

Client-Server-Architekturen

Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

Relational

ER Daten

SQL

DML

NoSQL

Recovery

Big Data

OODBMS DDL

EER-Modell

Transaktion

Datenbank

Normalisierung

Architektur

Modell

Relationale Algebra

Informationssysteme 1: Grundlagen von Datenbanken

Das ER-Modell

Wintersemester 2019/2020

Ralf Krause

Department für Informatik
Abteilung Informationssysteme

- **DB-Konzepte und –Architektur**
- **Modellierung von Datenbanken**
 - Das ER- und EER-Modell
- **Wichtige Grundlagen**
 - Das relationale Modell
 - Vom ER-Modell zum rel. Modell
 - Relationale Algebra und Kalküle
- **Abfrage und Administration**
 - SQL (DDL und DML)
- **Guter Entwurf**
 - Normalisierung + Normalformen

- **Datenbanken im Mehrbenutzerbetrieb**
 - Transaktionsverarbeitung
 - Recovery
- **Weitere Themen**
 - Aktive Datenbanken
 - Objektorientierte und objektrelationale Datenbanken
- **Blick über den Tellerrand**
 - Weiterführende Konzepte
 - Big Data und NoSQL

- Auftrag:
 - Erstellen einer neuen Datenbank(-anwendung)
 - Z.B. Online-Shop
- Fragen:
 - Wie geht man vor?
 - Welche Daten braucht man?
 - Welche Beziehungen zwischen den Daten sind wichtig?
 - Wie kann ich mit dem Auftraggeber kommunizieren?



- Die Phasen des DB-Entwurfs
- Modellierung
- Das **Entity-Relationship-Modell** (ER-Modell)
 - Anschauliches Beispiel (Online-Shop-DB)
 - Komponenten des ER-Modells
- Beispiele für ER-Schemata

Client-Server-Architekturen

Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

Relational

ER Daten SQL DML
NoSQL Recovery Big Data
OODBMS DDL EER-Modell
Transaktion Datenbank Normalisierung
Architektur

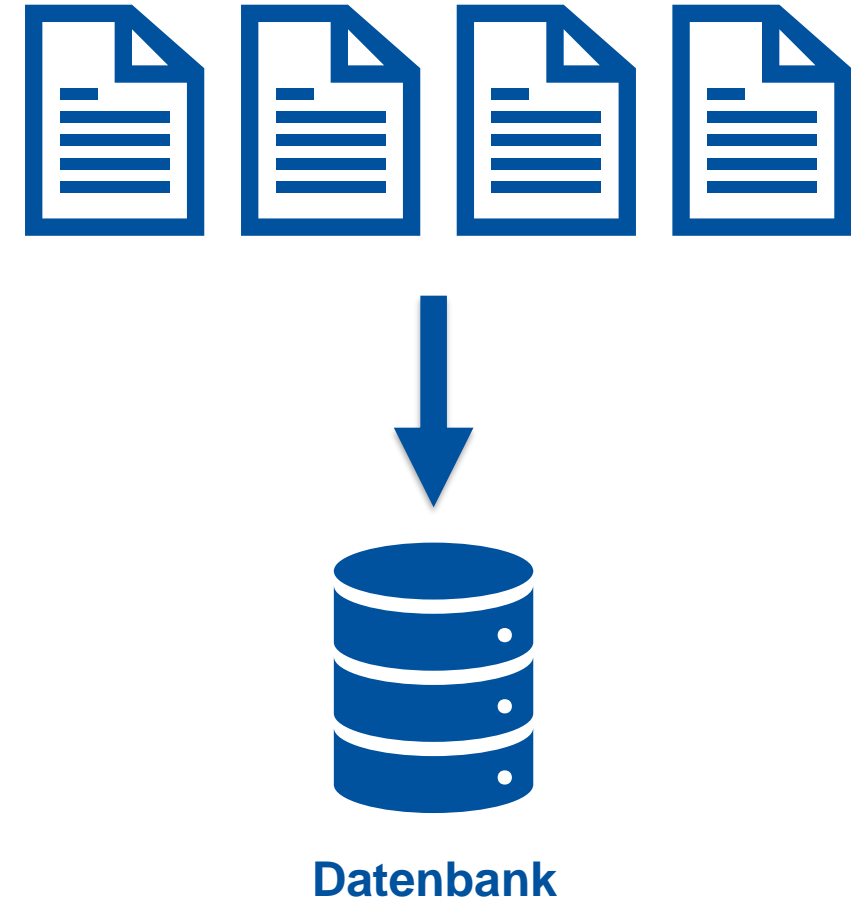
Modell

Relationale Algebra

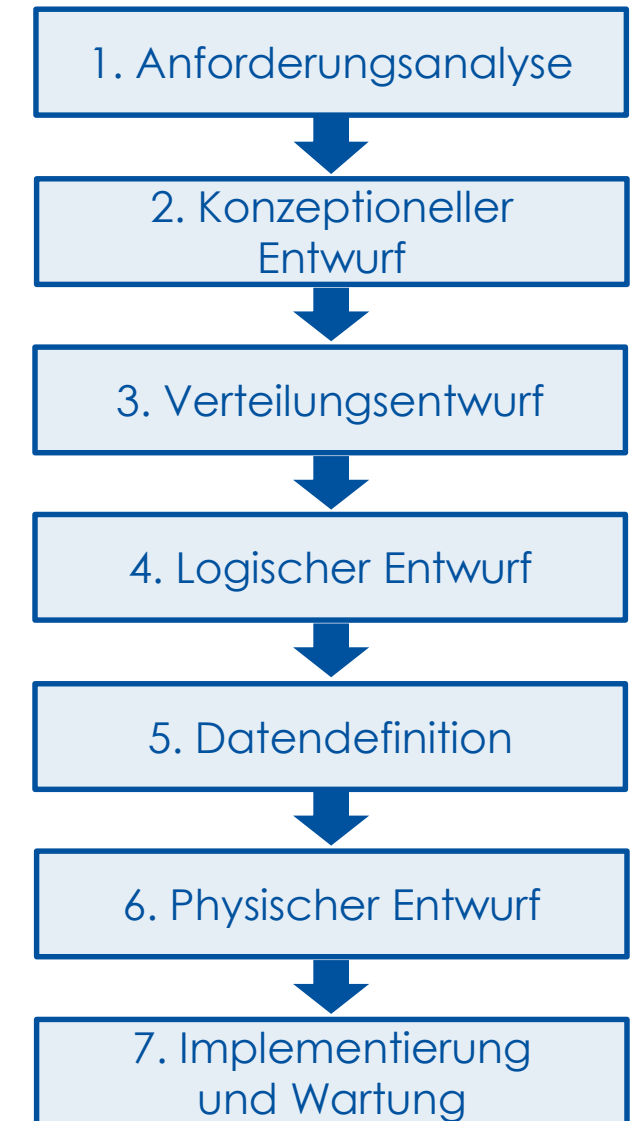
Entwurfsprozess von Datenbankentwurf

- 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
 - Verändert sich heute ein wenig
→ **Big Data**
 - Nicht redundante Speicherung

- **Abfolge von Entwurfsdokumenten**
 - Von Abstrakter Beschreibung
 - Bis zur tatsächlichen Realisierung in einem DBMS
 - Verschiedene Beschreibungsformalismen
 - ER, Relationenmodell, SQL DDL usw.
- In jedem Schritt
 - Informationserhaltung
 - Konsistenzerhaltung (Widerspruchsfreiheit)



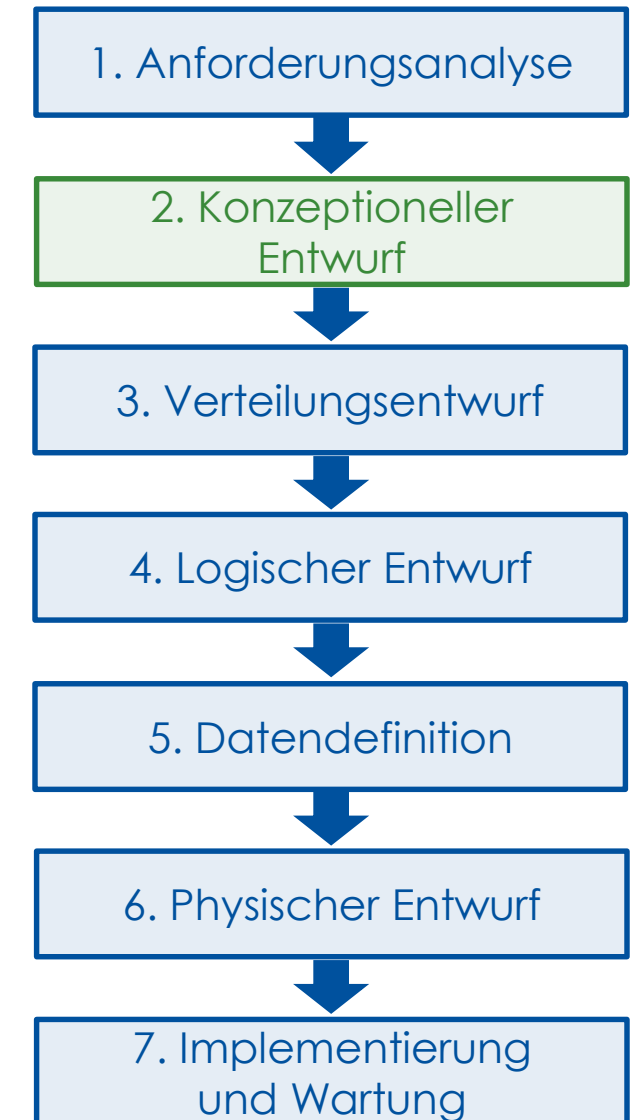
1. Anforderungsanalyse
 - Informelle Beschreibung
2. Konzeptioneller Entwurf
 - Entity-Relationship-Diagramm
3. Verteilungsentwurf
 - Partitionierung
4. Logischer Entwurf
 - Relationenschemata
5. Datendefinition
 - SQL DDL
6. Physischer Entwurf
 - Parameter und Indizes
7. Implementierung und Wartung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



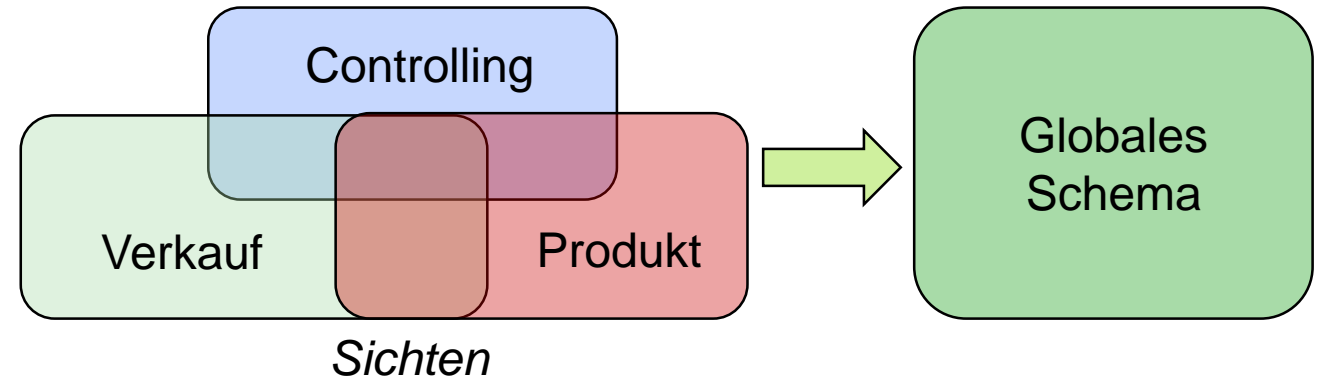
- **Vorgehensweise**
 - Sammlung des Informationsbedarfs
- „Klassischer“ DB-Entwurf
 - Nur Datenanalyse und Bestimmung der Folgeschritte
- Funktionsentwurf
 - (zunächst) Nicht Aufgabe des DB-Entwurfs
- **Ergebnis**
 - Informelle Beschreibung des Fachproblems
 - Texte, tabellarische Aufstellungen, Formblätter, ...
 - Trennen der Information über Daten (Datenanalyse) von den Informationen über Funktionen (Funktionsanalyse)



- Erste formale Beschreibung des Fachproblems
 - Sprachmittel: semantisches Datenmodell
- **Vorgehensweise**
 - Modellierung von Sichten z.B. für verschiedene Fachabteilungen
 - Analyse der vorliegenden Sichten in Bezug auf Konflikte
 - Namenskonflikte (Homonyme, Synonyme)
 - Typkonflikte
 - Bedingungskonflikte
 - Strukturkonflikte
 - Integration der Sichten in Gesamtschema
- **Ergebnis**
 - Konzeptionelles Gesamtschema, z.B. (E)ER-Diagramm

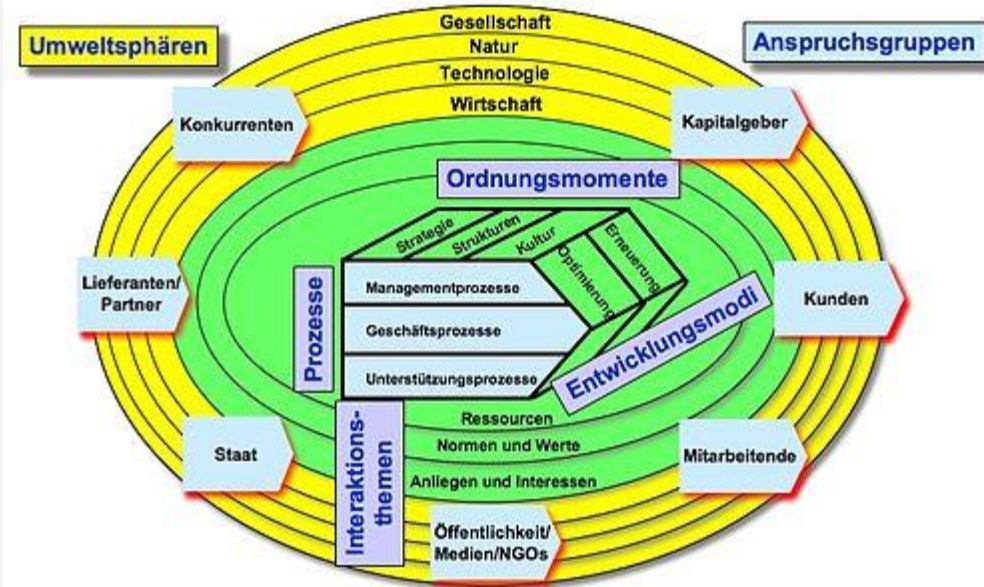


- Identifikation möglicher Konflikte in vorliegenden Sichten
- Integration der Sichten in ein Gesamtschema



In Wirklichkeit viel komplexer:

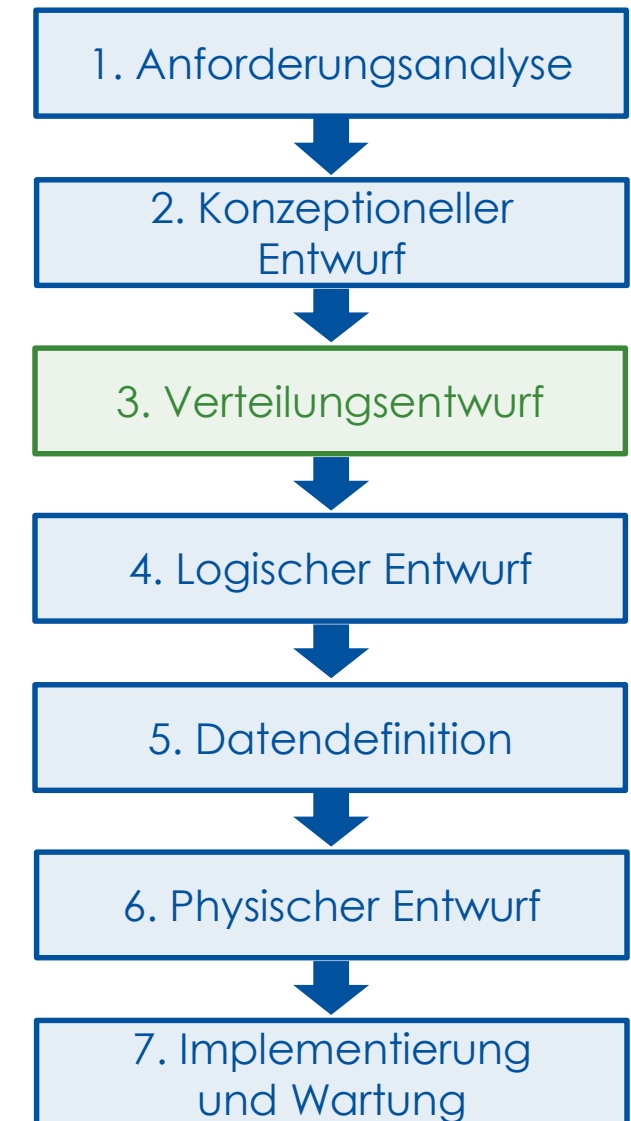
Die sechs Grundkategorien des neuen St. Galler Management-Modells



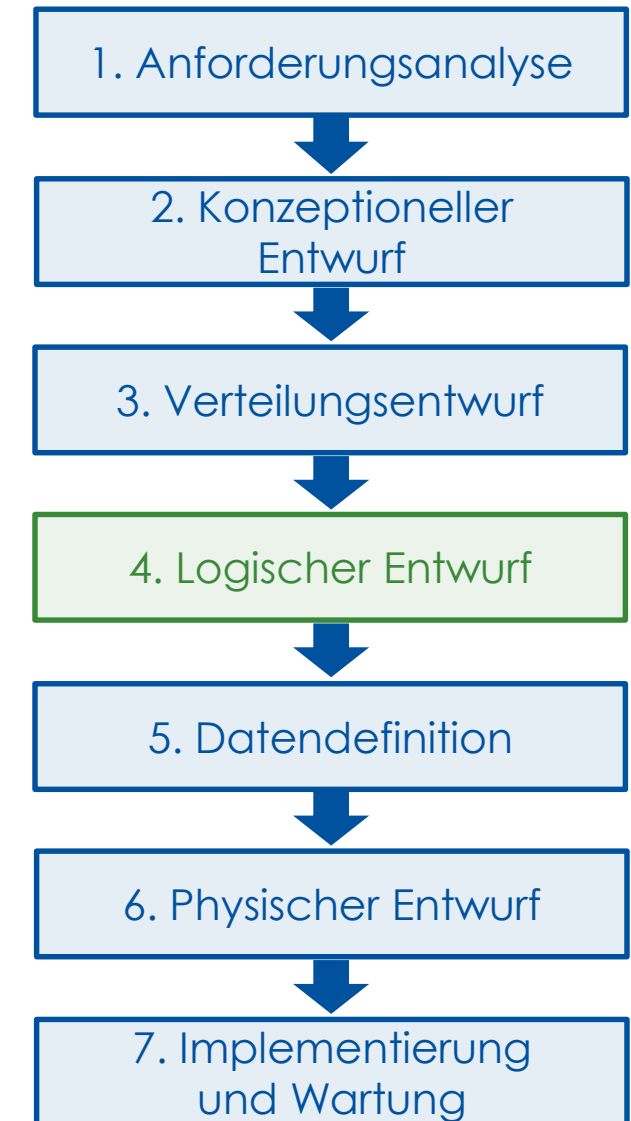
Quelle: Rüegg-Sturm, J. (2003): Das neue St. Galler Management-Modell. Grundkategorien einer integrierten Managementlehre. Der HSG-Ansatz. 2. Auflage, Bern/Stuttgart/Wien: Haupt, S. 22

- **Namenskonflikte**
 - Homonyme: Schloss, Bank
 - 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 oder widersprüchliche Anforderungen
- **Strukturkonflikte**
 - Gleicher Sachverhalt wird durch unterschiedliche Konstrukte ausgedrückt.

