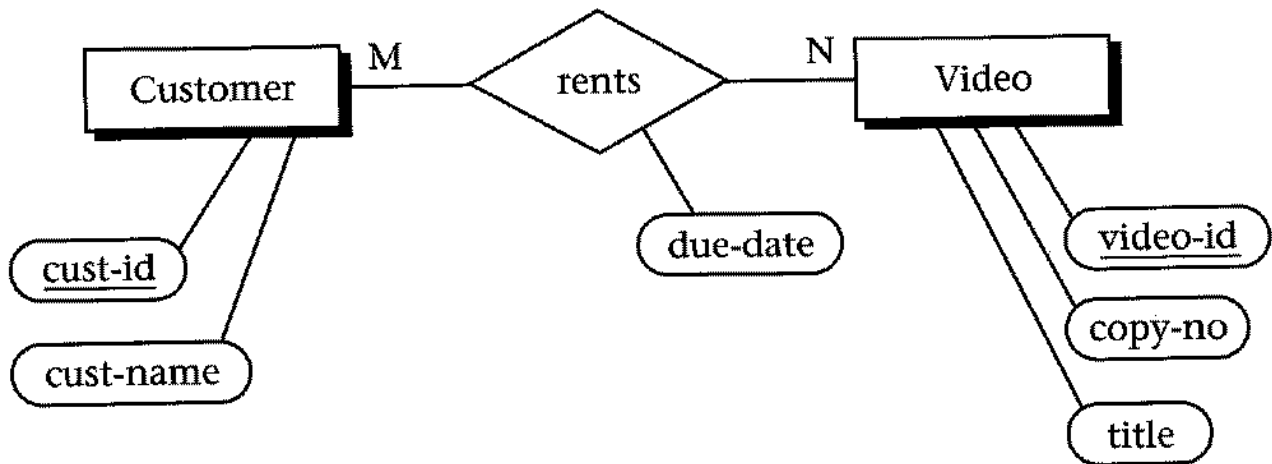


Beispiel ER-Diagramm



ER-Diagramme sind das Ziel des konzeptuellen Entwurfs. Es enthält alle Informationseinheiten, die im zu implementierenden Informationssystem enthalten sein sollen. Es dient auch der Kommunikation mit den späteren Benutzern, um die Vollständigkeit und Korrektheit der Modellierung zu validieren.

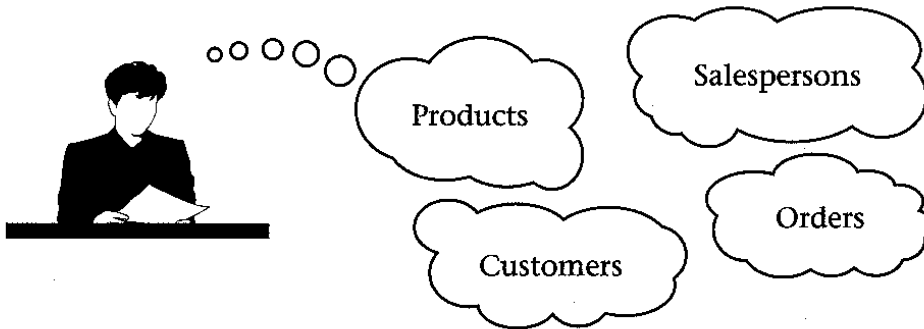
Aus einem ER-Diagramm gewinnt man leicht bspw. ein relationales Schema (oder ein Schema eines anderen durch eine Datenbanksystem unterstützten Modells).

Das Datenbanksystem wird durch Bekanntgabe des Schemas an die Anwendung "angepaßt": Es kann Daten, die dem Schema gehorchen, speichern und verarbeiten.

Vorteil: nicht jeder muß seine eigene Datenhaltung schreiben.

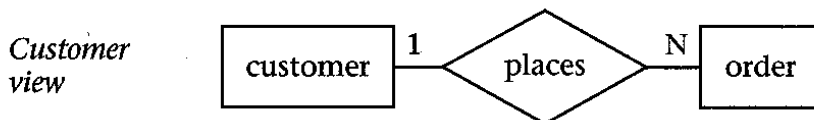
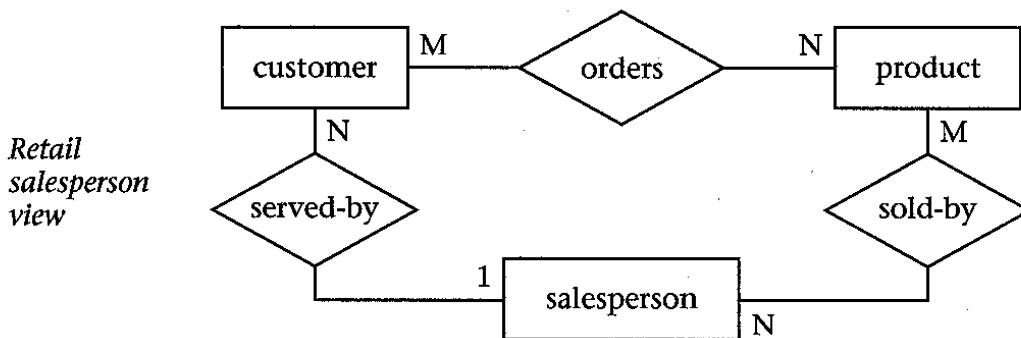
Datenbanklebenszyklus

Step I Information requirements (reality)

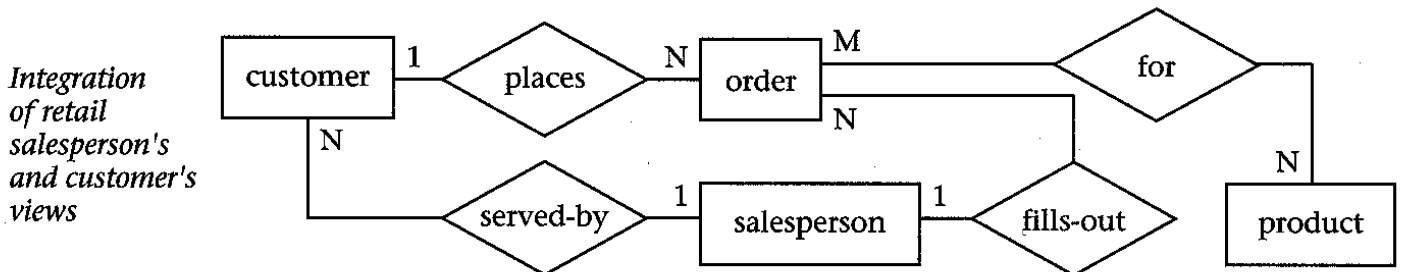


Step II Logical design

Step IIa ER modelling (conceptual)



Step IIb View integration



Beachte: Datenbank wird von mehreren Personen benutzt. Diese haben ihre eigene Sicht auf die Daten.

Datenbanklebenszyklus (Fortsetzung)

Step IIc Transformation of the ER diagram to SQL tables

Customer

cust-no	cust-name

Product

prod-no	prod-name	qty-in-stock

Order

order-no	sales-name	cust-no

Order-line

order-no	prod-no

Salesperson

sales-name	addr	dept	job-level	vacation-days

Step IIId Normalization of SQL tables (3NF, BCNF, 4NF, 5NF)

Decomposition of tables and removal of update anomalies

Salesperson

sales-name	addr	dept	job-level

Sales-vacations

job-level	vacation-days

Step III Usage refinement

Customer

cust-no	cust-name

Customer / refined

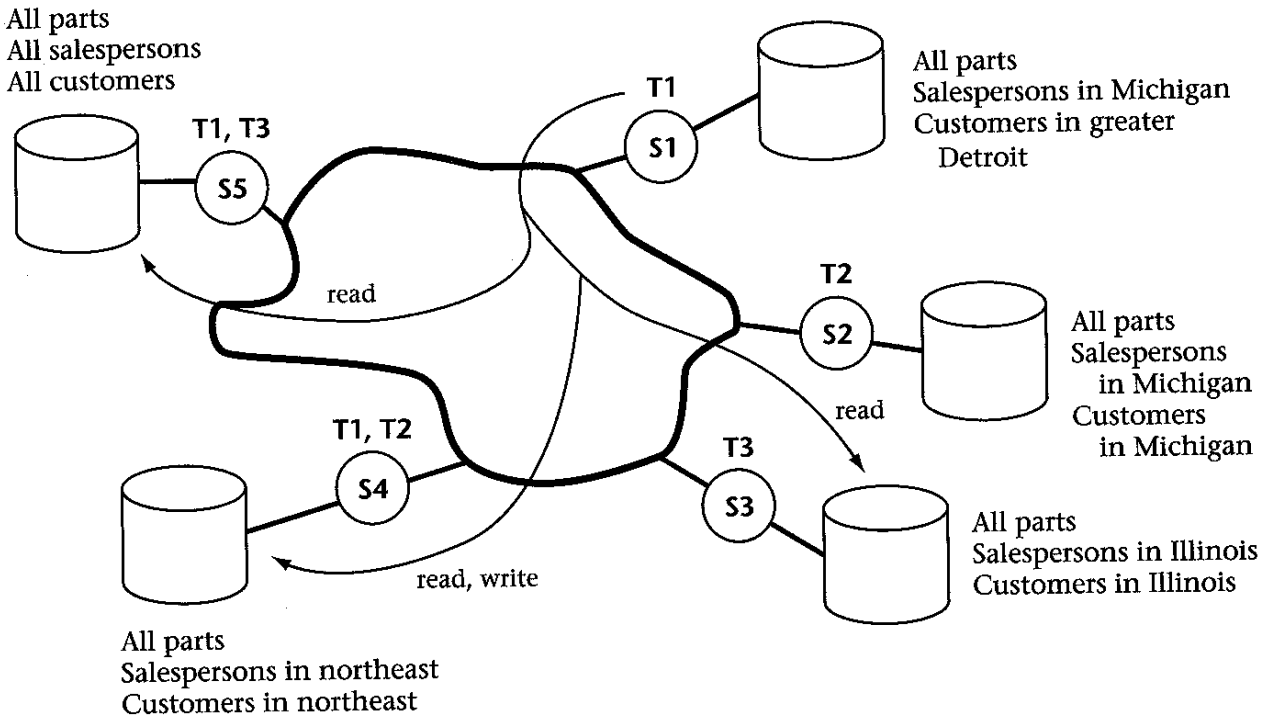
cust-no	cust-name	sales-name

Order

order-no	sales-name	cust-no

Datenbanklebenszyklus (Fortsetzung)

Step IV Data distribution



S1 = Ann Arbor, S2 = Detroit, S3 = Chicago, S4 = Boston, S5 = New York

Decisions: fragmentation, replication, allocation

Objectives: min response time, min communication cost, max availability

Step V Local DBMS schema and physical design

SQL (relational)

```

create table customer
(cust_no integer,
 cust_name char(15),
 cust_addr char(30),
 sales_name char(15),
 prod_no integer,
 primary key (cust_no),
 foreign key (sales_name)
 references salesperson,
 foreign key (prod_no)
 references product);
  
