

# SQL

# Grundlagen

Grundlagen relationaler Datenbanksysteme

ROLLBACK COMMIT RDBMS DELETE VALUES DATABASE DEFAULT CREATE LEFT OUTER JOIN INSERT OUTER SAVEPOINT JOIN TABLE DROP EXPLAIN PLAN DEADLOCK TRANSACTION NUMERIC ISOLATION LEVEL DATA DEFINITION LANGUAGE SELECT INDEX VIEW INNER EXTRACT TRIGGER DML

# Verwendung der Unterlagen



- Bitte beachte, dass die vorliegenden Unterlagen als Begleitmaterial für die Schulung erstellt worden sind. Sie sind daher nur eingeschränkt zum Selbststudium geeignet.
- Die meisten der vorgestellten Konzepte, Anweisungen, Syntaxdiagramme u.ä. bieten noch weitere als die hier vorgestellten Möglichkeiten. Die Darstellung ist im Wesentlichen auf den Umfang der Grundlagenschulung eingeschränkt.

# Voraussetzungen

- Keine

# Übersicht

- Datenbanksysteme allgemein
- Relationale Datenbank-Managementsysteme
- Datenmodellierung & Normalisierung
- Darstellung von Datenmodellen
- Datenbanksprachen – Einordnung von SQL

# Merkmale von Datenbanksystemen

- Sammlung von verwandten Daten
  - Aus Sicht des Benutzers zusammengehörende Daten, z.B. eine Personaldatenbank, eine Lagerinventardatenbank, Mediendatenbank,...
- Organisation nach natürlichen Gesichtspunkten, nicht anwendungsbezogen
- Trennung von Daten und Anwendungsprogramm
- Verwaltung durch ein Datenbank-Managementsystem (DBMS)
- Datenbanksystem = Datenbank + DBMS

# Datenbank-Managementsysteme (DBMS)

- Kapselt den Zugriff auf die Daten (kein direkter Zugriff für Benutzer)
- Überprüfung der Datenintegrität und -konsistenz
- Physische und logische Sicht voneinander unabhängig
- Daten können als Basis für andere Anwendungen genutzt werden

# Verschiedene Datenbankmodelle

- Ein Datenbankmodell definiert sich aus drei Eigenschaften (nach E.F. Codd):
  - generische Datenstruktur
  - generische Operatoren
  - Integritätsbedingungen

# Verschiedene Datenbankmodelle

- Beispiele für Datenbankmodelle:
  - Hierarchisches Datenbankmodell
  - Netzwerkdatenbankmodell (s.a. *CODASYL*, von Cobol beeinflusst)
  - Relationales Datenbankmodell
  - Objektrelationales Datenbankmodell
  - Objektorientiertes Datenbankmodell
  - Dokumentenorientiertes Datenbankmodell

# Eigenschaften von Verarbeitungsschritten

- ACID-Eigenschaften
  - Atomicity
    - Alles oder Nichts
  - Consistency
    - Eine Sequenz von Operationen auf einem konsistenten Datenbestand erzeugt einen konsistenten Datenbestand (erreicht durch Normalisierung und Integritätsbedingungen)
  - Isolation
    - Nebenläufige Operationen beeinflussen sich nicht
    - Der Grad der erlaubten Beeinflussung wird üblicherweise durch den Isolation Level bei Transaktionen gesteuert
  - Durability
    - Nach Abschluss einer Operation sind die Daten garantiert dauerhaft gespeichert