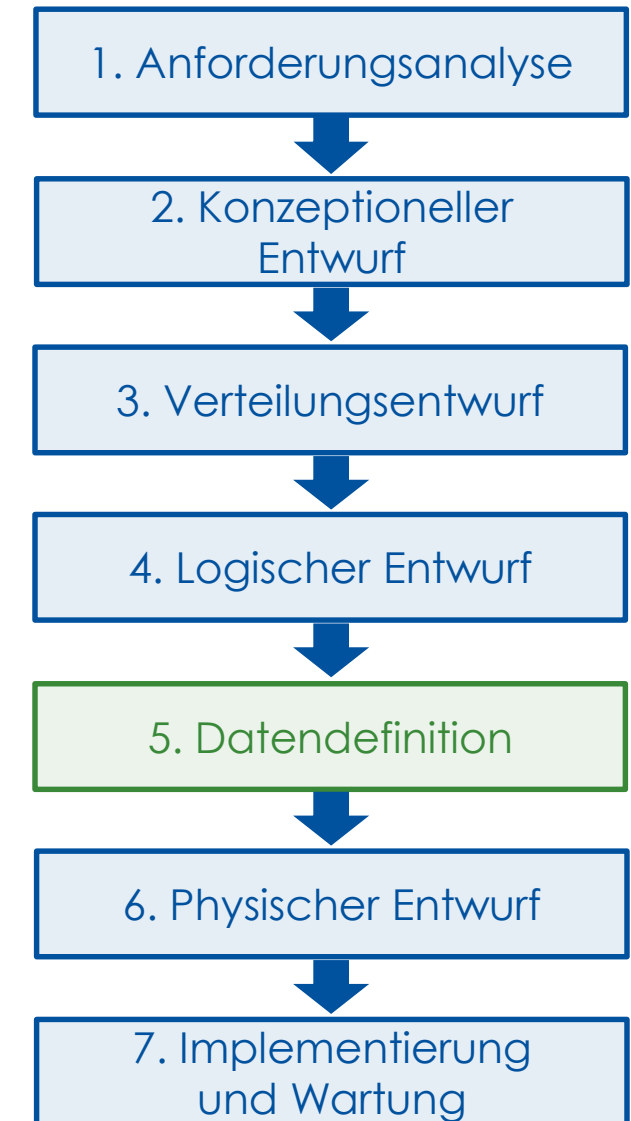
- Auch: Partitionierung
- Legt Datenverteilung auf mehrere Rechner fest
- **Vorgehensweise**
 - Horizontale Partitionierung
 - Zeilen werden auf verschiedene Rechner verteilt
 - Vertikale Partitionierung
 - Spalten werden auf verschiedene Rechner verteilt
- **Ergebnis** Partitionierungsentwurf



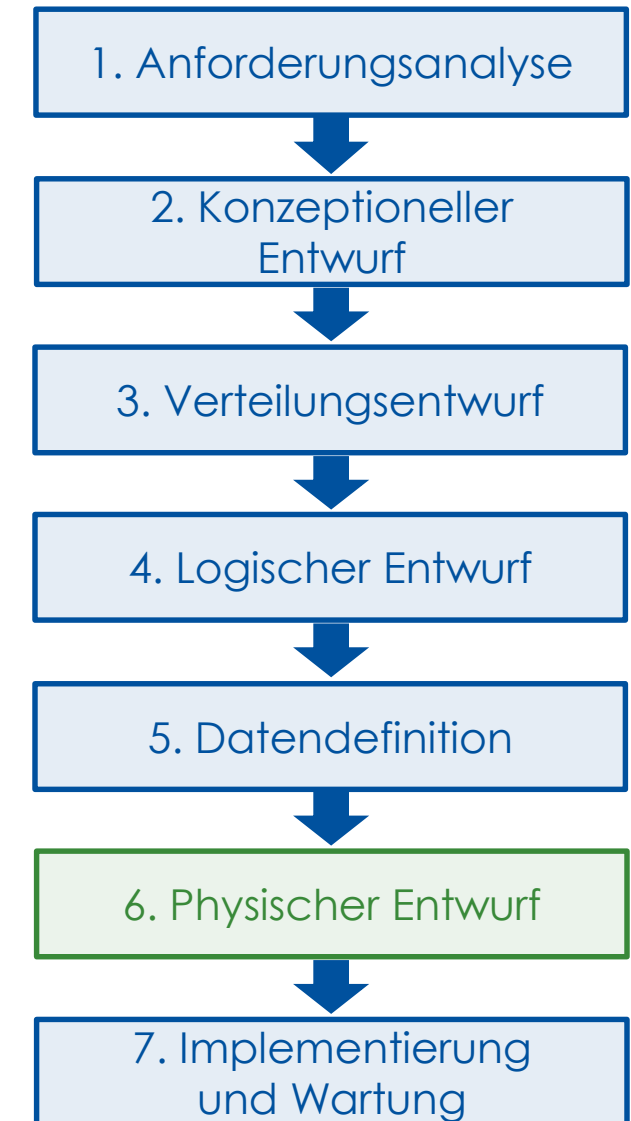
- Sprachmittel: Datenmodell des ausgewählten DBMS
 - Relationale Datenbanken: relationales Modell
 - XML-Datenbank: XML
 - Neuerdings tlw. keine festen Schemata
- **Vorgehensweise**
 - (Automatische) Transformation des konzeptionellen Schemas
 - Z.B. ER in relationales Modell
 - Verbesserung des relationalen Schemas anhand von Gütekriterien
 - Normalisierung, Redundanzvermeidung, ...
- **Ergebnis**
 - Logisches Schema, z.B. Menge von Relationenschemata



- Umsetzung des logischen Schemas in ein konkretes Schema
- Sprachmittel:
 - DDL (Data Definition Language), DML (Data Manipulation Language) eines DBMS
 - Datenbankdeklaration in der DDL des DBMS
 - CREATE TABLE...
 - Realisierung der Integritätssicherung
 - Schlüssel, Fremdschlüssel, Nebenbedingungen, Datentypen
 - Definition von Benutzersichten
 - CREATE VIEW...



- DBMS nimmt tlw. automatisiert physischen Entwurf vor
- Von uns ergänzt um Zugriffsunterstützung zur Effizienzverbesserung
 - z.B. Definition von Indizes
 - `CREATE INDEX ...`



- Beispiel

- Tabelle mit 10 GB Daten
- Festplattentransferrate ca. 50 MB/s
- Operation: Suchen einer Bestellung (Selektion)
- Implementierung: sequentielles Durchsuchen
- Aufwand: $10.240/50 = 205$ sec. (ca. 3,5 Minuten)

- Index

- Datenstruktur für effizienten, Suchschlüssel-basierten Zugriff auf Datensätze (<Schlüsselattributwert, Tupeladresse>)
- Meist als Baumstruktur realisiert



7. Implementierung und Wartung

- Wartung des DBMS
- Datenbank Tuning
- Anpassung an neue Anforderungen
- Anpassung an neue Systemplattformen
- Portierung auf neue DBMS
- Kostenaufwändigste Phase
- Transaktionsentwurf wird oft als Teil vom Software-Engineering betrachtet



Client-Server-Architekturen

Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

Relational

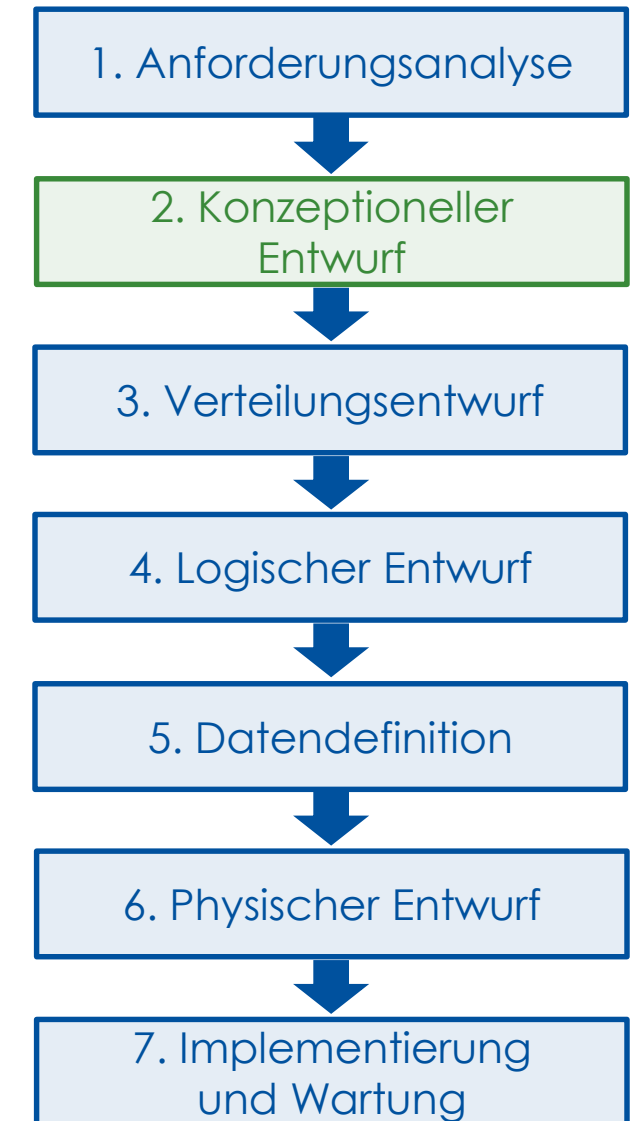
ER Daten SQL DML
NoSQL Recovery Big Data
OODBMS DDL EER-Modell
Transaktion Datenbank Normalisierung
Architektur

Modell

Relationale Algebra

Modellierung

1. Anforderungsanalyse
 - Informelle 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
 - Installation, Backups, ...



- Was ist ein Modell?
 - **Abbild** eines vorhandenen Originals, (z.B. der IT-Struktur einer Firma)
 - **Vorbild** für ein zu entwickelndes Original (z.B. eine neu zu instanzierende IT-Struktur in einer Firma)
- Verschiedene Klassen von Modellen/Originalen
 - **Konkrete / abstrakte Modelle**
 - Prototyp eines neuen Laptop / Erlösmodell
 - **Konkrete / abstrakte Originale**
 - Laptop / Personalentwicklung
- In der Informatik i.d.R.
 - abstrakte Abbilder oder Vorbilder
 - von/zu abstrakten oder konkreten Originalen

- Möglichkeiten aufgrund einer Modellierung
 - Durchführung von **Operationen, die am Original nicht oder nur mit einem größeren Aufwand durchgeführt** werden können
 - z.B. Erproben veränderter Produktdesigns statt am vollständigen Produkt besser an einem Modell des Produktes

- **Untersuchen und Verstehen komplexer Zusammenhänge**
 - z.B. anschauliche Simulation der IT-Struktur einer Firma
- **Kommunikationsgrundlage** zwischen Auftraggeber und Entwickler
 - z.B. bei der Softwareentwicklung und beim Datenbankentwurf
- **Fixieren von Anforderungen** für die Herstellung eines Originals
 - z.B. Anforderungsanalyse innerhalb der Softwareentwicklung → Produktspezifikation

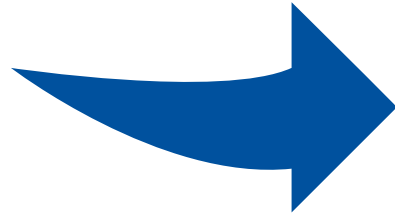
Diese Punkte sind hier von höherer Bedeutung!

- **Struktur**
 - Elemente eines Originals – z.B. Orga-Einheiten einer Firma
- **Eigenschaften**
 - Attribute der Elemente – z.B. Budget einer Orga-Einheit
- **Beziehungen**
 - Beziehungen zwischen Elementen – z.B. Mitarbeiter / Abt.-Leiter
- **Verhalten**
 - Dynamik der Elemente – z.B. Attributänderung beim Namen

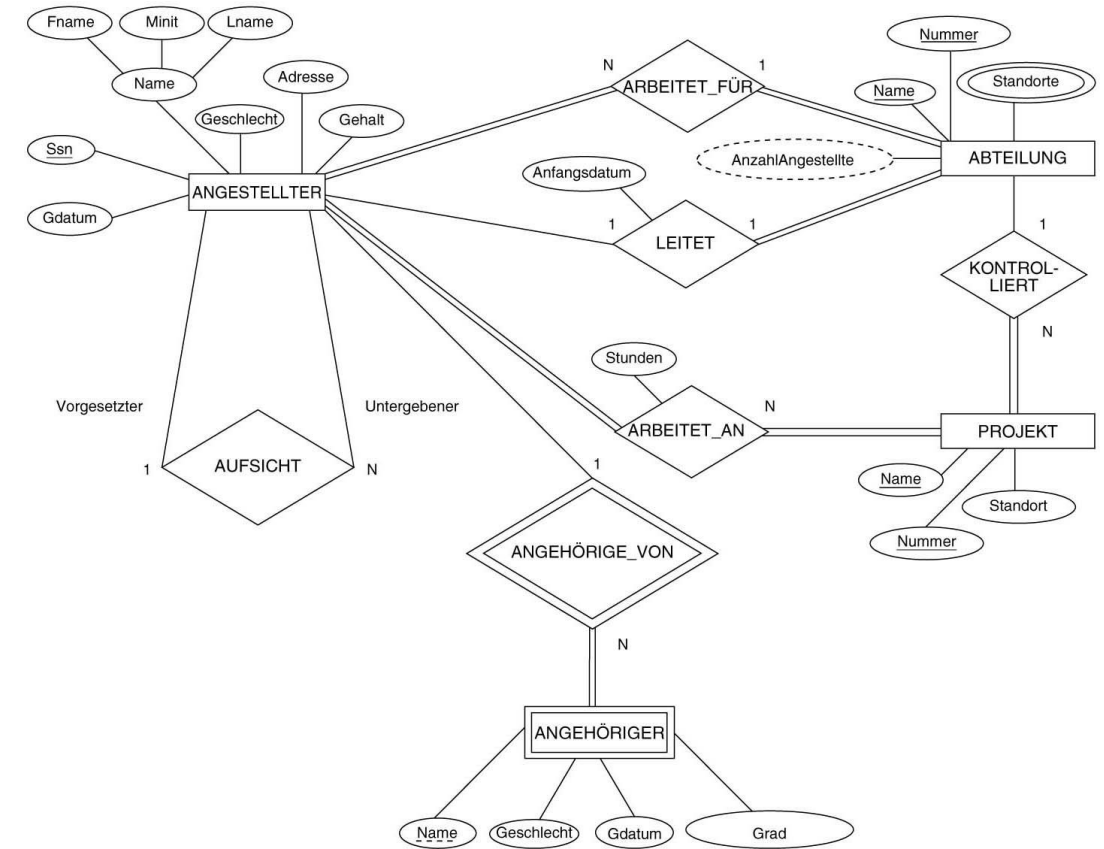
- **Entität** (entity)
 - Ein Ding / Objekt der realen oder der Vorstellungswelt
 - Nicht direkt darstellbar, sondern nur über Eigenschaften beobachtbar
- **Entitätstyp** (entity set)
 - Eine Klasse für gleichartige Objekte
- **Beziehung** (relationship)
 - Beschreibt Beziehungen zwischen Entitäten
 - Meist binär
- **Beziehungstyp** (relationship set)
 - Eine Klasse für gleichartige Beziehungen

- **Attribut**

- repräsentiert eine Eigenschaft von Entitäten oder von Beziehungen
- Zunächst nur primitive Datenwerte (String, Integer, ...) und Operationen darauf
- Später auch komplexe Attribute



ER-Schemadiagramm für die Datenbank FIRMA.



