

Datenbankmanagement

Theorie: Modellierung

Prof. Dr. Gregor Hülsken

- Was ist die Bedeutung der Datenmodellierung
- Von der Realität zum konzeptionellen Datenmodell
- Welche Notationen sind üblich
- Wie werden die Objekte und Eigenschaften identifiziert

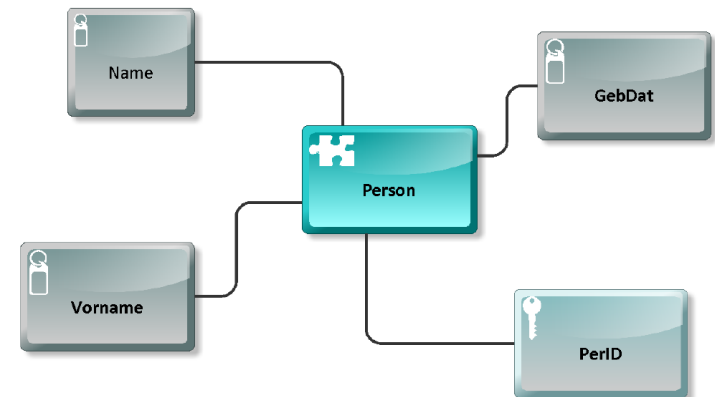
1	Einführung und Überblick
2	Modellierung
2.1	Grundbegriffe
2.2	Modellierung - Notationen
2.3	Semantische Datenbankmodellierung
2.4	Logische Datenbankmodellierung
3	Normalisierung
4	Relationale Algebra
5	Lookup etc. in der Praxis
6	SQL – Data Definition Language
7	SQL – Data Manipulation Language
8	SQL – Trigger
9	SQL – Funktionen / Prozeduren
10	SQL – Datenschutz
11	Transaktionen

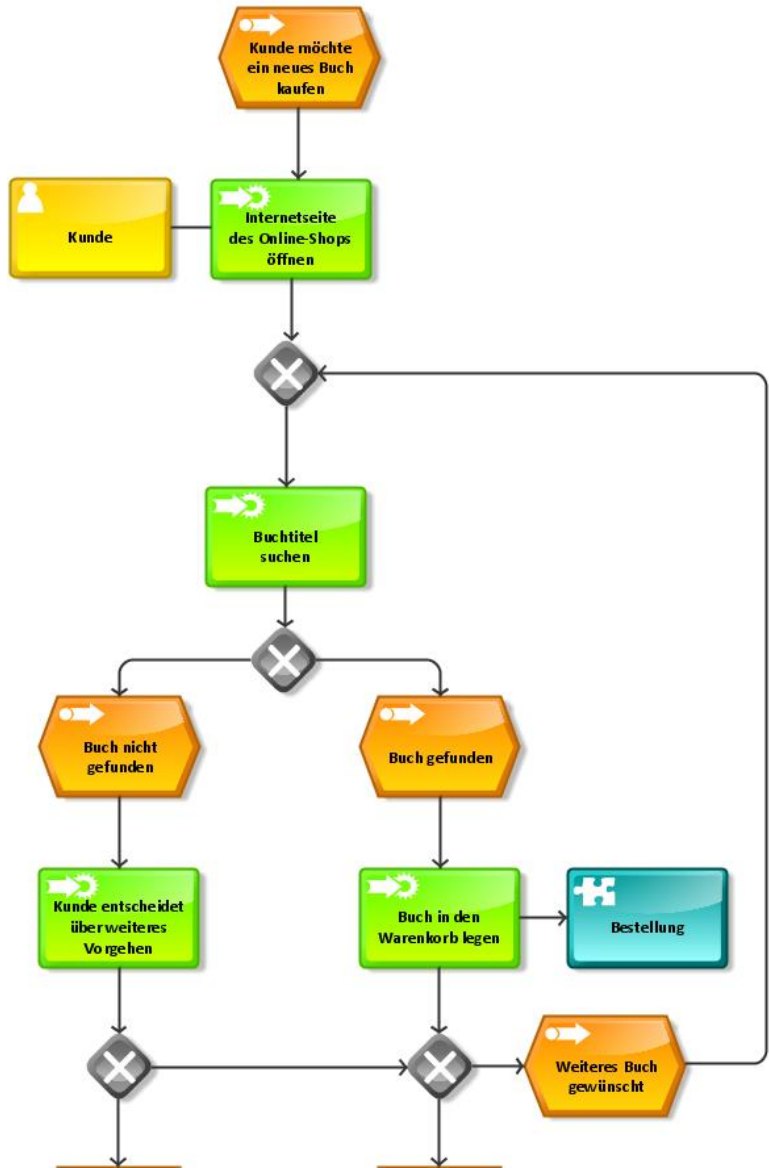
Gegenstand der Datenmodellierung ist es, alle **relevanten** Informationen eines Systems, z.B. eines betrieblichen Funktionsbereiches (Vertrieb), zu beschreiben. Man benötigt dazu eine **Syntax**, die den Sachverhalt möglichst einfach, mit wenigen Syntaxelementen aber genügend formal, um **exakt** sein zu können, zu beschreiben.

- Ebene 1 Reale Welt
- Ebene 2 Betriebswirtschaftliche Modellierung (ARIS)

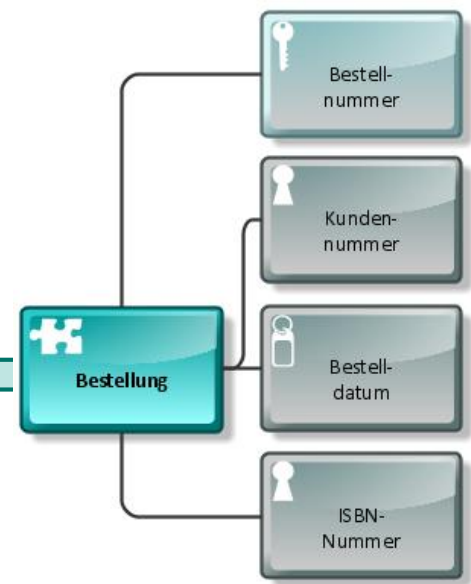
Ebenen der Datenbankmodellierung:

- Ebene 3 Semantische Modellierung
- Ebene 4 Logische Modellierung
- Ebene 5 Physische Modellierung





- Kante
- Ereignis
- Aktivität
- UND-Regel
- XOR-Regel
- ODER-Regel
- Organisationseinheit
- Rolle
- Person
- Standort
- Entität
- Datenbank
- Dokument
- IT-System
- Produkt
- Risiko
- Prozessschnittstelle



Bestelldaten "MyFavoriteBookExpress.com"

Typ: Datenmodell

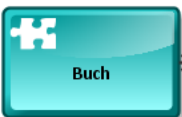
Das **Datenmodell** (IE Datenmodell, information engineering) beschreibt auf abstrakter Ebene, wie Daten miteinander in Verbindung stehen.

Es gehört zur Entity Relationship-Modellierung und wird z. B. zum Beschreiben von Datenbanktabellen verwendet.

Die **Kardinalitäten** von Beziehungen zwischen Entitäten beschreiben, in welcher Anzahl diese miteinander in Beziehung stehen.

Im Beispiel hat eine Bestellung beliebig viele Bücher (mindestens eines), verdeutlicht durch das Symbol am Kantenende des Objekts "Buch". Eine Buch hingegen ist beliebig vielen Bestellungen zugeordnet (oder keiner), verdeutlicht durch die Null am Objekt "Bestellung".

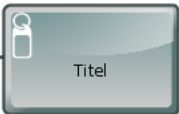
Die Kardinalitäten werden durch die Beziehungsattribute eingestellt (Ansicht/Attribute).



