

Einführung in Datenbanksysteme

Tutorium 05 Logischer Entwurf

Tutoren

Mit Folienmaterial aus der Vorlesung und anderen Quellen



Fachgebiet Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

<http://www.dima.tu-berlin.de/>

QISPOS-Anmeldung bis 31.Mai

- Relationales Modell
- Vom EER zum Relationen Datenbank Schema
 - Entitytyp mit Attributen
 - Relationstyp
 - 1:1 Beziehungen
 - 1:n Beziehungen
 - m:n Beziehungen
 - Rekursive Beziehungen
 - Mehrstellige Beziehungen
 - Schwache Entitytypen
 - Generalisierung/Spezialisierung

- Was ist eine Relation?
 - Eine Tabelle, d.h.
 - Eine geordnete Menge an Attributen (Spalten) und
 - eine ungeordnete Menge an Tupel (Zeilen)

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

- Attribute: Titel Jahr Länge Typ
- Tupel: z.B. (Basic Instinct, 1992, 127, Farbe)
- Relationenname: Film
- Relationenschema: Film (Title, Jahr, Länge, Typ)
- Datenbankschema: mehrere Relationenschemata

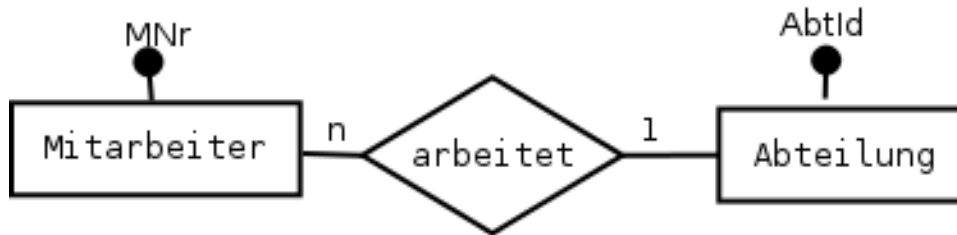
- Bedingungen, die von einer Datenbank zu jedem Zeitpunkt erfüllt sein müssen
- Einschränkungen der möglichen **DB-Zustände (Ausprägungen der Relationen)**
- IB werden überwacht von
 - DBMS
 - Anwendung
- Beispiele IB die von DBMS überwacht werden
 - Schlüsselattribut
 - UNIQUE Key
 - Fremdschlüssel
 - Trigger

- Primärschlüssel
 - Ein Attribut oder eine Kombination mehrerer Attribute einer Relation, die die Eindeutigkeit der Tupel dieser Relation sicherstellen, ein Primärschlüssel soll minimal sein
 - Es existiert ein Primärschlüssel und ggf. weitere Schlüsselkandidaten.
- Unique Key
 - Hat alle Eigenschaften vom Primärschlüssel, stellt aber lediglich die Eindeutigkeit sicher.
- Fremdschlüssel
 - Ein Attribut, das auf einen Tupel einer anderen Relation verweist.
 - referentielle Integritätsbedingungen beim Ändern/Löschen eines Datensatzes in der Haupttabelle
 - restricted/no action(nicht zulässig)
 - cascade(weitergeben)
 - set null

- $R(\underline{a}, b)$
 - a ist ein Schlüsselattribut
- $\text{Buch}(\underline{\text{ISBN}}, \text{Titel})$
 - ISBN ist ein eingebetteter Primärschlüssel
- $\text{Artikel}(\underline{\text{ArtikelNr}}, \text{Bezeichnung})$
 - ArtikelNr ist nicht eingebettet sondern ein künstlicher Schlüssel
- $R(\underline{a}, \underline{b})$
 - a, b ist ein zusammengesetzter Schlüssel
- $R(\underline{A(x, y, z)}, c)$
 - $A(x, y, z)$ ist ein komplexer Schlüssel
- $R(\underline{a}, b \rightarrow X)$ und $X(\underline{b}, c)$
 - b ist ein Fremdschlüssel, zeigt auf den Primärschlüssel b der Relation X
 - b ist auch ein eingebetteter Fremdschlüssel, da er zum Datensatz gehört

- Darstellung aller Informationen des ER-Diagramms
- Exaktheit
 - Das Datenbankschema kann genauso viele Instanzen wie das ER-Diagramm darstellen.
 - Das Datenbankschema kann nicht mehr Instanzen als das ER-Diagramm darstellen.
- Erhaltung und Einhaltung der Informationskapazität!
 - Genau die gleichen Daten können abgespeichert werden
 - Nicht mehr, nicht weniger.