- Nomen
 - Hinweis auf Entitäten oder Attribute
- Adjektive
 - Hinweis auf Attribute
- Verben
 - Hinweis auf Beziehungen
 - häufig zwischen Entitäten
 - aber auch zwischen Attributen
- Typische Problemfälle
 - Synonyme: verschiedene Worte für den selben Begriff (Buchtitel, Exemplar)
 - Homonyme: gleiches Wort mit verschiedenen Bedeutungen (Bank, unsaubere Definitionen z. B. Entität)

Hinweis

Qualität des Ausgangstextes zur Analyse ist ein eigenes Thema und nicht Teil des Moduls

Client-Server-Architekturen

Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

Relational

ER Daten SQL DML
NoSQL Recovery Big Data
OODBMS DDL EER-Modell
Transaktion Datenbank Normalisierung
Architektur

Modell

Relationale Algebra

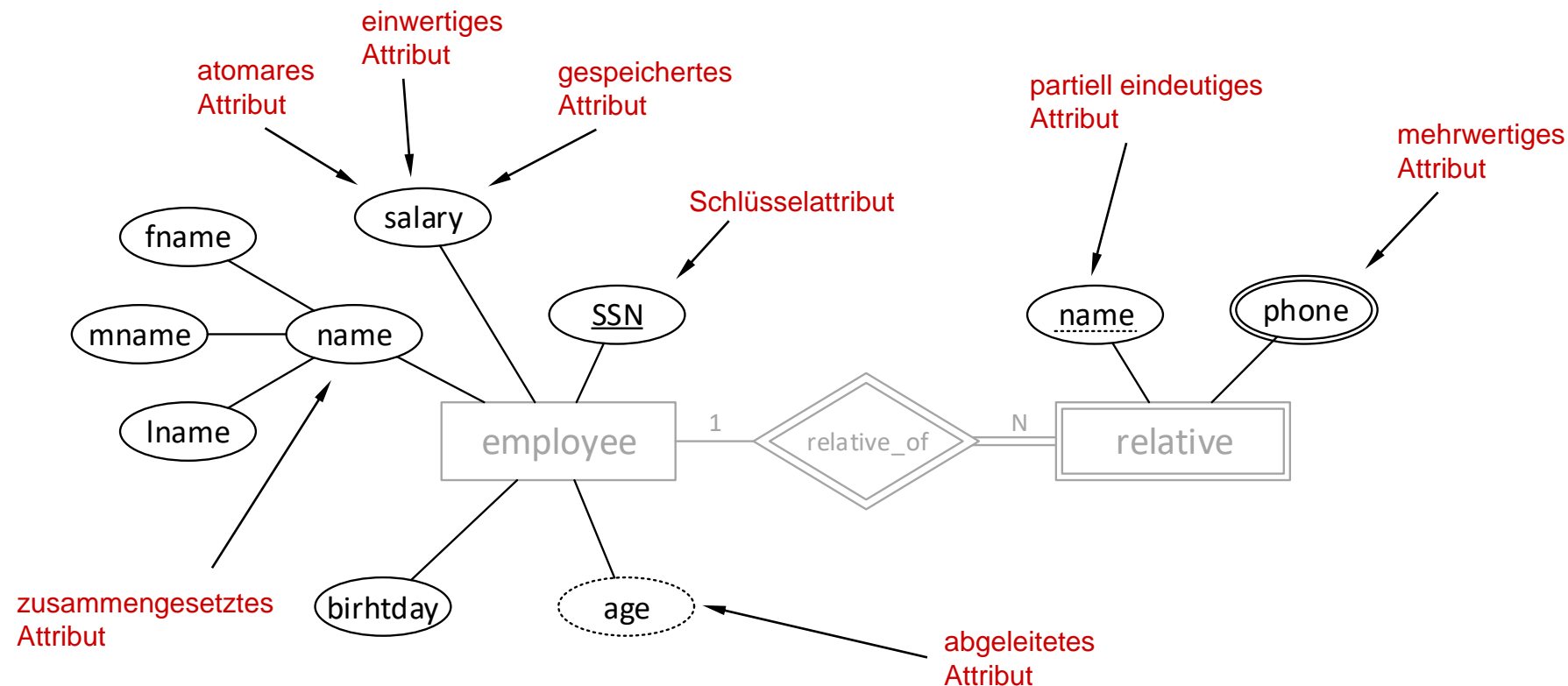
Komponenten von Entity-Relationship- Diagrammen

- Beschreiben **wesentliche Konzepte („abgrenzbare Dinge“)** **realer oder gedachter Miniwelten**
 - Besitzen dort i.d.R. eine eigenständige Existenz



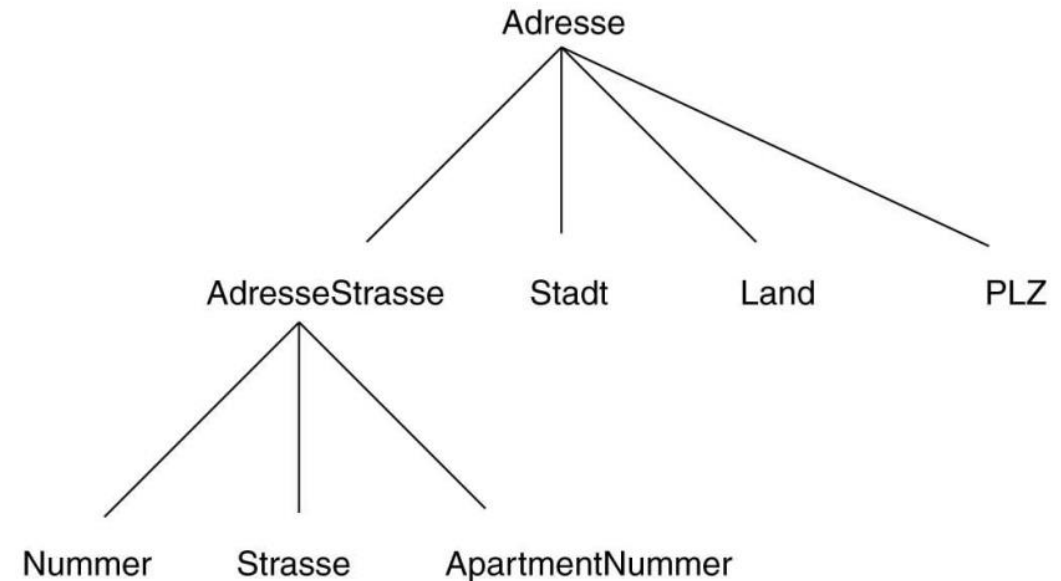
- (Starke) Entitäten
 - Besitzen ausgezeichnete Attribute zur eindeutigen Identifikation
- Schwache Entitäten
 - Können nicht über eigene Attribute eindeutig identifiziert werden
 - Starke identifizierende Entität zur Identifikation stets notwendig
 - Steht mit der schwachen Entität direkt oder transitiv in Verbindung

- Beschreiben **Eigenschaften** von Entitäten (und Beziehungen)
- Es gibt unterschiedliche Attributtypen



- Atomare Attribute
 - Sind unteilbar, d.h. lassen sich nicht sinnvoll zerlegen
- Zusammengesetzte Attribute
 - Gebildet aus mehreren Attributen
 - jeweils unabhängige Bedeutung
 - Können zusammengesetzt oder atomar sein

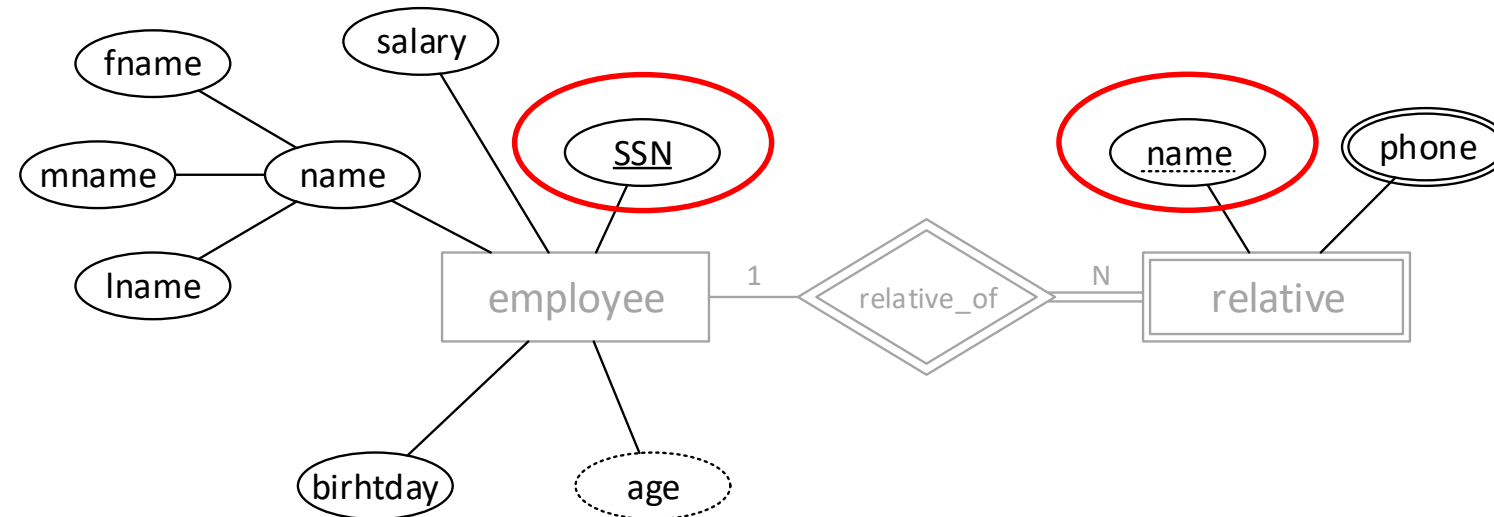
Eine Hierarchie aus zusammengesetzten Attributen; die Komponente AdresseStrasse einer Adresse besteht aus den weiteren Komponenten Nummer, Strasse und ApartmentNummer.



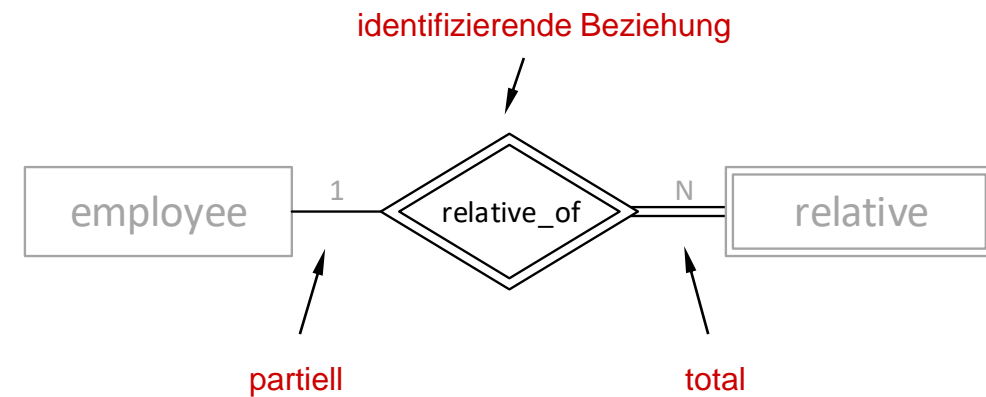
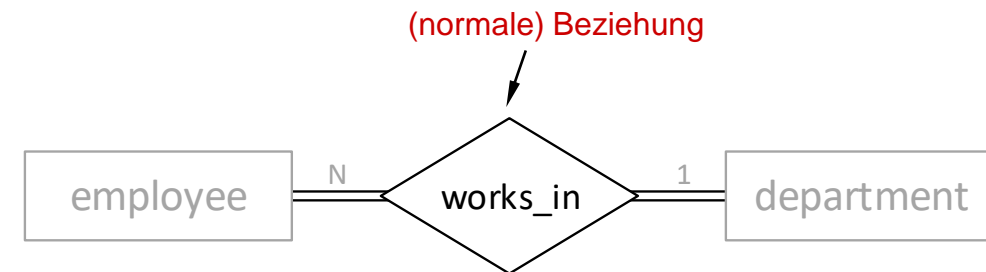
- Einwertige Attribute
 - Haben zu einem Zeitpunkt genau einen Wert
 - Z.B. das Alter einer Person
- Mehrwertige Attribute
 - Können zu einem Zeitpunkt mehrere Werte besitzen
 - Z.B. akademische Titel einer Person
- Abgeleitete Attribute
 - Werden nicht gespeichert sondern aus anderen Attributen berechnet
 - Beispiel: „Alter“ wird aus „Geburtsdatum“ berechnet
- Komplexe Attribute
 - Zusammengesetzte und mehrwertige Attribute können beliebig verschachtelt werden

```
{AdresseTelefon({Telefon(Vorwahl, Telefonnummer)} ,  
Adresse(AdresseStrasse(Nummer, Strasse, ApartmentNummer), Stadt, Land, PLZ))}
```

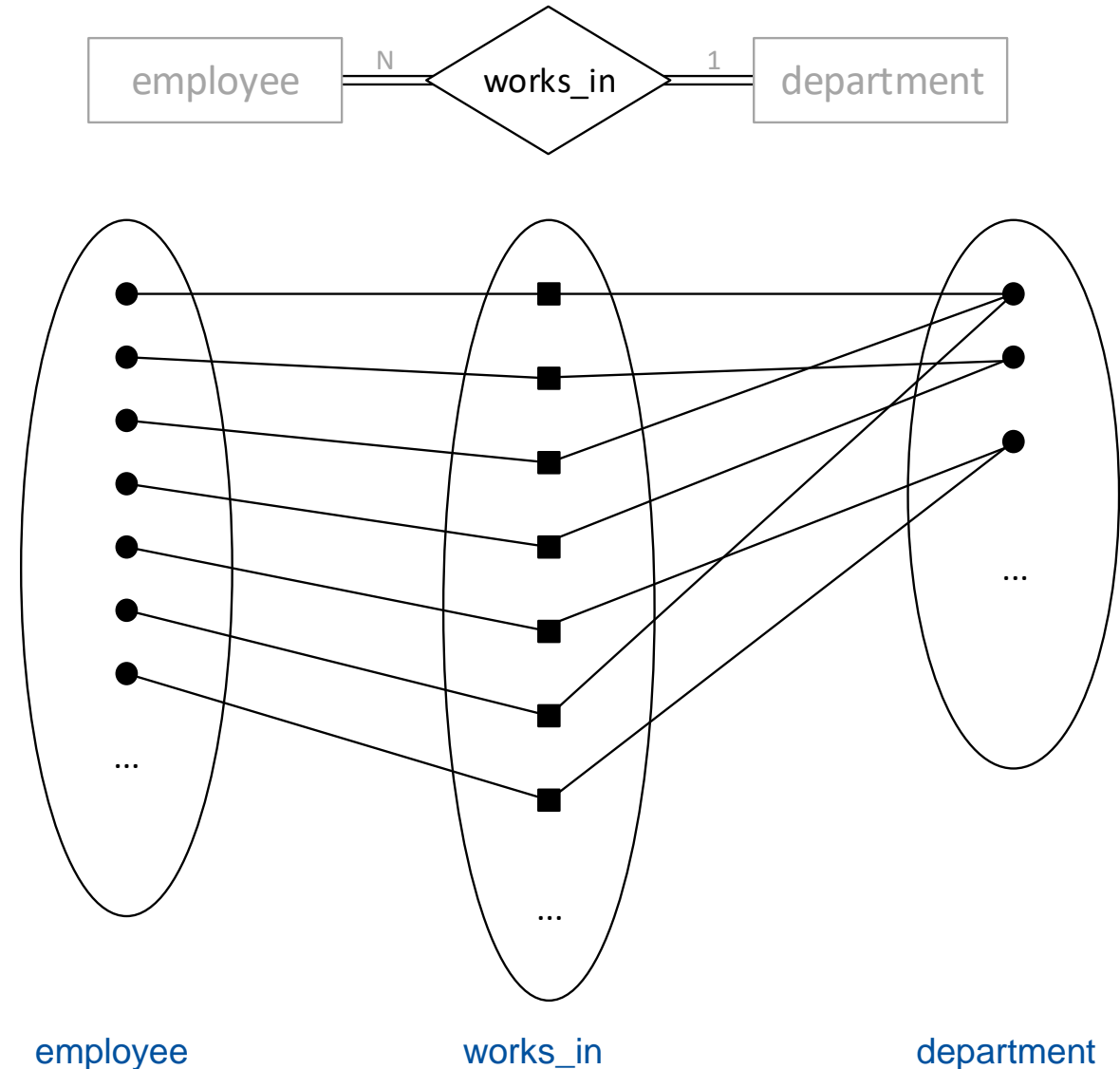

- Schlüsselattribute
 - Werte ermöglichen eindeutige Identifikation von DB-Instanzen
 - Beispiele:
 - Matrikelnummer für Studierende
 - Sozialversicherungsnummer für Arbeitnehmer
- Partiiell eindeutige Attribute
 - Nur für schwache Entitäten
 - Ermöglichen mit Schlüsselattributen der identifizierenden Entität die eindeutige Identifikation schwacher Entitäten



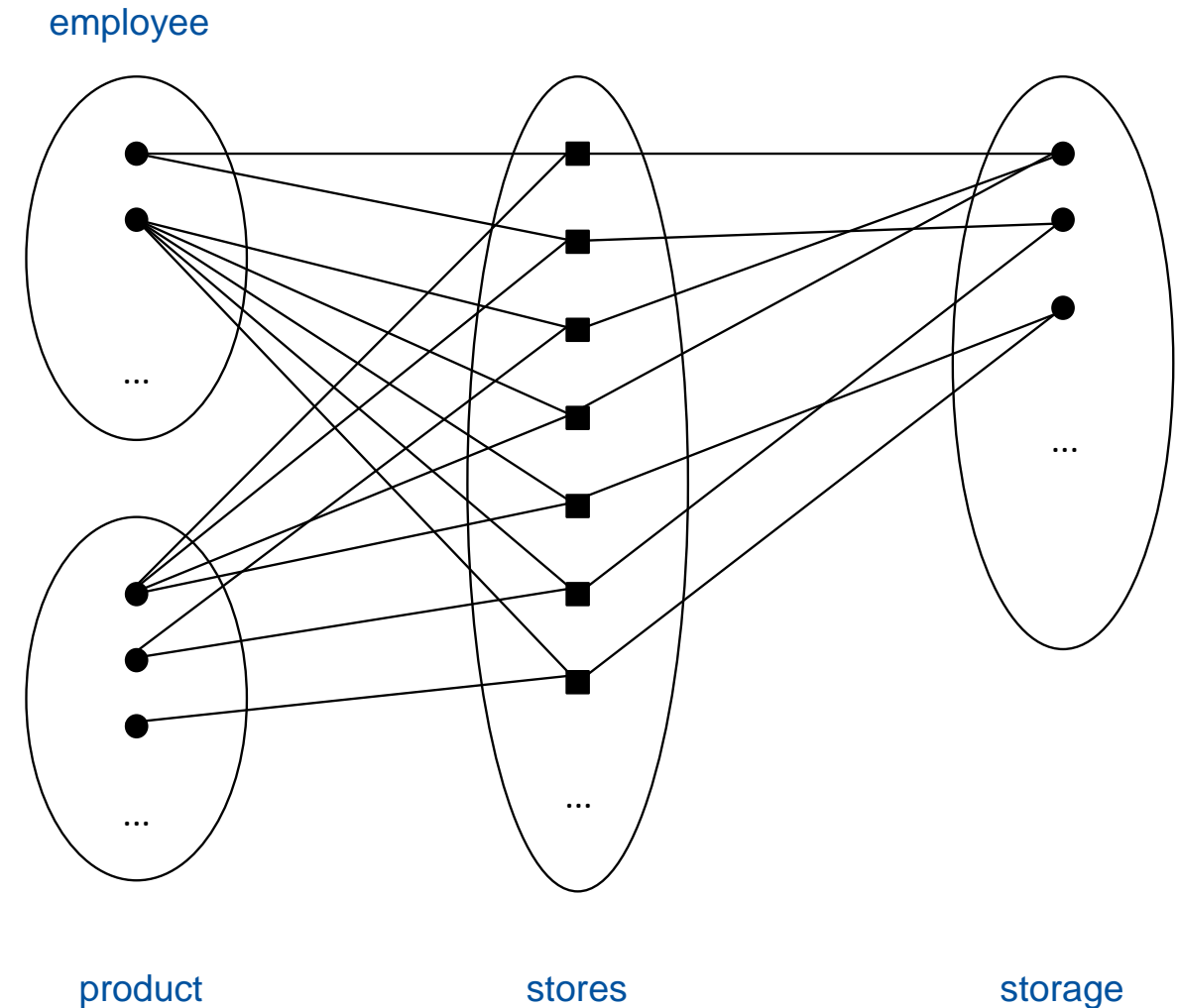
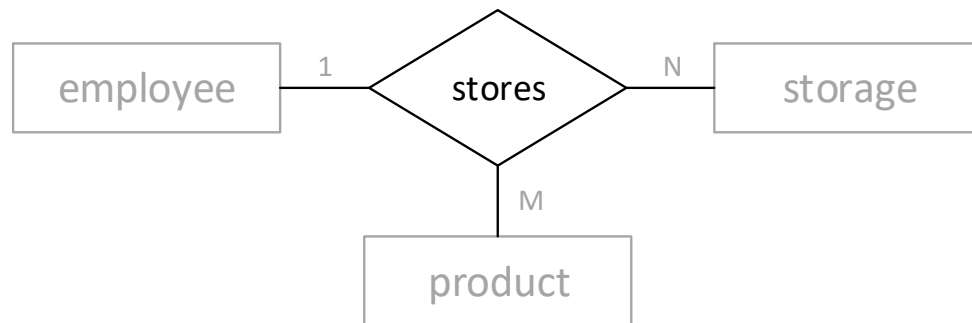
- Spezifizieren einen Zusammenhang zwischen Entitäten
 - Normale Beziehungen
 - Identifizierende Beziehungen
- Totale Teilnahme
 - Alle Instanzen einer Entität nehmen an Beziehung teil
- Partielle Teilnahme
 - Nicht jede Instanz einer Entität muss an Beziehung teilnehmen