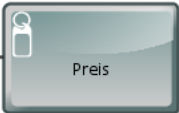
Buch



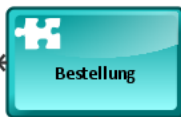
ISBN-Nummer



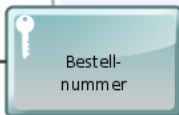
Titel



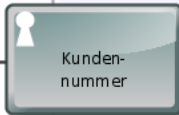
Preis



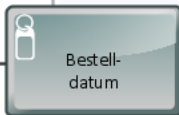
Bestellung



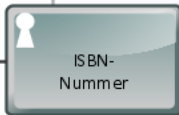
Bestellnummer



Kundennummer



Bestelldatum



ISBN-Nummer

Attribute in einem Datenmodell beschreiben eine Eigenschaft eines Datenobjekts (Entitäts), z. B. einer Spalte einer Datenbanktabelle. Es gibt verschiedene Typen:

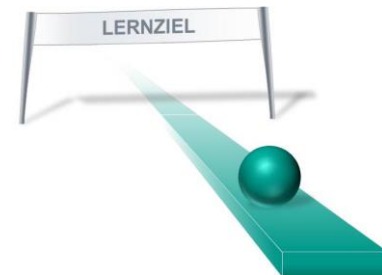
Der **Primärschlüssel** (hier: Bestellnummer) ist ein eindeutiger Identifizierer eines Objekts.

Der **Fremdschlüssel** ist ein Verweis auf den Primärschlüssel eines anderen Datenobjekts. Die Kundennummer ist z. B. ein Verweis auf ein Datenobjekt vom Typ "Kunde".

"Bestelldatum" hingegen ist ein **Attribut**, welches eine Eigenschaft des Datentyps "Bestellung" beschreibt.

Im Anschluss an diesen Themenblock sollen Sie:

- Wissen, was die Bedeutung der Datenmodellierung ist
- Von der Realität zum konzeptionellen Datenmodell kommen
- Wissen, welche Notationen sind üblich
- Objekte und Eigenschaften identifizieren können



2.1 Grundbegriffe

Datenmodellierung Definition:

Gegenstand der Datenmodellierung ist es, alle relevanten Informationen eines Systems, z.B. eines betrieblichen Funktionsbereiches (Vertrieb), zu beschreiben. Man benötigt dazu eine Syntax, die den Sachverhalt möglichst einfach, mit wenigen Syntaxelementen aber genügend formal, um exakt sein zu können, zu beschreiben.

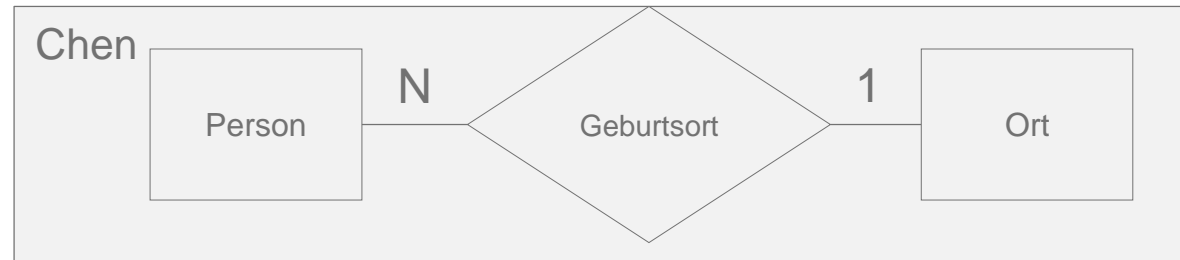
Begriffe

- Homonyme Begriffe
Begriffe mit unterschiedlicher Bedeutung (Bank/ Bank)
- Äquipollenzen
Unterschiedliche Sichtweisen der selben Objekte (Mitarbeiter, Aufträge
(Auftrag aus rechtlicher organisatorischer, buchhalterischer Sicht)
- Vage Begriffe
Dinge, die jeder erwähnt, die jedoch nicht genau definiert sind (Multimedia)
- Bezeichner
Problem der Identifikation von z.B. Artikelnummern (unterschiedlich in
verschiedenen Bereichen)

2.2 Modellierung - Notationen

Alternative ER-Diagramme

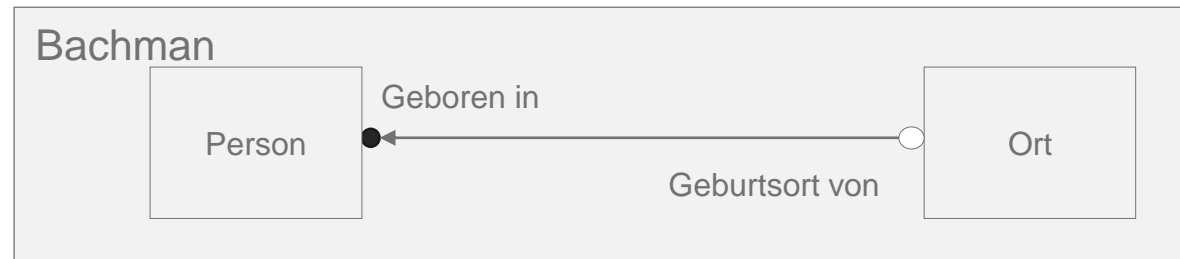
Die Chen-Notation von Peter Chen, dem Entwickler der ER-Diagramme, 1976.



Die IDEF1X: langjähriger de-facto Standard bei U.S. amerikanischen Behörden.



Die Bachman-Notation von Charles Bachman als weit verbreitete Werkzeug-Diagramm-Sprache.



Alternative ER-Diagramme

Die Martin-Notation (Krähenfuß - Notation) als weit verbreitete Werkzeug-Diagramm-Sprache (Information Engineering).



Die (min,max)-Notation von Jean-Raymond Abrial, 1974.



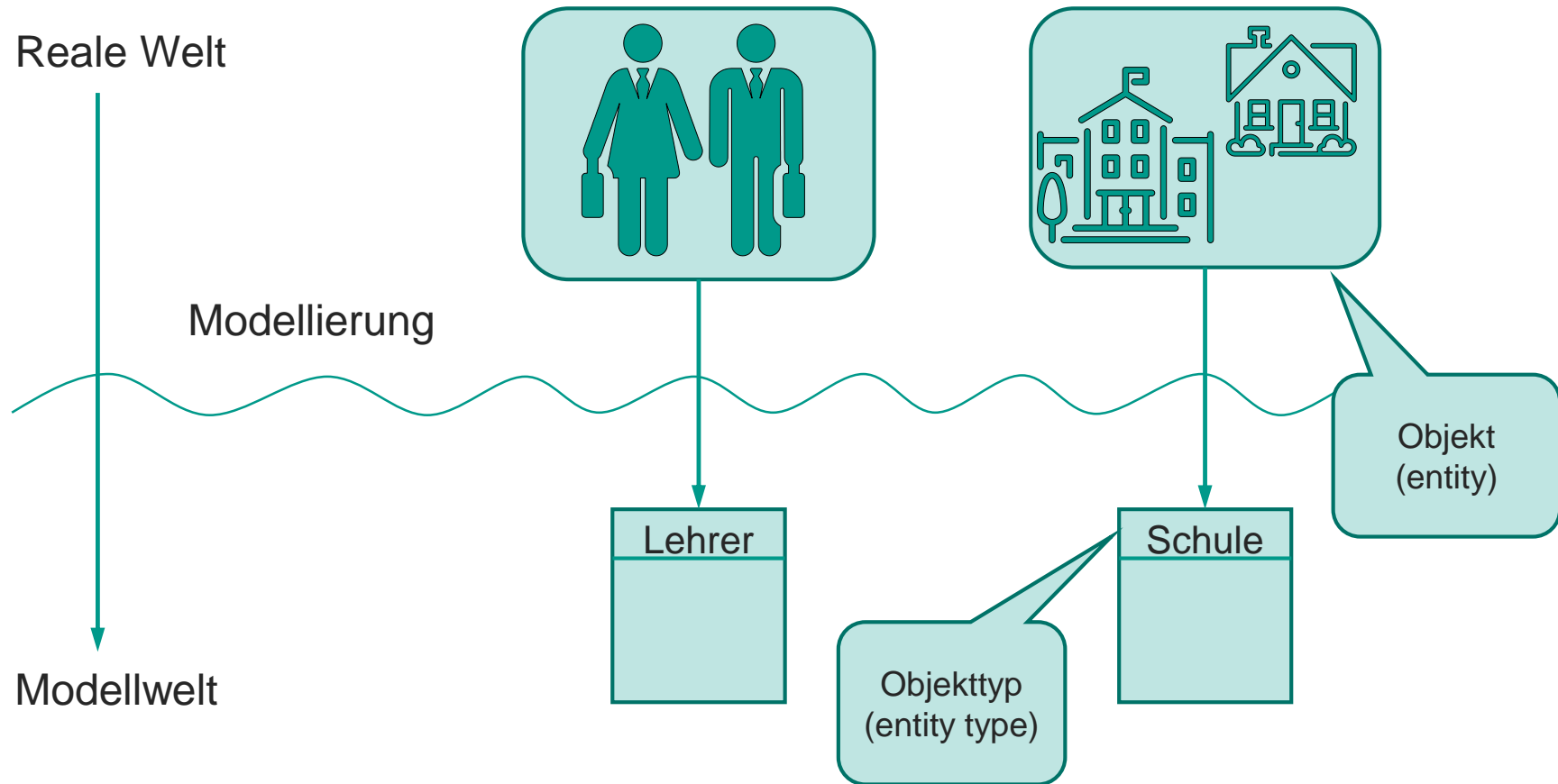
UML als Standard, den selbst ISO in eigenen Normen als Ersatz für ER-Diagramme verwendet.



