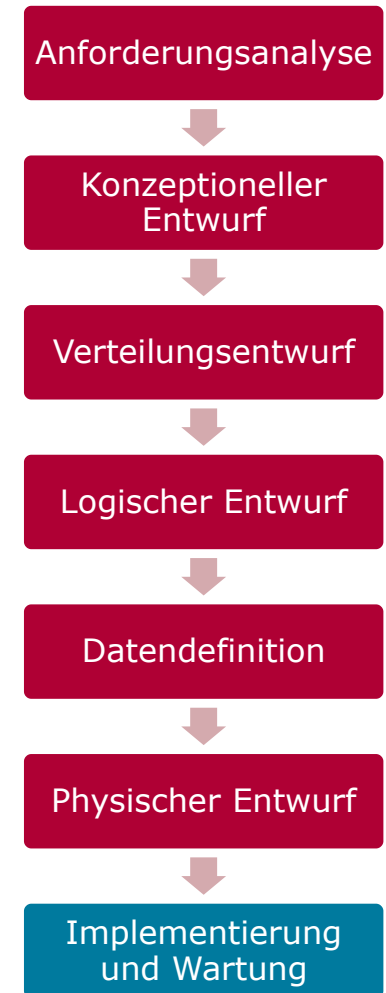
- DBMS nimmt automatisiert physischen Entwurf vor.
- Ergänzen um Zugriffsunterstützung zur Effizienzverbesserung
  - z.B. Definition von Indizes
  - **CREATE INDEX ...**
- Index
  - Datenstruktur für effizienten, Suchschlüssel-basierten Zugriff auf Datensätze (<Schlüsselattributwert, Tupeladresse>)
  - Meist als Baumstruktur realisiert
- Beispiel
  - Tabelle mit 10 GB Daten
  - Festplattentransferrate ca. 50 MB/s
  - Operation: Suchen einer Bestellung (Selektion)
  - Implementierung: sequentielles Durchsuchen
  - Aufwand:  $10.240/50 = 205 \text{ sec.} = 3,5 \text{ min.}$
- Nicht in dieser VL



# Implementierung und Wartung

15

- **Wartung des DBMS**
  - Parameter, Festplatten, etc.
- **Datenbank Tuning**
  - Weitere Optimierung der physischen Ebene
- **Anpassung an neue Anforderungen**
- **Anpassung an neue Systemplattformen**
- **Portierung auf neue Datenbankmanagementsysteme**
  
- **Kostenaufwändigste Phase**
- **Software Engineering**
  
- **Nicht in dieser VL**

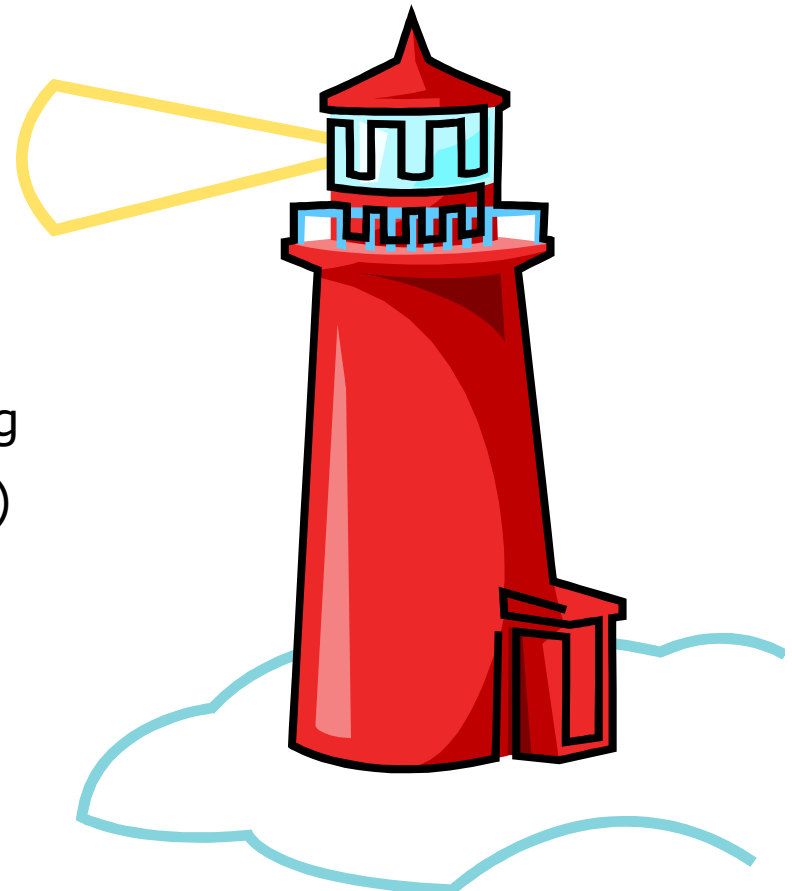


# Überblick

16

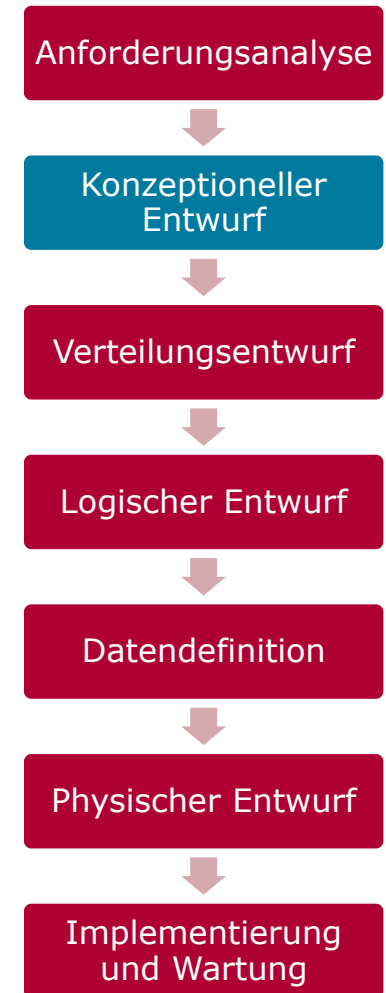


- Motivation und Einbettung
- Wiederholung: ER-Diagramme
- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Konvertierung von Spezialisierung
- Funktionale Abhängigkeiten (FDs)
- Ableitungsregeln für FDs
- Normalformen





1. Anforderungsanalyse
  - Informale Beschreibung
2. **Konzeptioneller Entwurf**
  - **Entity-Relationship-Diagramm**
3. Verteilungsentwurf
  - Partitionierung
4. Logischer Entwurf
  - Relationenschemata
5. Datendefinition
  - SQL DDL
6. Physischer Entwurf
  - Parameter und Indizes
7. Implementierung und Wartung
  - Installationen, Backups, etc.



# Das Entity-Relationship-Modell

18

- Nach Peter P. Chen 1976
  - The entity-relationship model – towards a unified view of data.  
ACM TODS
- Standardmodell in der frühen Entwurfsphase

## The Entity-Relationship Model—Toward a Unified View of Data

PETER PIN-SHAN CHEN

Massachusetts Institute of Technology

---

A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of database design and description using the model and the diagrammatic technique is given. Some implications for data integrity, information retrieval, and data manipulation are discussed.