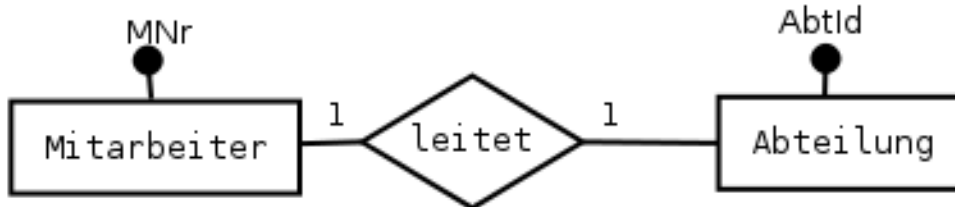
1. Wandle jeden Entitytypen in eine Relation mit den gleichen Attributen um.
 2. Wandle jeden Relationshiptypen in eine Relation um, deren Attribute die zugehörigen Attribute und die Schlüsselattribute der beteiligten Entitytypen sind.
 3. Verfeinere den Entwurf
 1. Zusammenlegung von Relationen
 2. Normalisierung (**nächste Woche**)
- Ausnahmen
- Schwache Entitytypen
 - vererbte Schlüssel einbeziehen
 - Relation zusammenlegen, wenn keine eigenen Attribute
 - IST Relationships



Mitarbeiter(MNr, AbtId->Abteilung)
Abteilung(AbtId)



Kunde(KundenNr)
bestellt(KundenNr, PNr)
Produkt(PNr)

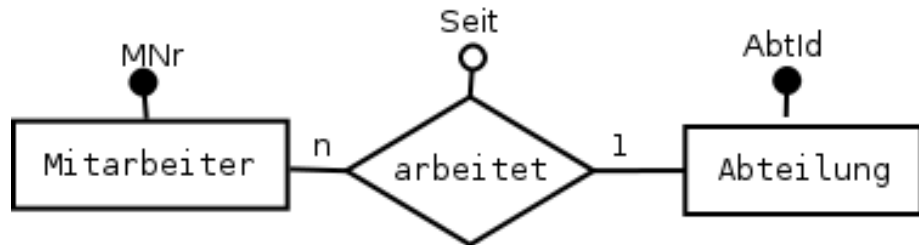


Mitarbeiter(MNr, AbtId)

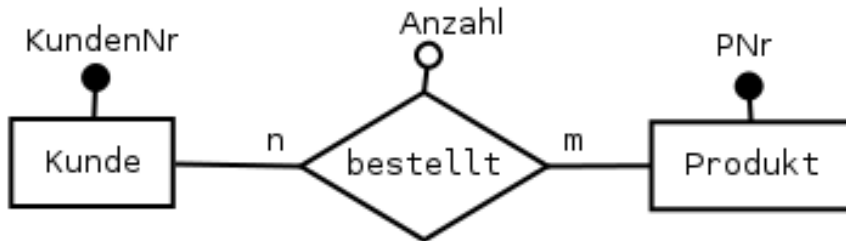
Welche Attribute sind Fremdschlüsseln, welche Schlüssel?

Warum haben wir bei 1:1 Beziehung nur eine Tabelle?

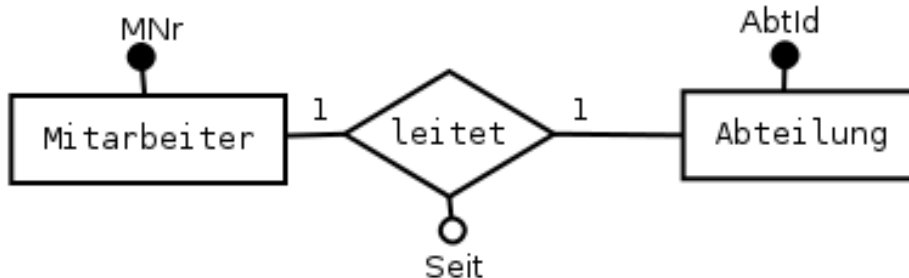
Warum bei 1:n nur zwei Tabellen?