```

CODASYL (network)

```


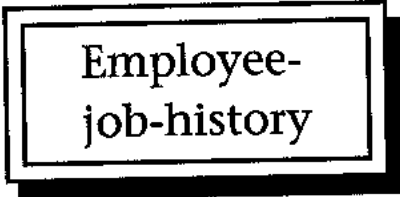


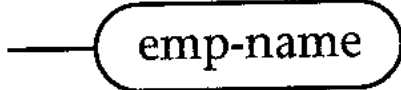

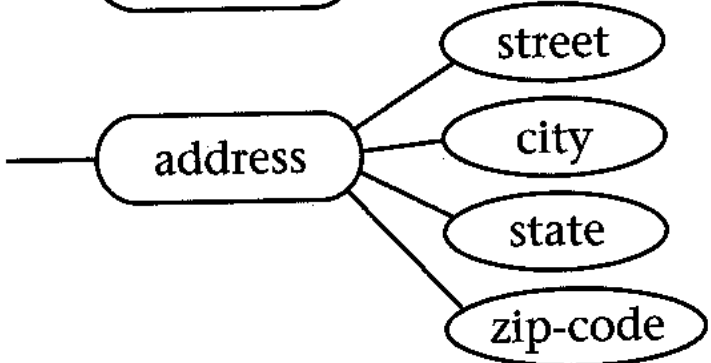
schema name is order-db

record name is customer
location mode is calc using cust-no
duplicates not allowed
02 cust-no pic x(10)
02 cust-name pic x(20)
02 cust-addr pic x(25)

set name is served-by
owner is salesperson
member is customer
  
```

Physical design parameters: Indexing, access methods, clustering

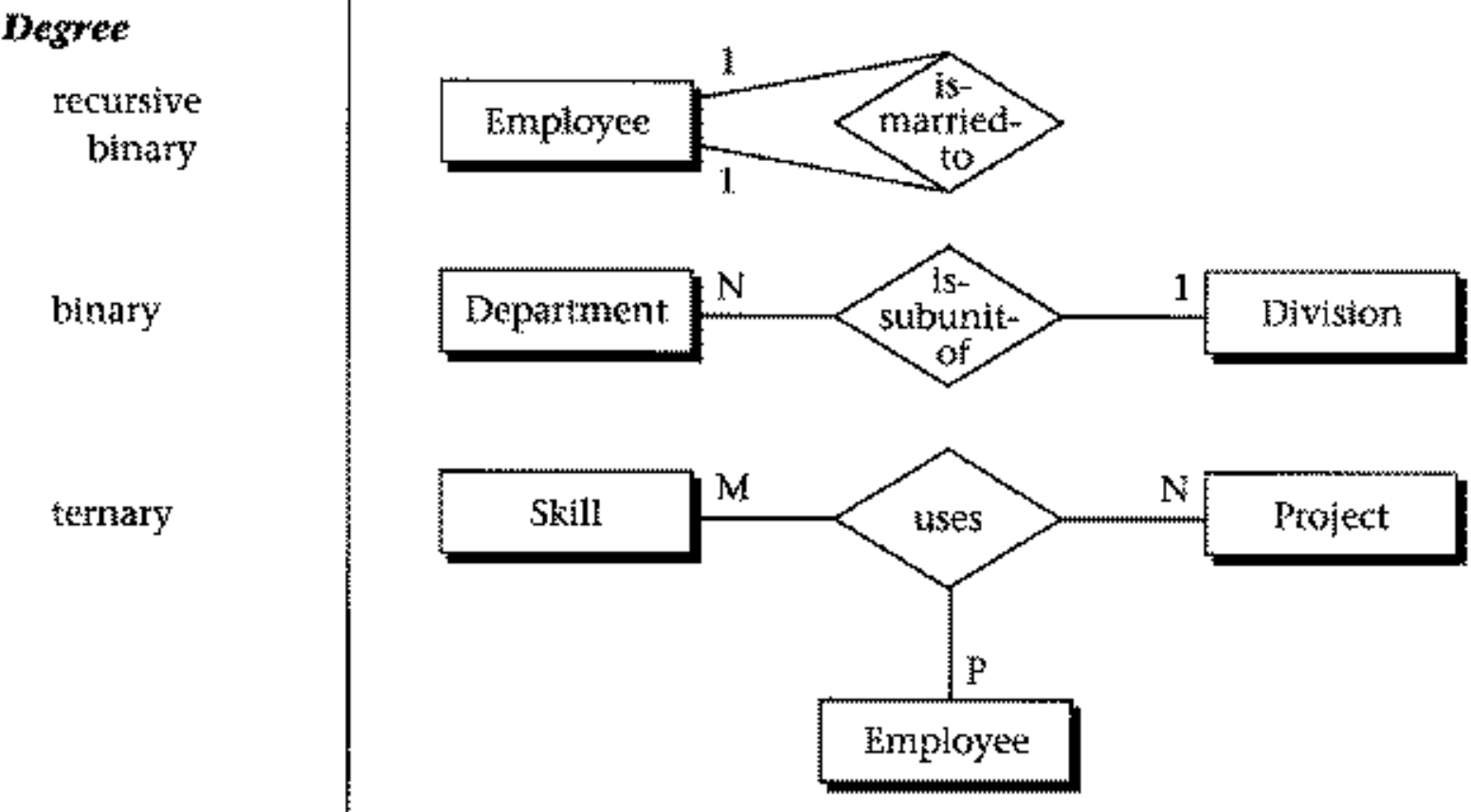
(Basis-)Konzepte des Entity-Relationship-Diagramms

Concept	Representation & Example
Entity	
Weak entity	
Relationship	
Attribute	
indentifier (key)	
descriptor (nonkey)	
multivalued descriptor	
complex attribute	

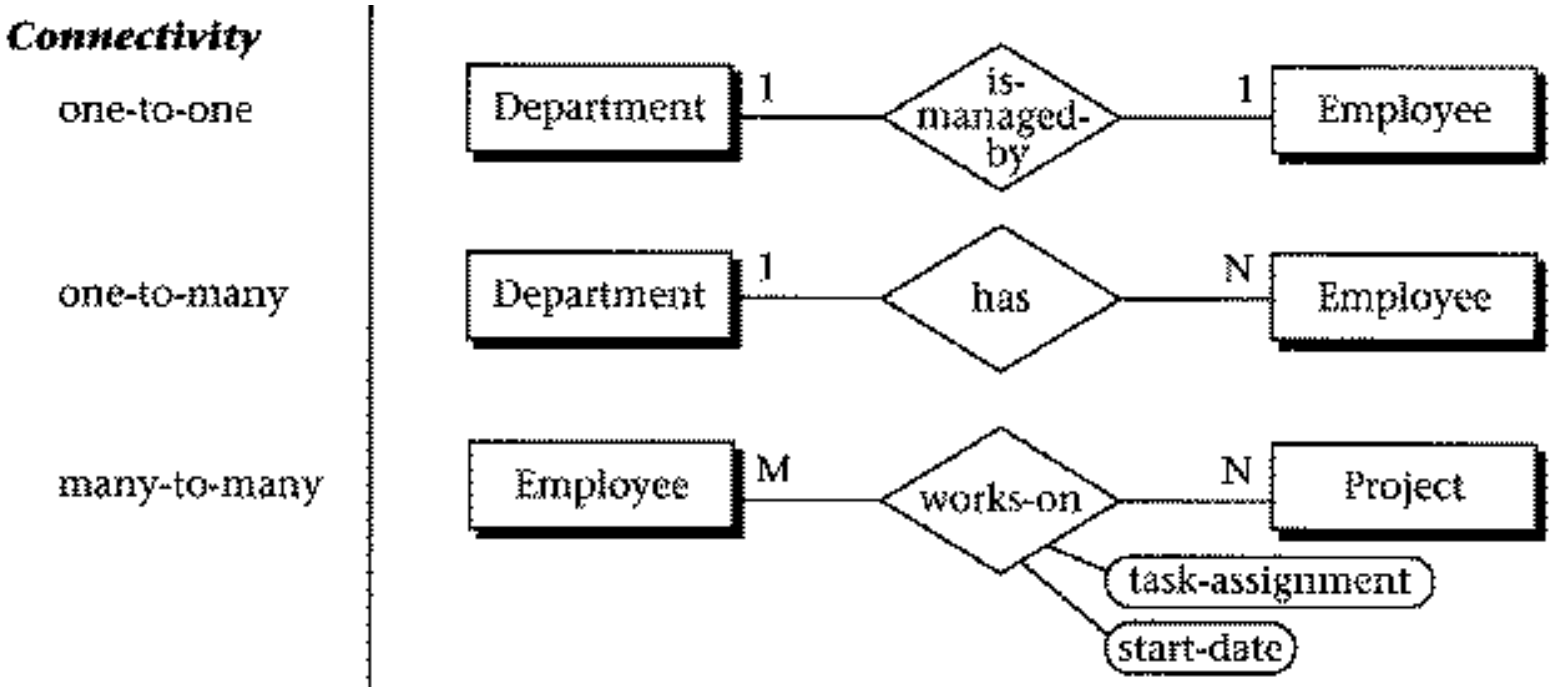
Entitäten

- Entitäten sind diejenigen Objekte, über die Informationen gesammelt und verarbeitet werden sollen
- Oft Personen, Plätze, Ereignisse
- Unterscheidung:
 - Entität: eine Person (auch Instanz oder Ausprägung genannt)
 - Entitätstyp: abstrakte Beschreibung aller möglichen Personen bzw. ihrer Daten
 - schwache Entitäten sind solche, die nicht alleine existieren können (z. B. Räume ohne Gebäude)

Beziehungstypen: Grad



Beziehungstypen: Funktionalität



Beziehungstypen: Existenz

Existence

optional



mandatory



unknown



Beziehungstypen

- Beziehungen erfassen Beziehungen (im weitesten Sinne) zwischen Entitäten
- Beispiel: verheiratet mit, angestellt bei, arbeitet für, ...
- konkrete Beziehung heißt auch Ausprägung
- Die Rolle beschreibt den Part einer Entität in einer Beziehung

Beziehungstypen und ihre Funktionalität

Oft unterliegen Beziehungen Einschränkungen.

verheiratet mit: Jede Person mit genau einer anderen Person, mehr geht nicht (1:1)

arbeitet für: Jede Person arbeitet für höchstens einen Arbeitgeber, jeder Arbeitgeber beschäftigt aber mehrere Personen (1:N)

befreundet mit: beliebig (N:M)

Attribute

Attribute beschreiben Entitäten oder besser: deren Eigenschaften.

- Alter
- Größe
- Gewicht
- Gehalt

Sowohl Entitäten als auch Beziehungen können Attribute haben.

Charakteristika von Beziehungen

Wertigkeit einer Beziehung:

- Bis jetzt nur zweiwertige Beziehungen, d. h. Beziehungen zwischen zwei Entitätstypen.
- Allgemein: n-wertige Beziehungen, also Beziehungen zwischen n Entitäten ($n > 2$).