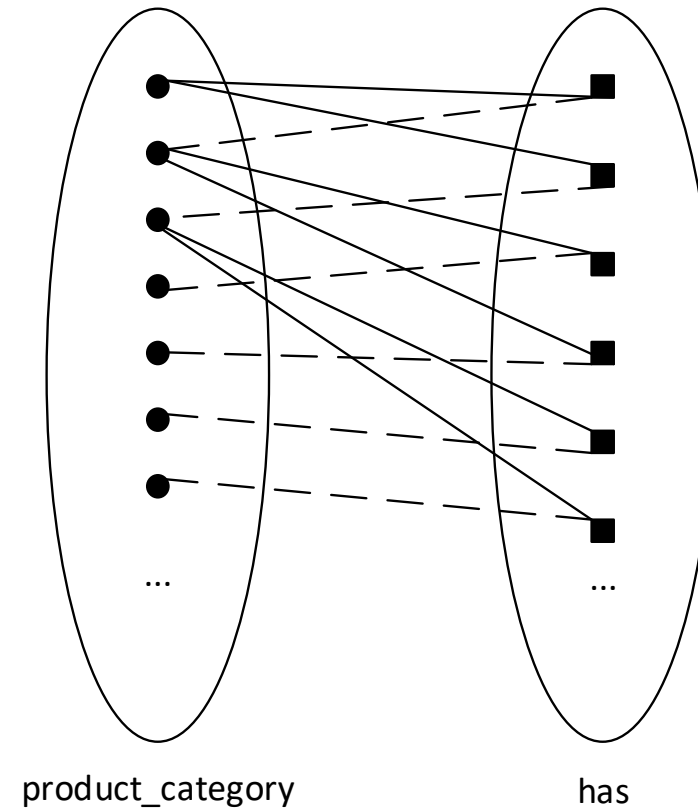
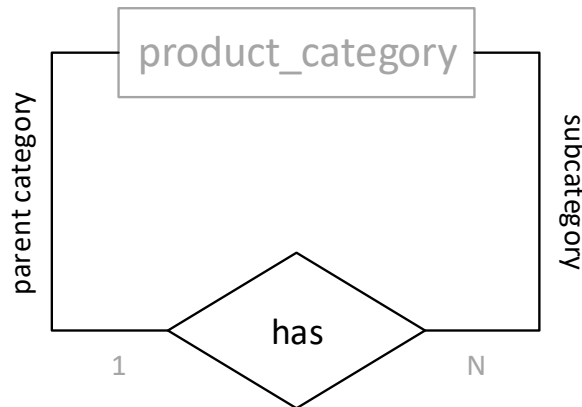
- Beispiel einer Beziehung instanzierter Entitäten ...
 - Einige Instanzen der Beziehung **works_in** zwischen den Entitäten **employee** und **department**



- Grad (bzw. Stelligkeit) einer Beziehung
 - Anzahl der beteiligten Entitäten
 - I.d.R. binäre Beziehungen
 - Seltener ternäre oder höhere Grade
 - Können (mit Semantikverlust!) auf binäre Beziehungen abgebildet werden



- Rekursive Beziehungen
 - DB-Instanzen können zu sich selbst in Beziehung stehen
 - Spezifikation von Rollen

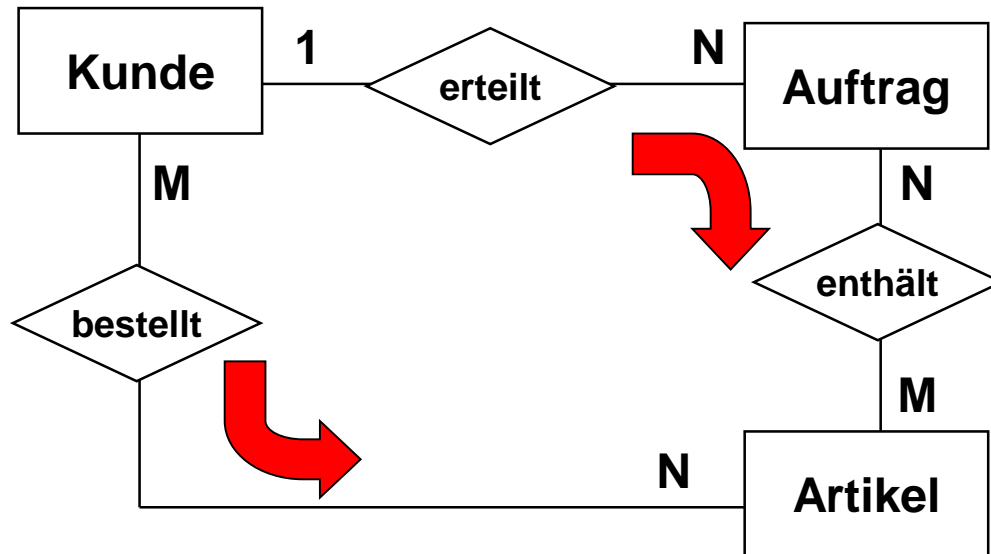


Rollen:

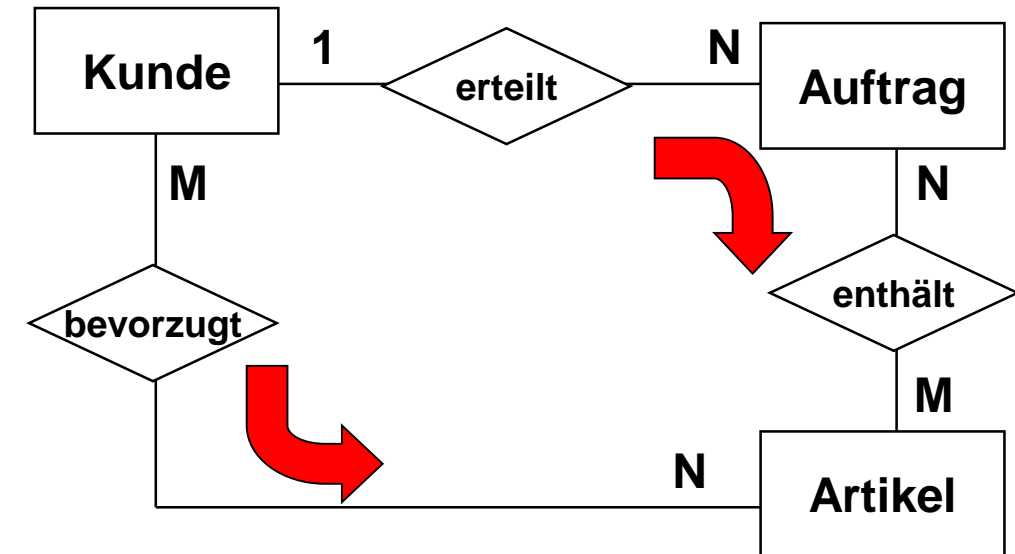
————— parent category

- - - - - subcategory

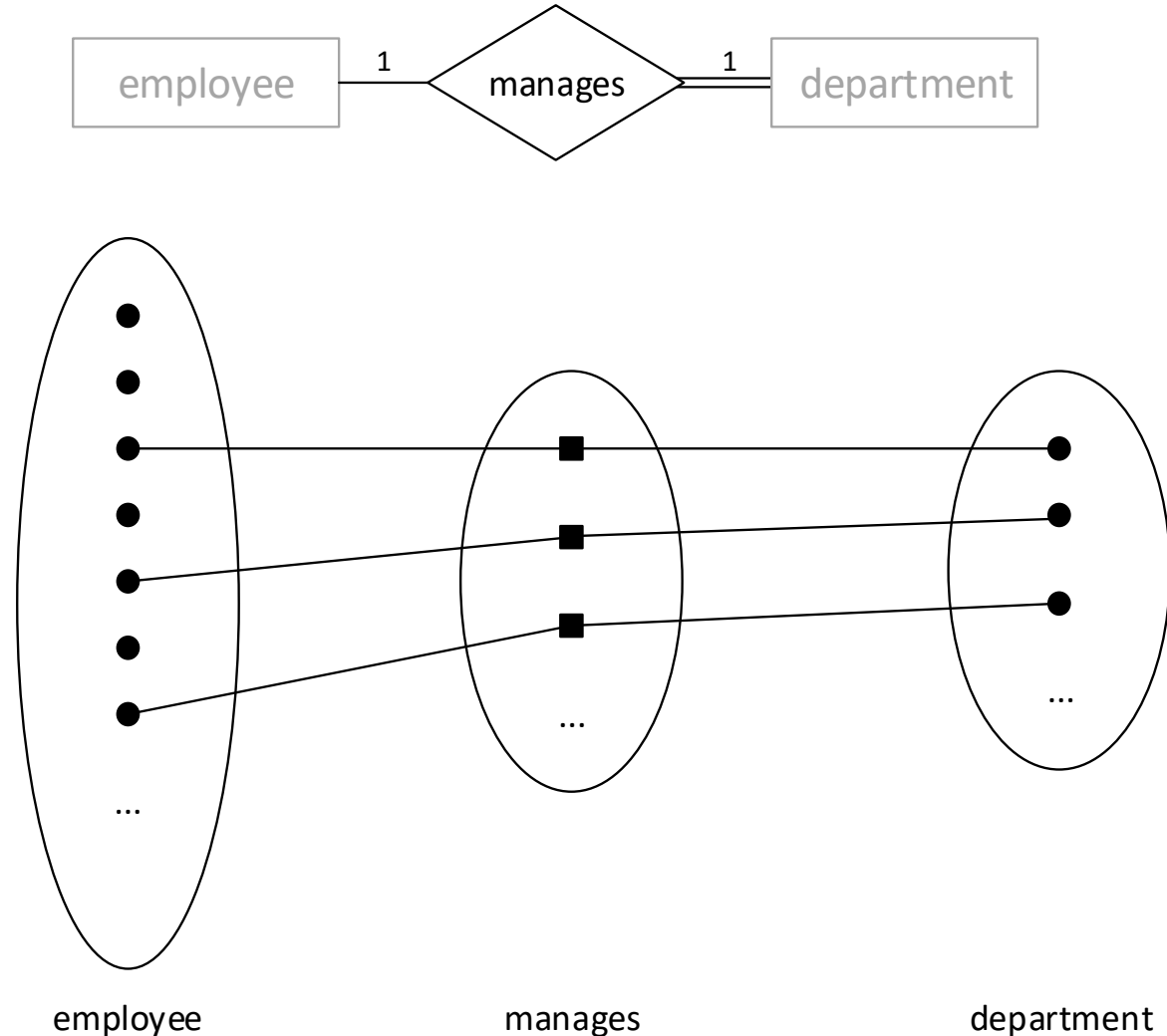
- Keine Modellierung **redundanter Informationen**



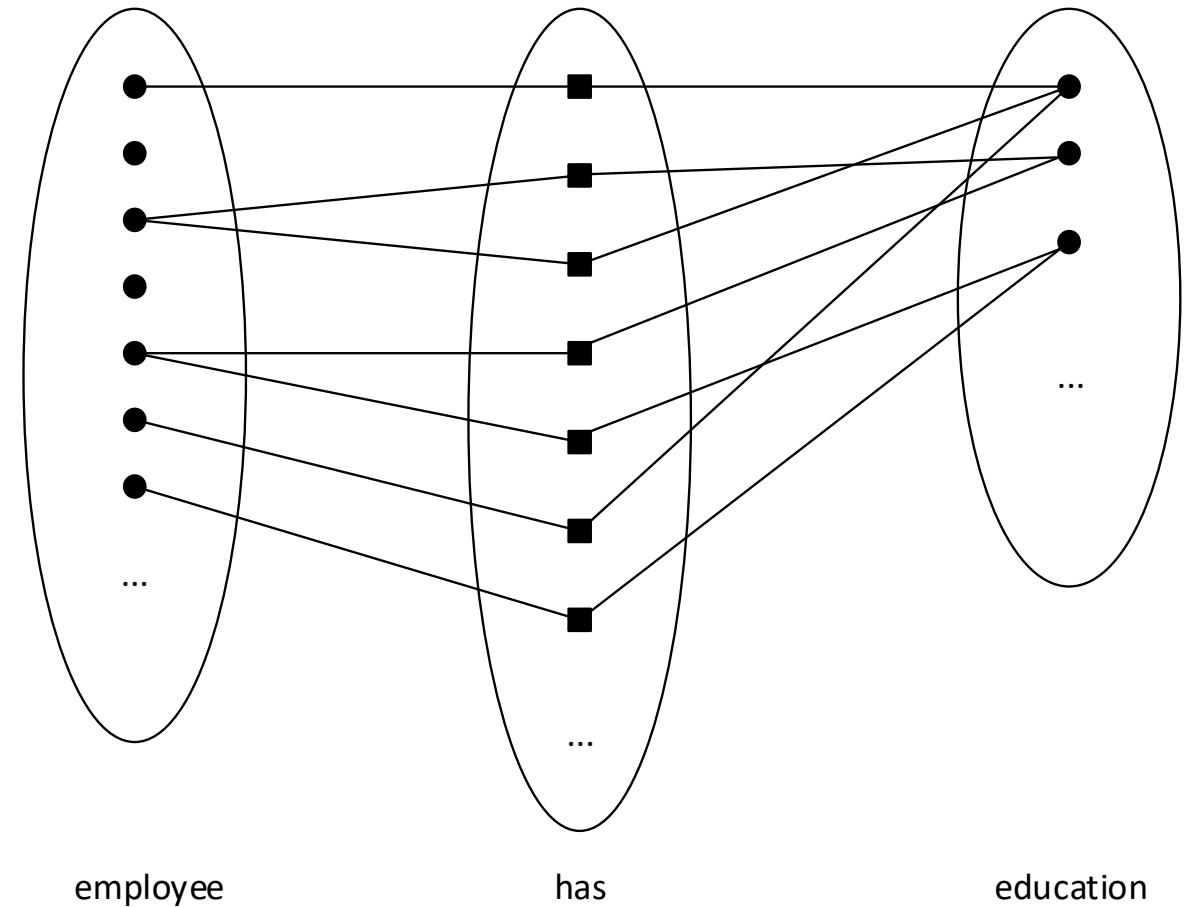
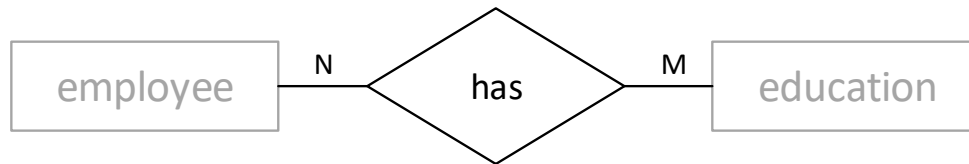
- Zyklen zur Darstellung **unterschiedlicher Zusammenhänge** sinnvoll



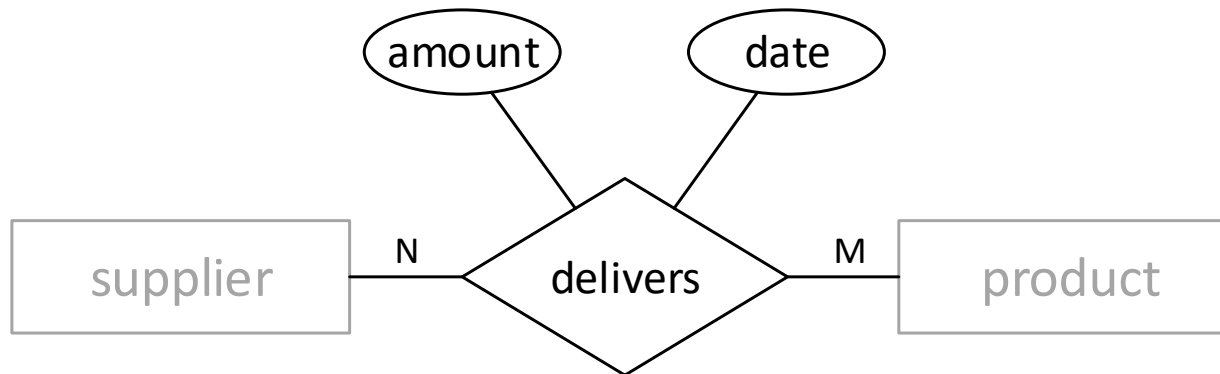
- Genauere Spezifikation binärer Beziehungen
 - 1:1
 - 1:N
 - N:1
 - M:N
- Beispiel für 1:1