Quelle: Eigene Darstellung

2.3 Semantische Datenbankmodellierung

Klassifizierung von Objekttypen



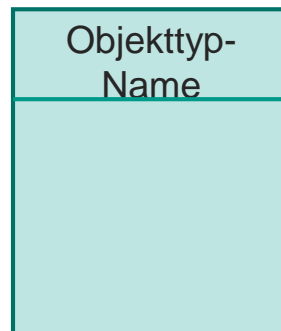
Objekte / Objekttypen

Ein Objekt (engl.: entity) ist eine Exemplar von Personen (z.B. Lehrer, Schüler), Gegenständen (z.B. Ort, Schule) oder nichtmateriellen Dingen (z.B. Beschäftigungsverhältnis, Unterrichtsverpflichtung), über das Informationen gespeichert werden.

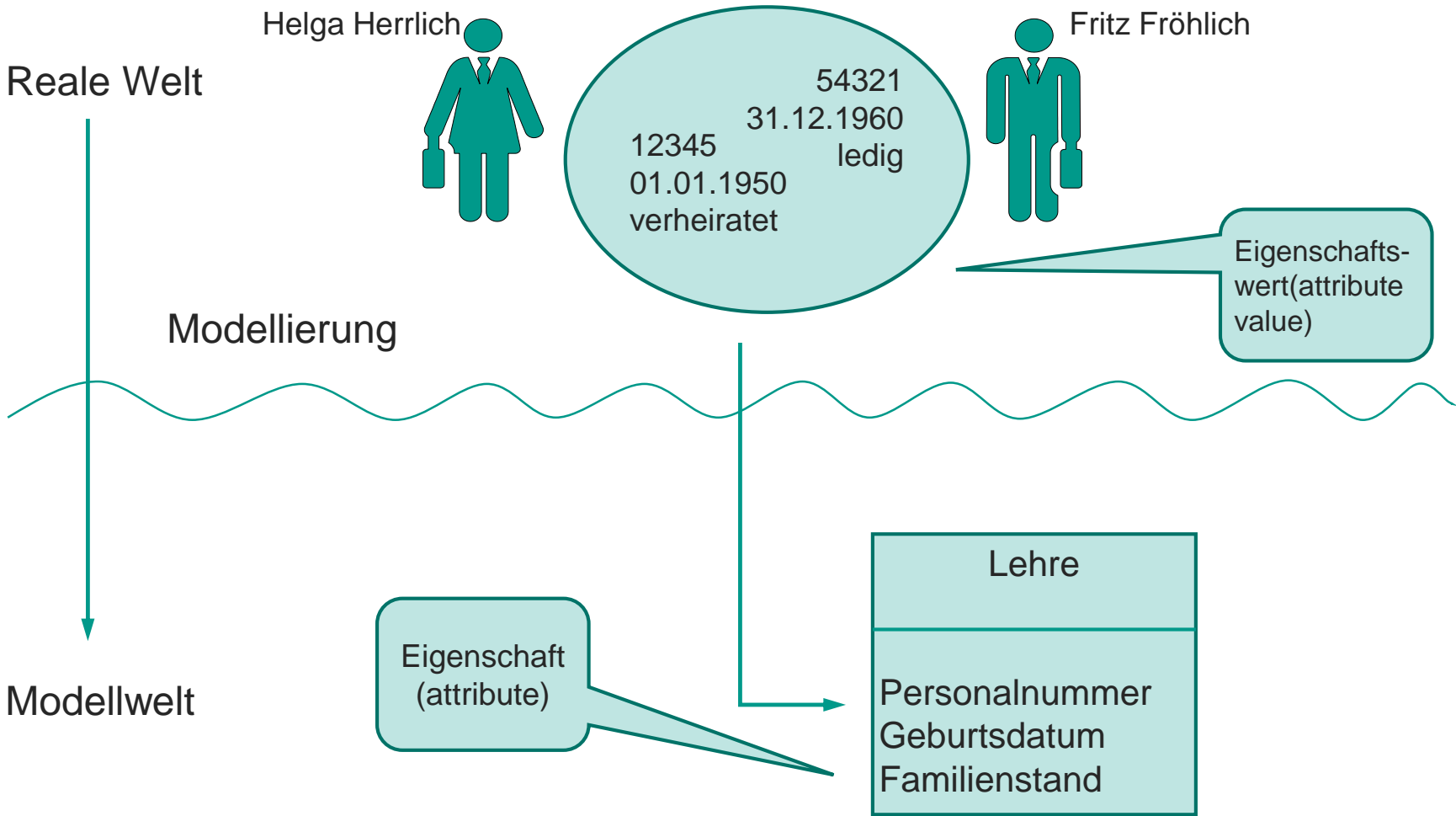
Ein Objekttyp (engl. entity type) ist eine durch einen Objekttyp-Namen eindeutig benannte Klasse von Objekten, über die dieselben Informationen gespeichert werden und die in prinzipiell gleicher Weise verarbeitet werden.

Klassifizierung von Objekttypen 1

- Ein Objekttyp wird durch eine zweigeteilte Box dargestellt, in deren Kopfteil der Objekttyp-Name eingetragen wird.
- Die Größe und die Position der Box sind bedeutungslos.
- Der Objekttyp-Name steht im Singular und muss für das gesamte Datenmodell eindeutig sein!



Festlegen der relevanten Eigenschaften



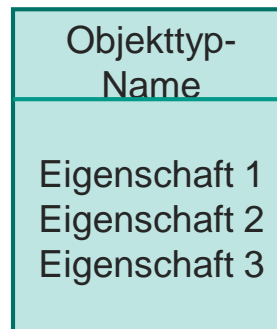
Eigenschaften / Eigenschaftswerte

Eine Eigenschaft (engl.: attribute) ist die Benennung für ein relevantes Merkmal aller Objekte, die in einem Objekttyp zusammengefasst werden.

Ein Eigenschaftswert (engl.: attribute value) ist eine spezielle Ausprägung, die eine Eigenschaft für ein konkretes Objekt annimmt.