# Eigenschaften von Verarbeitungsschritten

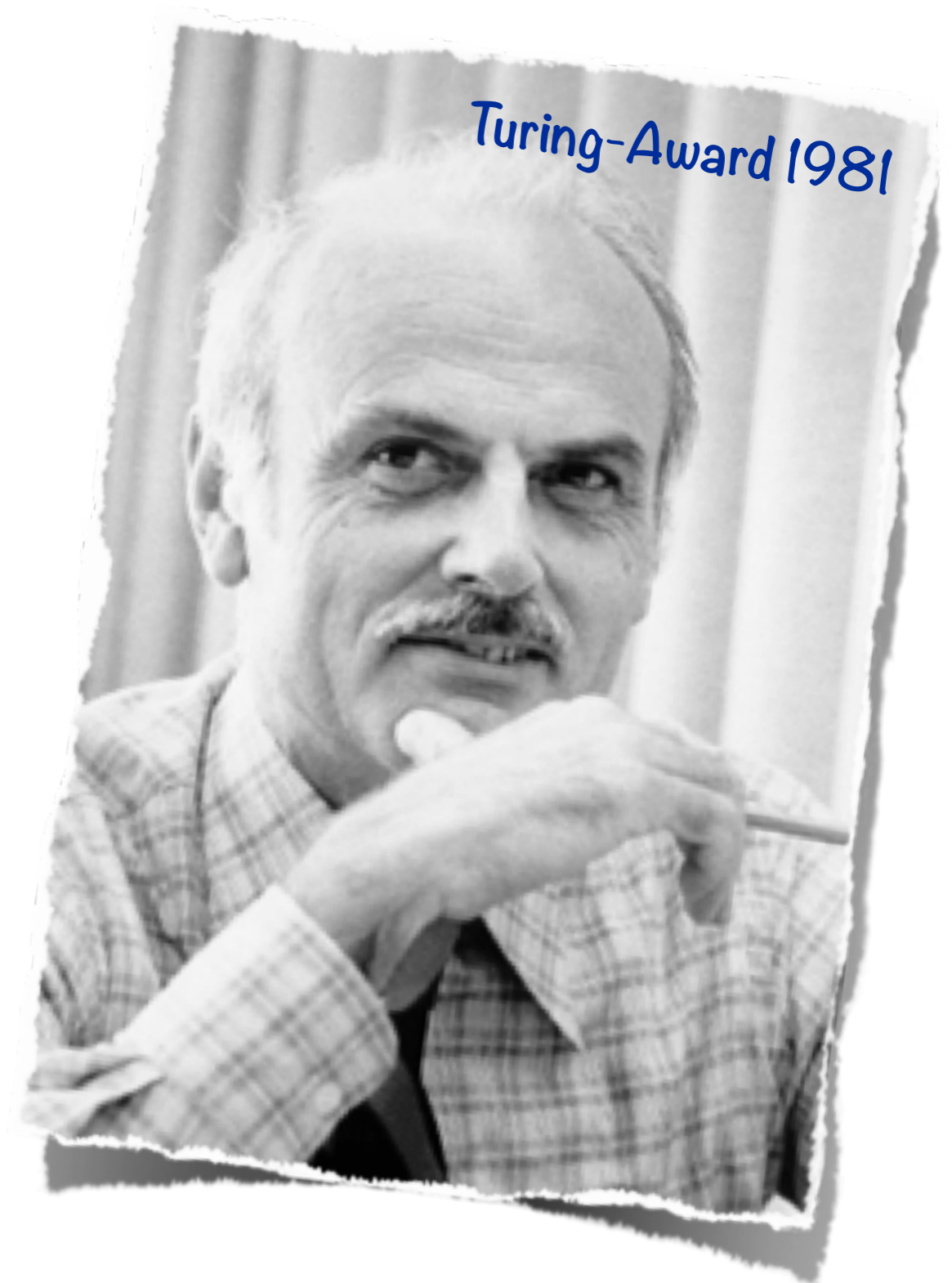
- In verteilten Datenbanken ist die Erfüllung aller ACID-Eigenschaften problematisch, daher wird für Datenbanksysteme mit entsprechendem Anforderungsprofil oft das BASE-Prinzip verfolgt
  - Basically Available
  - Soft state
  - Eventual consistency

# Mögliche Infrastruktur

- Datenbanksysteme sind oftmals als Client / Server Systeme aufgebaut – Datenbank und DBMS werden auf einem Server zur Verfügung gestellt, der Zugriff erfolgt von den Clients
- Datenbanken können auch lokal sein, z.B. zur Datenhaltung einer lokal installierten Applikation (Einstellungen, Daten,...)
- Der Zugriff ist durch die Abstraktion des DBMS gleich

# Relationale Datenbanksysteme (1)

- Das Konzept wurde in den frühen 70er Jahren von Edgar Frank Codd (1923 – 2003) entwickelt.
- Daten werden in Relationen (Tabellen) gespeichert.
- Das relationale Modell basiert auf der Relationenalgebra. Daten werden als Mengen betrachtet. Das mathematische Fundament ist die Mengenlehre.



# Relationale Datenbanksysteme (2)

Theorie	Praxis
Relation	Tabelle (table)
Tupel	Zeile (row)
Tupelelement	Spalte (column)
Wertebereich	Datentyp (type)

# Tabellen – Grundlagen (1)

- Eine relationale Datenbank enthält Tabellen, in denen die Daten abgelegt werden
- Tabellen werden durch ihren Namen eindeutig identifiziert
- Tabellen bestehen aus Zeilen und Spalten

# Tabellen – Grundlagen (2)

- Spalten haben einen pro Tabelle eindeutigen Namen, einen Datentyp (und weitere Attribute)
- Alle Werte in einer Spalte sind von gleicher Art
- Zeilen enthalten die Daten für die Spalten
- Ein einzelner Datensatz (Zeile) lässt sich über seinen Primärschlüssel eindeutig identifizieren
- Die Reihenfolge der Zeilen und Spalten ist beliebig

# Beispiele



# Datenmodellierung – „Naiver“ Ansatz

- Fiktiver Anwendungsfall: Eine Datenbank, mit der eine kleine Kette von Boutiquen für Zirkus- und Freizeitbedarf die Etikettierung innerhalb der Filialen abwickelt.
- Erster Ansatz: Eine Tabelle mit Artikelnummer, Artikelbezeichnung, dem Preis des Artikels, der Produktart des Artikels sowie der Adresse der Filiale.