- Beispiel für N:M

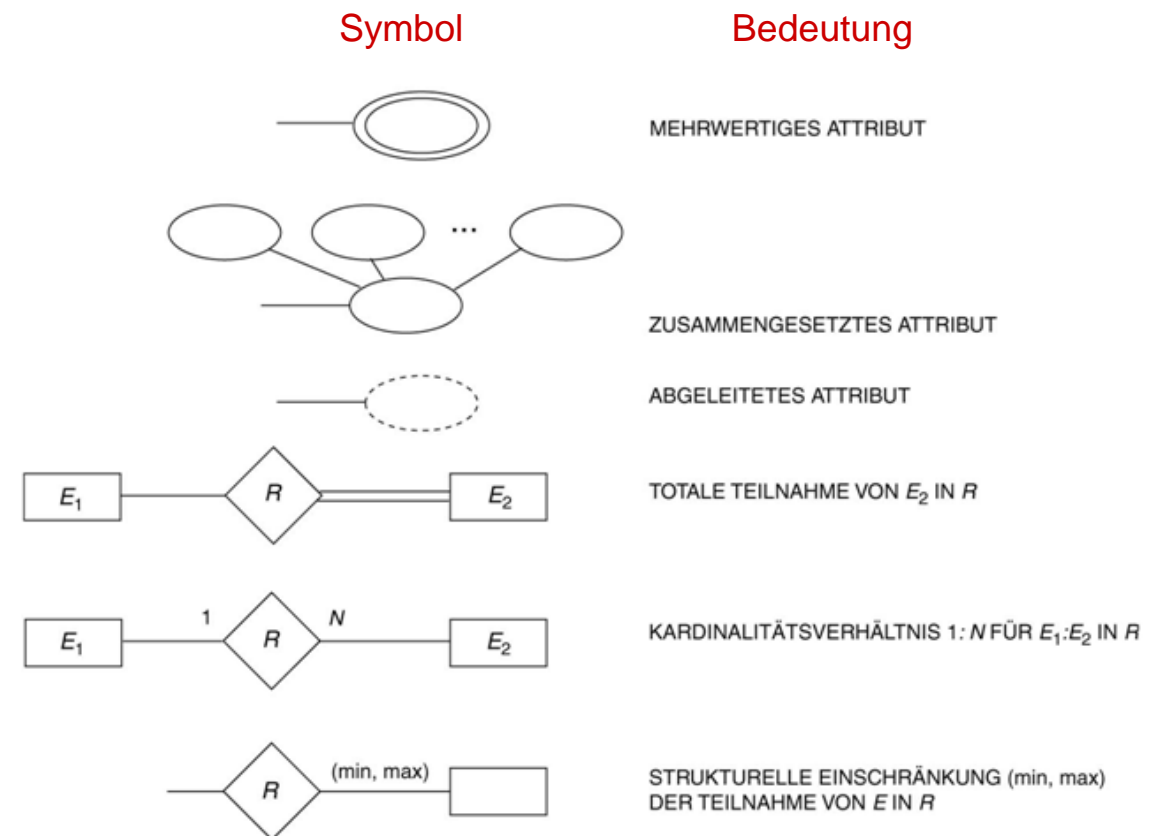
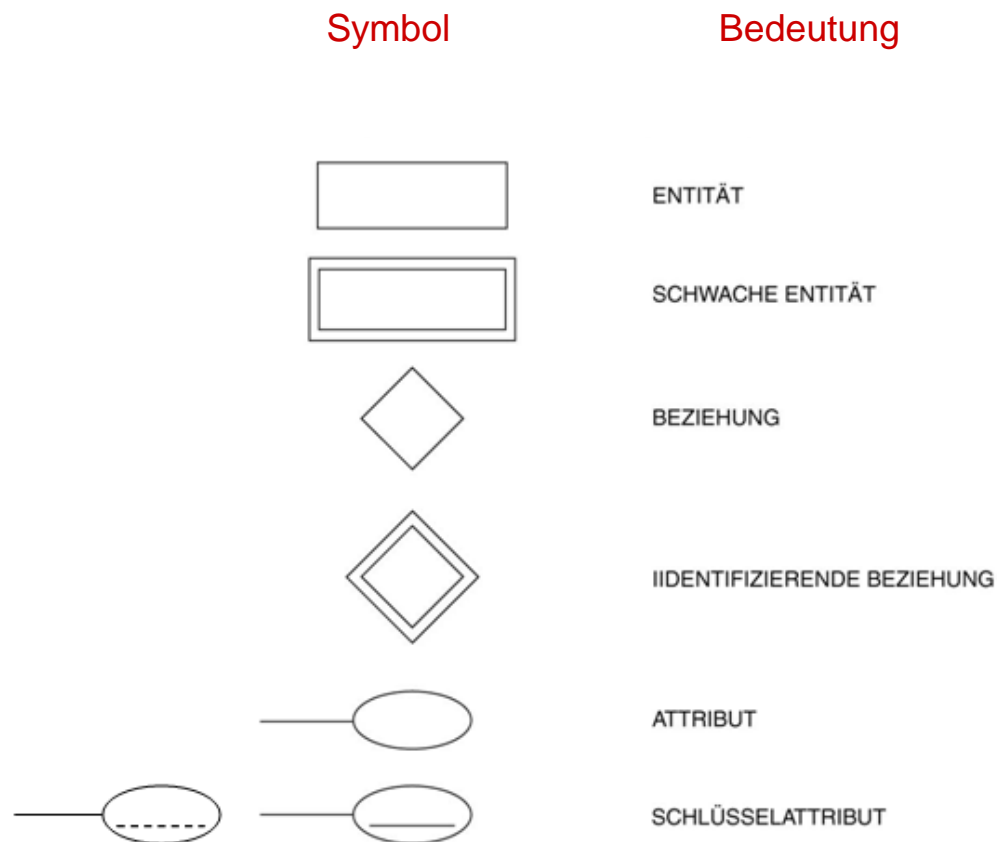


- Auch Beziehungen können Eigenschaften (Attribute) haben



- Beispiel:
 - Details zu einzelnen Lieferungen werden mit der Beziehung gespeichert
 - Gelieferte Menge (amount)
 - Lieferdatum (date)
 - Zuordnung zu supplier oder product nicht sinnvoll
 - Liefermenge und –datum sind weder Lieferanten- noch produktspezifisch

- Übersicht über die Notation von ER-Schemata



Client-Server-Architekturen

Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

Relational

ER Daten SQL DML
NoSQL Recovery Big Data
OODBMS DDL EER-Modell
Transaktion Datenbank Normalisierung
Architektur

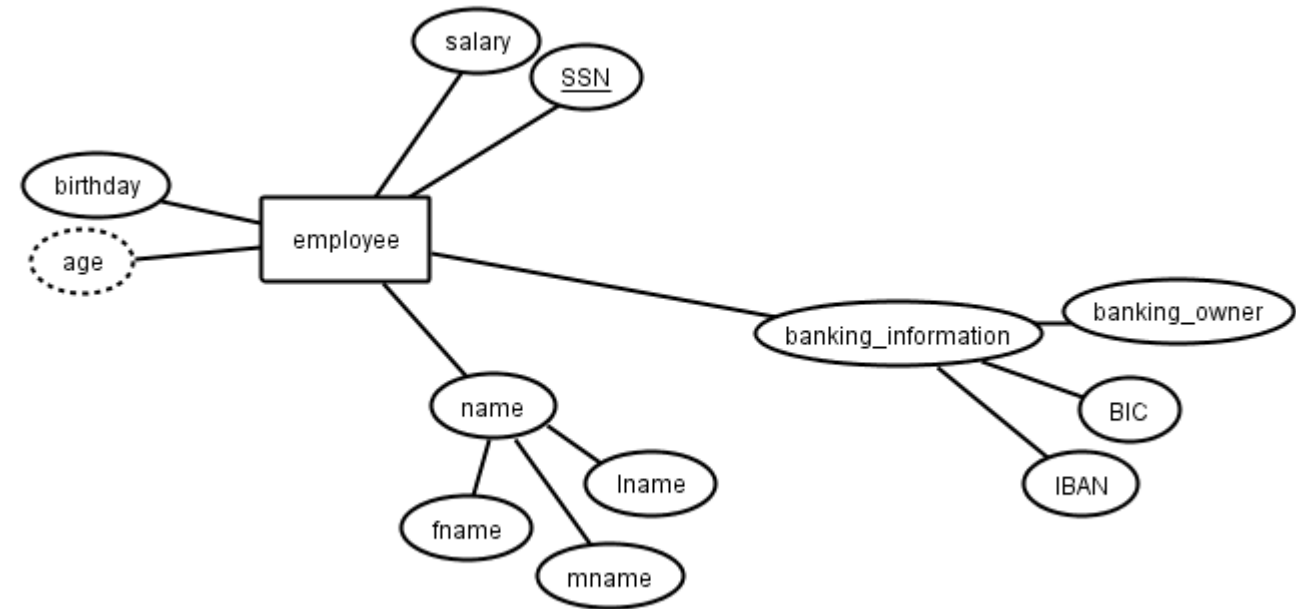
Modell

Relationale Algebra

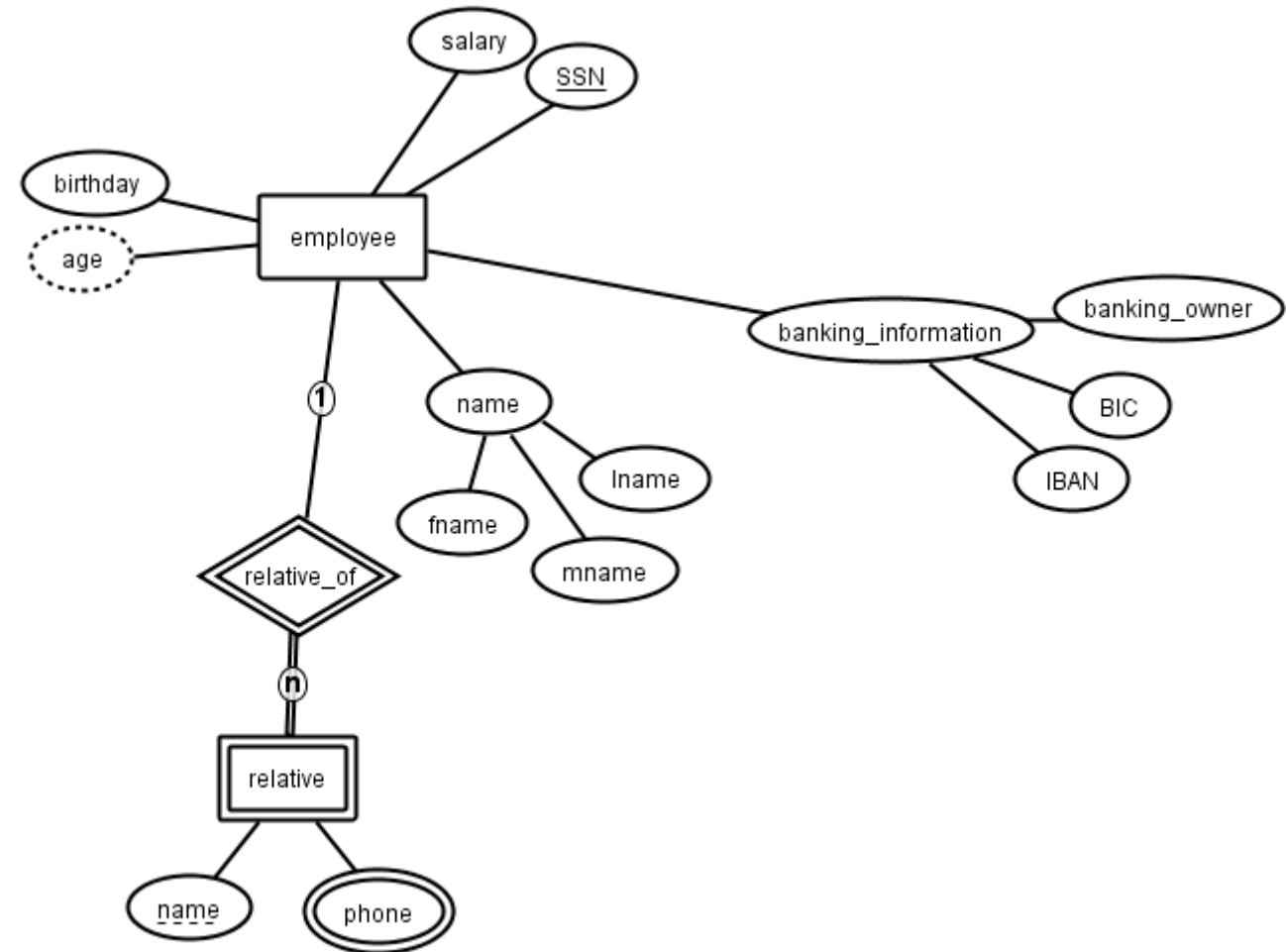
Beispiel: Modellierung eines Online-Shops

- Textuelle Beschreibung nicht immer eindeutig
 - Ggf. auch unvollständig
 - Viel Spielraum für Interpretation
- Abbildung von Konzepten auf ER-Konstrukte nicht eindeutig
 - Ist die Adresse eines Kunden ein Attribut oder eine eigene Entität?
- Konsequenz
 - Nicht die „eine richtige“ Lösung
 - Verschiedene Modellierungsalternativen
 - Abwägung von Vor- und Nachteilen
- Modellierungsbeispiel
 - Basiert auf Lösung einer Aufgabe aus dem DB-Praktikum
 - Keine „Musterlösung“

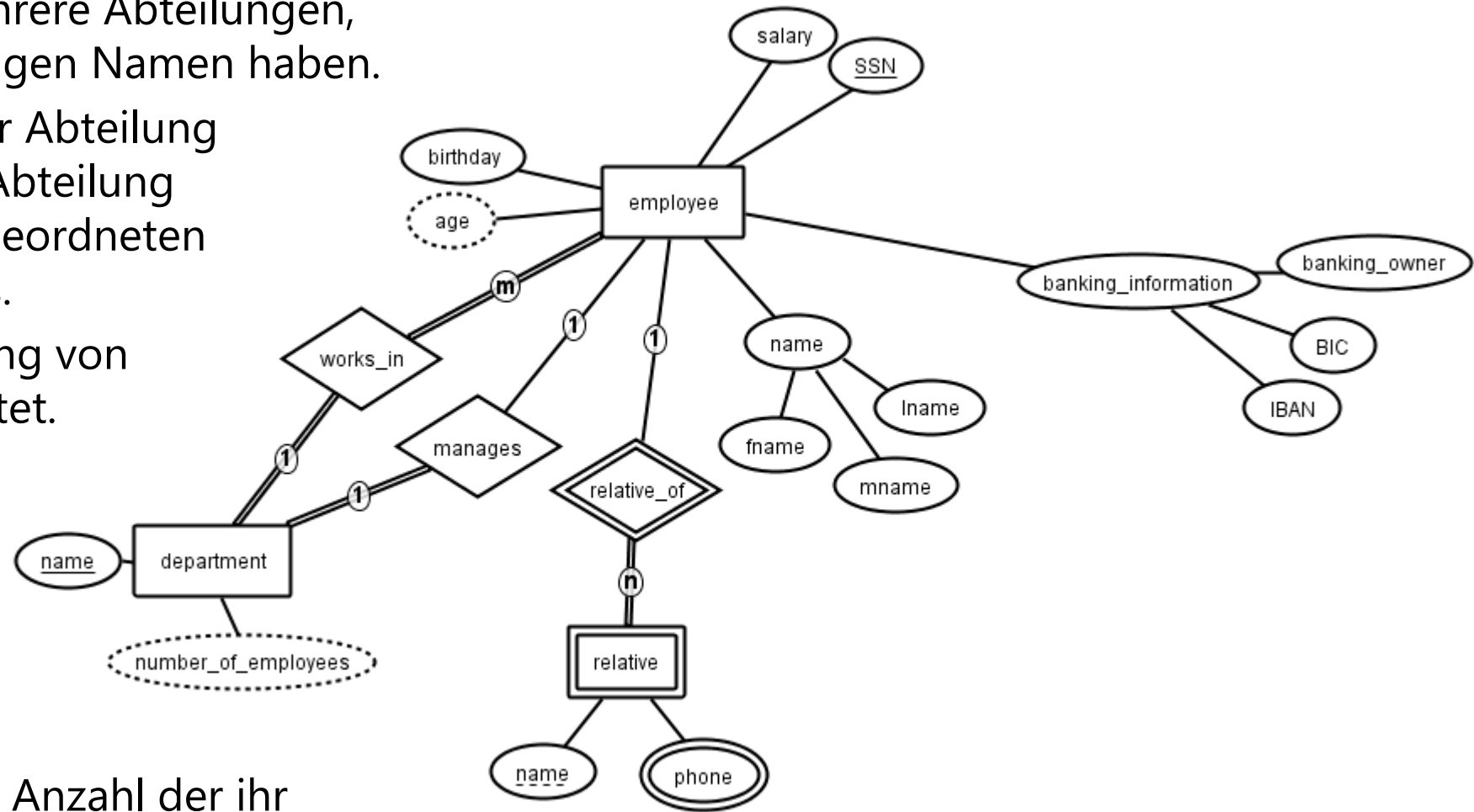
- Der Online-Shop hat Angestellte.
- Jeder Angestellte hat eine eindeutige Sozialversicherungsnummer, ein Alter, ein Geburtsdatum und ein Einkommen.
- Jeder Angestellte hat außerdem einen Namen, der aus Vorname, Nachname und einem optionalen Mittelteil besteht.
- Für die Gehaltszahlungen sind für jeden Mitarbeiter Bankinformationen bestehend aus IBAN, BIC und dem Namen des Kontoinhabers bekannt.



- Jeder Mitarbeiter kann ein oder mehrere Verwandte mit Name und Telefonnummern als Notfallkontakt angeben.