The entity-relationship model can be used as a basis for unification of different views of data: the network model, the relational model, and the entity set model. Semantic ambiguities in these models are analyzed. Possible ways to derive their views of data from the entity-relationship model are presented.

Key Words and Phrases: database design, logical view of data, semantics of data, data models, entity-relationship model, relational model, Data Base Task Group, network model, entity set model, data definition and manipulation, data integrity and consistency

CR Categories: 3.50, 3.70, 4.33, 4.34

---



# Ein erstes ER Diagramm

19

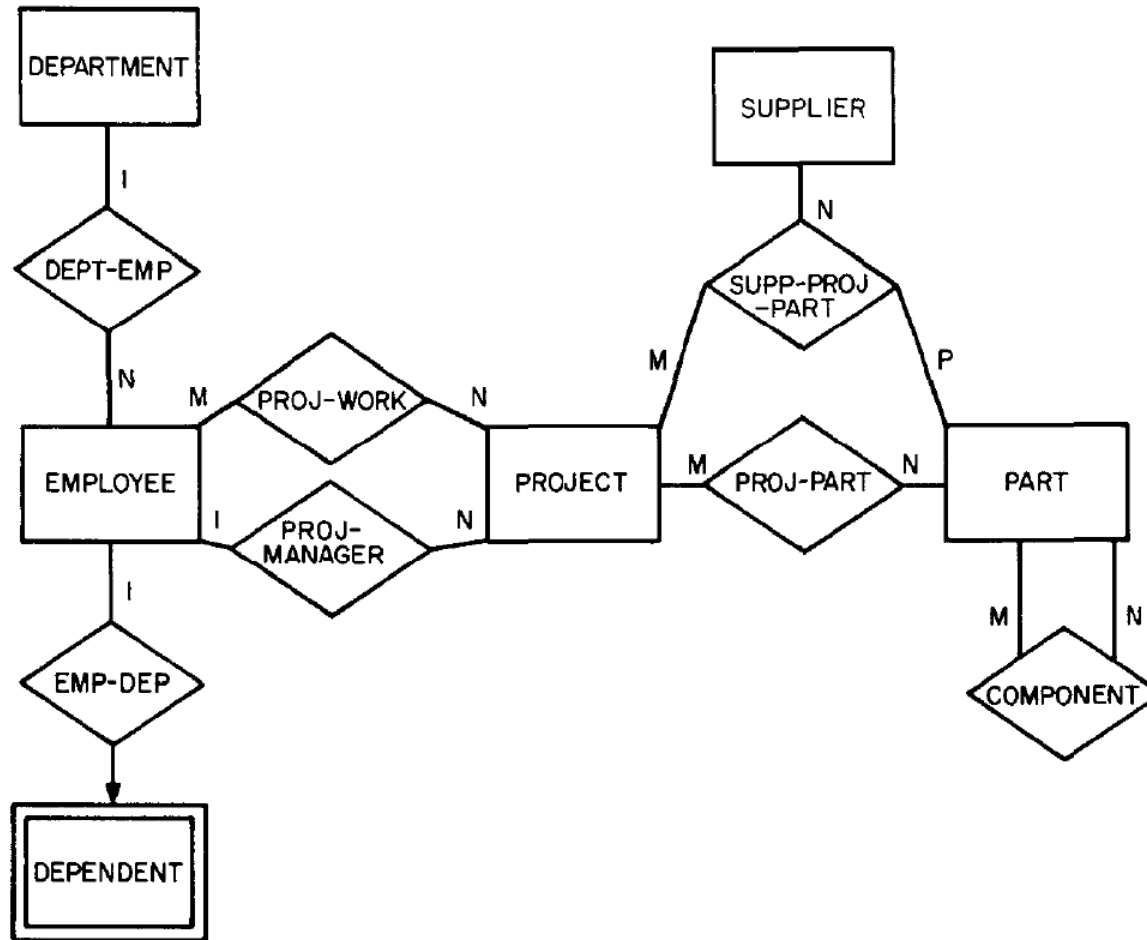


Fig. 11. An entity-relationship diagram for analysis of information in a manufacturing firm

Quelle: Peter P. Chen, The Entity-Relationship Model. ACM Transactions on Database Systems 1(1) 1976

# Begriffe

20

- Entity
  - Ein Ding / Objekt der realen oder der Vorstellungswelt
  - Nicht direkt darstellbar, sondern nur über Eigenschaften beobachtbar
- Entitytyp (*entity set*)
  - Eine Klasse für gleichartige Objekte
- Relationship
  - Beschreibt Beziehungen zwischen Entities
  - Meist binär
- Relationshiptyp
  - Eine Klasse für gleichartige Beziehungen
- Attribut
  - repräsentiert eine Eigenschaft von Entities oder von Relationships
  - Zunächst nur primitive Datenwerte (String, Integer, ...) und Operationen darauf