Mitarbeiter(MNr, Seit, AbtId → Abteilung)
Abteilung(AbtId)

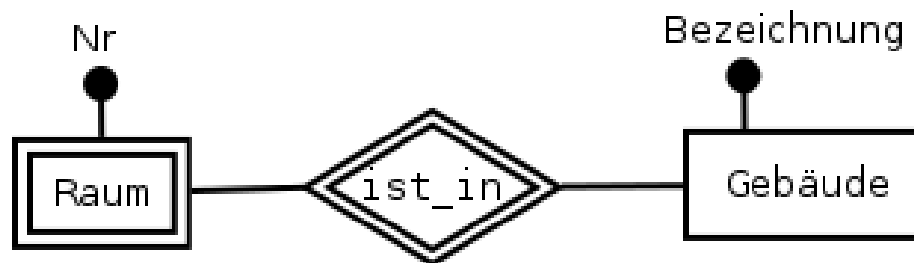


Kunde(KundenNr)
bestellt(KundenNr, PNr, Anzahl)
Produkt(PNr)



Mitarbeiter(MNr, AbtId, Seit)

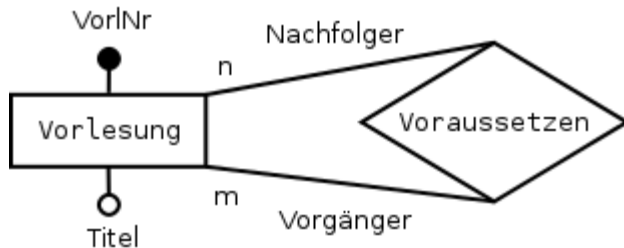
- Die Relation eines *schwachen Entitytypen* muss nicht nur die eigenen Attribute, sondern auch die Schlüsselattribute aller Entitytypen, die über unterstützende Relationshiptypen erreicht werden, enthalten.



Gebäude(Bezeichnung)

Raum(Nr, Bezeichnung → Gebäude)

- Rekursive Beziehungen können genauso wie vorher behandelt werden.



Vorlesung(VorlNr, Titel)

Voraussetzen(Vorgänger, Nachfolger)

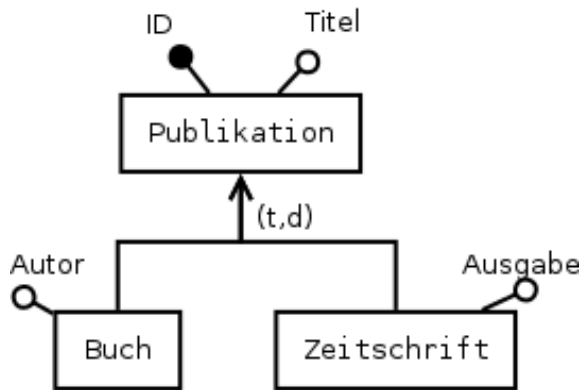
Voraussetzen

<u>Vorgänger</u>	<u>Nachfolger</u>
5001	5041
5001	5043
5041	5052
5043	5052

Wieviele Relationen bei rekursiver Beziehungen
Bei 1:1, 1:n?

■ ER-Stil

- in den Relationenschemata der Spezialisierungen wird der Primärschlüssel der Generalisierung als Fremdschlüssel und zudem als Primärschlüssel aufgenommen



Welche Vorteile?
Welche Nachteile?

Publikation

<u>ID</u>	Titel
1	Der Spiegel
2	Database systems

Buch

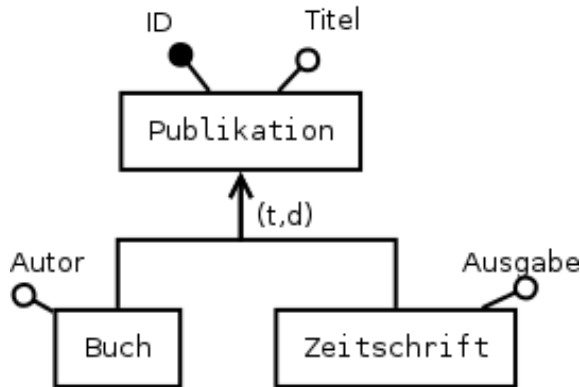
<u>ID</u>	Autor
2	Hector G.-M.