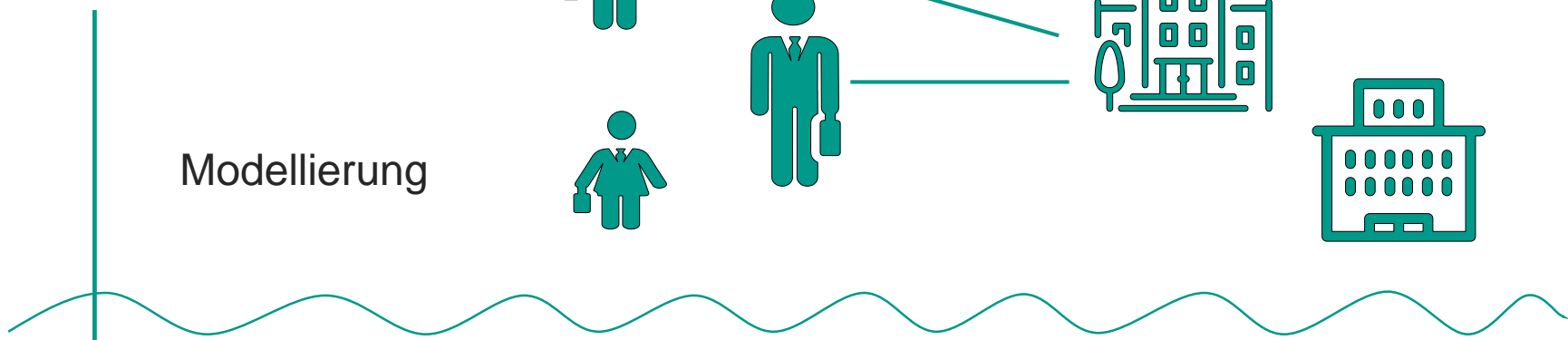
Klassifizierung von Objekttypen 2

- Die Benennung einer Eigenschaft wird in den unteren Teil der Box des Objekttyps eingetragen, für den die Eigenschaft als relevante Angabe gespeichert werden soll.
- Die Reihenfolge der Eigenschaften ist bedeutungslos.
- Die Benennung der Eigenschaft steht im Singular und muss für den Objekttyp eindeutig sein.



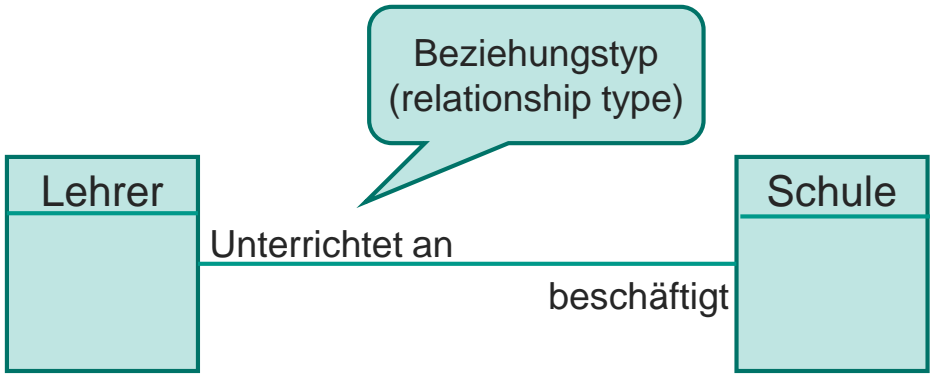
Beziehungstypen

Reale Welt



Modellierung

Modellwelt

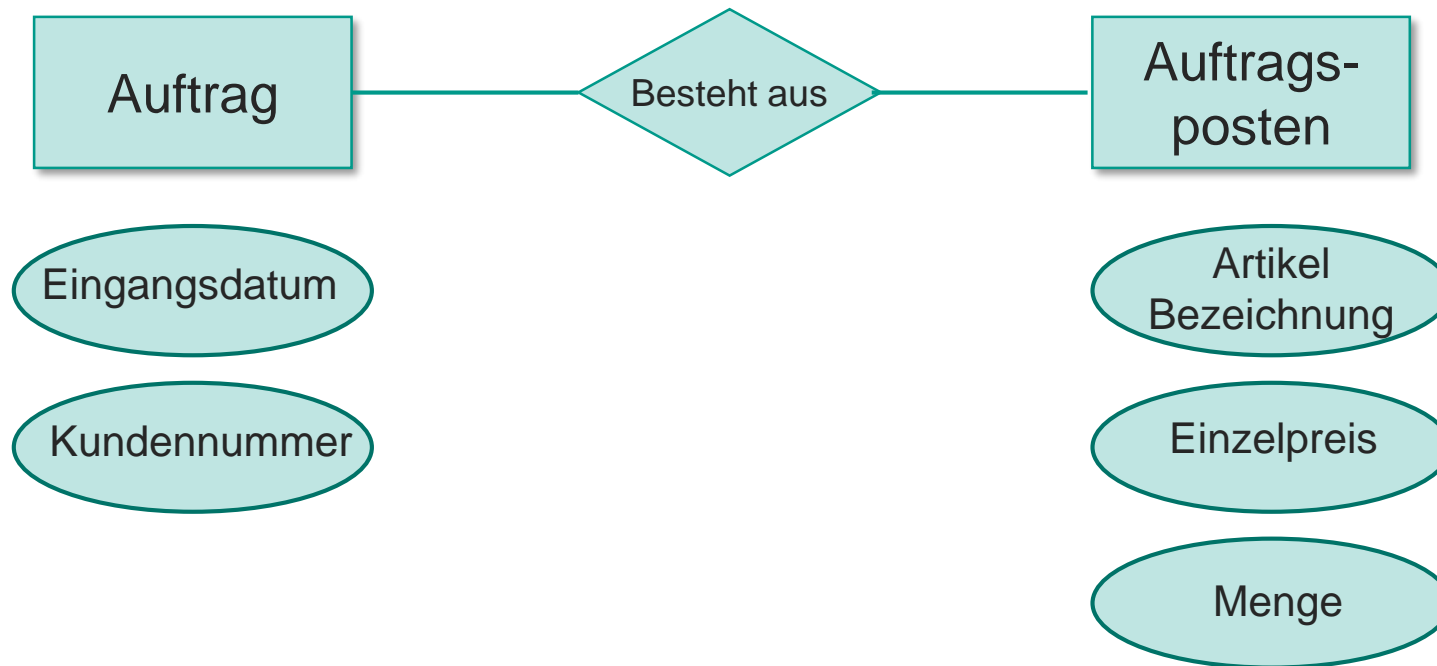


Chen-Notation 1:1 Beziehung



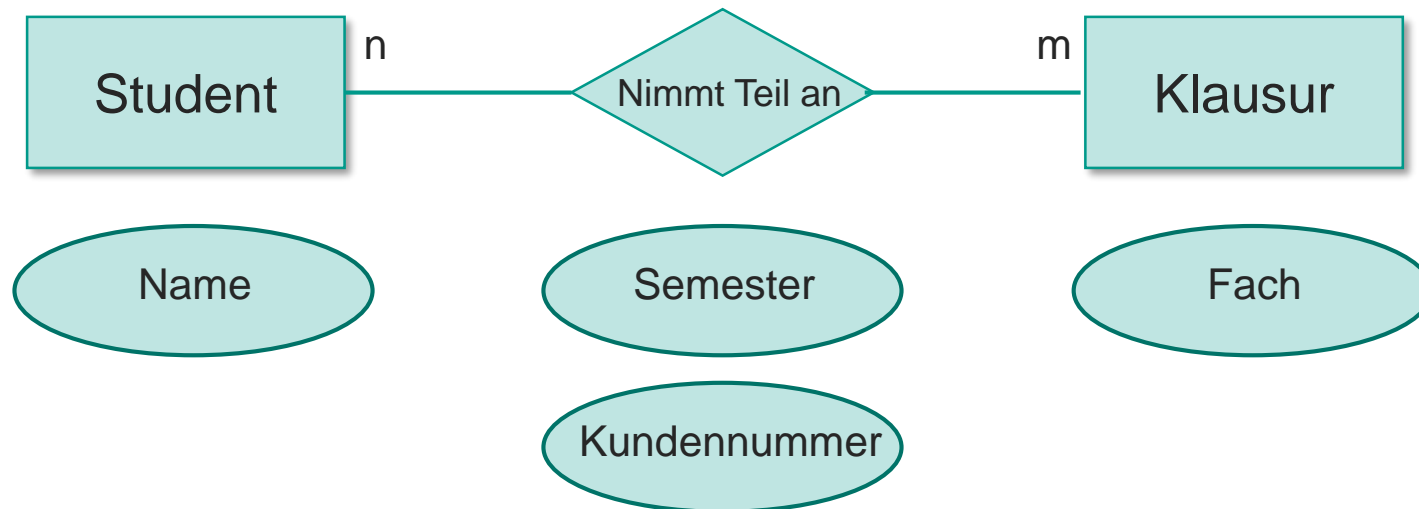
- Ein Beamer befindet sich in einem Raum
- In einem Raum befindet sich kein oder ein Beamer