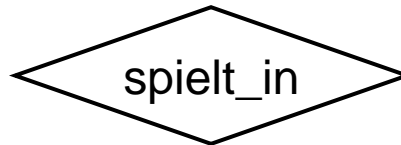
# Entity-Relationship-Diagramm

21

- Entitytyp: Rechteck



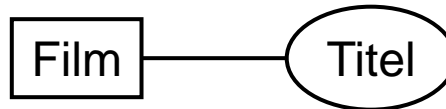
- Relationshiptyp: Raute



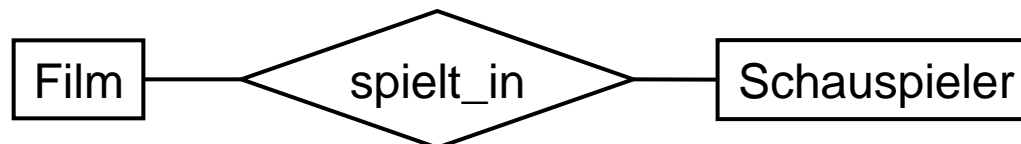
- Attribut: Oval



- Kanten verbinden Entitytypen mit Attributen

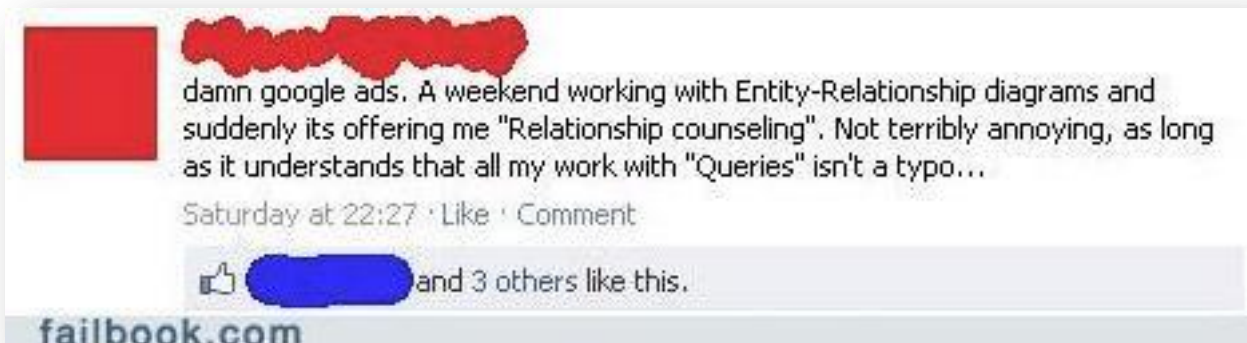
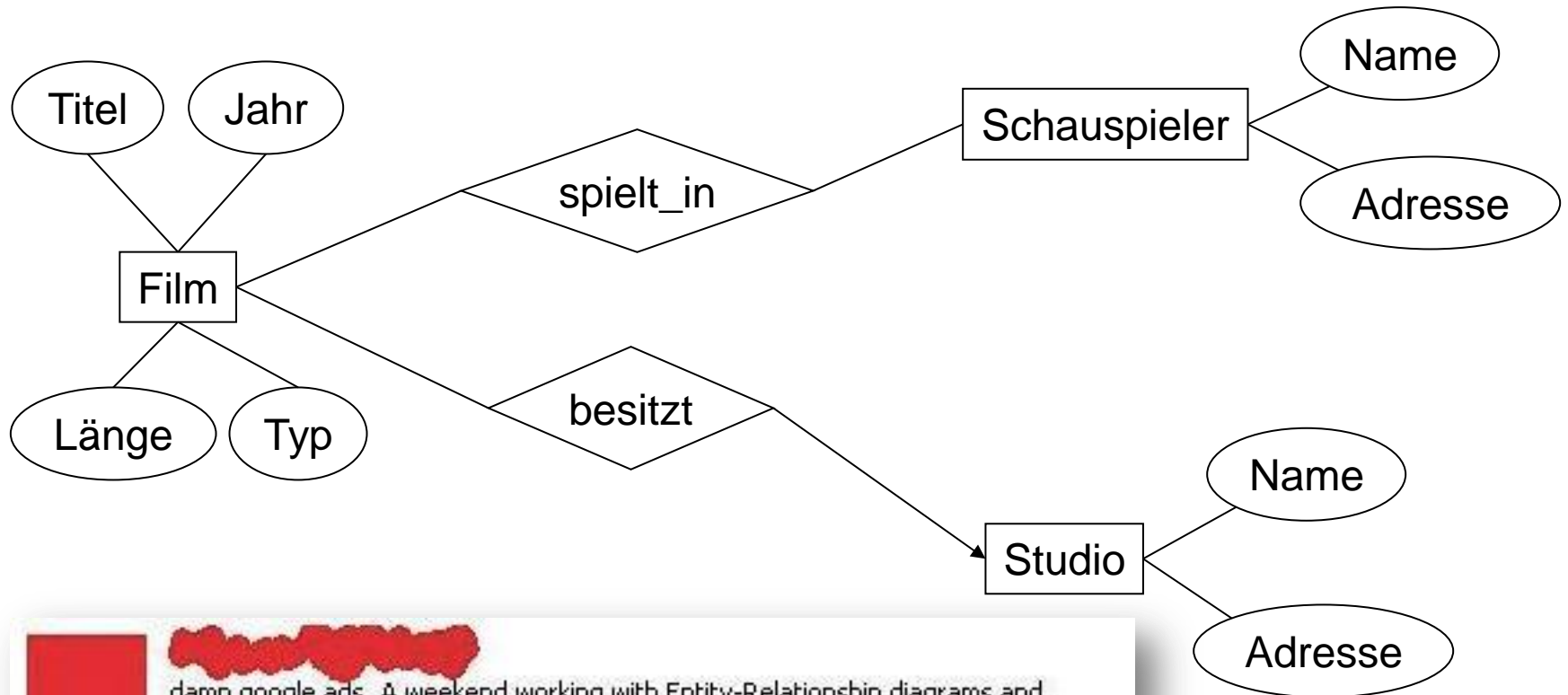