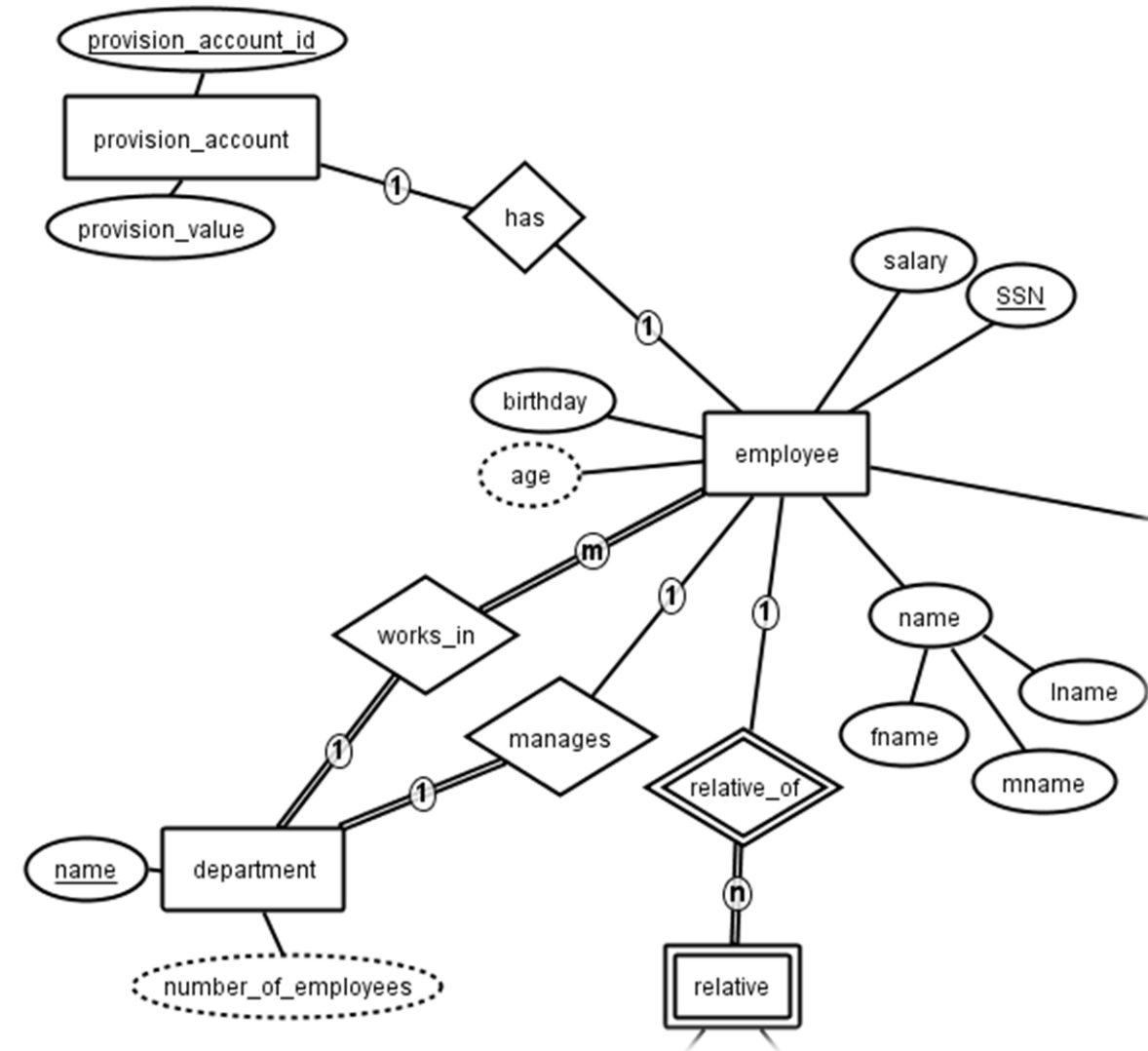


- Der Online-Shop hat mehrere Abteilungen, die jeweils einen eindeutigen Namen haben.
- Jeder Angestellte ist einer Abteilung zugeordnet, wobei eine Abteilung mindestens einen ihr zugeordneten Angestellten haben muss.
- Zudem wird jede Abteilung von einem Angestellten geleitet.

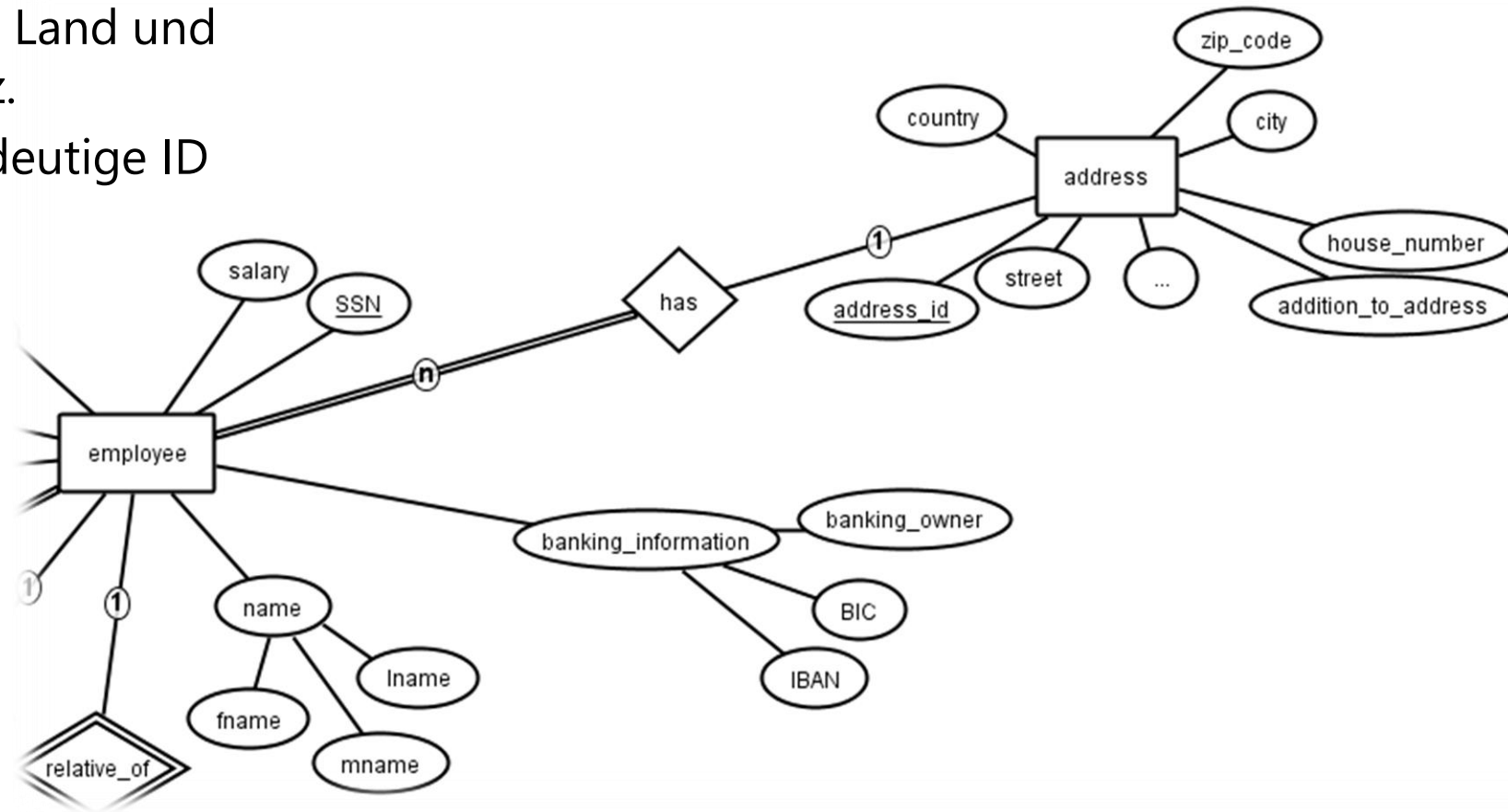


- Für jede Abteilung ist die Anzahl der ihr zugeordneten Mitarbeiter bekannt.

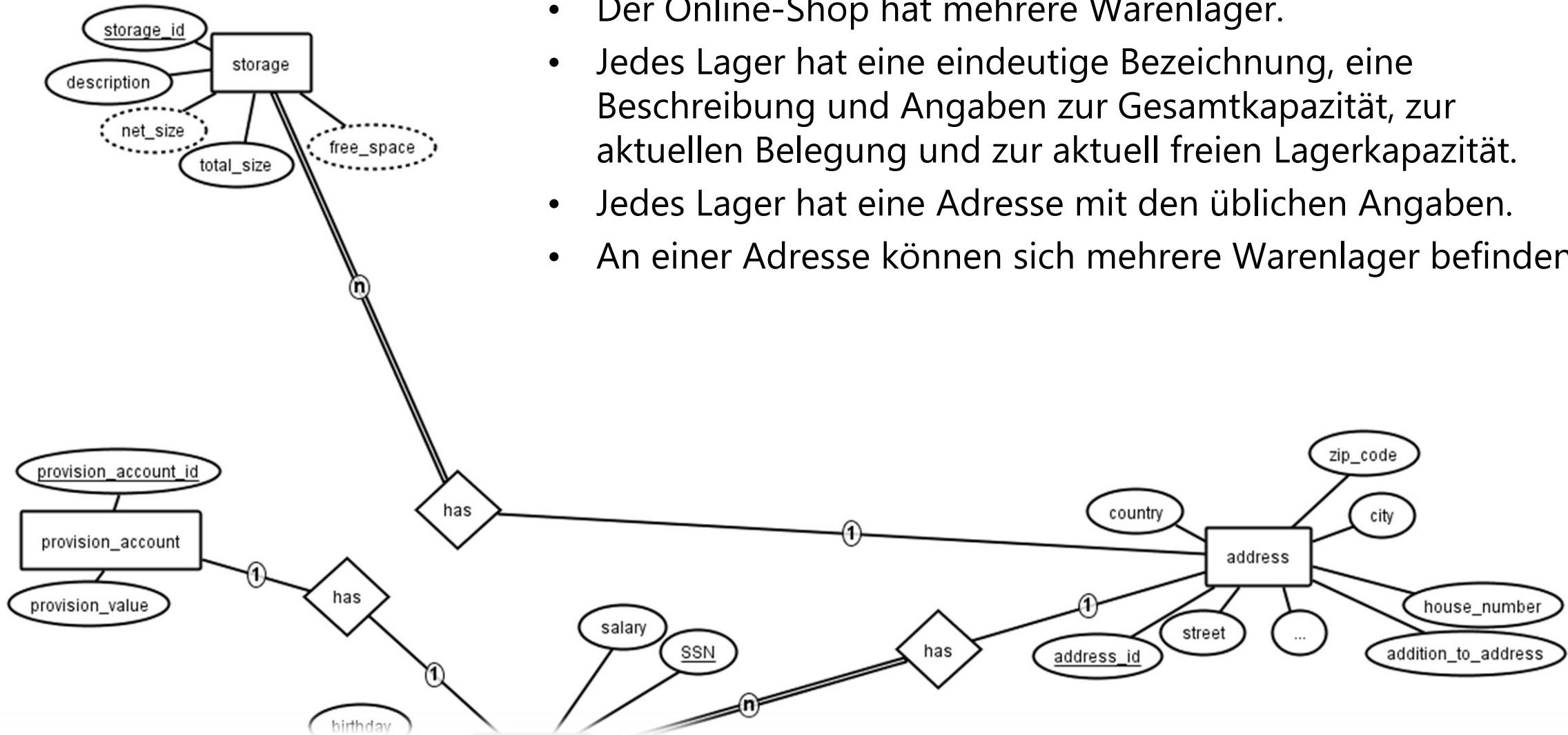
- Ein Angestellter kann ein Provisionskonto haben, das eine eindeutige ID und den Wert der Provision hat.



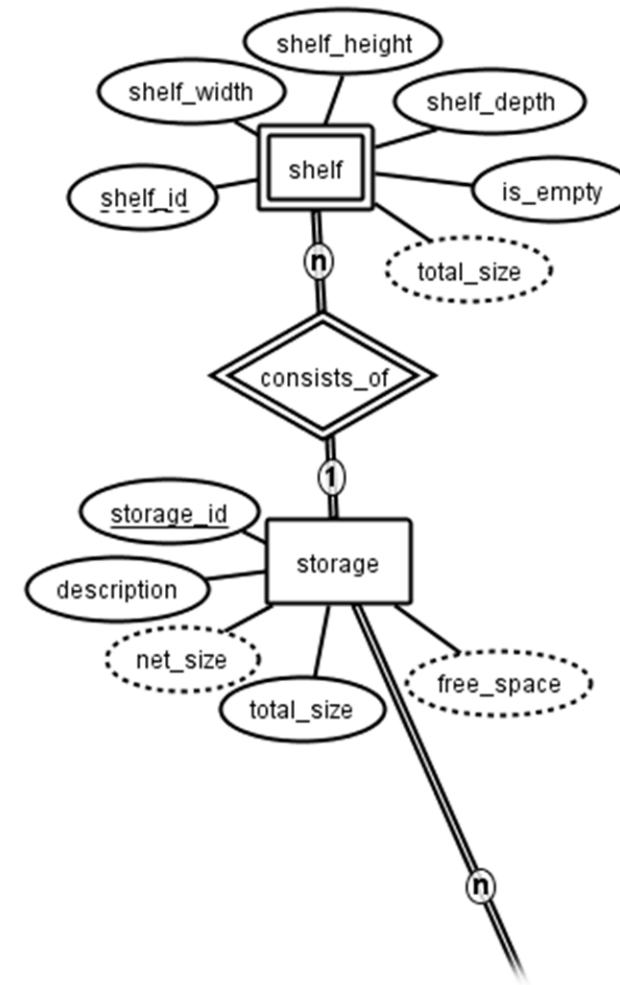
- Jeder Angestellte hat eine Adresse bestehend aus Angaben zu Straße, Hausnummer, Postleitzahl, Ort, Land und einem optionalen Adresszusatz.
- Für jede Adresse wird eine eindeutige ID gespeichert. [...]



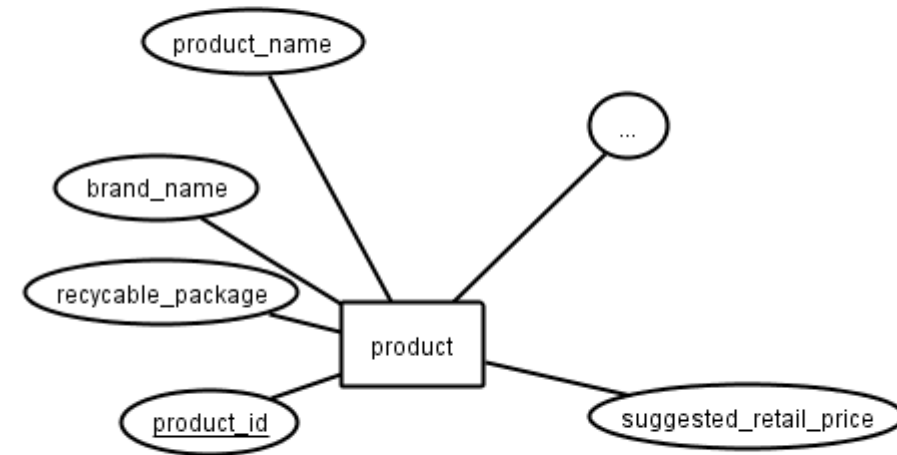
- Der Online-Shop hat mehrere Warenlager.
- Jedes Lager hat eine eindeutige Bezeichnung, eine Beschreibung und Angaben zur Gesamtkapazität, zur aktuellen Belegung und zur aktuell freien Lagerkapazität.
- Jedes Lager hat eine Adresse mit den üblichen Angaben.
- An einer Adresse können sich mehrere Warenlager befinden.



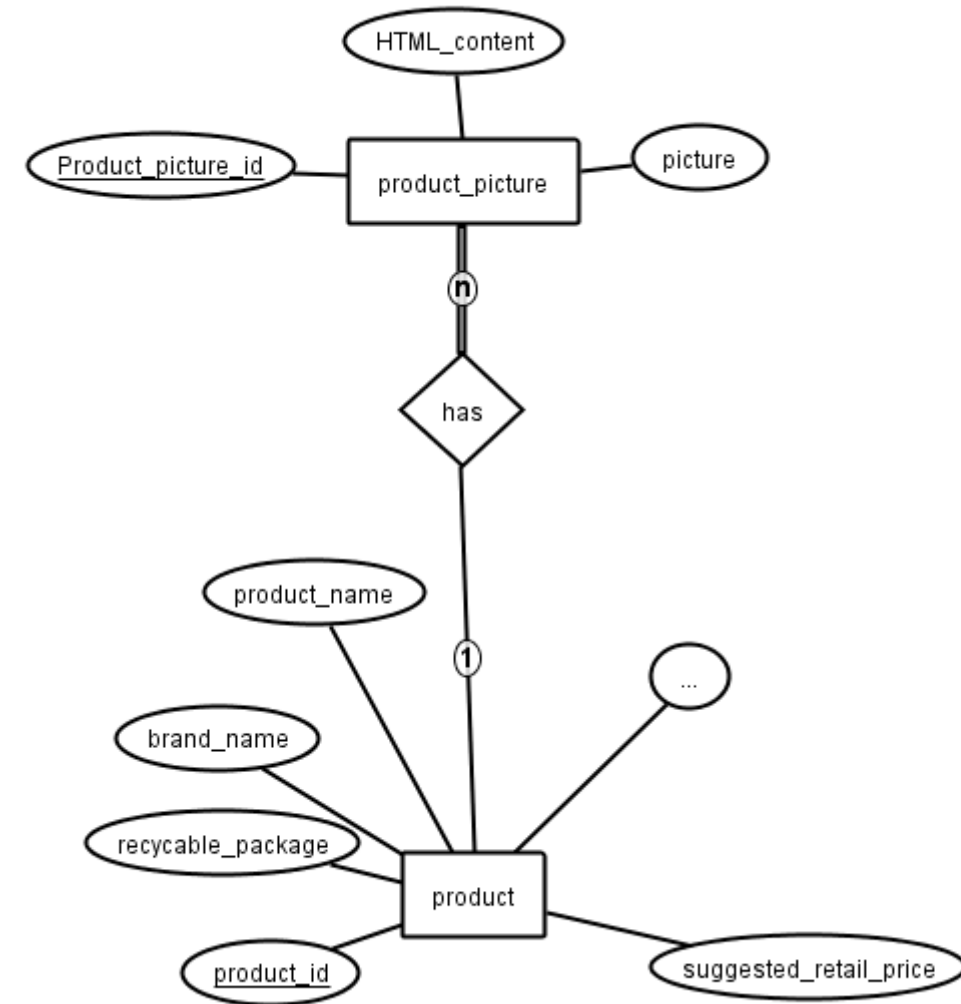
- In jedem Warenlager steht eine Vielzahl von Regalen.
- Jedes Regal hat eine Nummer, die für das Warenlager eindeutig ist.
- Für jedes Regal sind Höhe, Breite, Tiefe sowie die Lagerkapazität bekannt.
- Zudem ist bekannt, ob das Regal leer ist.