Chen-Notation 1:n Beziehung



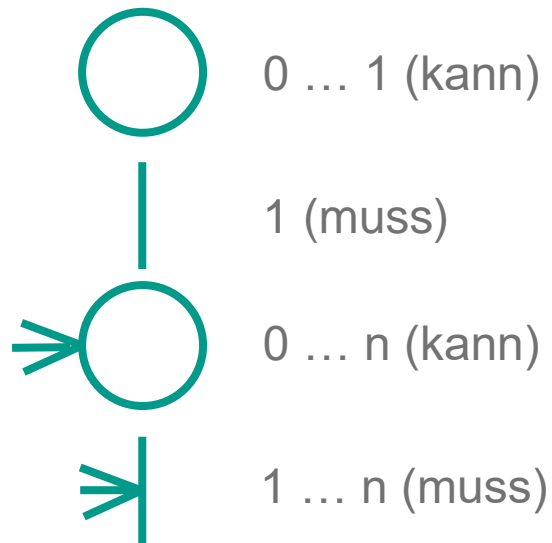
- Ein Auftrag besteht aus n Auftragspositionen
- Eine bestimmte Auftragsposition gehört immer zu einem bestimmten Auftrag!

Chen-Notation n:m Beziehung

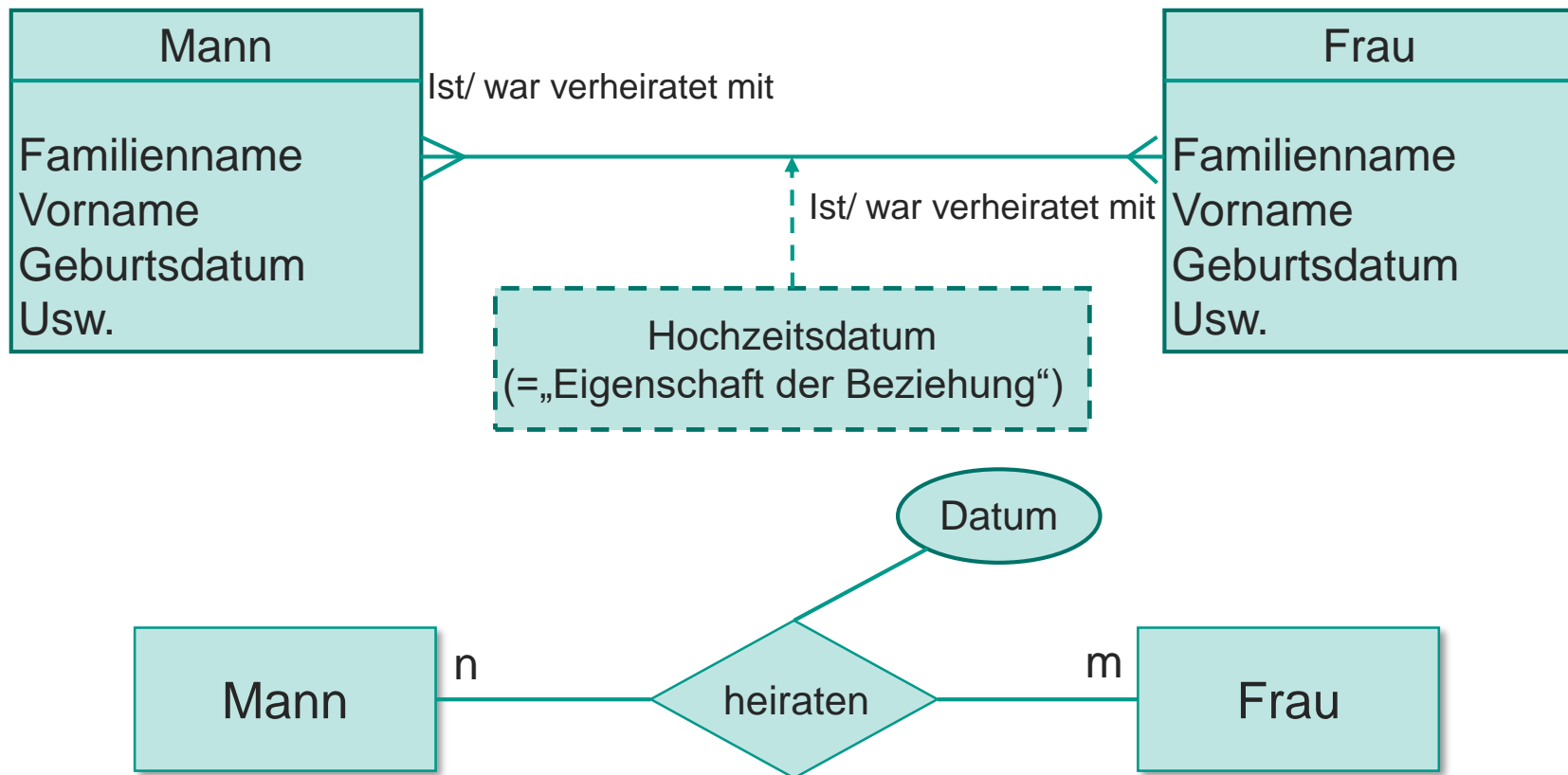


- Ein Student nimmt an keiner, einer oder mehrerer Klausuren teil, die mit einer Note bewertet werden.
- An einer Klausur in einem Semester nehmen n Studenten teil.

Mögliche Kardinalitäten

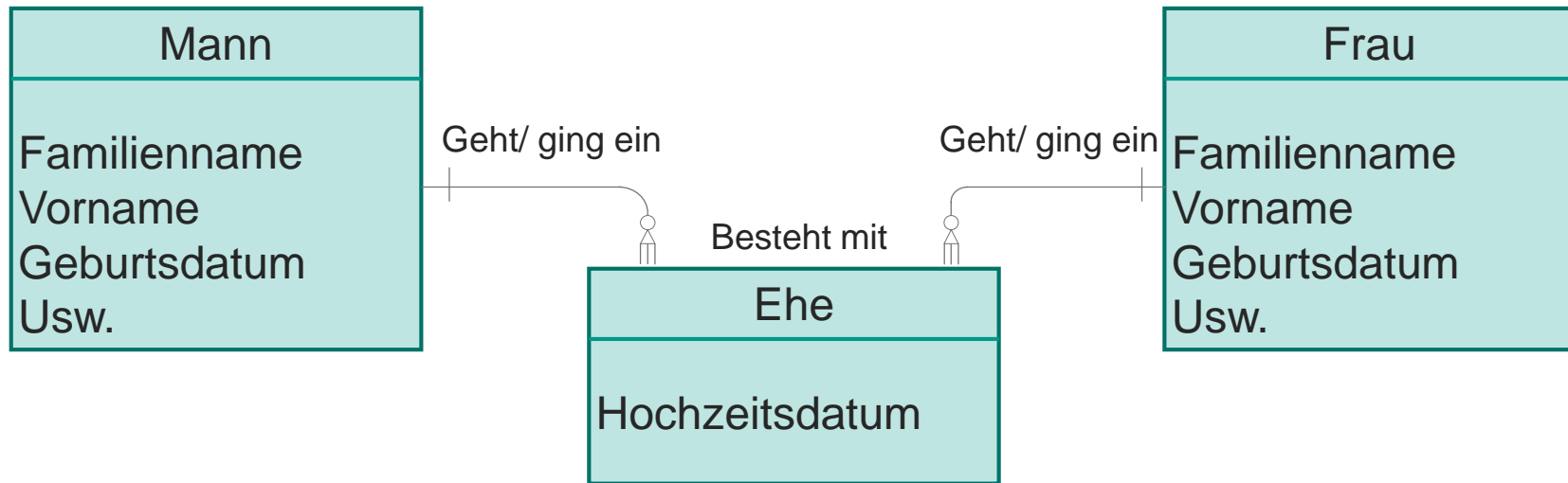


Eigenschaften von Beziehungstypen 1



Eigenschaften von Beziehungstypen lassen sich in der Krähenfußnotation nicht darstellen. Anders in der Chen-Notation!