Zeitschrift

<u>ID</u>	Ausgabe
1	Nr. 15 / 2.4.1

■ Nullwerte

- alle Attribute der Spezialisierungen werden in das Relationenschema hinzugefügt – Nullwerte für die Instanzen, die zu einer Klasse gehören in denen die Attribute nicht vorhanden sind



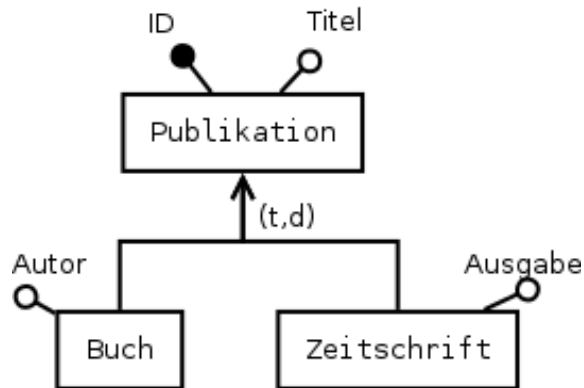
Vorteile?
Nachteile?

Publikation

<u>ID</u>	Titel	Typ	Autor	Ausgabe
1	Der Spiegel	Zeitschrift	NULL	Nr. 15 / 2.4.1
2	Database systems	Buch	Hector G.-M.	NULL

■ Objektorientierter Stil

- Erzeuge Relation für jeden Teilbaum
- Diese Relation repräsentiert die Entities, die genau diese Komponente der Hierarchie besitzen
- Objekte gehören zu genau einer Klasse



Buch	<u>ID</u>	Titel	Autor
	2	Database Systems	Hector G.-M.

Zeitschrift	<u>ID</u>	Titel	Ausgabe
	1	Der Spiegel	Nr. 15 / 2.4.1

■ ER Stil

- Schlüsselattribute vererben
- Relation für jede „Sorte“

Vor-/Nachteil:

- Viele Tupel pro Entity (da Zusammensetzung über vererbte Schlüssel notwendig)
- n Relationen

■ Objektorientierter Stil

- Relation für jeden Teilbaum
- Achtung bei overlapping: Alle Kombinationen berücksichtigen

Vor-/Nachteil:

- Minimaler Speicherbedarf (nur 1 Tupel pro Entity)
- Jeweils nur so viele Attribute wie nötig
- 2^n Relationen!

■ Mit Nullwerten

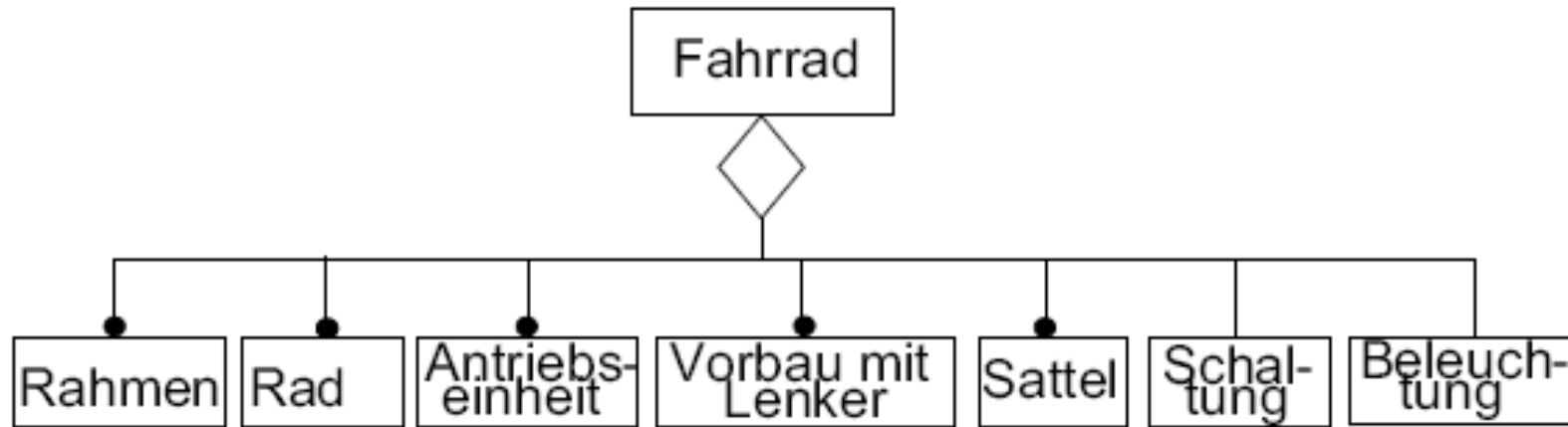
- Ein ET der für „nicht Spezialfälle“ NULL werte der Attribute speichert.