Artikelnummer	Bezeichnung	Preis	Produktart	Produktbez	Filiale
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	16, Rue Albert Henri, 75001 Paris
100000	Bean-Bag Neon, Kunstleder, 65mm, 130g	4,90	BNBG	Bean-Bags	19, Rue Aristide Briand, 13002 Marseille
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	17, Rue Blanche Vitte, 69003 Lyon
100300	Bean-Bag Echtleder	15,60	BNBG	Bean-Bags	16, Rue Albert Henri, 75001 Paris
100400	Bean-Bag Silikon	6,00	BNBG	Bean-Bags	17, RUE BLANCHE VITTE, 69003 LYON
102300	Turbo Bouncing Ball	10,90	STGB	Bühnenbälle	16, Rue Albert Henri, 75001 Paris

# Datenmodellierung – Problemanalyse

Artikelnummer	Bezeichnung	Preis	Produktart	Produktbez	Filiale
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	16, Rue Albert Henri, 75001 Paris
100000	Bean-Bag Neon, Kunstleder, 65mm, 130g	4,90	BNBG	Bean-Bags	19, Rue Aristide Briand, 13002 Marseille
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	17, Rue Blanche Vitte, 69003 Lyon
100300	Bean-Bag Echtleder	15,60	BNBG	Bean-Bags	16, Rue Albert Henri, 75001 Paris
100400	Bean-Bag Silikon	6,00	BNBG	Bean-Bags	17, RUE BLANCHE VITTE, 69003 LYON
102300	Turbo Bouncing Ball	10,90	STGB	Bühnenbälle	16, Rue Albert Henri, 75001 Paris

- Hohes Datenvolumen durch mehrfach vorhandene, gleiche Information (Redundanz)
- Abweichende Schreibweisen oder Tippfehler machen es z.B. schwierig, alle Artikel der Filiale in Lyon zu finden
- In der Spalte Filiale sind viele Informationen zusammengefasst, die in jeder Zeile neu erfasst werden müssen (Mehraufwand / Probleme z.B. bei Adressänderung)
- Widersprüchliche Informationen z.B. beim Artikel 100000



# Normalisierung

- Das Speichern vieler oder gar aller Daten in einer einzelnen Tabelle birgt das Risiko logischer Fehler
- Die Daten sollten so modelliert werden, dass
  - Redundanzen (mehrfache Speicherung gleicher Inhalte) vermieden werden
  - Daten einfach erfasst und bearbeitet werden können
  - Anomalien vermieden werden
    - Einfügeanomalie: Daten können nicht eingefügt werden, weil davon unabhängige Daten fehlen
    - Änderungsanomalie: Bei der Änderung von Daten werden unbeabsichtigt auch davon unabhängige Daten verändert
    - Löschanomalie: Werden Daten gelöscht, gehen davon unabhängige Daten verloren

# Normalisierung – Normalformen

- Die durch E.F. Codd begründete Theorie der Normalisierung beschreibt fünf sogenannte Normalformen
- Die Datenbanken werden schrittweise in die jeweils nächsthöhere Normalform versetzt
- In der Praxis wird üblicherweise zumindest die dritte Normalform angestrebt

# Normalisierung – 1. Normalform

- Eine Relation ist in der ersten Normalform, wenn jedes Attribut der Relation einen atomaren Wertebereich hat.
  - D.h. jeder Kreuzungspunkt von Zeile und Spalte besitzt höchstens einen Wert.
  - Es gibt keine Wiederholungsgruppen (wie z.B. preis\_1, preis\_2, preis\_3)
- Verletzungen der ersten Normalform im Beispiel:
  - Die Spalte Filiale enthält die Attributwertebereiche Straße, Postleitzahl und Ort.