Eigenschaften von Beziehungstypen 2



In der Krähenfußnotation muss der Beziehungstyp in einen Objekttyp transformiert werden, damit die Eigenschaften (des Beziehungstyps) zugeordnet werden können!

Eigenschaften von Eigenschaften

Datenmodelle müssen von Zeit zu Zeit angepasst werden!

Beispiel:

- Problem:
Für die jeweilige Marke soll ein Mindestpreis gespeichert werden.
- Lösung:
Die Marke wird in einem eigenen Objekttyp umgewandelt

Auto
Polizeiliches Kennzeichen
Marke
Farbe
Kilometerstand
Usw.



Das relationale Datenbank-Modell (12 Regeln von Codd) 1

1. Darstellung von Informationen:
Alle Informationen in einer relationalen Datenbank (einschließlich Namen von Tabellen und Spalten) sind explizit als Werte in Tabellen darzustellen.
2. Zugriff auf Daten:
Jeder Wert einer relationalen Datenbank muss durch eine Kombination von Tabellename, Primärschlüssel und Spaltenname auffindbar sein.
3. Systematische Behandlung von Nullwerten:
Das DBMS behandelt Nullwerte durchgängig gleich als unbekannte oder fehlende Daten und unterscheidet diese von Standardwerten.
(Probleme bei der Syntax „Null“ vs. „“)
4. Struktur einer Datenbank:
Die Datenbank und ihre Inhalte werden in einem sogenannten Systemkatalog auf derselben logischen Ebene wie die Daten selbst - also in Tabellen - beschrieben. Demzufolge lässt sich der Katalog mit Hilfe der Datenbanksprache abfragen.

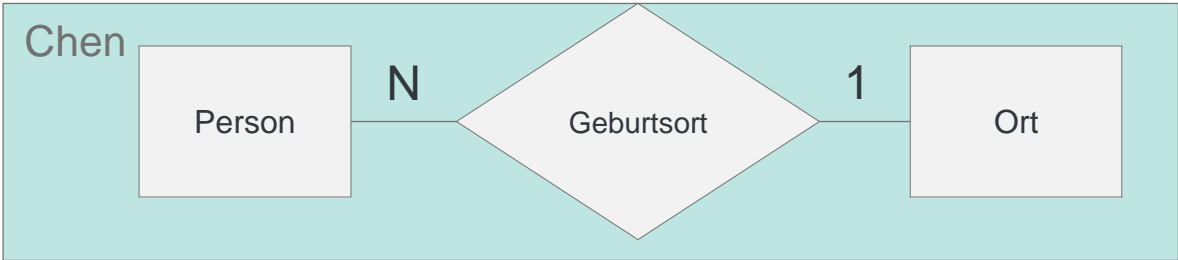
Das relationale Datenbank-Modell (12 Regeln von Codd) 2

5. Abfragesprache:
Zu einem relationalen System gehört mindestens eine Abfragesprache mit einem vollständigen Befehlssatz für Datendefinition, Manipulation, Integritätsregeln, Autorisierung und Transaktionen.
6. Aktualisieren von Sichten:
Alle Sichten, die theoretisch aktualisiert werden können, lassen sich auch vom System aktualisieren.
7. Abfragen und Bearbeiten ganzer Tabellen:
Das DBMS unterstützt nicht nur Abfragen, sondern auch die Operationen für Einfügen, Aktualisieren und Löschen in Form ganzer Tabellen.
8. Physikalische Datenunabhängigkeit:
Der logische Zugriff auf die Daten durch Anwendungen und Ad-Hoc-Programme muss unabhängig von den physikalischen Zugriffsmethoden oder den Speicherstrukturen der Daten sein.

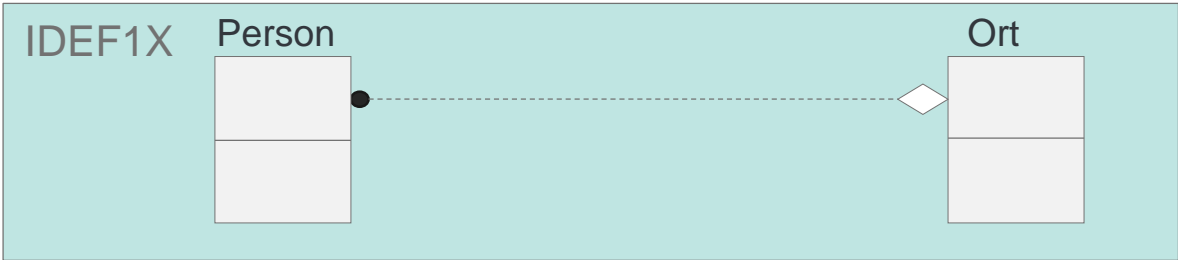
Das relationale Datenbank-Modell (12 Regeln von Codd) 3

9. Logische Datenunabhängigkeit:
Änderungen der Tabellenstrukturen dürfen keinen Einfluss auf die Logik der Anwendungen und Ad-Hoc-Programme haben.
10. Unabhängigkeit der Integrität:
Integritätsregeln müssen sich in der Datenbanksprache definieren lassen. Die Regeln müssen im Systemkatalog gespeichert werden. Es darf nicht möglich sein, die Regeln zu umgehen
11. Verteilungsunabhängigkeit:
Der logische Zugriff auf die Daten durch Anwendungen und Ad-Hoc-Programme darf sich beim Übergang von einer nicht-verteilten zu einer verteilten Datenbank nicht ändern.
12. Kein Unterlaufen der Abfragesprache:
Integritätsregeln, die über die Datenbanksprache definiert sind, dürfen sich nicht mit Hilfe von Low-Level- Sprachen umgehen lassen.

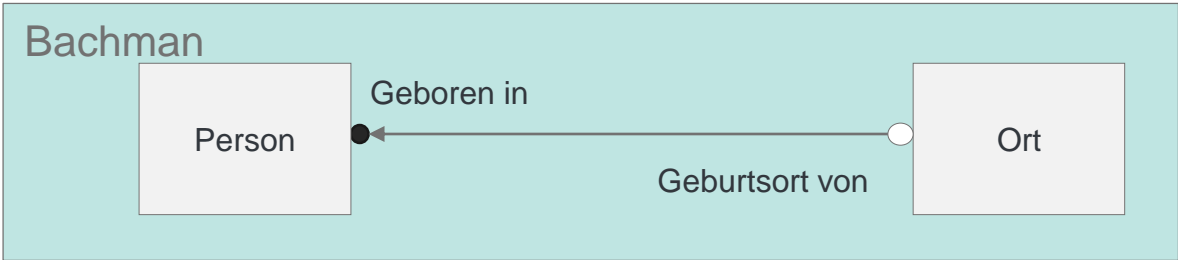
Die Chen-Notation von Peter Chen, dem Entwickler der ER-Diagramme, 1976.



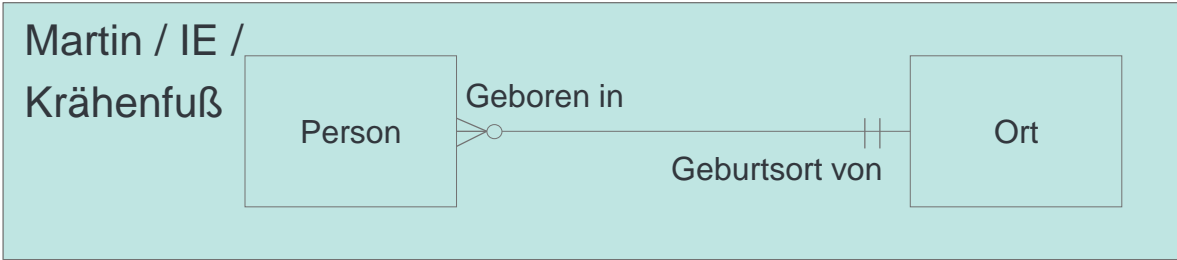
Die IDEF1X: langjähriger de-facto Standard bei U.S. amerikanischen Behörden.