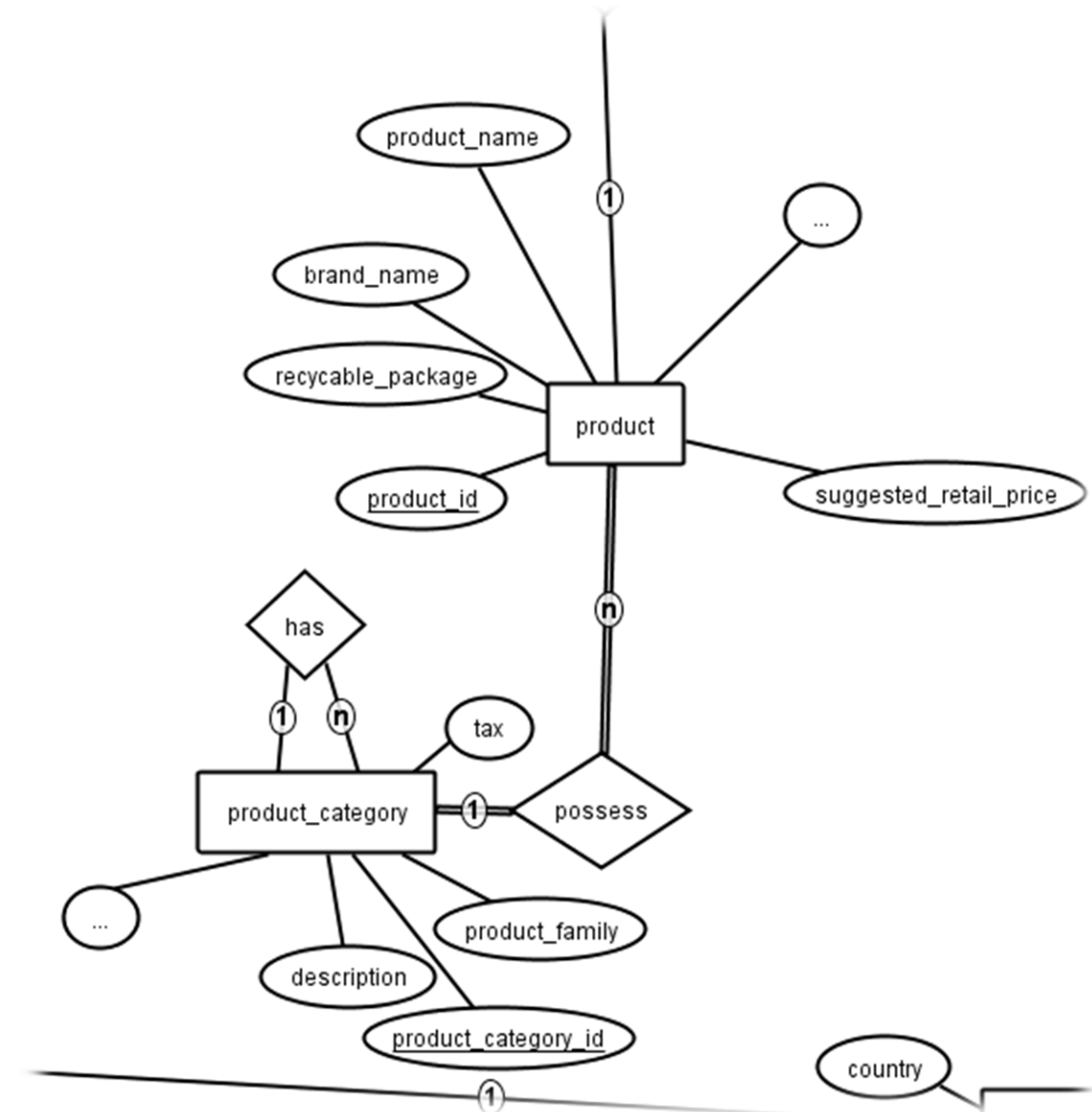
- Im Online-Shop werden verschiedene Produkte angeboten.
- Jedes Produkt hat eine eindeutige Produkt-ID, einen Produktnamen und einen Markennamen des Herstellers.
- Zudem ist bekannt, ob die Verpackung recyclebar ist.
- Ein Produkt hat eine unverbindliche Preisempfehlung des Herstellers. [...]



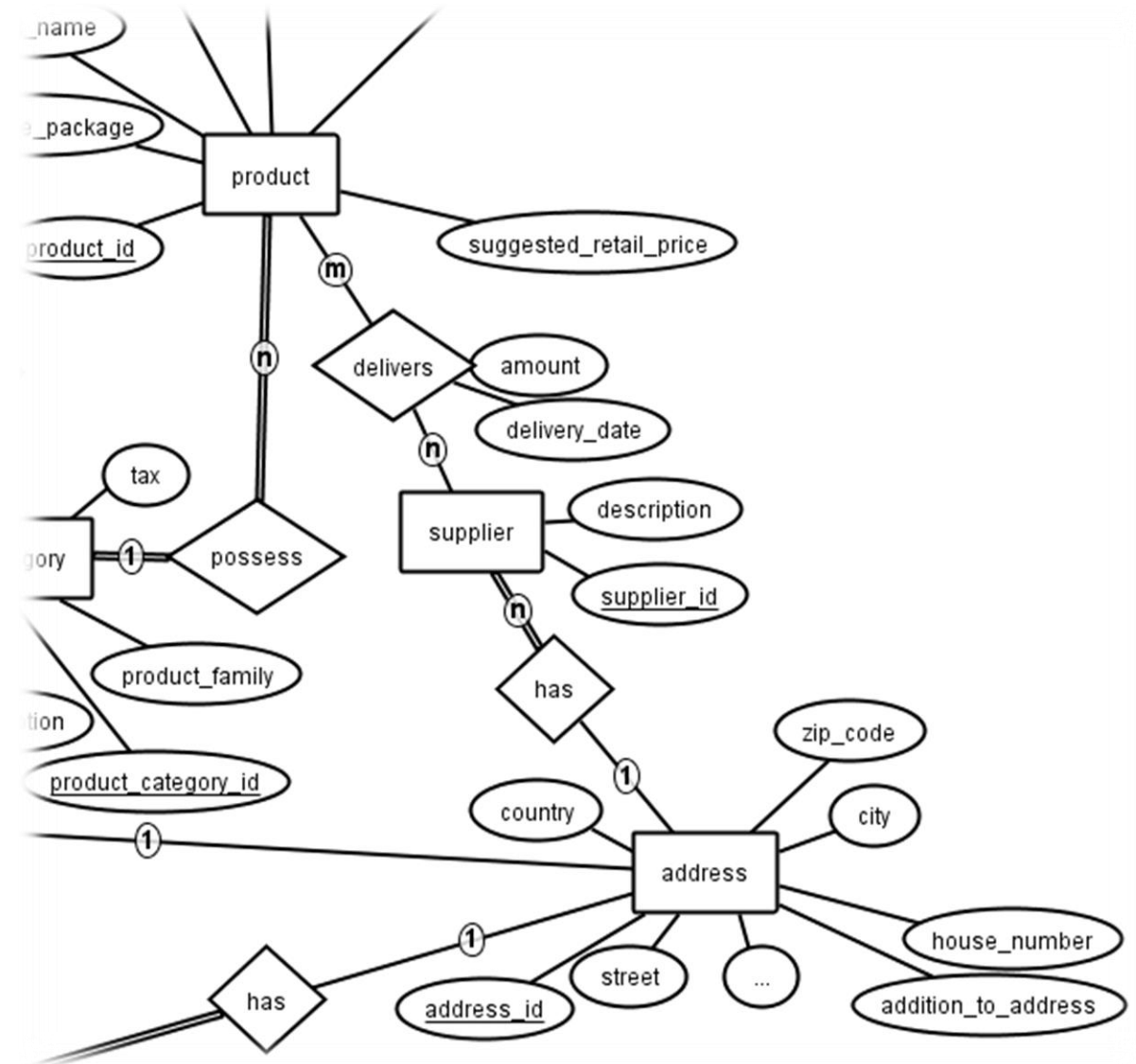
- Für jedes Produkt gibt es ein oder mehr Bilder, die im Online-Shop angezeigt werden.
- Jedes Bild hat eine eindeutige ID.
- Neben dem Dateipfad zur Bilddatei gibt es auch eventuelle Ergänzungen zur Beschreibung des Bildes auf der Homepage.



- Jedes Produkt ist einer Produktkategorie zugeordnet, wobei eine Produktkategorie beliebig viele Produkte umfassen kann.
- Jede Produktkategorie hat eine eindeutige ID.
- Die Produktkategorien können hierarchisch gegliedert sein, sodass eine Produktkategorie viele Subkategorien haben kann, aber jede Produktkategorie höchstens einer Oberkategorie zugeordnet ist.
- Für jede Produktkategorie gibt es einen festgelegten Mehrwertsteuersatz, eine Beschreibung sowie die Bezeichnung der entsprechenden Produktfamilie.

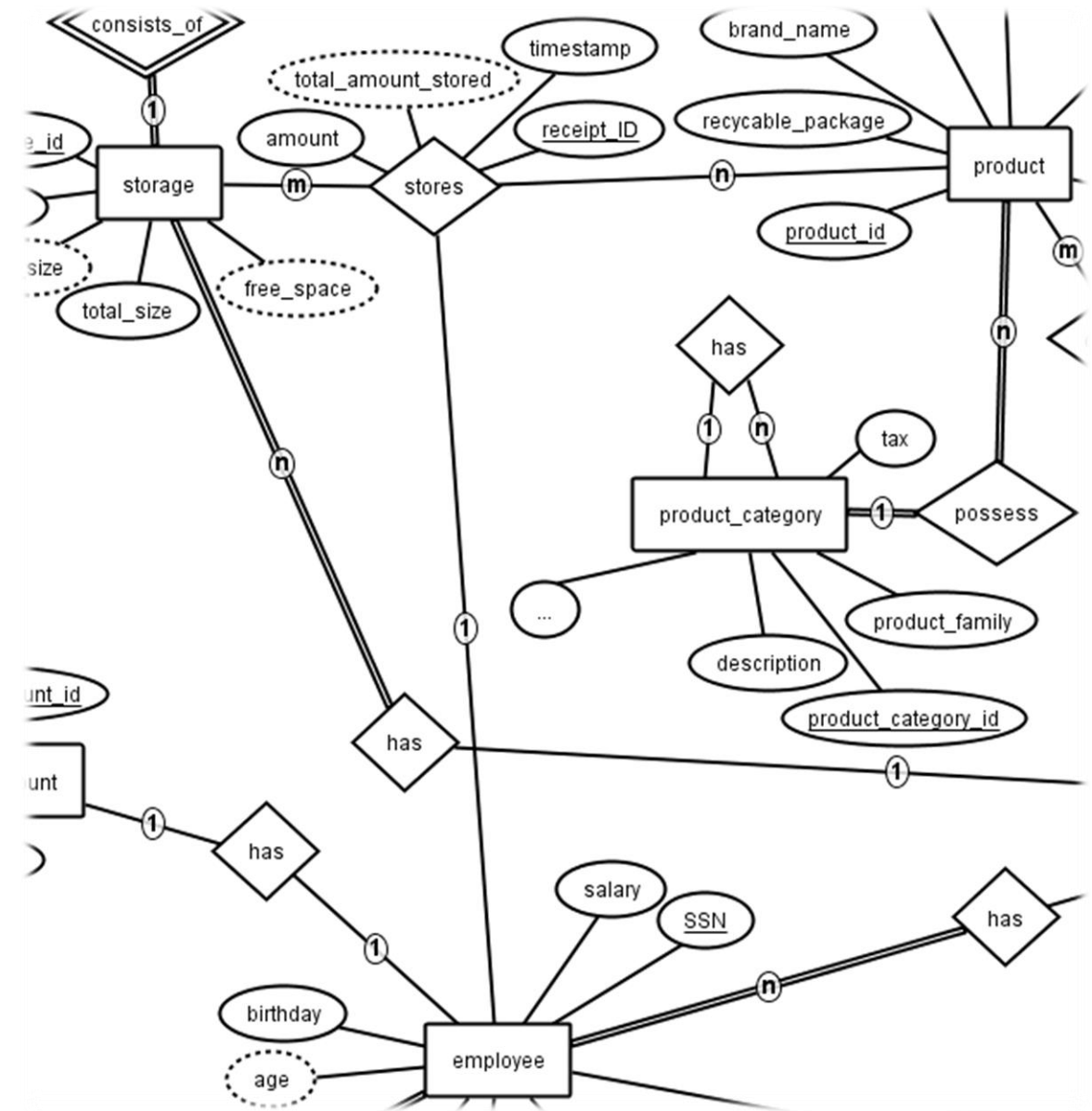


- Produkte werden von einem Lieferanten an einem bestimmten Datum in einer bestimmten Menge geliefert.
- Jeder Lieferant hat eine eindeutige ID und eine Beschreibung.
- Zudem hat jeder Lieferant eine Adresse mit den üblichen Angaben.

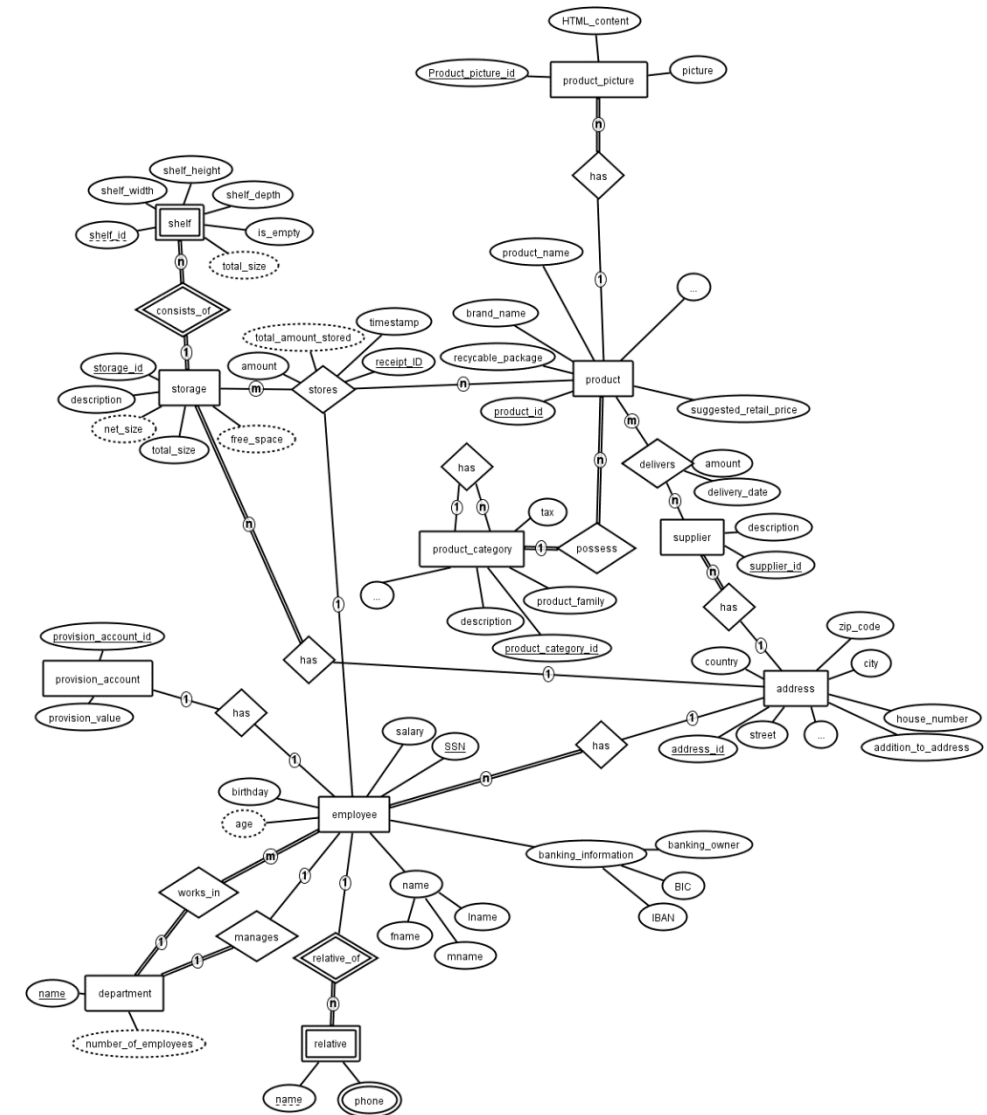


Beispiel: Modellierung eines Online-Shops

- Gelieferte Produkte werden von einem Mitarbeiter in einem Warenlager eingelagert.
- Dabei kann der Mitarbeiter unterschiedliche Produkte auf verschiedene Warenlager verteilen.
- Für jede eingelagerte Lieferung werden die Menge, der Zeitpunkt der Einlagerung, sowie eine eindeutige Empfangsnummer gespeichert.
- Zudem ist die Menge aller in den verschiedenen Warenlagern eingelagerten Produkte bekannt.



- Das war nur ein kleiner Teil...
- Modellierung weiterer Teile (Kunden, Bestellungen, Lieferungen, ...) in den Übungen!