Teilnahme einer Entität an einer Beziehung:

- optional
- zwingend (mandatory)

Modellierung

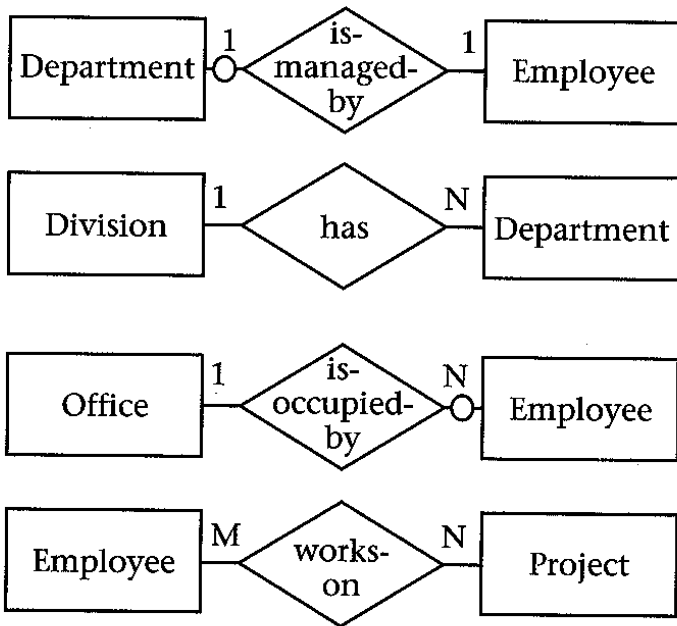
- dient der Kommunikation mit den zukünftigen Benutzern
- dient dem Realisierungsteam als Vorlage

gut zur Kommunikation: Grafik mit genormten Bestandteilen

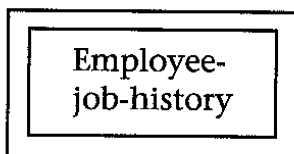
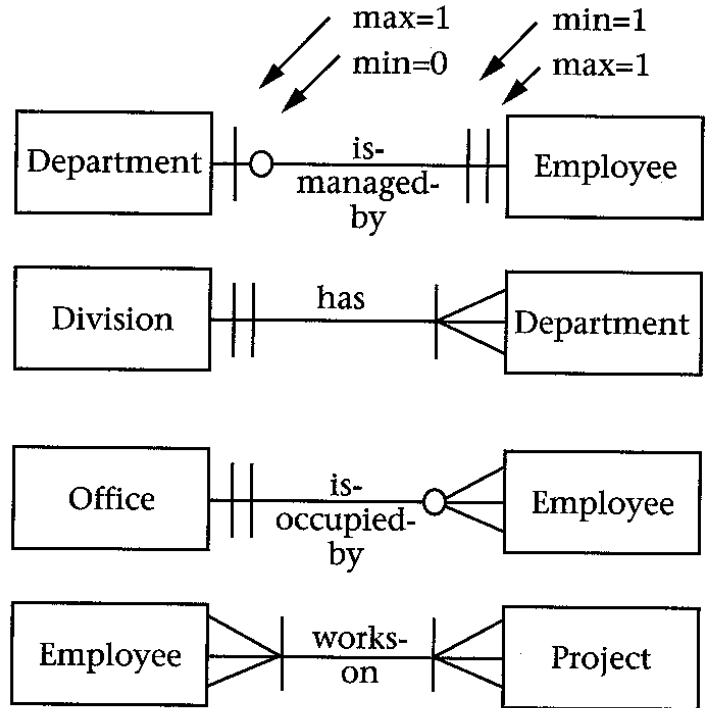
weitere Beispiele: Schaltpläne

Verschiedene grafische Notationen

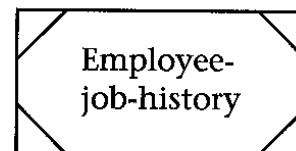
ER model constructs using the Chen notation



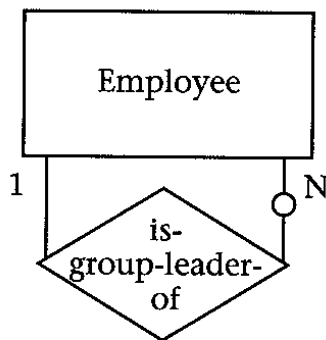
ER model constructs using the "crow's foot" approach [Ever86, Knowledgeware]



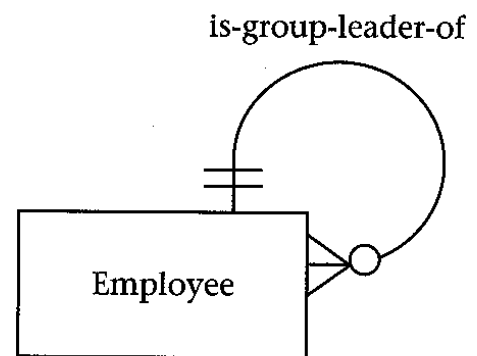
weak entity



intersection entity



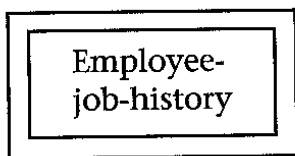
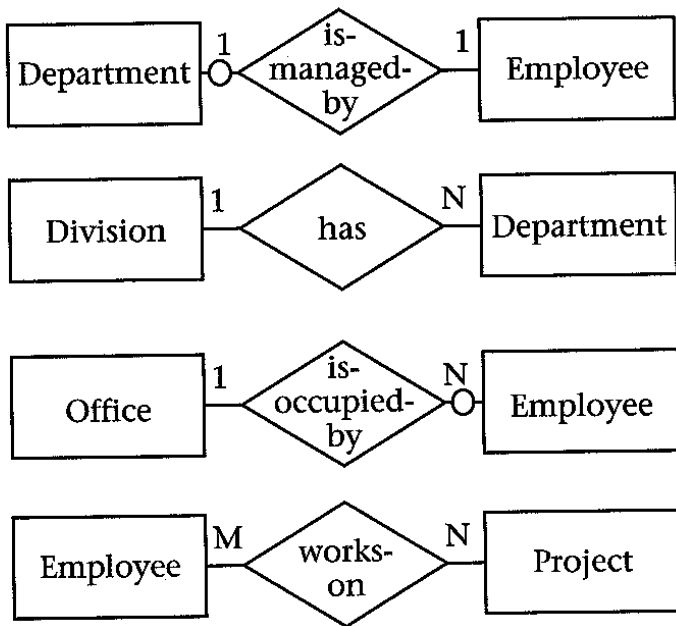
Recursive binary relationship



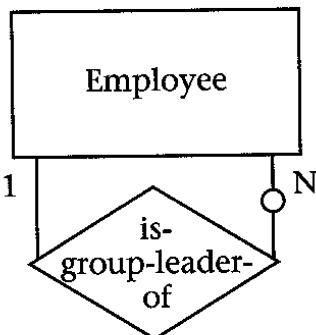
Recursive entity

Verschiedene grafische Notationen

ER model constructs using the Chen notation

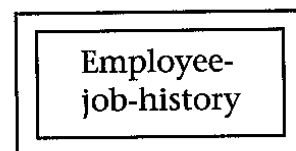
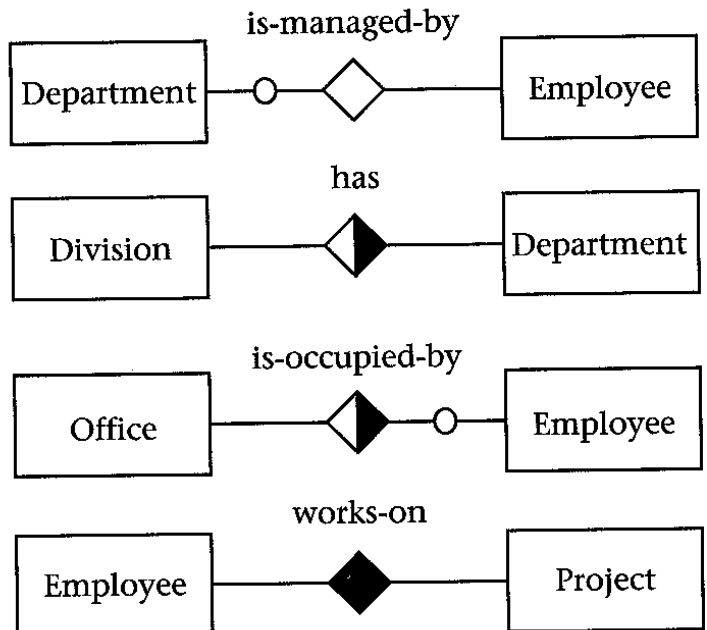


weak entity

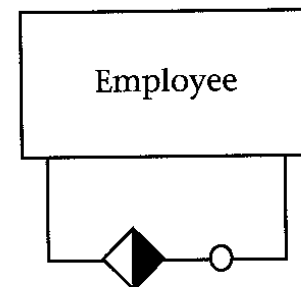


Recursive binary relationship

ER model constructs using shading [Rein85, TYF86]



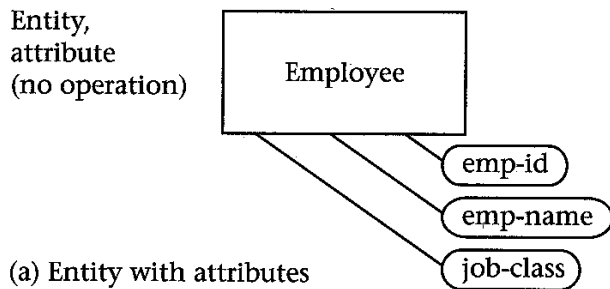
weak entity or intersection entity



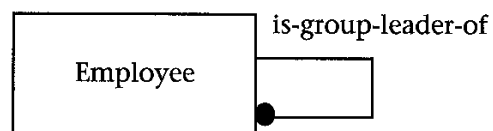
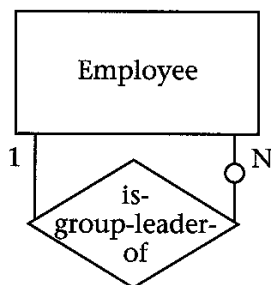
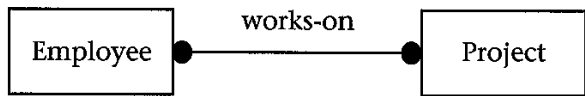
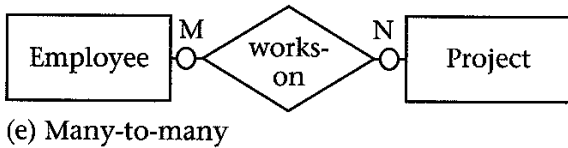
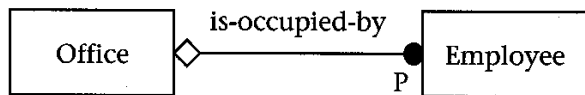
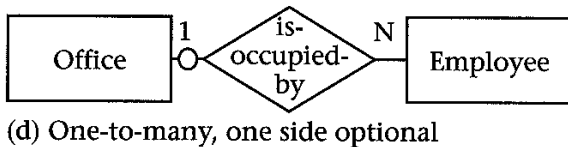
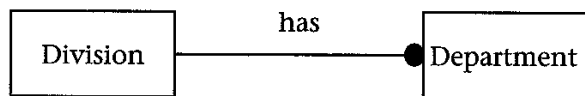
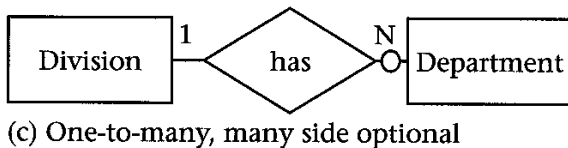
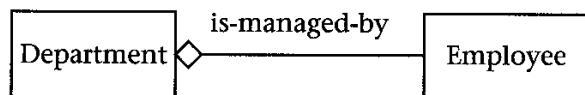
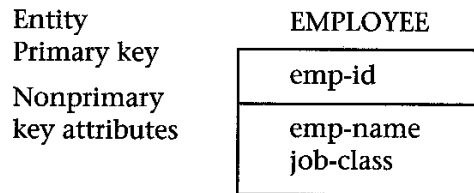
is-group-leader-of
Recursive binary relationship

Verschiedene grafische Notationen

ER model constructs using the Chen notation



ER model constructs using IDEF1X [Bruc92]



(f) Recursive binary relationship

Weitere Modellierungskonzepte

Bisher nur Basiskonzepte.