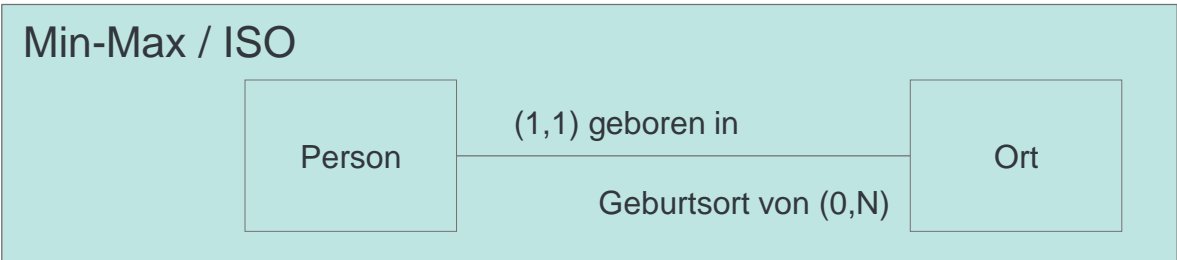
Die Bachman-Notation von Charles Bachman als weit verbreitete Werkzeug-Diagramm-Sprache.



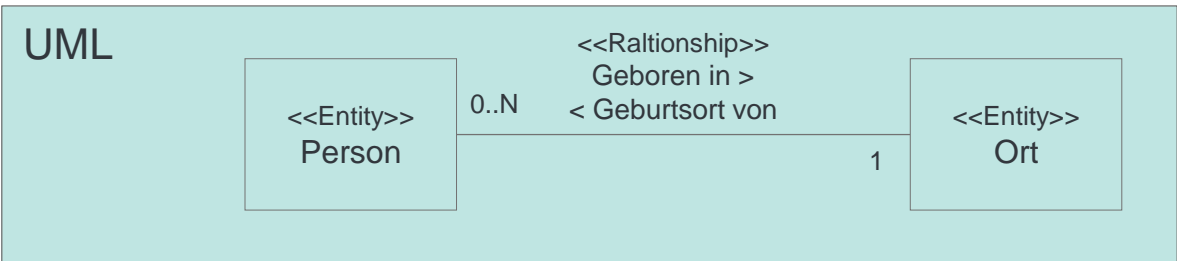
Die Martin-Notation (Krähenfuß - Notation) als weit verbreitete Werkzeug-Diagramm-Sprache (Information Engineering).



Die (min,max)-Notation von Jean-Raymond Abrial, 1974.



UML als Standard, den selbst ISO in eigenen Normen als Ersatz für ER-Diagramme verwendet.

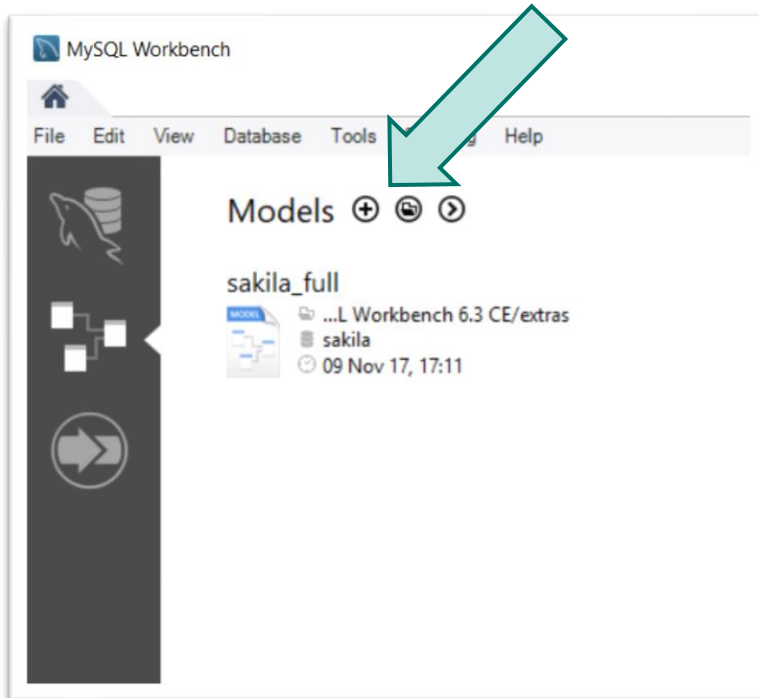


Pilot – Flug

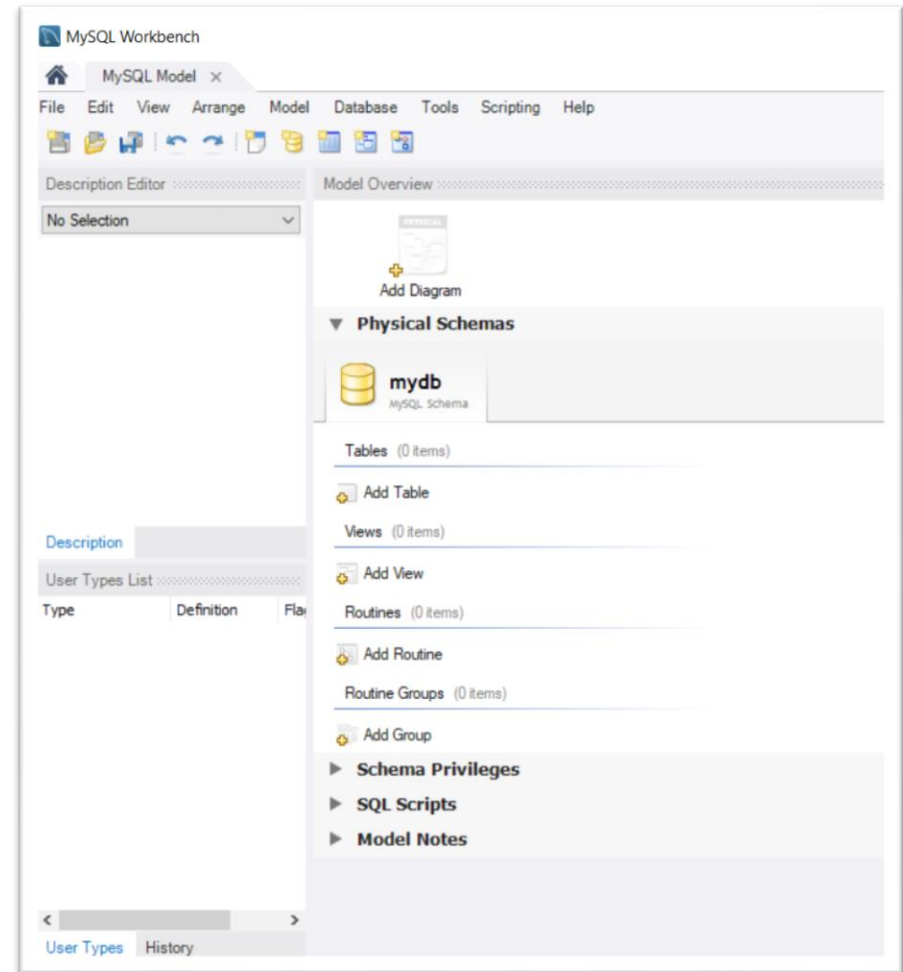
- Der Pilot hat eine Personalnummer und einen Name
 - Ein Flug findet an verschiedenen Tagen statt.
 - Ein Flug wird immer von einem Piloten durchgeführt.
 - Ein Flug hat einen Start-Ort und ein Ziel.
 - Ein Flug hat immer eine Distanz.
1. **Definieren Sie die sachlogische Zusammenhänge**
 2. **In einem ERM**

Erste Übungen in der MySQL-Workbench

Modellierung



Öffnen Sie die MySQL-Workbench und erstellen ein einfaches Modell



File Edit View Arrange Model Database Tools Scripting Help

Bird's Eye
Zoom: 100%

Catalog Tree
mydb
 Tables
 Views
 Routine Groups

Diagram

```

    graph TD
      pilot[pilot] -- idpilot INT --> roster[roster]
      flug[flug] -- idflug INT --> roster
      roster[roster] -- idpilot INT --> pilot
      roster[roster] -- idflug INT --> flug
  