- Kanten verbinden Entitytypen mit Relationshiptypen



# Entity-Relationship-Diagramm

22



23

- Kardinalitäten



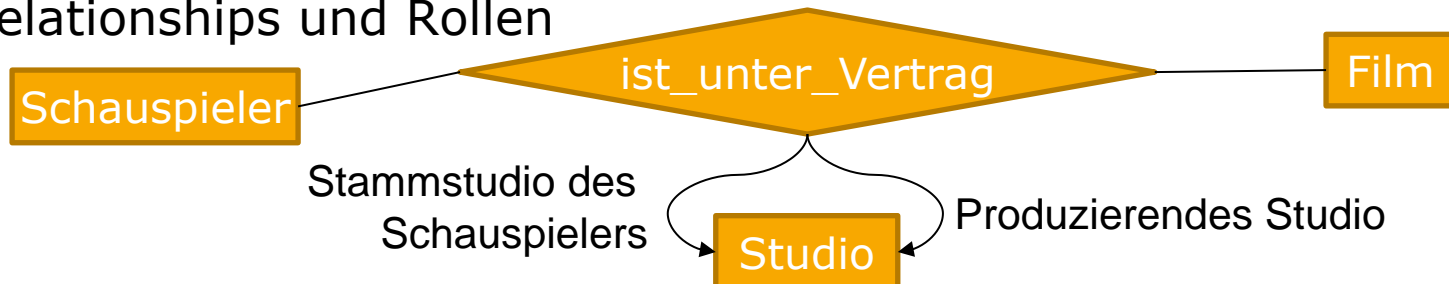
- Referentielle Integrität



- Schwache Entitytypen



- N-äre Relationships und Rollen



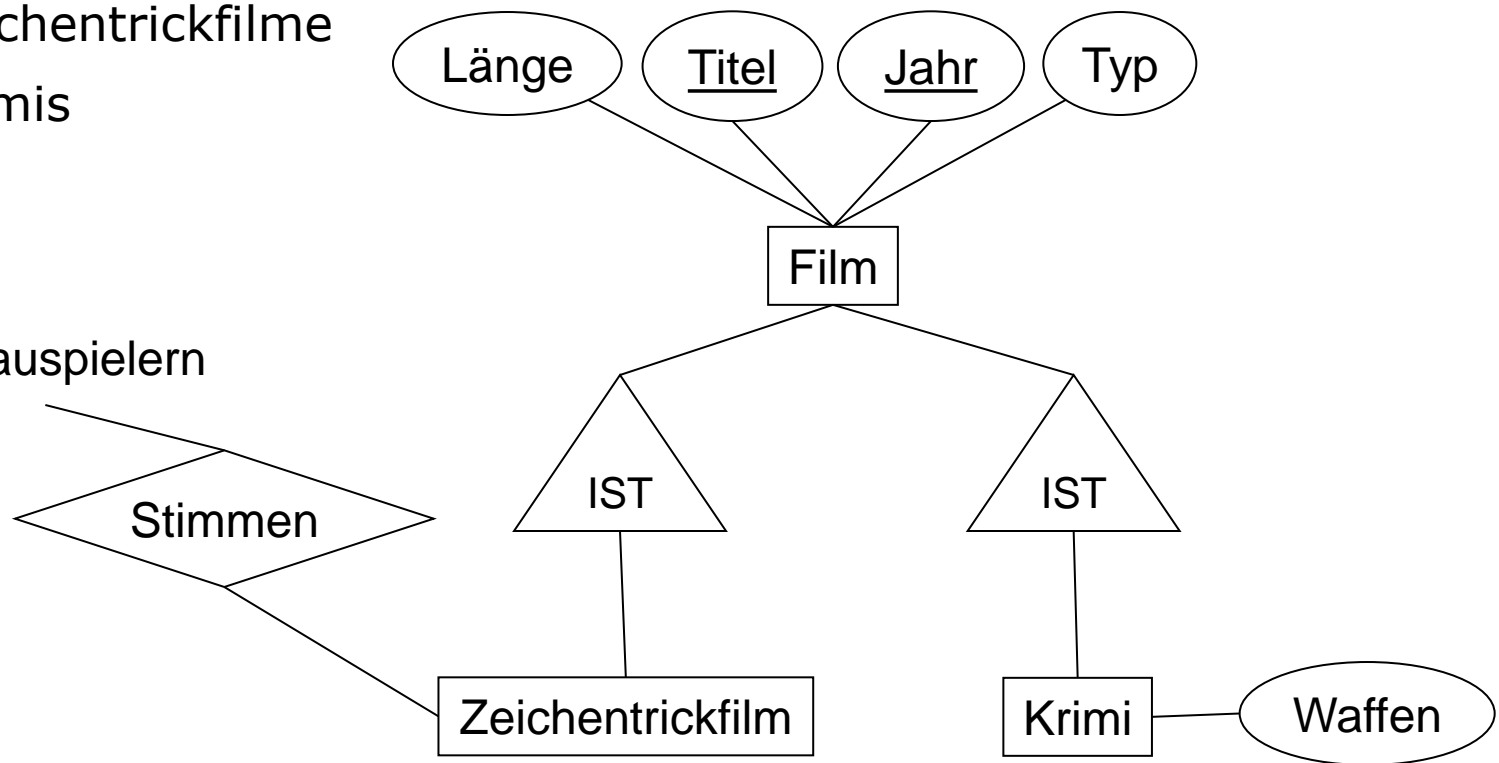
# Die IST-Beziehung

24

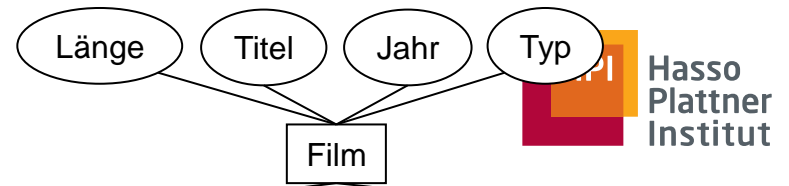
## Subklassen zu Filmen

- Zeichentrickfilme
- Krimis

zu Schauspielern







- IST-Beziehungen nur als Bäume
  - Keine Mehrfachvererbung
- Ein Entity kann aus mehreren Komponenten des IST-Baumes bestehen.
  - „Krieg der Sterne“ hat vier Attribute.
  - „Cinderella“ hat vier Attribute und „Stimmen“-Relationships.
  - „Der dritte Mann“ hat vier Attribute und zusätzlich das Attribut „Waffen“.
  - „Roger Rabbit“ hat vier Attribute, zusätzlich das Attribut „Waffen“ und „Stimmen“-Relationships.
- Anders als objekt-orientierte Modelle
  - In OO sind Objekte immer in genau einer Klasse; Subklassen erben von Superklasse(n).
  - In ER sind Entities in allen Subklassen repräsentiert, in die sie gehören.
  - In ER ist ein Entity in einer Subklasse auch automatisch in den Superklassen repräsentiert.



# Nebenbedingungen (Constraints)

26

- Schlüssel
  - Ein oder mehrere Attribute
  - Werte identifizieren eindeutig ein Entity.
- Referentielle Integrität
  - Existenz des referenzierten Entities
- Domänen
  - Einschränkung des Wertebereichs
- Allgemeine Nebenbedingungen (*assertions*)
  - Z.B. nicht mehr als 10 Schauspieler pro Film
- Nebenbedingungen sind Teil des Schemas. Sie leiten sich nicht aus den Daten ab!

# Schlüssel

27

- Ein Schlüssel ist eine (minimale) Menge von Attributen eines Entitytyps, für die gilt, dass keine zwei Entities gleiche Werte in allen Schlüsselattributen haben.
  - Einige Attributwerte können übereinstimmen.
  - Oft nur ein Attribut
- Für jeden Entitytyp muss ein Schlüssel angegeben werden.
- Es kann mehr als einen Schlüssel für einen Entitytyp geben.
  - Üblich: Primärschlüssel auswählen
- Bei IST-Beziehungen muss die Wurzel-Superklasse sämtliche Schlüsselattribute enthalten.
  
- Darstellung durch Unterstreichen der Attributnamen

# Schlüssel

28

- Filme
  - „Titel“ als Schlüssel?
    - ◇ King Kong
  - Titel und Jahr als Schlüssel?
    - ◇ Eventuell kann ein Film dann nicht gespeichert werden.
    - ◇ In IMDB: 275 Doppelte
- Schauspieler
  - „Name“ oder „Name“ und „Adresse“
- Studio
  - Name
- In der Praxis: Modellierung eines speziellen, numerischen Schlüsselattributs.
  - ISBN, SSN, Imma-Nummer,...

Steuerzentrale - DB2COPY1

Steuerzentrale Ausgewählt Editieren Sicht Tools Hilfe

Objektsicht Befehlseditor 1 X

Befehle Abfrageergebnisse Zugriffsplan

Diese Ergebnisse werden mit Hilfe von Aktualisierungen und Löschungen mit Su...