Weitere sind wichtig:

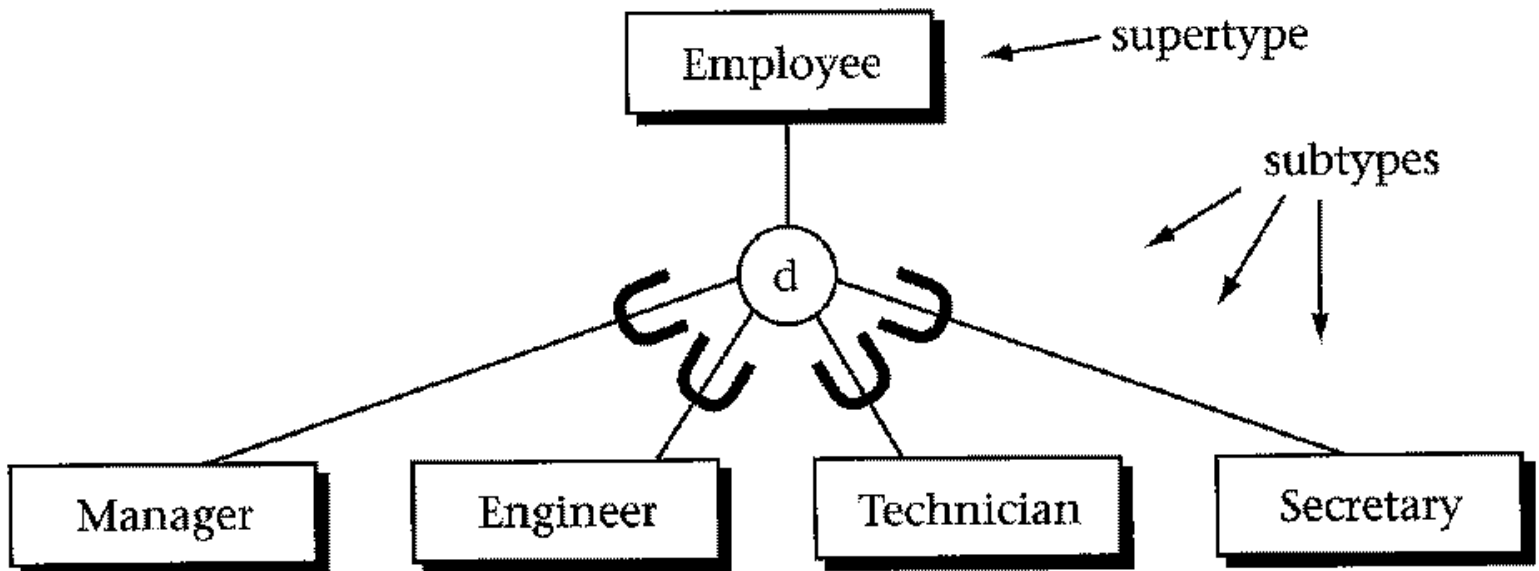
Spezialisierung/Generalisierung

Wichtige Eigenschaften zur weiteren Klassifikation einer Generalisierung

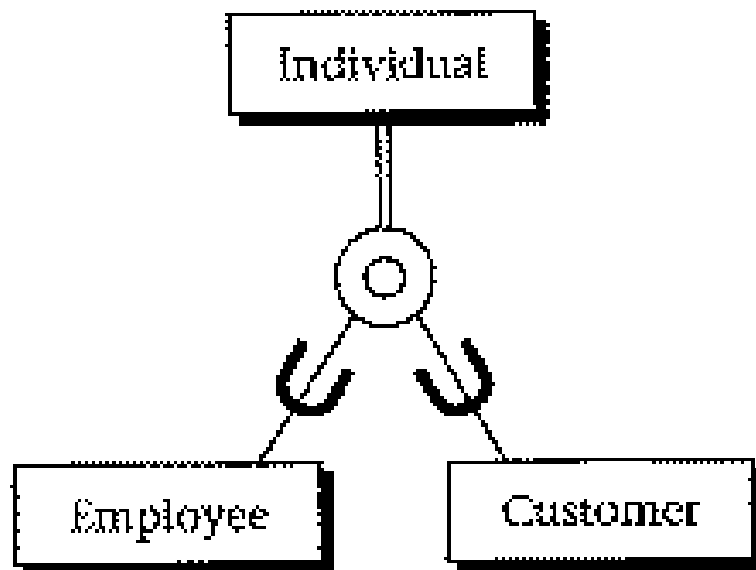
- Vollständigkeit
- Disjunktheit

Beispiel: Professoren und wissenschaftliche Mitarbeiter sind Universitätsmitarbeiter

Generalisierungen



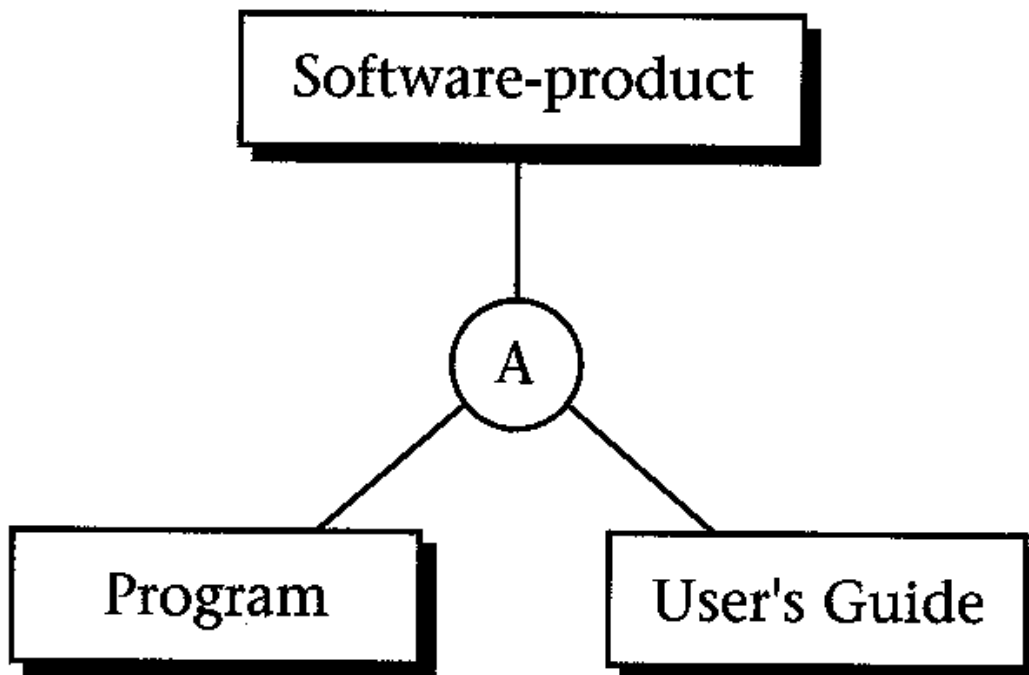
(a) Generalization with disjoint subtypes



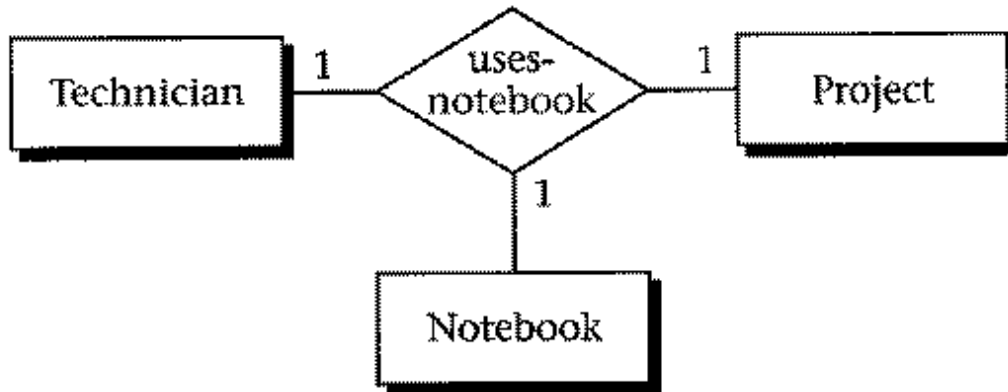
(b) Generalization with overlapping subtypes and completeness constraint

Aggregation

- auch als Part-of-Beziehung bezeichnet
- Fahrrad besteht aus Rahmen, Sattel, Lenker und Rädern
- Das Ganze ist mehr als die Summe der Teile
 - “Ganzes” ist aggregierte Entität



Dreiwertige Beziehungen und ihre Funktionalitäten



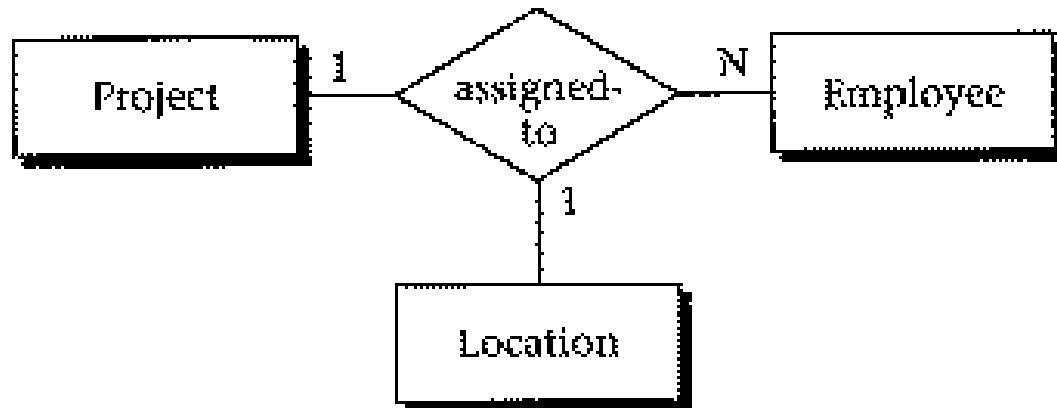
A technician uses exactly one notebook for each project. Each notebook belongs to one technician for each project. Note that a technician may still work on many projects and maintain different notebooks for different projects.

Functional dependencies

$\text{emp-id, project-name} \rightarrow \text{notebook-no}$
 $\text{emp-id, notebook-no} \rightarrow \text{project-name}$
 $\text{project-name, notebook-no} \rightarrow \text{emp-id}$

(a) one-to-one-to-one ternary relationship

Dreiwertige Beziehungen und ihre Funktionalitäten



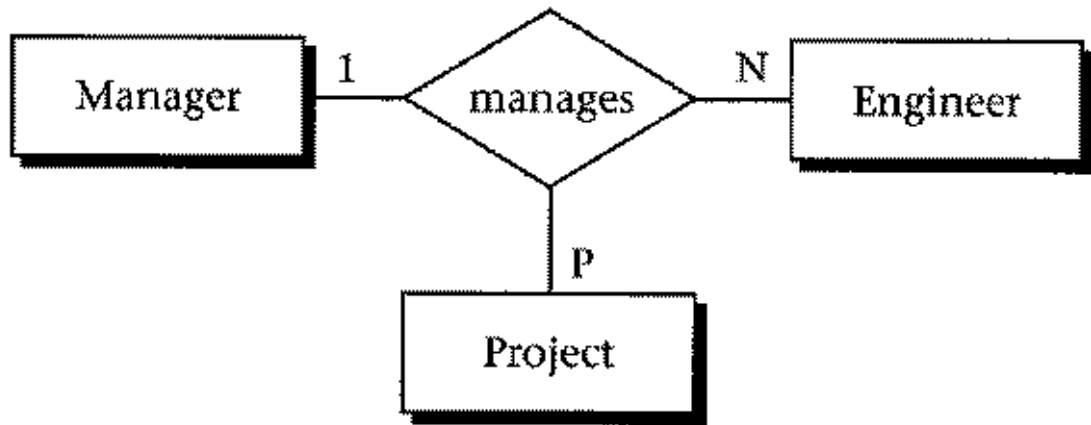
Each employee assigned to a project works at only one location for that project, but can be at different locations for different projects. At a particular location, an employee works on only one project. At a particular location, there can be many employees assigned to a given project.