# Normalisierung – 1. Normalform Beispiel

- Überführung des Beispiels in die erste Normalform:

Artikelnummer	Bezeichnung	Preis	Produktart	Produktbez	Filiale
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	16, Rue Albert Henri, 75001 Paris
100000	Bean-Bag Neon, Kunstleder, 65mm, 130g	4,90	BNBG	Bean-Bags	19, Rue Aristide Briand, 13002 Marseille
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	17, Rue Blanche Vitte, 69003 Lyon
100300	Bean-Bag Echtleder	15,60	BNBG	Bean-Bags	16, Rue Albert Henri, 75001 Paris
100400	Bean-Bag Silikon	6,00	BNBG	Bean-Bags	17, RUE BLANCHE VITTE, 69003 LYON
102300	Turbo Bouncing Ball	10,90	STGB	Bühnenbälle	16, Rue Albert Henri, 75001 Paris



Artikelnummer	Bezeichnung	Preis	Produktart	Produktbez	Strasse	PLZ	Ort
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	16, Rue Albert Henri	75001	Paris
100000	Bean-Bag Neon, Kunstleder, 65mm, 130g	4,90	BNBG	Bean-Bags	19, Rue Aristide Briand	13002	Marseille
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	17, Rue Blanche Vitte	69003	Lyon
100300	Bean-Bag Echtleder	15,60	BNBG	Bean-Bags	16, Rue Albert Henri	75001	Paris
100400	Bean-Bag Silikon	6,00	BNBG	Bean-Bags	17, RUE BLANCHE VITTE	69003	LYON
102300	Turbo Bouncing Ball	10,90	STGB	Bühnenbälle	16, Rue Albert Henri	75001	Paris

# 1. Normalform – Streitfragen

- Und was ist im Beispiel mit der Hausnummer?
  - Wenn Straße und Hausnummer eventuell getrennt betrachtet werden – z.B. in einer Anwendung mit Adressprüfung (eine PLZ kann z.B. in Deutschland je nach Hausnummer in einer Straße unterschiedlich sein) – dann reicht das vorgestellte Beispiel **nicht** als 1. Normalform; die Hausnummer muss dann getrennt von der Straße modelliert werden.
  - Wenn man davon ausgehen kann, dass die Hausanschrift immer als Einheit angesprochen wird, dann kann sie als einzelnes Textattribut modelliert werden.

# Normalisierung – 2. Normalform

- Eine Relation ist in der zweiten Normalform, wenn die erste Normalform vorliegt und jedes Nichtschlüsselattribut von jedem Schlüsselkandidaten voll funktional abhängig ist.
  - D.h. jedes Attribut, das nicht Teil eines Schlüssels ist, ist jeweils vom ganzen Schlüssel abhängig, nicht nur von einem Teil eines Schlüssels.
  - “B ist von A funktional abhängig” wenn gilt, aus A folgt B ( $A \Rightarrow B$ ).
- Praktischer Nutzen der zweiten Normalform: Jede Relation modelliert nur einen Sachverhalt.

# Normalisierung – 2. Normalform

- Übergang zur zweiten Normalform:
  - Alle nicht voll funktional abhängigen Nichtschlüsselattribute auslagern
  - Neue Relationen für die logisch zusammenhängenden Attributkombinationen mit dem korrespondierenden Schlüssel als Primärschlüssel
  - Duplikate eliminieren
- Die Beispieltabelle wird zerlegt in Artikel, Filiale, Preis und die bislang implizite Zuordnung zwischen Artikel und Filiale (Listung)

# Normalisierung – 2. Normalform Beispiel

Artikelnummer	Bezeichnung	Preis	Produktart	Produktbez	Strasse	PLZ	Ort
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	16, Rue Albert Henri	75001	Paris
100000	Bean-Bag Neon, Kunstleder, 65mm, 130g	4,90	BNBG	Bean-Bags	19, Rue Aristide Briand	13002	Marseille
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	4,90	BNBG	Bean-Bags	17, Rue Blanche Vitte	69003	Lyon
100300	Bean-Bag Echtleder	15,60	BNBG	Bean-Bags	16, Rue Albert Henri	75001	Paris
100400	Bean-Bag Silikon	6,00	BNBG	Bean-Bags	17, RUE BLANCHE VITTE	69003	LYON
102300	Turbo Bouncing Ball	10,90	STGB	Bühnenbälle	16, Rue Albert Henri	75001	Paris



Artikelnummer	Bezeichnung	Produktart	Produktbez
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	BNBG	Bean-Bags
100300	Bean-Bag Echtleder	BNBG	Bean-Bags
100400	Bean-Bag Silikon	BNBG	Bean-Bags
102300	Turbo Bouncing Ball	STGB	Bühnenbälle

Artikelnummer	Filialnummer
100000	1000
100300	1000
100400	1000
102300	1000
100000	1010
100000	1020

Annahme: Einheitlicher  
Preis in allen Filialen

Artikelnummer	Preis
100000	4,90
100300	15,60
100400	6,00
102300	10,90

Filialnummer	Strasse	PLZ	Ort
1000	16, Rue Albert Henri	75001	Paris
1010	19, Rue Aristide Briand	13002	Marseille
1020	17, Rue Blanche Vitte	69003	Lyon

# Normalisierung – 3. Normalform

- Eine Relation ist in der dritten Normalform, wenn die zweite Normalform vorliegt und kein Nichtschlüssel von einem Schlüsselkandidaten transitiv abhängt.
- Die dritte Normalform behebt in der zweiten Normalform verbliebene thematische Durchmischungen der Relationen.