TITLE	DATE	3
Shadow Chasers	1988	2
Shogun	1980	2
Shriners Hospit...	2000	2
Sinatra: The Cl...	2002	2
Sirens	1995	2
Skag	1980	2
Skin Deep	2000	2
Skin Deep	2003	2
Skyport	1960	2
Something Is O...	1988	2
Space Invaders	1999	2
Sparks	1998	2
Special Delivery	2003	2
Spring	2002	2
Srečna porodica	1979	2
Sternbergs - Är...	1999	2
Still Life	2004	2
Straight Up	1997	2
Strange Days	1998	2
Strange World	1999	2
Striker	1976	2
Studio 5-B	1989	2
Summerland	2004	2
Sunset Beat	1990	2
Supercarrier	1988	2
Sweet Heart	1996	2
Take Five	1987	2
Talk to Me	2000	2
Tarzan: The E...	1996	2
Tæskeholdet	1997	2
Temptation	1968	2
Tenshi no tama...	1985	2
Tenspeed and ...	1980	2
Test, The	2001	2
Top of the Hill	1989	2

# Wo ist der Schlüssel?

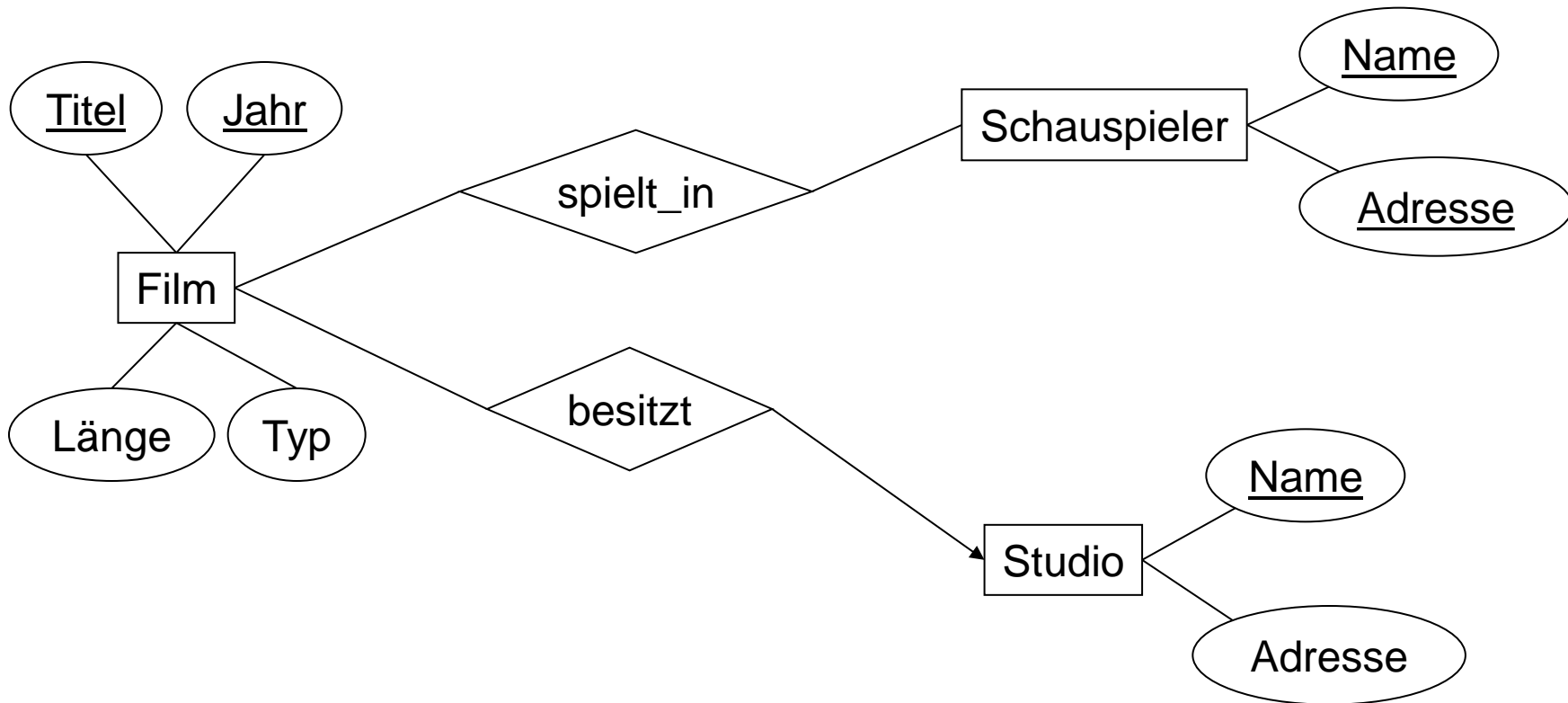
29

The screenshot shows a Mozilla Firefox browser window with the title "Star Wars (1977) - Mozilla Firefox". The address bar contains "http://www.imdb.com/title/tt0076759/". The browser's toolbar includes various icons and a search bar with "Star Wars" entered. The IMDb website is displayed, featuring the IMDb logo and navigation links like "NOW PLAYING", "MOVIE / TV NEWS", "MY MOVIES", "DVD / VIDEO", "IMDb TV", "MESSAGE BOARDS", "SHOWTIMES & TICKETS", and "GAME BASE". The main content area is titled "Star Wars (1977)" and includes a search bar, a result count of "1 of 108", and a list of navigation options such as "Overview", "main details", "combined details", "full cast and crew", "company credits", "Awards & Reviews", "user comments", "external reviews", "newsgroup reviews", "awards & nominations", "user ratings", "recommendations", "message board", "Plot & Quotes", "plot summary", "plot keywords", "Amazon.com summary", and "memorable quotes". The main content also features a "Photo Gallery (35 photos)" with a "more" link, a "Genre" section listing "Action / Adventure / Fantasy / Sci-Fi (more)", and a "Tagline" section with the text "A long time ago in a galaxy far, far away... (more)". A "SHOP STAR WARS" section on the right side of the page offers various media formats like DVD, VHS, and CD for different countries, along with "Memabilia Books" and "All Products" from Amazon.com.

# Schlüssel

30

- In ER-Diagrammen kann nur ein Schlüssel notiert werden.
  - Der sich über mehrere Attribute eines Entitytyps erstrecken kann.



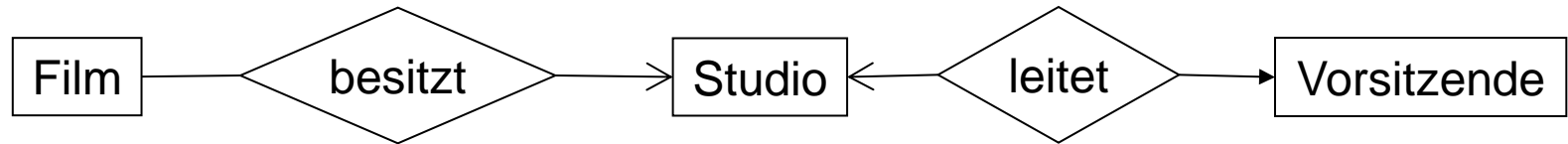
# Referentielle Integrität

31

- Schlüssel: Höchstens ein bestimmter Wert für ein Attribut
  - Bzw. höchstens eine Wertekombination bei mehreren Attributen im Schlüssel
- Referentielle Integrität: Genau ein bestimmter Wert
  - Bsp. n:1 Relationship zwischen „Film“ und „Studio“
    - ◇ Ein Film kann zu höchstens einem Studio gehören.
    - ◇ Aber ein Film muss zu keinem Studio gehören.
    - ◇ Auch wenn ein Film zu einem Studio gehört, muss dieses nicht in der DB repräsentiert sein.
    - ◇ Referentielle Integrität erzwingt die Existenz und Repräsentation des Studios
- „Erzwingen“
  - Bei Einfügen/Ändern eines Films muss entsprechendes Studio vorhanden sein.
  - Ein Studio darf nicht gelöscht werden, solange es noch Filme besitzt.
  - Oder: Wenn ein Studio gelöscht wird, werden auch alle entsprechenden Filme gelöscht.
  - Verschiedene Einstellungen im DBMS