Functional dependences

$\text{emp-id, loc-name} \rightarrow \text{project-name}$
 $\text{emp-id, project-name} \rightarrow \text{loc-name}$

(b) one-to-one-to-many ternary relationship

Dreiwertige Beziehungen und ihre Funktionalitäten



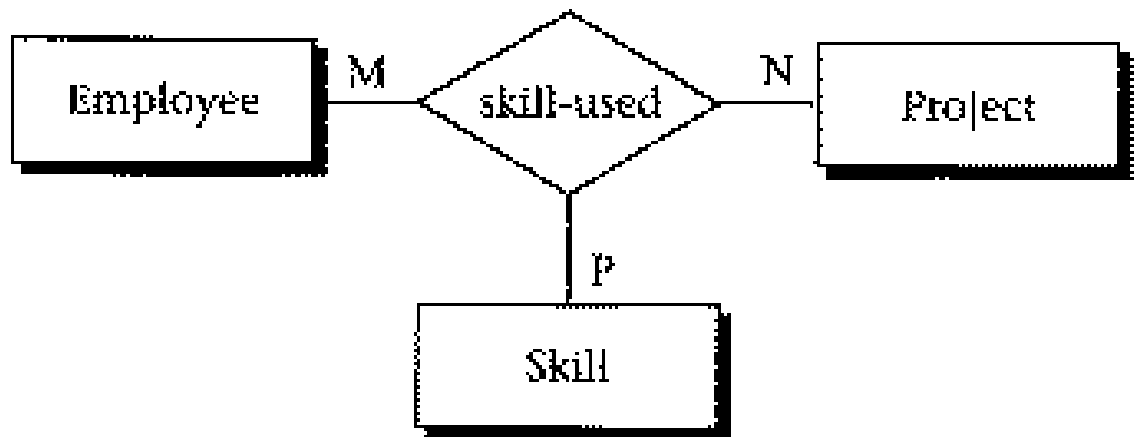
Each engineer working on a particular project has exactly one manager, but each manager of a project may manage many engineers, and each manager of an engineer may manage that engineer on many projects.

Functional dependencies

project-name, emp-id \rightarrow mgr-id

(c) one-to-many-to-many ternary relationship

Dreiwertige Beziehungen und ihre Funktionalitäten



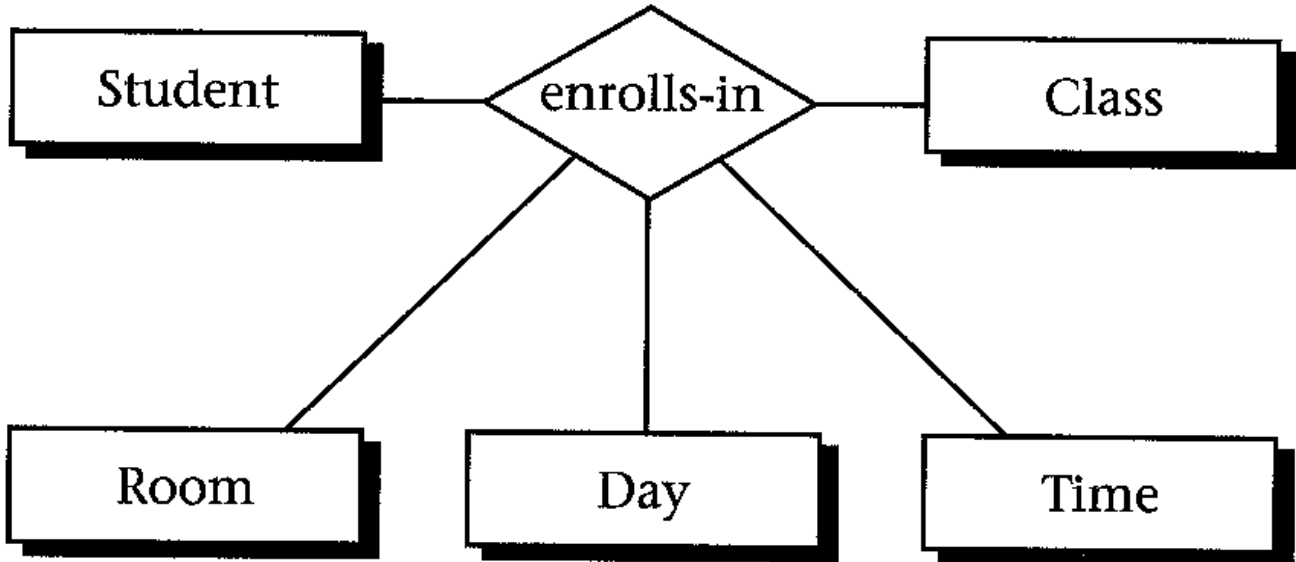
Employees can use many skills on any one of many projects, and each project has many employees with various skills.

Functional dependencies

none

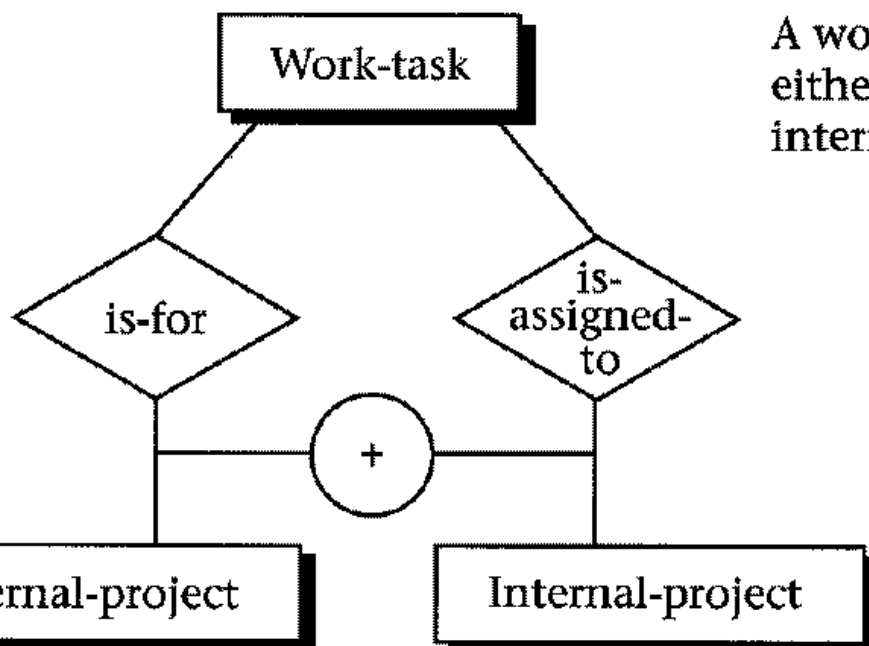
(d) many-to-many-to-many ternary relationship

N-äre Beziehung



Constraints/Zwangsbedingungen

- Oft sind besondere Bedingungen (z. B. Gesetze) einzuhalten
- Nicht jede Bedingung kann grafisch dargestellt werden
- Drei weitere Bedingungen werden jedoch oft benutzt:
 - Exklusions Constraint (Ausschluß)
 - Uniqueness Constraint (Eindeutigkeit)
 - Existence Constraint (Existenz dieser Funktionalität) (bereits gehabt)



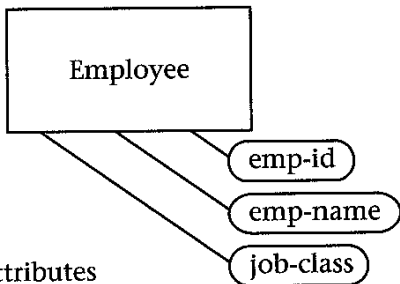
A work task can be assigned to either an external project or an internal project, but not both.

Objektorientierte Modellierung

- jetzt modern
- Konzepte nicht anders, nur anders benannt und andere Darstellung(en)

ER diagram notation

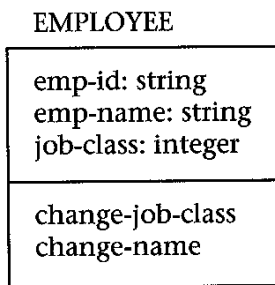
Entity, attribute (no operation)



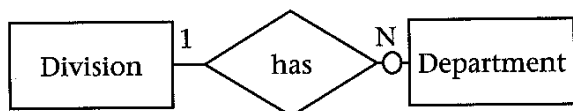
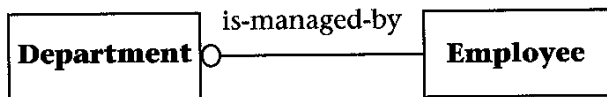
(a) Entity with attributes

Object diagram using the Rumbaugh notation [Rumb91]

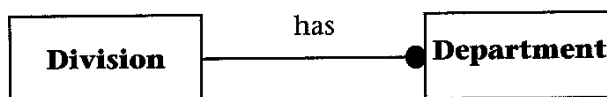
Object, attribute, operation



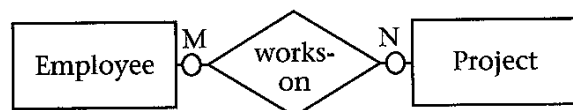
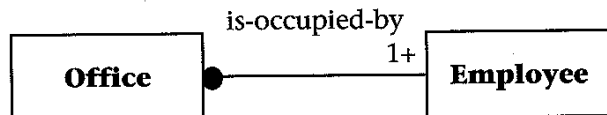
(b) One-to-one



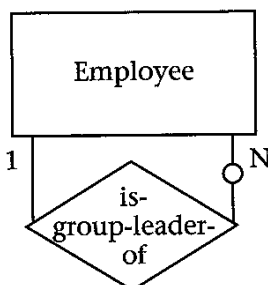
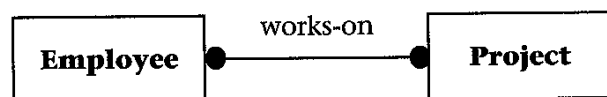
(c) One-to-many, many side optional



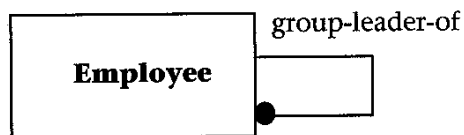
(d) One-to-many, one side optional



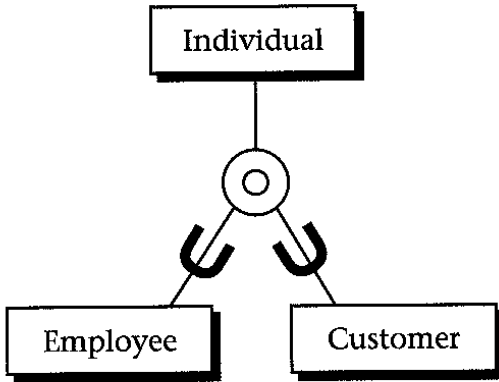
(e) Many-to-many



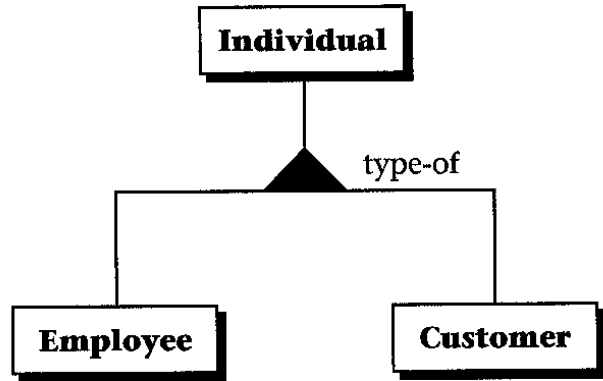
(f) Recursive binary relationship



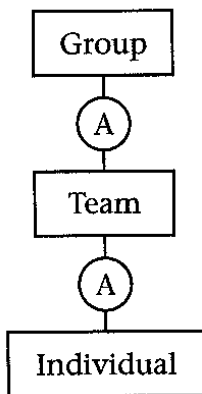
ER diagram notation



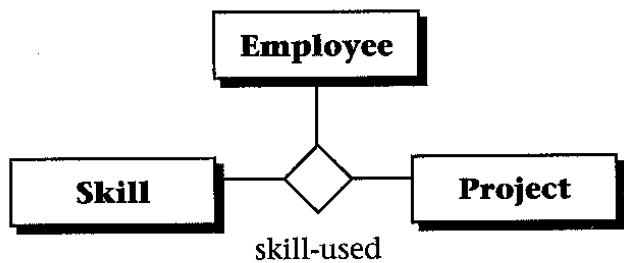
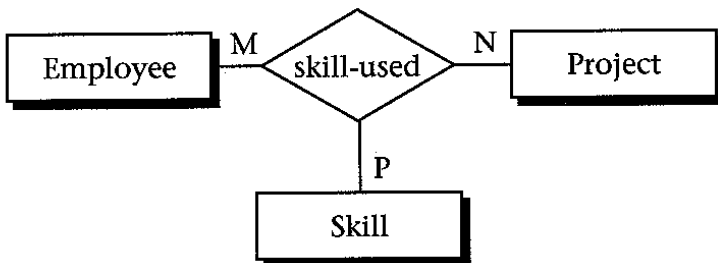
Object diagram using the Rumbaugh notation [Rumb91]