# Normalisierung – 3. Normalform Beispiel

- Der Zusammenhang zwischen der Bezeichnung der Produktgruppe und dem Artikel ist indirekt – die Bezeichnung hängt von der Produktart ab, die wiederum vom Artikel abhängig ist. Die Bezeichnung hängt also transitiv von der Artikelnummer ab.

Artikelnummer	Bezeichnung	Produktart	Produktbez
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	BNBG	Bean-Bags
100300	Bean-Bag Echtleder	BNBG	Bean-Bags
100400	Bean-Bag Silikon	BNBG	Bean-Bags
102300	Turbo Bouncing Ball	STGB	Bühnenbälle



Artikelnummer	Bezeichnung	Produktart	Produktart	Bezeichnung
100000	Bean-Bag Neon, Kunstleder, 70mm, 130g	BNBG	BNBG	Bean-Bags
100300	Bean-Bag Echtleder	BNBG	STGB	Bühnenbälle
100400	Bean-Bag Silikon	BNBG		
102300	Turbo Bouncing Ball	STGB		

# 3. Normalform – Streitfragen

- Und was ist im Beispiel mit der Abhängigkeit des Ortsnamens von der Postleitzahl?
- Ähnlich wie bei der Frage nach der Modellierung von Straße und Hausnummer ist abzuwägen, inwiefern die Abhängigkeit zwischen Postleitzahl und Ortsname bedeutsam ist. Wenn z.B. postalisch „unkorrekte“ Teilortsnamen erlaubt sein sollen, dann ist die Modellierung als einfache Attribute sinnvoll.

# Normalisierung – Zusammenfassung

- Zusammenfassung der Normalformen:
  - Jede Spalte hat einen und nur einen Wert
  - Der Wert jeder Spalte hängt vom Wert des Primärschlüssels ab
  - Der Wert jeder Spalte hängt nur vom Wert des Primärschlüssels ab
- Merksatz: “*The key, the whole key, and nothing but the key – so help me Codd.*”