```

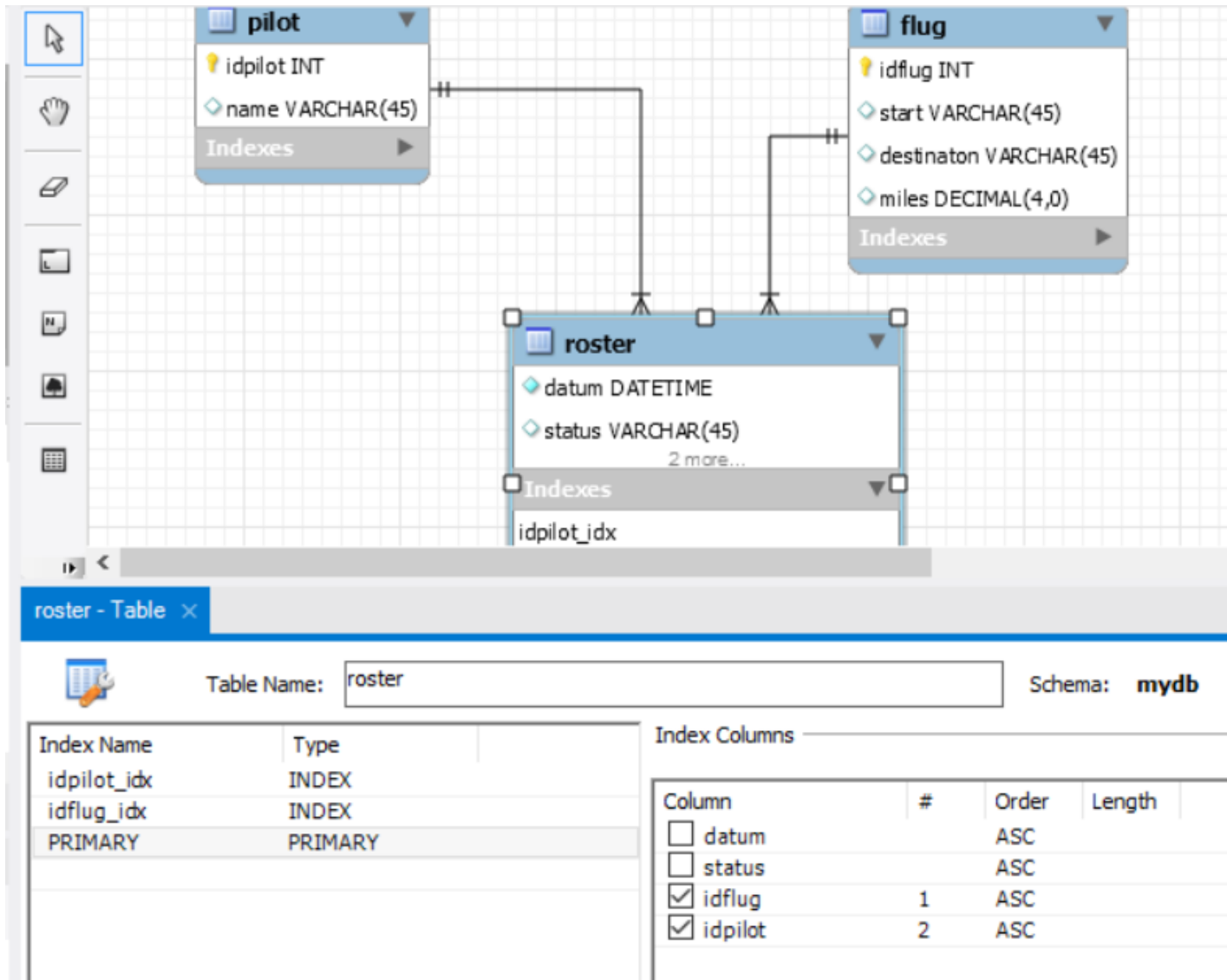
roster - Table x Relationship Relationship

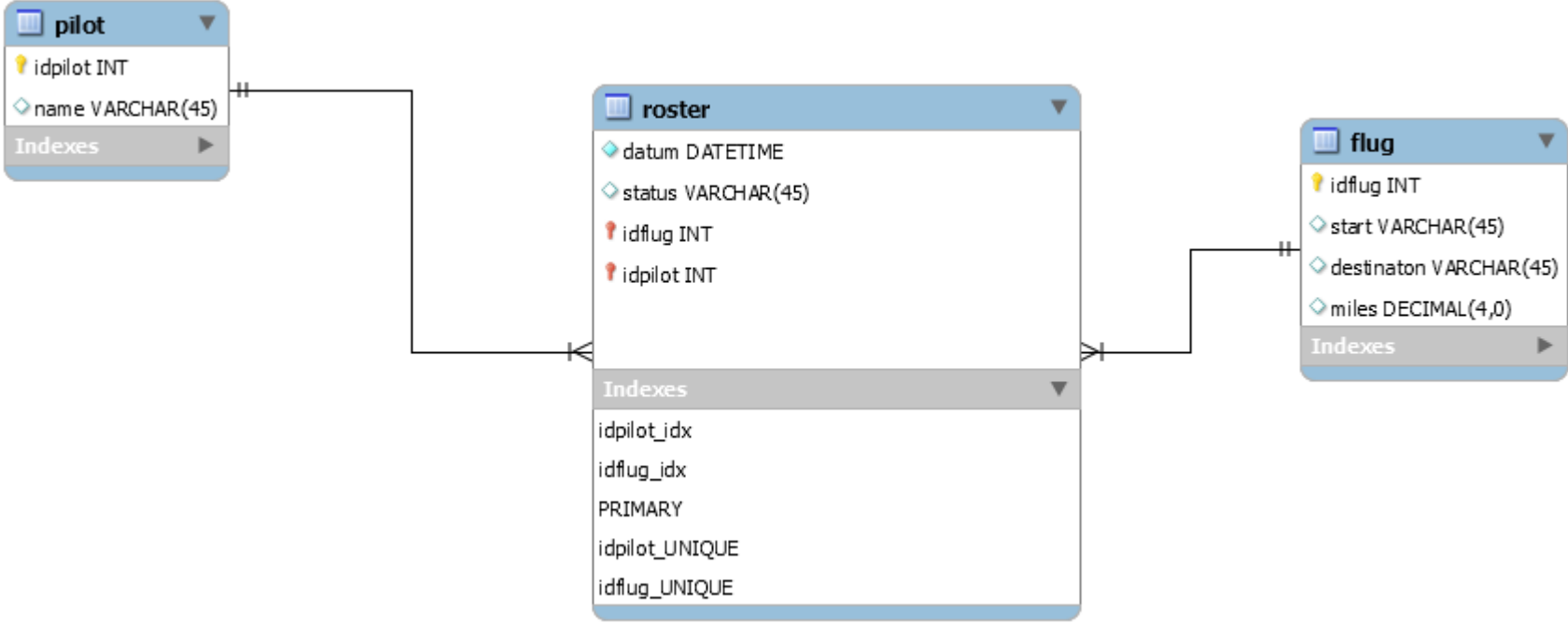
Table Name: roster Schema: mydb

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
datum	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
status	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
idflug	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
idpilot	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: idflug
Collation: Table Default
Comments:

Data Type: INT
Default:
Storage: ☐ Virtual ☐ Stored
☐ Primary Key ☐ Not Null ☐ Unique
☐ Binary ☐ Unsigned ☐ Zero Fill
☐ Auto Increment ☐ Generated





Übung

Für den Pizzaservice der Fa. Sator soll ein Datenbankentwurf angefertigt werden. Erstellen Sie für die folgenden Bedingungen ein entsprechendes Entity- Relationship-Modell und fügen Sie sinnvolle Attribute dem Model hinzu.

- der Pizzaservice hat mehrere Standorte
- der Pizzaservice liefert verschiedene Kategorien wie Wein, Pizza und Salate
- jedes Produkte wird in max. einer Kategorie zugeordnet
- die Kundendaten sollen mit Name, Vorname, Adresse und Telefonnummer gespeichert werden
- eine Bestellung kann mehrere Produkte enthalte

Aufgabe

Erstellen Sie das entsprechende Datenmodell