(a) Generalization ("is-a") relationship with supertype and nondisjoint subtypes



(b) Aggregation ("part-of") relationship



(c) ternary relationship

Datenbankzyklus

1. Anforderungsanalyse (requirements analysis)
2. (a) konzeptuelles Modell (ER-Modell)
(b) Sichtintegration (view integration)
3. logische Modellierung
4. physische Modellierung

ad 1: Anforderungsanalyse

- Erarbeiten der Datenanforderungen des Unternehmens in Termini primitiver Objekte
- Beschreibung der Informationen über Objekte und Beziehungen, die für die Modellierung notwendig sind
- Bestimmung der Transaktionstypen und deren Interaktion mit den Objekten
- definiere alle Leistungen, Integritäts-, Administrationsbedingungen
- spezifiziere HW + SW
- dokumentiere sorgfältig!!!, insbesondere alle Entscheidungen

ad 2 (a): konzeptuelles Modell

1. klassifiziere Entitäten und Attribute
2. identifiziere die Generalisierungshierarchie
3. definiere Beziehungen

Klassifiziere Entitäten und Attribute

- Was unterscheidet Entität von Attribut?
 1. Entitäten sollten beschreibende Information enthalten
 2. Klassifiziere mehrwertige Attribute als Entitäten
 3. Füge Attribute denjenigen Entitäten hinzu, die sie am direktesten beschreiben

Beispiele hierzu:

ad (1) Entität hat nur Identifier → Attribut

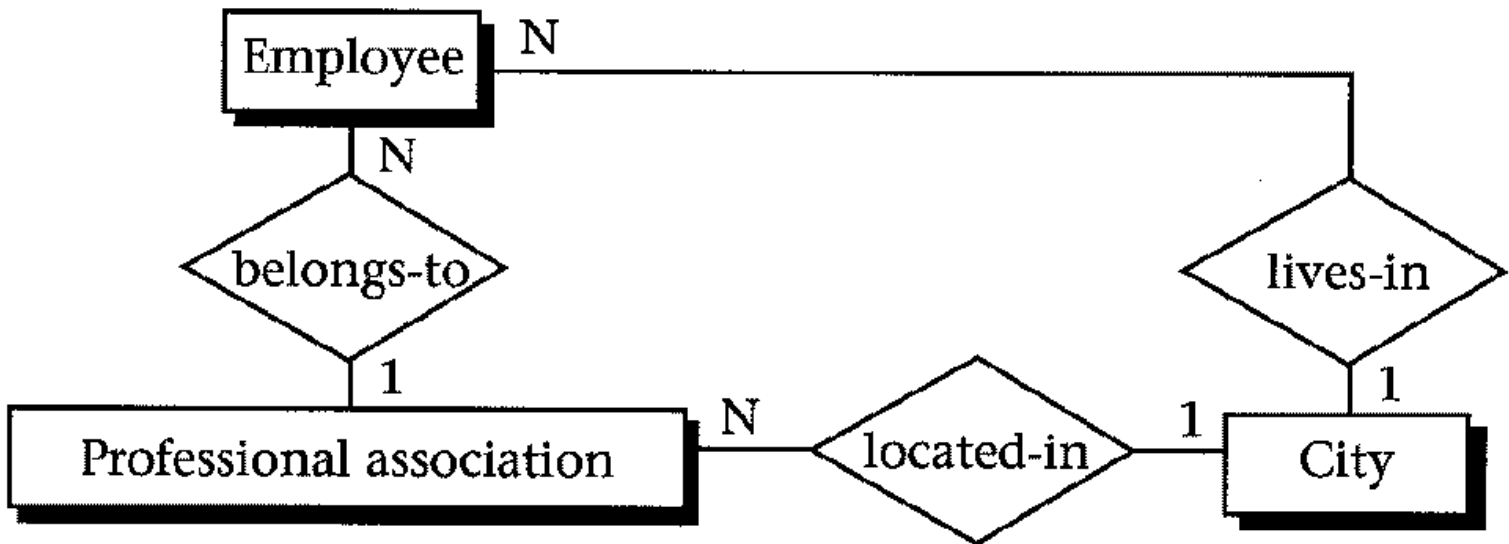
ad (2) Ein Unternehmen hat mehrere Büroräume

ad (3) Büroraumnummer sollte zu Abteilung gehören und nicht zu Angestellten

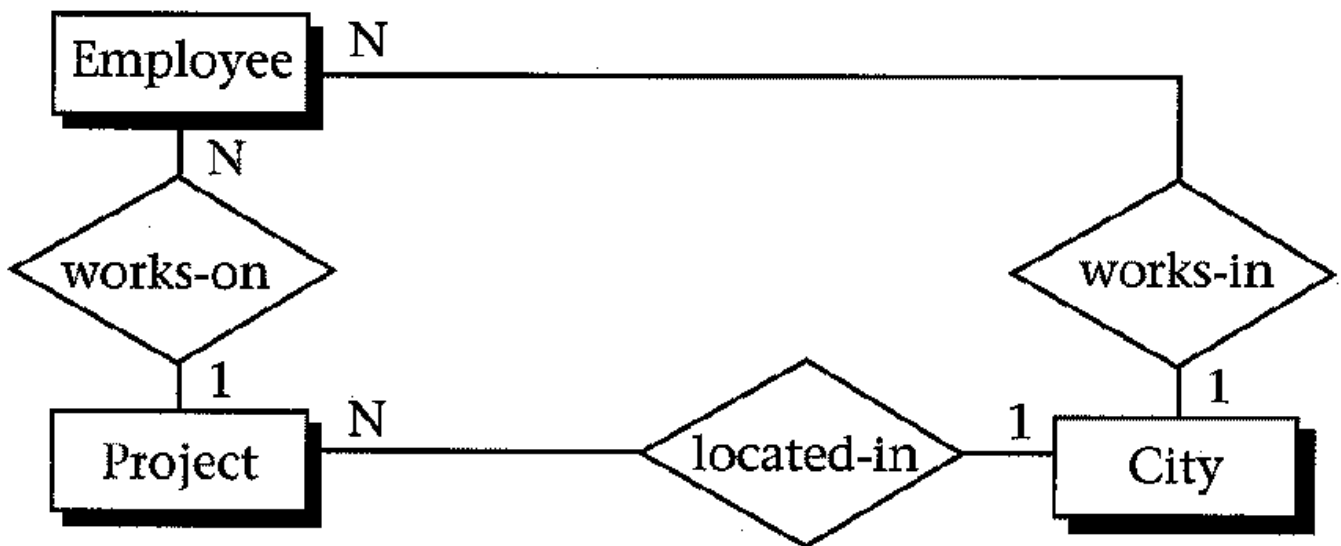
Identifiziere Generalisierungshierarchie

- über Teilmengenbeziehungen
- über Faktorisierung gemeinsamer Attribute
- Reflektiert genau die Konsequenzen der Generalisierungshierarchie:
 - Vererbung
 - Substituierbarkeit
 - Teilmengenbeziehung
- Beispiel:
 - Person mit Attr: *p-id*, *address*, *geb.-datum*
 - Angestellter mit Attr: *e-id*, *a-name*, *gehalt*
 - Kunde mit Attr: *k-id*, *k-name*, *firma*