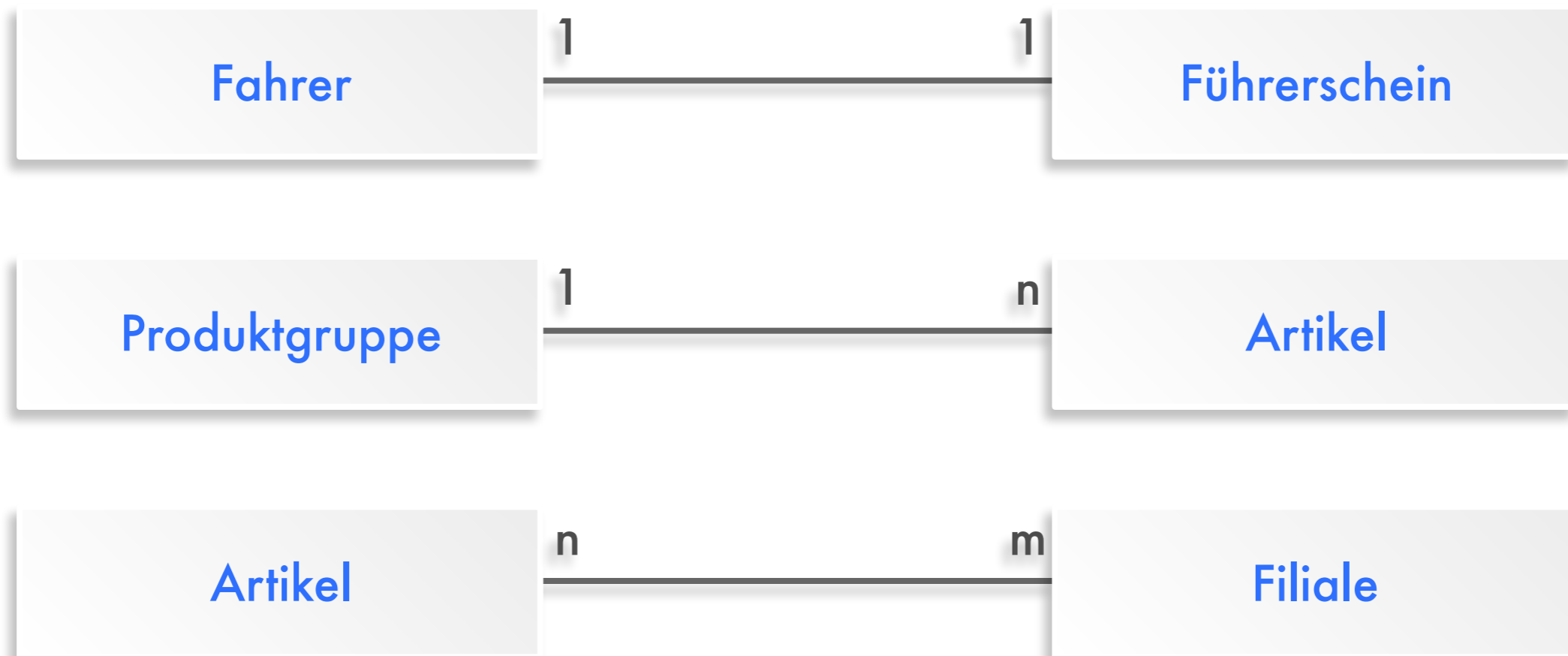
# Normalisierung – Ausnahmen

- In speziellen Anwendungsfällen kann es sinnvoll sein, im Detail eine gezielte Denormalisierung zuzulassen.
  - Performance
  - Abfragen vereinfachen
  - Besonderheiten des Geschäftsprozesses
- Beispiel Adressen – je nach Kontext können verschiedene Modellierungen eines Sachverhalts sinnvoll sein.
- Grundsätzlich ist jedoch eine normalisierte Datenbank vorzuziehen!

# Tabellenbeziehungen (1)

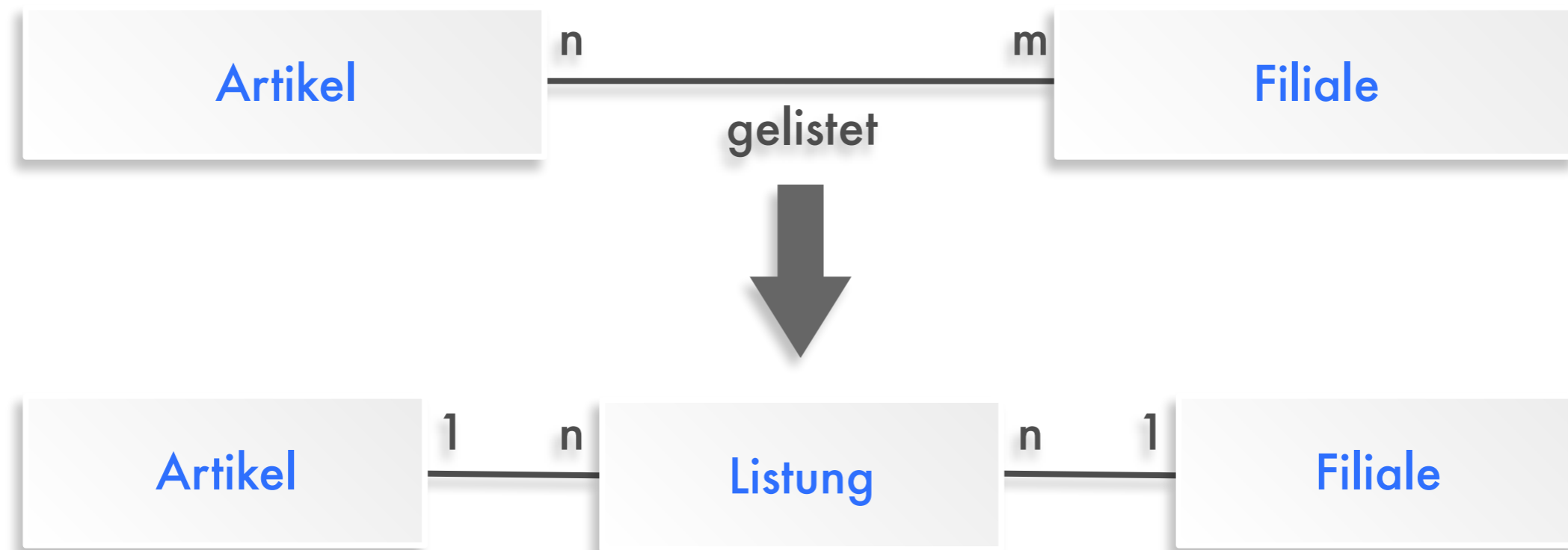
- 1:1
  - Jeder Datensatz der einen Tabelle ist mit höchstens einem Datensatz der anderen Tabelle verbunden
- 1:n
  - Jeder Datensatz der einen Tabelle kann mit mehreren Datensätzen der anderen Tabelle verbunden sein
- n:m
  - Jeder Datensatz der einen Tabelle kann mit mehreren Datensätzen der anderen Tabelle verbunden sein und umgekehrt