# Referentielle Integrität

32

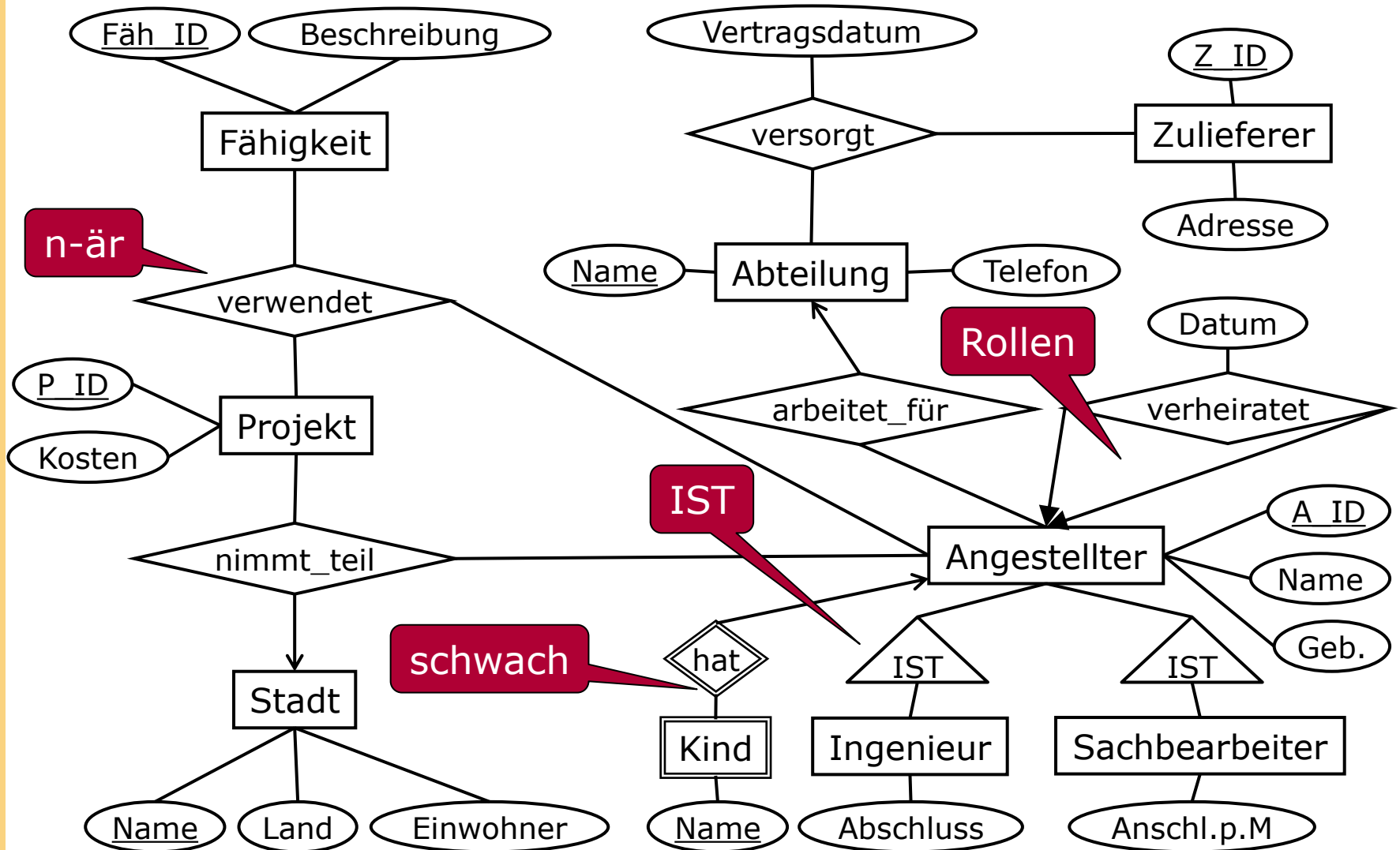


- Jeder Film muss von genau einem Studio besessen werden.
- Jeder Vorsitzende muss genau ein Studio leiten.
- Jedes Studio wird von höchstens einem Vorsitzenden geleitet.
  - Eventuell aber von keinem
- Ein Studio kann mehrere Filme besitzen.