redundante Beziehungen

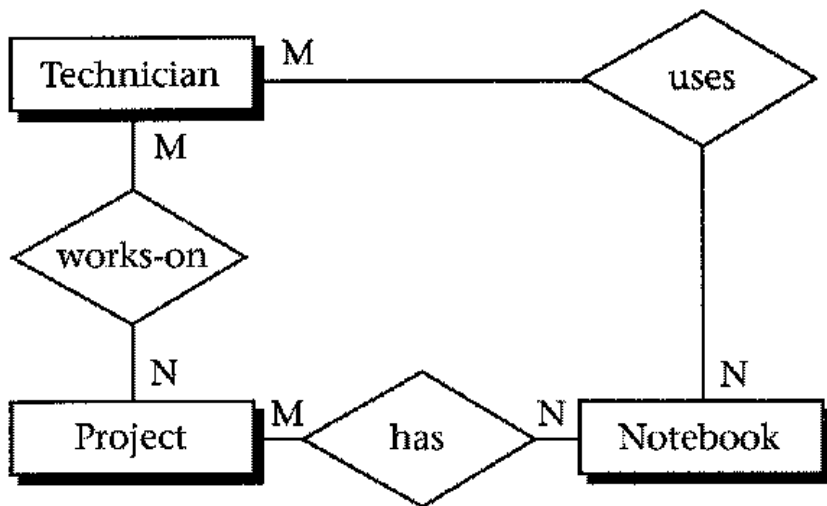


(a) Nonredundant relationships

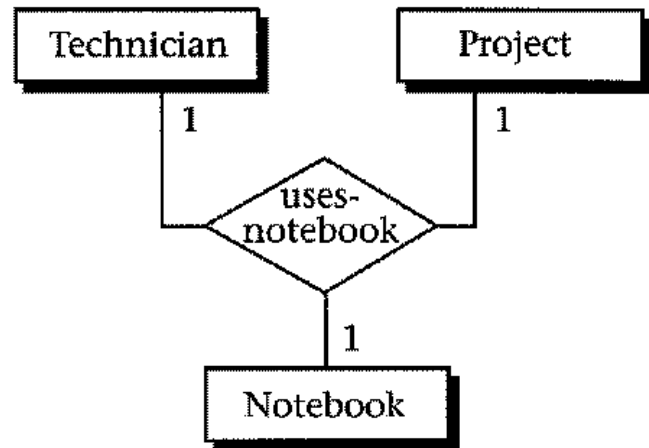


(b) Redundant relationships using transitivity

dreiwertige Beziehungen



(a) Binary relationships



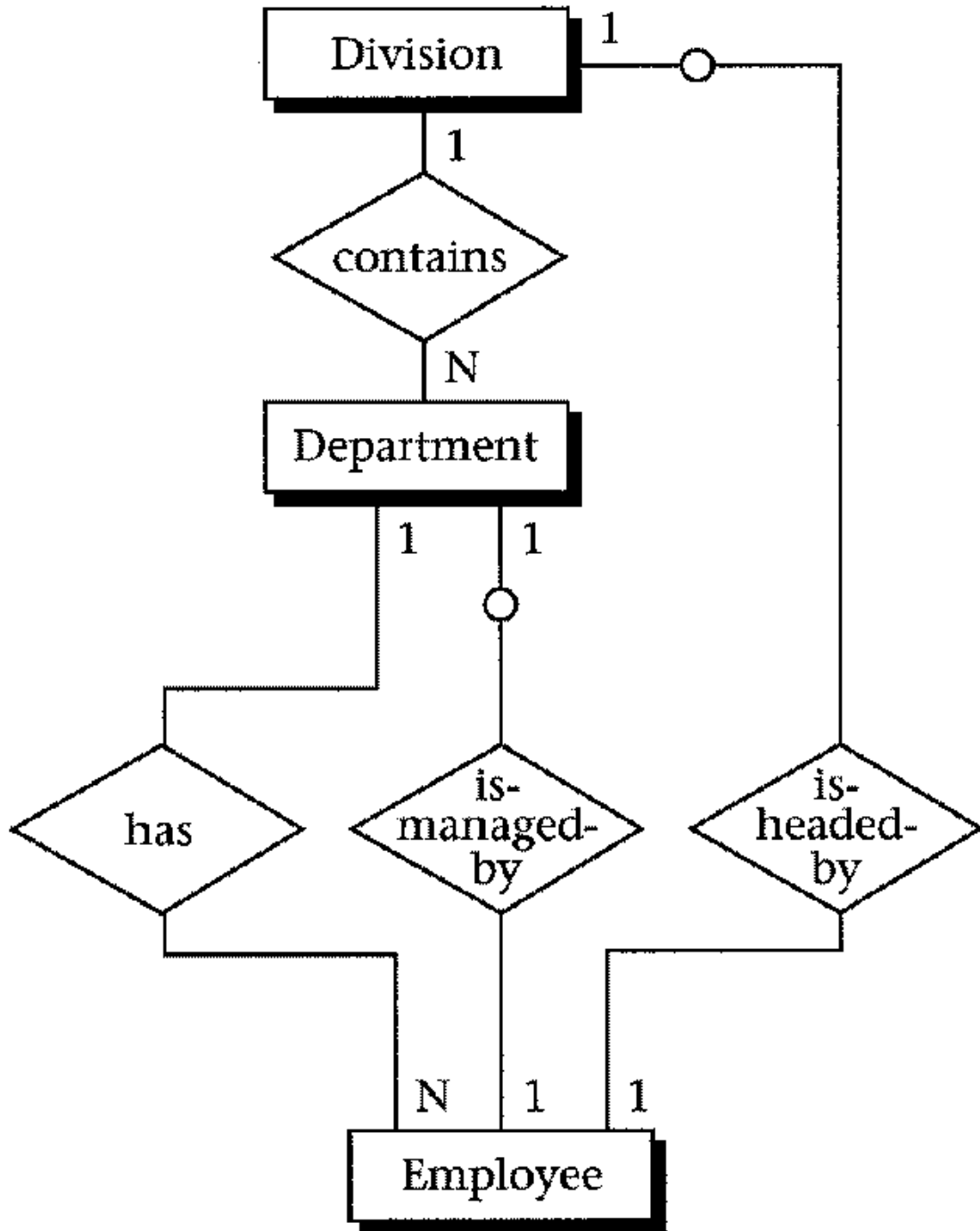
(b) Different meaning using a ternary relationship

Beispiel

Anforderungsanalyse

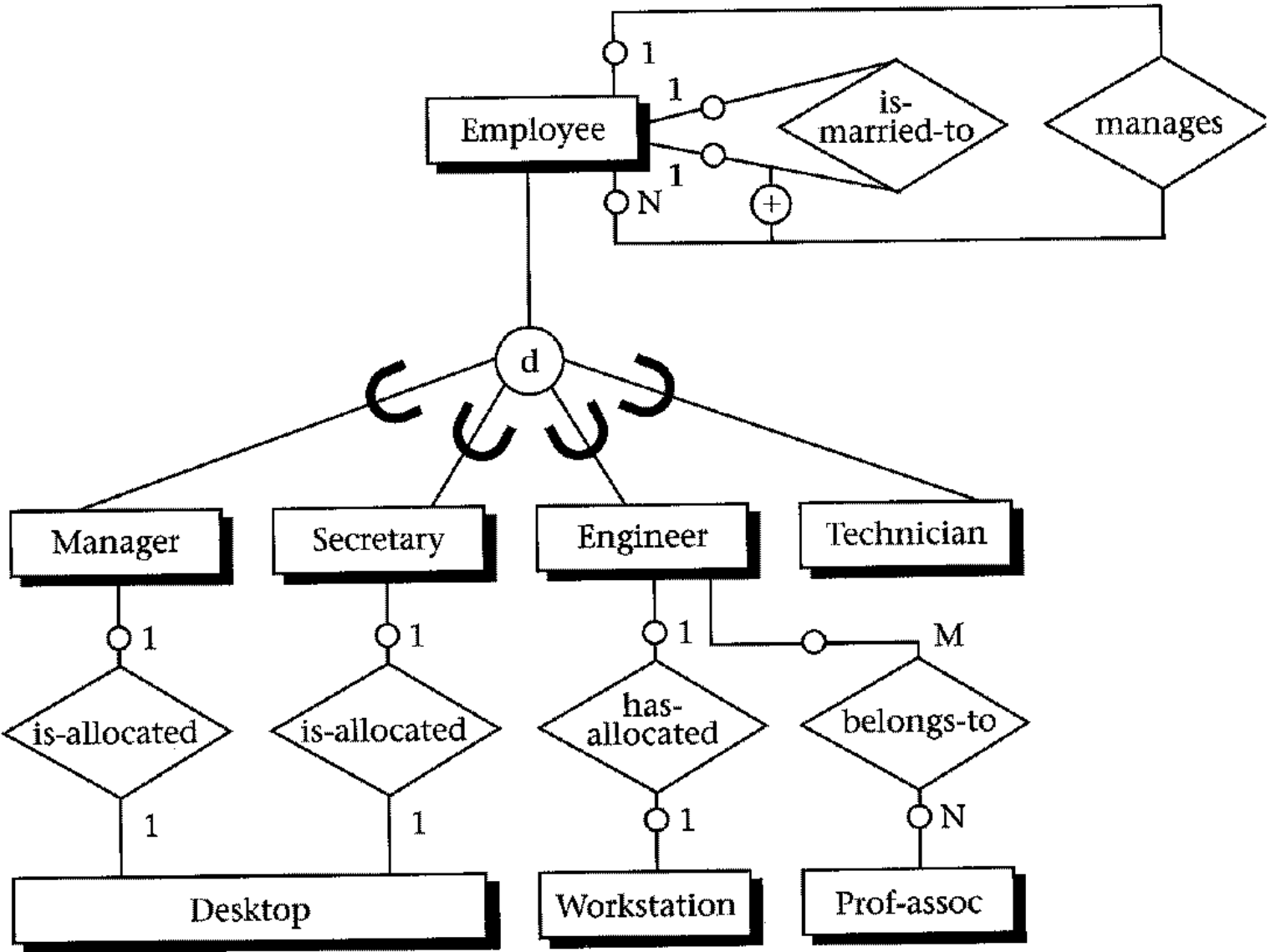
- Unternehmensweite Datenbank für Personal, Fähigkeiten, Projekte, Abteilungen, Organisationen, in denen Personal Mitglied sein kann
- Erste Sicht: Management Sicht
- Zweite Sicht: Angestellte und deren Untertypen
- Dritte Sicht: Zuweisung von Angestellten zu Projekten
- Globales Schema