# Tabellenbeziehungen (2)



# Auflösung von $n:m$ Beziehungen

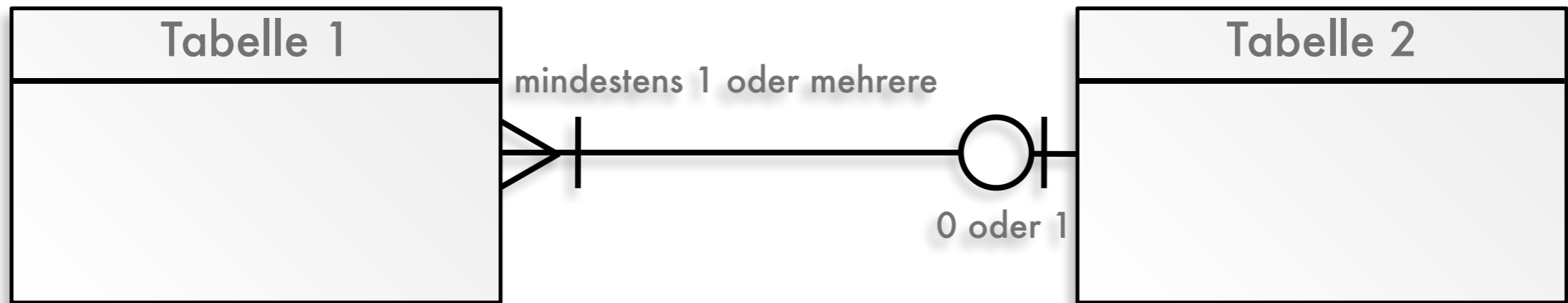
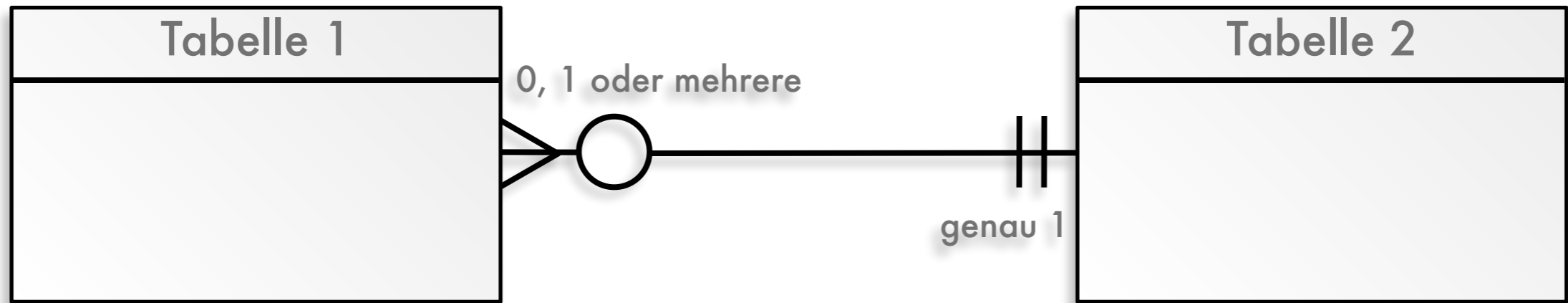
- $n:m$  Beziehungen können technisch nicht direkt abgebildet werden
- Der Bezug wird über eine Zwischentabelle abgebildet, die beide Ausgangstabellen jeweils über  $1:n$  Beziehungen verknüpft



# Datenbankdiagramme

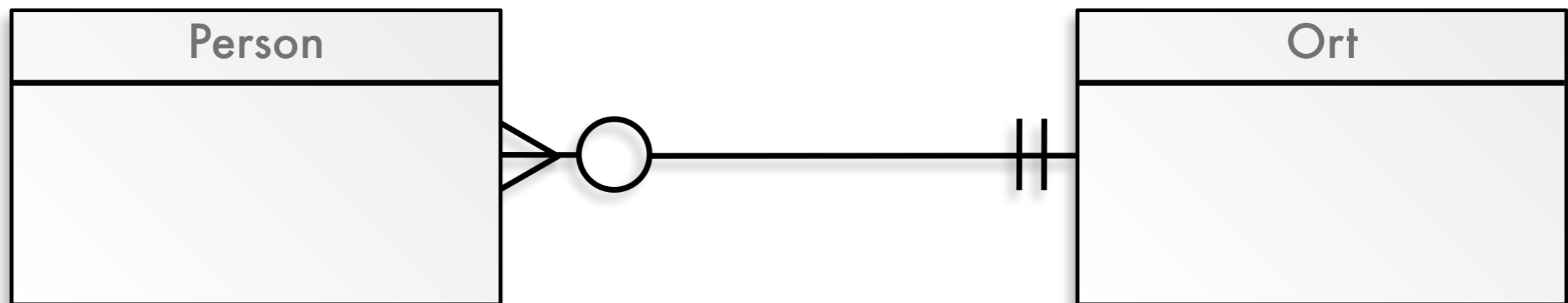
- Die „Krähenfuß-“ oder „Martin-Notation“
  - Drei verschiedene Symbole für Kardinalitäten
    - Ring für 0
    - Strich für 1
    - „Krähenfuß“ für mehrere (n)
  - Die Kombination aus je einem Symbol für die minimale und die maximale Kardinalität ergibt die vier (aussagekräftigen) Kombinationen:
    - 0 oder 1
    - genau 1
    - 0, 1 oder mehrere
    - mindestens 1 oder mehrere