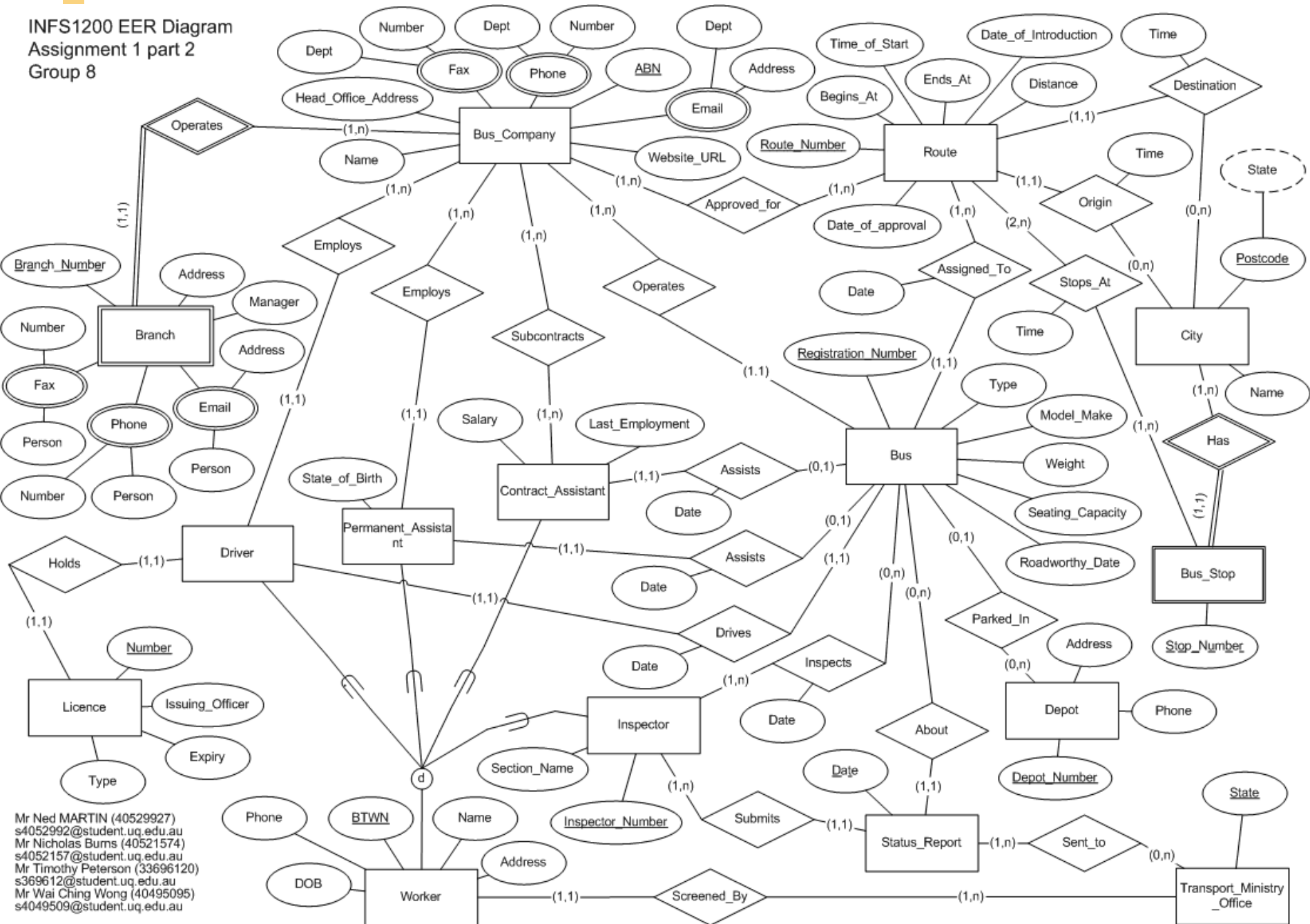


# ER-Diagramm mit komplexen Relationshiptypen

33



INFS1200 EER Diagram  
 Assignment 1 part 2  
 Group 8

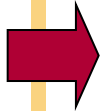


Mr Ned MARTIN (40529927)  
 s4052992@student.uq.edu.au  
 Mr Nicholas Burns (40521574)  
 s4052157@student.uq.edu.au  
 Mr Timothy Peterson (33696120)  
 s369612@student.uq.edu.au  
 Mr Wai Ching Wong (40495095)  
 s4049509@student.uq.edu.au

# Überblick

35

- Motivation und Einbettung
- Wiederholung: ER-Diagramme
- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Konvertierung von Spezialisierung
- Funktionale Abhängigkeiten (FDs)
- Ableitungsregeln für FDs
- Normalformen



# Die Relation in der Mathematik

36

- Eine Relation  $R$  ist eine Menge von  $n$ -Tupeln.
- Dinge, die in der Relation  $R$  zueinander stehen, bilden ein  $n$ -Tupel, das Element von  $R$  ist.
- Teilmenge des Kartesischen Produkts
  - $R \subseteq A_1 \times \dots \times A_n$
  - $A_1 \times \dots \times A_n := \{(a_1, \dots, a_n) \mid a_1 \in A_1 \wedge \dots \wedge a_n \in A_n\}$
- Für Datenbanken:
  - Mengen  $A_1, \dots, A_n$  sind Domänen (Wertebereiche)
    - ◇ Z.B.: Integer, String, Boolean, usw.
    - ◇ Entspricht Spalten einer Tabelle
  - Tupel  $(a_1, \dots, a_n)$  sind die Datenwerte
    - ◇ Z.B. (4, Naumann, Potsdam, m)
    - ◇ Entspricht Zeilen einer Tabelle

# Die Relation

37

- Konzeptuell ist eine Datenbank eine Menge von Tabellen.
  - Relation zwischen Werten der Attributdomänen
  - Tabellen = Relationen

Titel	Jahr	Länge	Typ
Basic Instinct	1992	127	Farbe
Total Recall	1990	113	Farbe
Dead Man	1995	121	s/w

- Die Relation ist das einzige Konstrukt des relationalen Modells
  - Sehr einfach
  - Einfach in einer DB abzubilden (zwei-dimensional)
  - Relationen können nicht nur Entities sondern auch Relationships darstellen.
  - Entspricht oft unserer Vorstellung der Daten
  - Ist das abstrakte Modell hinter SQL

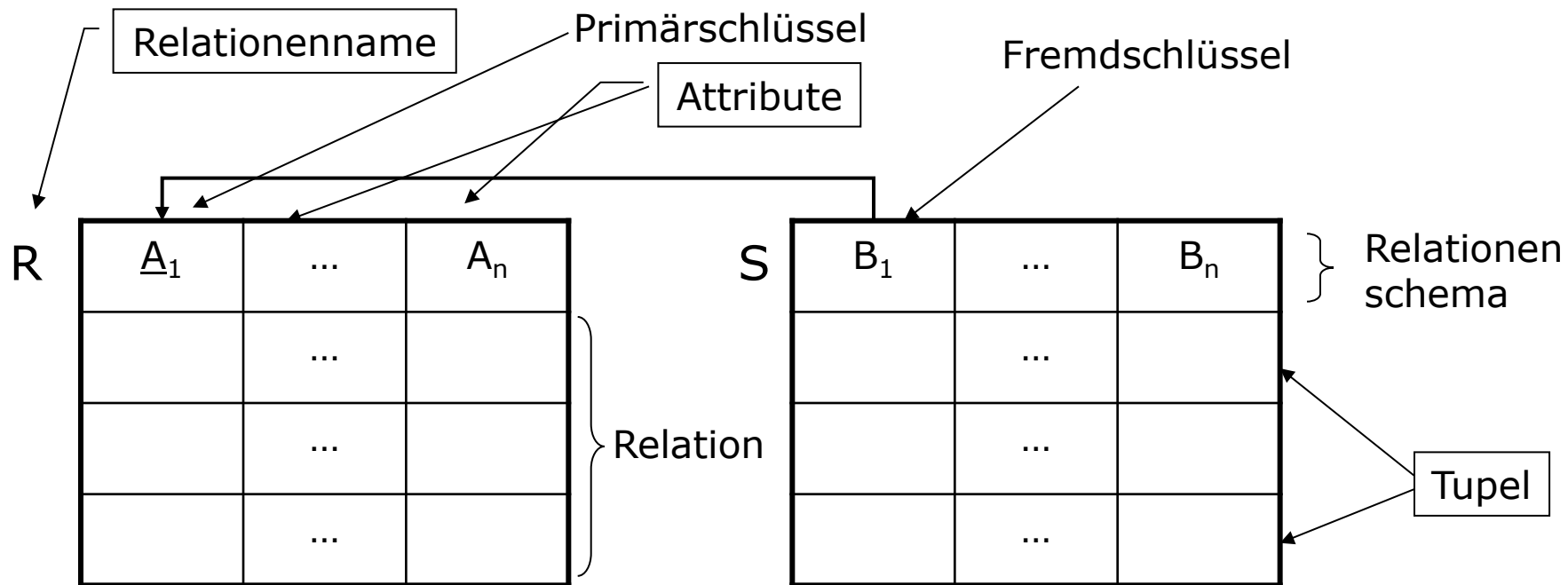
# Elemente des Relationalen Modells

38

- *Datenbankschema*
  - Besteht aus einem oder mehreren Relationenschemata.
- *Relationenschema*
  - Weitere Einträge in der Tabelle: Die „*Relation*“
  - Besteht aus keinem oder mehr Tupeln.
- Eine Zeile der Tabelle: *Tupel*
  - Tupel bilden eine Menge (nicht eine Liste).
- Eine Spaltenüberschrift: *Attribut*
  - Attribute bilden eine Menge (nicht eine Liste).
- Ein Eintrag: *Attributwert*
  - Atomar
  - Stammt aus einer elementaren Domäne (Integer, String, ...)