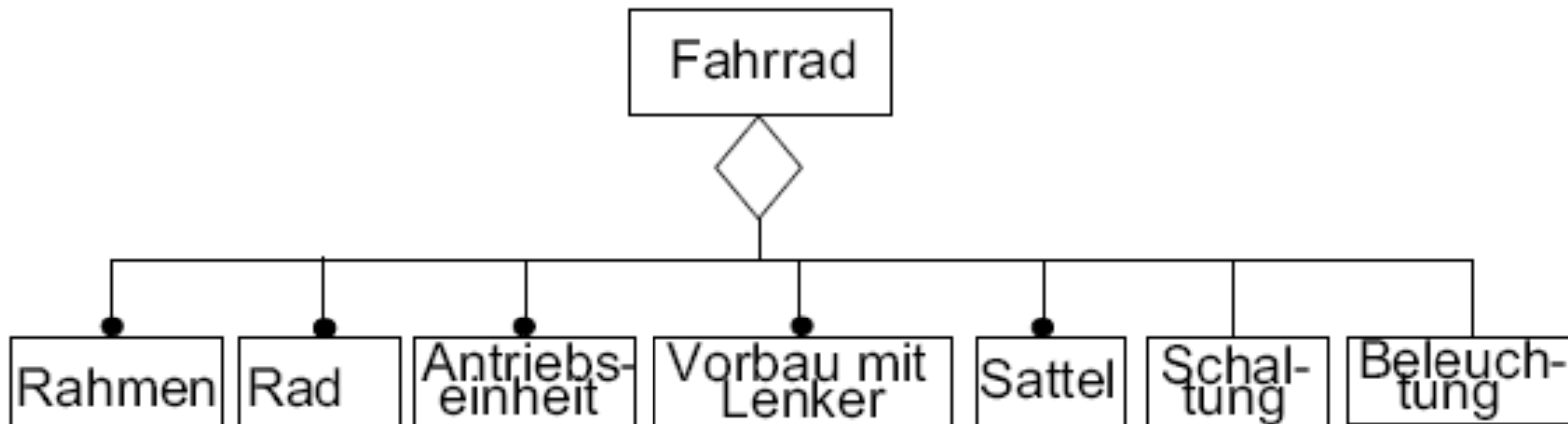
Vor-/Nachteil:

- Nur eine Relation
- Lange Tupel
- viele NULL möglich

=> Die Wahl des Stils ist im Einzelfall zu entscheiden

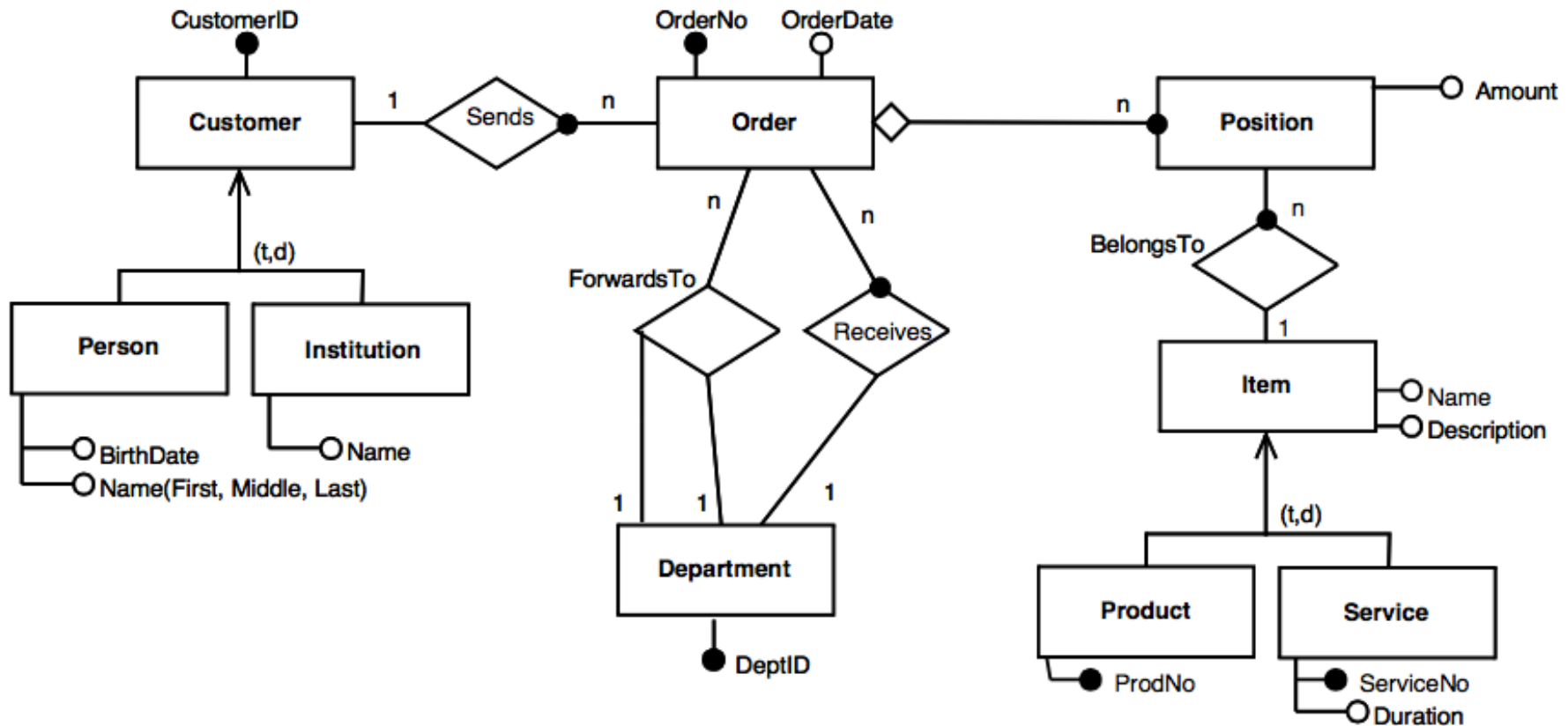
- Bildet das EER-Typ-Diagramm in ein relationales Schema ab.
Attribute übersichtshalber weggelassen





- **Fahrrad** (FNr)
- **Rahmen** (RNr, FNr)
- **Rad** (RadNr, FNr)
- **Antriebseinheit** (ANr, FNr)
- **Vorbau_mit_Lenker** (VmLNr, FNr)
- **Sattel** (SNr, FNr)
- **Schaltung** (SchNr, FNr)
- **Beleuchtung** (BNr, FNr)

- Führen Sie den logischen Entwurf in Richtung einer relationalen Datenbank durch, d.h. bilden Sie das EER-Typ-Diagramm in ein relationales Schema ab.



- **Customer** (CustomerID, BirthDate, PersName (First, Middle, Last), InstName, Type)
- **Department** (DeptID)
- **Order** (OrderNo, OrderDate, ReceivingDeptID, SendingCustomerID)
- **ForwardsTo** (OrderNo, SendingDeptID, ReceivingDeptID)
- **Item** (ItemID, Name, Description)
- **Product** (ItemID, ProdNo)
- **Service** (ItemID, ServiceNo, Duration)
- **Position** (PosID, Amount, ItemID, OrderNo)