# “Krähenfuß”-Notation – Beispiele



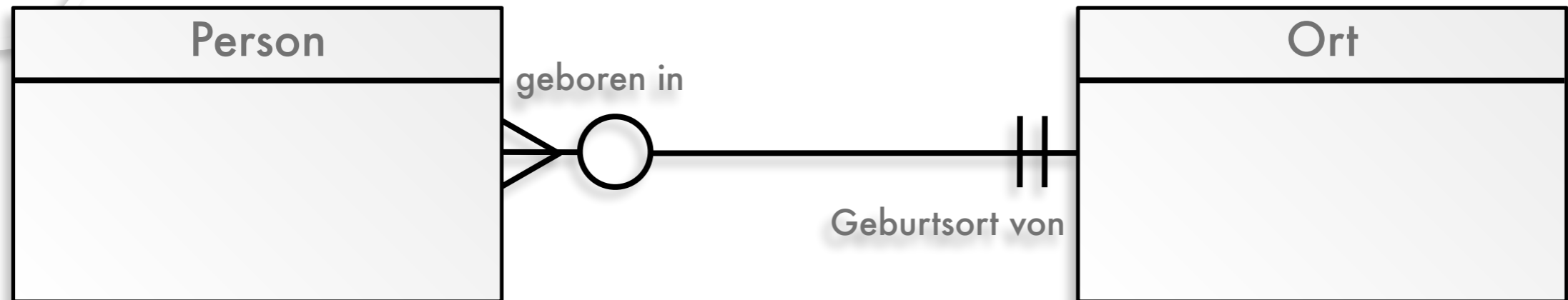
# “Krähenfuß”-Notation – Lesart

- Die “Krähenfuß”-Notation wird wie folgt gelesen:
  - Die zwei in Beziehung stehenden Entitäten auswählen
  - Diejenige der beiden auswählen, für die eine Aussage getroffen werden soll
  - Die Notation auf Seite der anderen Entität liefert die gewünschte Aussage
- Beispiel Geburtsort einer Person:
  - Eine Person hat genau einen Geburtsort, in einem Ort können beliebig viele Personen geboren worden sein



# Verschiedene Darstellungsformen (1)

Martin

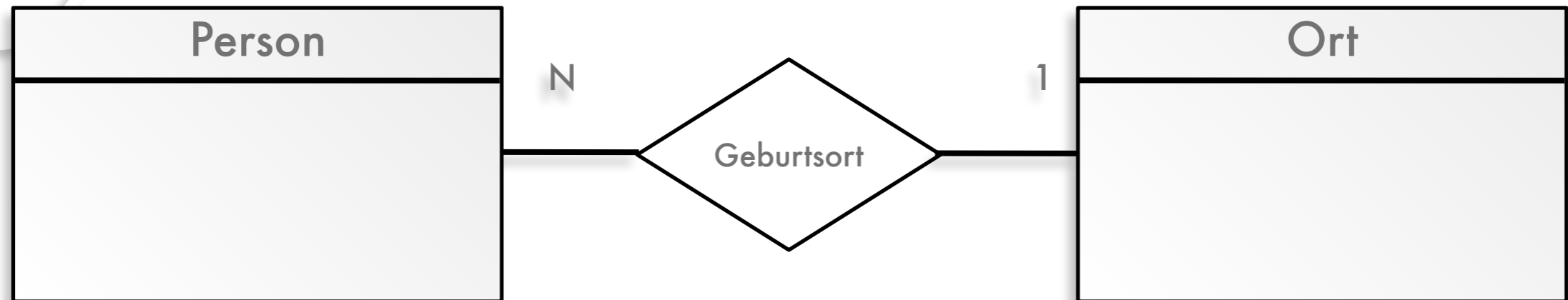


Min -  
Max /  
ISO



# Verschiedene Darstellungsformen (2)

Chen



UML



# Anforderungen an Datenbanksprachen

- Daten definieren
  - Wie sind die Daten angeordnet, welche Werte können sie annehmen
- Daten abfragen und verändern
- Zugriff steuern
- Konsistenz gewährleisten
- Die derzeit bekannteste und am weitesten verbreitete Datenbanksprache ist SQL