# Elemente einer Relation

39



# Formal

40

- Domänen  $D_1, \dots, D_n$
- Relation  $R \subseteq D_1 \times \dots \times D_n$

## Beispiel

- Relationenschema: Film(Titel, Jahr, Länge, Typ)
- Domänen: String, Integer, Integer, String
- Tupel: (Star Wars, 1977, 124, farbig)



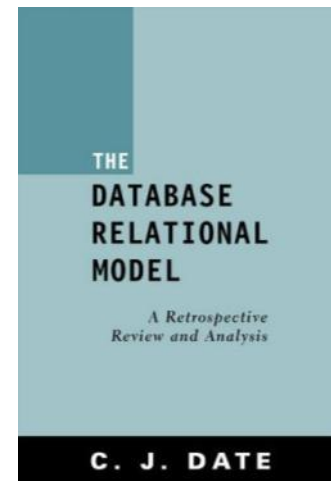
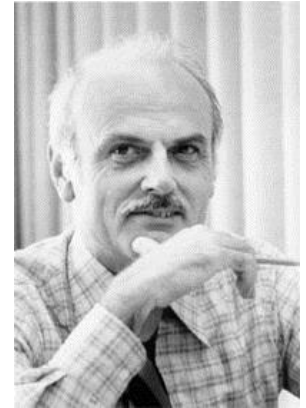


# Edgar F. Codd

41

[http://en.wikipedia.org/wiki/Edgar\\_F.\\_Codd](http://en.wikipedia.org/wiki/Edgar_F._Codd)

- Promotion an der University of Michigan Ann Arbor
- Entwicklung des Relationalen Modells bei IBM (Almaden)
- „A Relational Model of Data for Large Shared Data Banks" (1970)
- Artikelserie
- Literaturhinweis:
  - The Database Relational Model: A Retrospective Review and Analysis :
    - ◇ A Historical Account and Assessment of E. F. Codd's Contribution to the Field of Database Technology
    - ◇ Chris J. Date
  - ISBN: 0-201-61294-1 (9.99 EUR)



## *Contributions* (nach C.J.Date)

42

- Transformation von Datenmanagement zu einer Wissenschaft
  - Entsprechende Klarheit und Strenge
- Nicht nur das relationale Modell, sondern überhaupt das Konzept eines Datenmodells
  - Unterscheidung zwischen Modell und Implementierung
- Relationale Algebra und relationales Kalkül
- Informell: Anfragesprache Alpha
  - Angelehnt: SEQUEL von Chamberlin und Boyce
- Funktionale Abhängigkeiten
- Normalformen
  - Erste bis dritte Normalform

599

DERIVABILITY, REDUNDANCY AND CONSISTENCY OF RELATIONS  
STORED IN LARGE DATA BANKS

E. F. Codd  
Research Division  
San Jose, California

ABSTRACT: The large, integrated data banks of the future will contain many relations of various degrees in stored form. It will not be unusual for this set of stored relations to be redundant. Two types of redundancy are defined and discussed. One type may be employed to improve accessibility of certain kinds of information which happen to be in great demand. When either type of redundancy exists, those responsible for control of the data bank should know about it and have some means of detecting any "logical" inconsistencies in the total set of stored relations. Consistency checking might be helpful in tracking down unauthorized (and possibly fraudulent) changes in the data bank contents.

RJ 599 (# 12343)  
August 19, 1969

## INTRODUCTION

The first part of this paper is concerned with an explanation of a relational view of data. This view (or model) of data appears to be superior in several respects to the graph or network model [1, 2] presently in vogue. It provides a means of describing data with its natural structure only: that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level retrieval language which will yield maximal independence between programs on the one hand, and machine representation and organization of data on the other. A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations -- these are discussed in the second part of this paper. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations. Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present management information systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system.

1. A Relational View of Data

The term relation is used here in its accepted mathematical sense. Given sets  $S_1, S_2, \dots, S_n$  (not necessarily distinct),  $R$  is a relation on these  $n$  sets if it is a set of  $n$ -tuples, each of which has its first element from  $S_1$ , its second element from  $S_2$ , and so on. We shall refer to  $S_j$  as the  $j^{\text{th}}$  domain of  $R$ . As defined above,  $R$  is said to have degree  $n$ . Relations of degree 1 are often called unary, degree 2 binary, degree 3 ternary, and degree  $n$  n-ary.

For expository reasons, we shall frequently make use of an array representation of relations, but it must be remembered that this particular representation is not an essential part of the relational view being expounded. An array which represents an  $n$ -ary relation  $R$  has the following properties:

- (1) Each row represents an  $n$ -tuple of  $R$ ;
- (2) The ordering of rows is immaterial;
- (3) All rows are distinct;
- (4) The ordering of columns is significant - it corresponds to the ordering  $S_1, S_2, \dots, S_n$  of the domains on which  $R$  is defined;
- (5) The significance of each column is partially conveyed by labeling it with the name of the corresponding domain.

2.

The example in Figure 1 illustrates a relation of degree 4 called ship which reflects the shipments-in-progress of parts from specified suppliers to specified projects in specified quantities.

<u>ship</u>	( <u>supplier</u>	<u>part</u>	<u>project</u>	<u>quantity</u> )
	1	2	5	17
	1	3	5	23
	2	3	7	9
	2	7	5	4
	4	1	1	12

FIGURE 1: A Relation of Degree 4

One might ask: If the columns are labeled by the name of the corresponding domains, why should the ordering of columns matter? As the example in Figure 2 shows, two columns may have identical headings (indicating identical domains), but possess distinct meanings with respect to the relation. The relation depicted is called component. It is a binary relation, each of whose two domains is called part. The meaning of component (x, y) is that part x is an immediate component (or subassembly) of part y.

<u>component</u>	( <u>part</u>	<u>part</u> )
	1	5
	2	5
	3	5
	2	6
	3	6
	4	7
	6	7

Figure 2: A Relation with Two Identical Domains

We now assert that a data bank is a collection of time-varying relations. These relations are of assorted degrees. As time progresses, each n-ary relation may be subject to insertion of additional n-tuples, deletion of existing ones, and alteration of components of any of its existing n-tuples.

Consider, for example, a data bank which contains information about parts, projects, and suppliers. The individual description for an individual object (such as a particular part) is called an entity [3]. The prototype description for a class of objects is called an entity type. The set of entities of a given entity type can be viewed as a relation, and we shall call such a relation an entity type relation. In the example under consideration, there might be an entity type relation called part defined on the following domains:

# Überblick

46

- Motivation und Einbettung
- Wiederholung: ER-Diagramme
- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Konvertierung von Spezialisierung
- Funktionale Abhängigkeiten (FDs)
- Ableitungsregeln für FDs
- Normalformen