Client-Server-Architekturen

Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

Relational

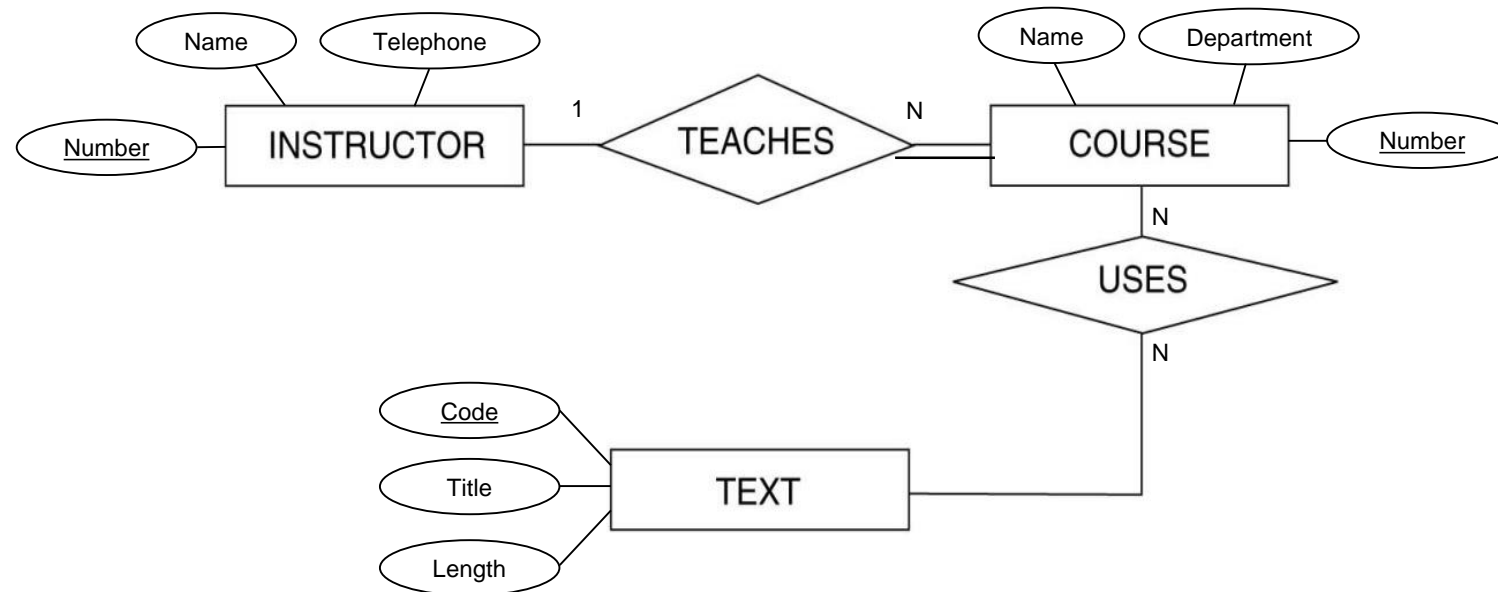
ER Daten SQL DML
NoSQL Recovery Big Data
OODBMS DDL EER-Modell
Transaktion Datenbank Normalisierung
Architektur

Modell

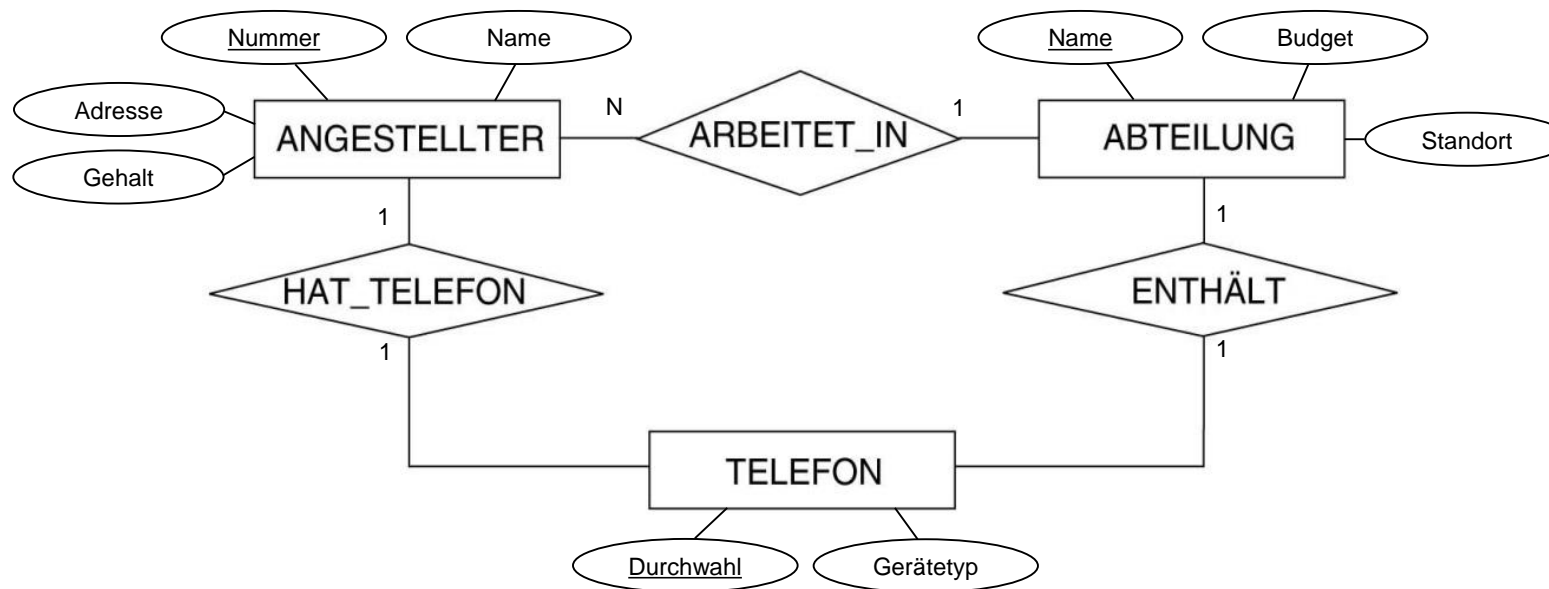
Relationale Algebra

Weitere Beispiele zur ER-Modellierung

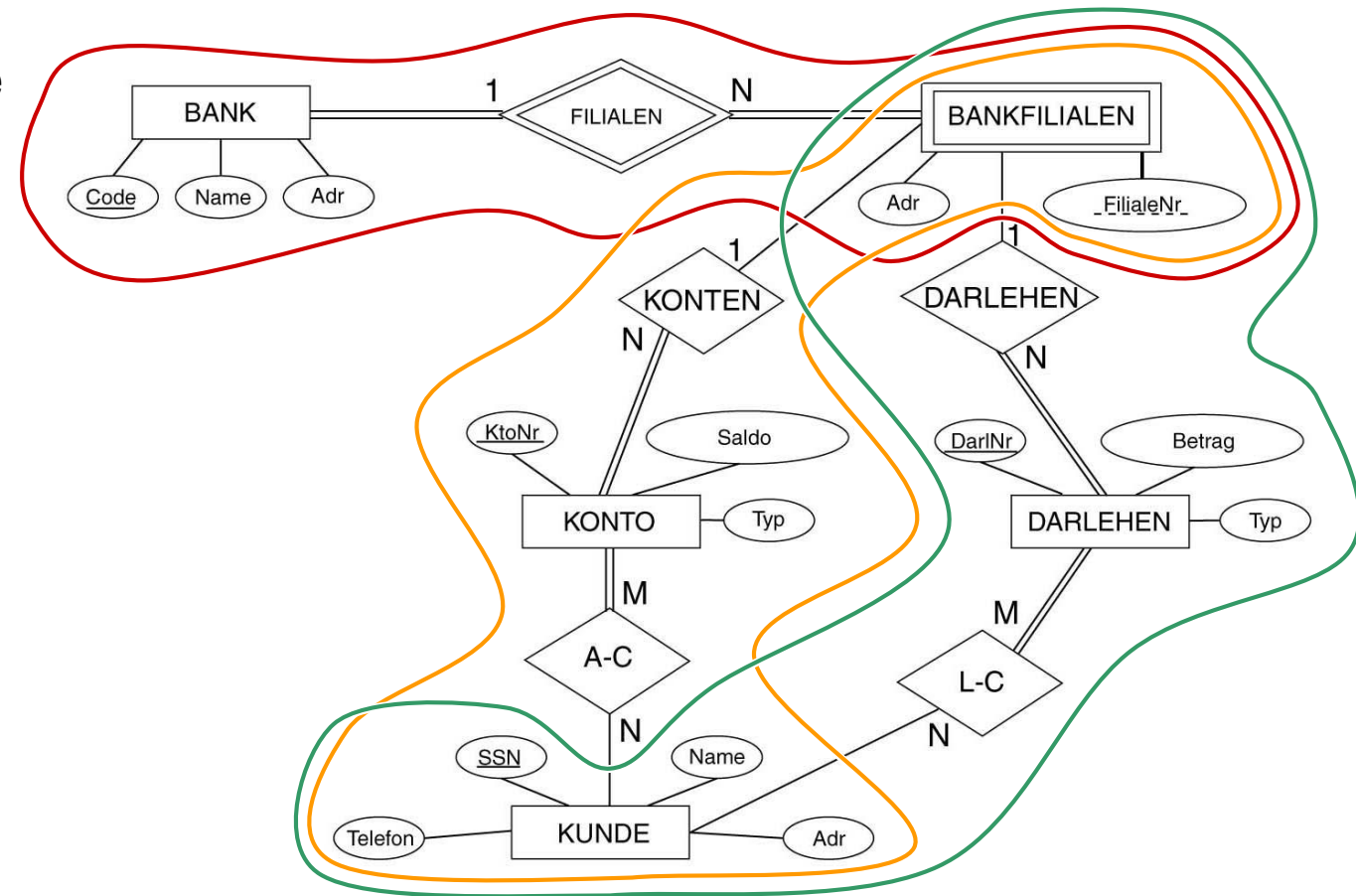
- Ein Dozent unterrichtet mehrere Kurse. In diesen Kursen werden Lehrbücher verwendet.



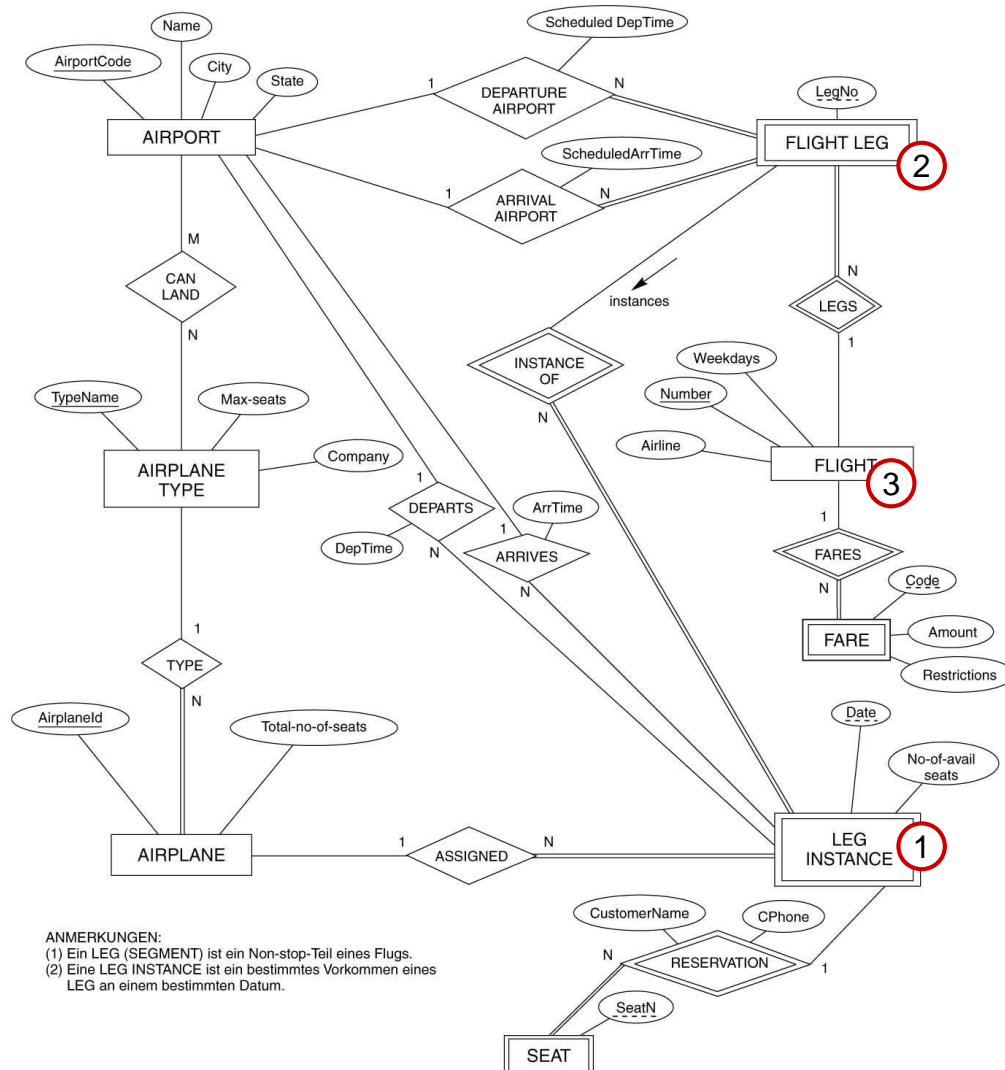
- In einer Abteilung arbeiten mehrere Angestellte. Jeder Angestellte hat genau ein eigenes Telefon. Darüber hinaus gibt es in jeder Abteilung an zentraler Stelle ein Abteilungstelefon.



- Eine Bank wird durch einen Code eindeutig identifiziert; sie hat einen Namen und eine Adresse. Die Tagesgeschäfte der Bank werden in mehreren Bankfilialen abgewickelt, die jeweils eine eigene (bei ihrer Bank eindeutige) Filial-Nummer und eine eigene Adresse haben.
- Ein Kunde kann mehrere Konten besitzen, ebenso können bei einem Konto mehrere Kontoinhaber eingetragen sein. Die Bank speichert für jeden Kunden seine Sozialversicherungsnummer (SSN) als Identifikator, außerdem seinen Namen, seine Adresse und Telefonnummer. Für jedes Konto, das bei genau einer Bankfiliale geführt wird, werden Kontonummer, Kontotyp und Saldo angegeben.
- Kunden können (alleine oder gemeinsam) bei ihrer Bankfiliale mehrere Darlehen aufnehmen. Jedes Darlehen hat eine Darlehensnummer, einen Darlehentyp und Darlehensbetrag.

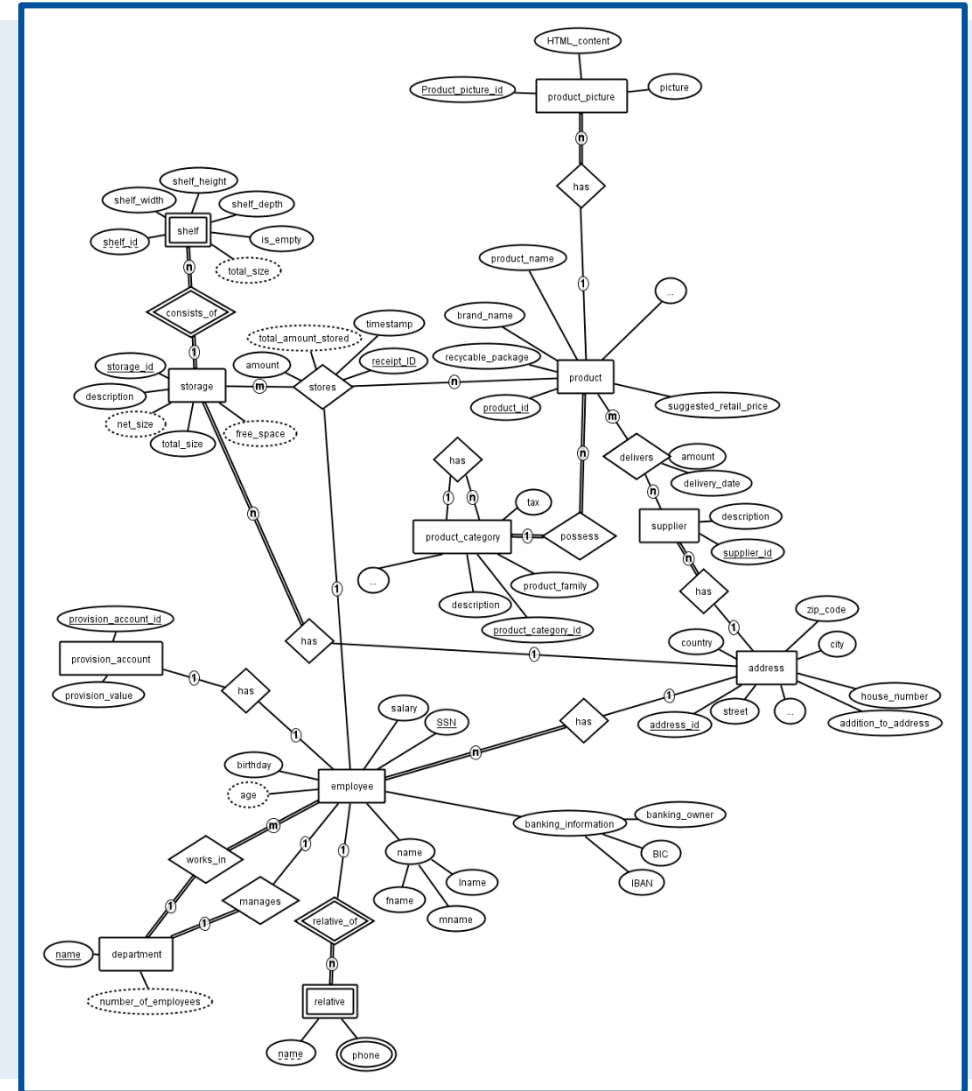


ER-Diagramm einer Datenbank AIRLINE.



- Beachte hier: Identifikation schwacher Entitäten durch starke Entitäten
 - Identifikation über mehrere schwache Entitäten hinweg (transitiv)
 - Der „Seat“ wird über die Beziehung zu einer konkreten „Leg Instance“ ① identifiziert, diese wiederum über die abstrakte „Flight Leg“ ②, welche zu einem bestimmten „Flight“ ③ (mit Flugnummer) gehört
 - So erfolgt die Identifikation eines Sitzes letztlich auch immer transitiv über die Flugnummer

- Phasen des DB-Entwurfs
- Modellierung
- ER-Modellierung
 - ER-Notation:
 - Entitäten
 - Beziehungen
 - Attribute
 - Kardinalitäten
 - Beispiele



- Enhanced-Entity-Relationship-Modell (EER-Modell)
 - Erweiterung vom ER-Modell
 - Spezialisierung und Generalisierung
- Vorgehensmodelle zur (E)ER-Modellierung
- Dokumentation von (E)ER-Modellen
- Qualitätskriterien für (E)ER-Schemata