Beispiel: Erste Sicht



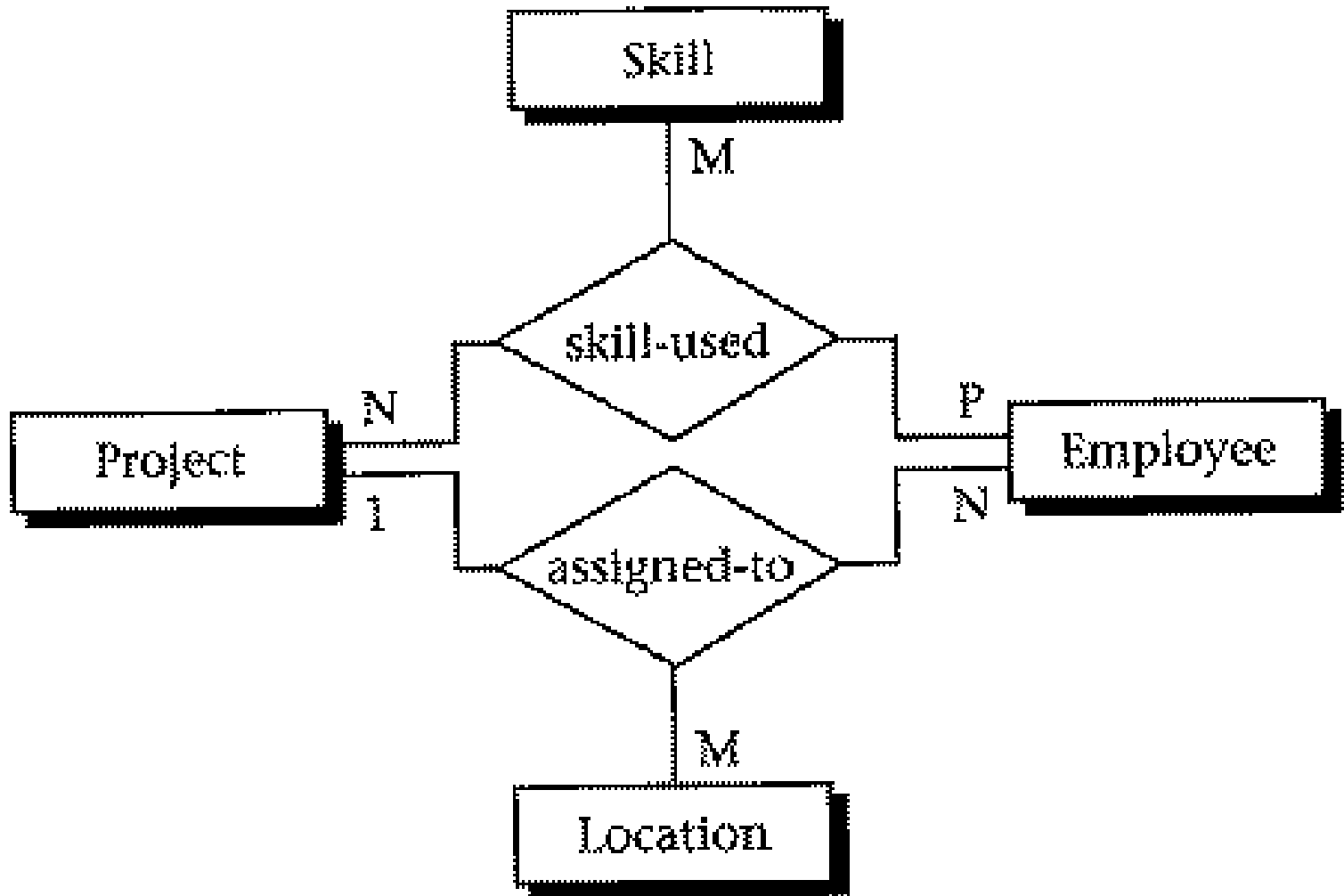
(a) Management view

Beispiel: Zweite Sicht



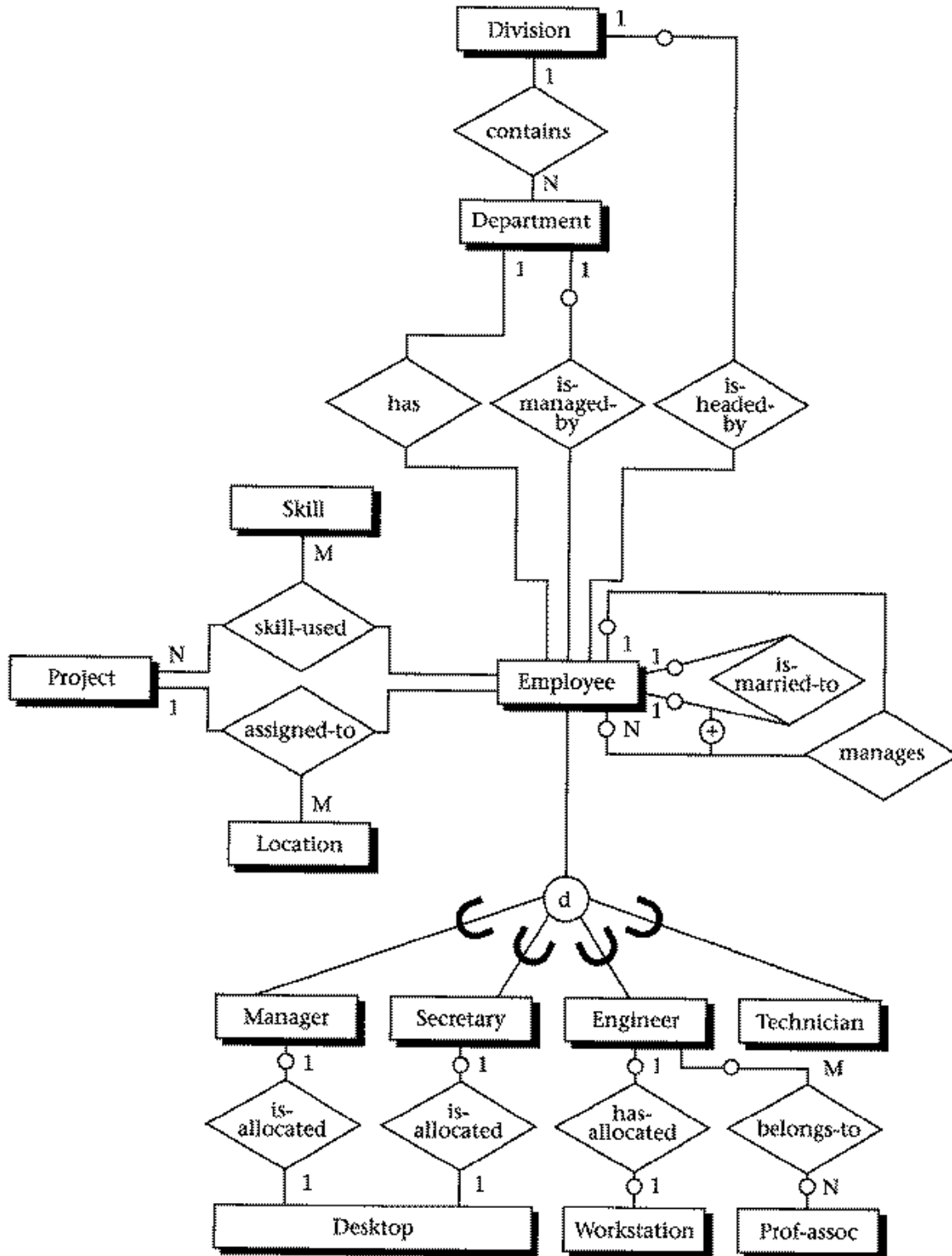
(b) Employee view

Beispiel: Dritte Sicht



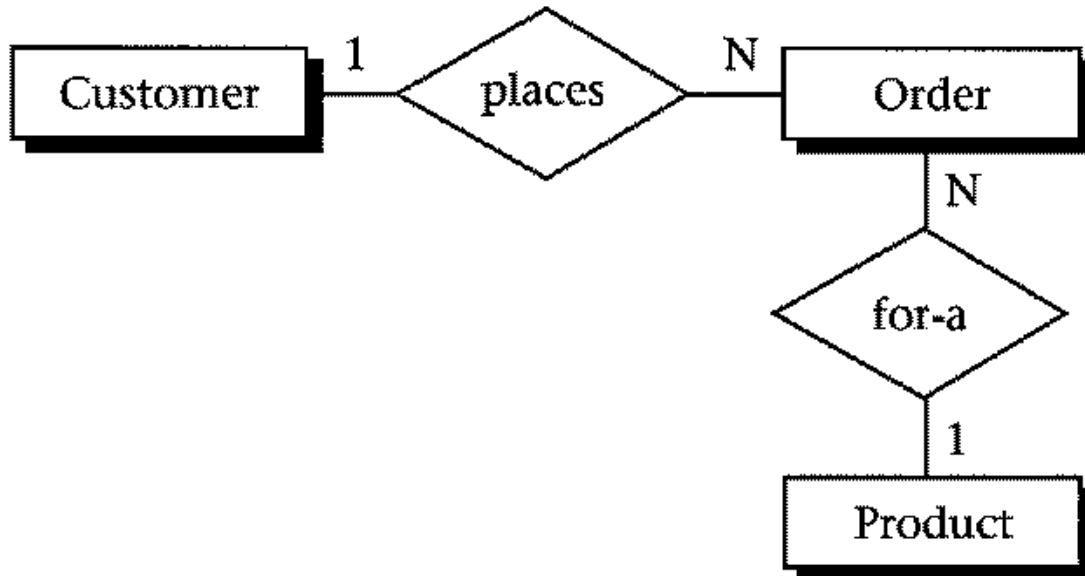
(c) Employee assignment view

Beispiel: Globale Sicht



(d) Global ER schema

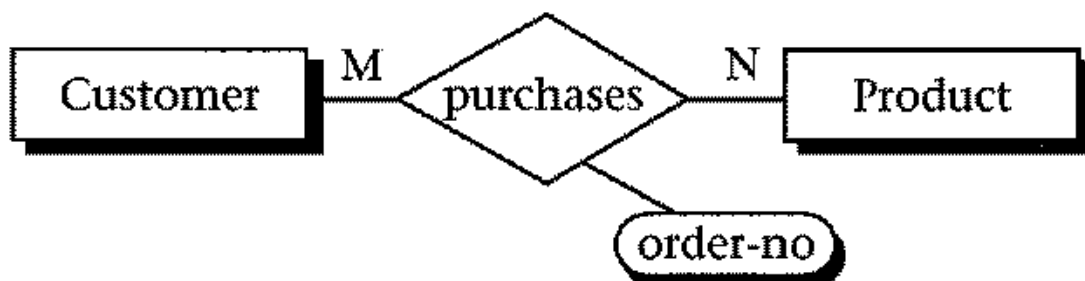
Sichtintegration



(a) The concept of order as an entity



(b) The concept of order as a relationship



(c) The concept of order as an attribute

Vier Schritte

1. Analyse der Sichten
2. Vergleich der Sichten
3. Angleichen der Sichten
4. Vereinen und Restrukturieren der Sichten

ad 1. (a) Binärer Ansatz

(b) n-ärer Ansatz

Entscheidung ist zu fällen.

ad 2. Korrespondenzen zwischen Entitäten

Erkennen von Konflikten (Schemaunterschiede)

Synonyme/Homonyme (Datenwörterbuch)

Strukturelle Differenzen/Konflikte

- Typkonflikte
- Abhängigkeitskonflikte (verschiedene Kardinalitäten)
- Schlüsselkonflikte (SSNO, EmpNO)
- Verhaltenskonflikte (verschiedene Integritätsbedingungen, $=/\neq$ Null)

Angleichen/Auflösen der Konflikte

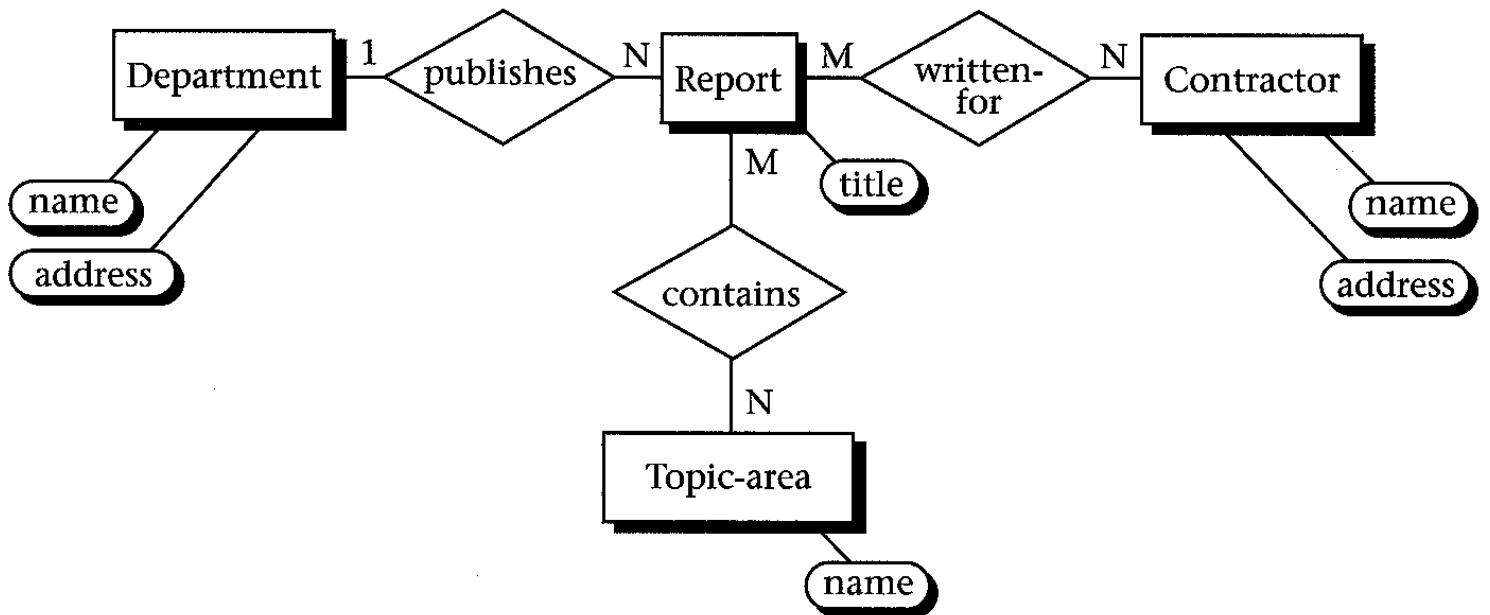
Mittel:

- Umbenennungen
- Generalisierungen
- Restrukturierungen

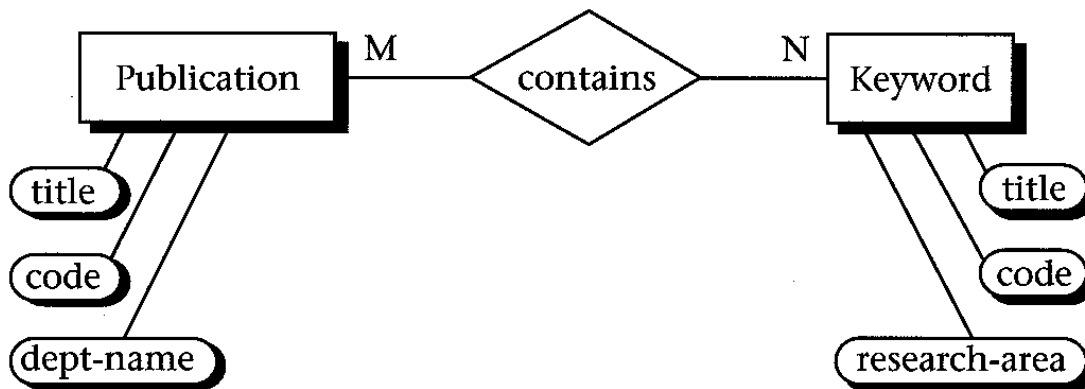
Vereinen und Restrukturierung

Ziele: Vollständigkeit, Minimalität (Entfernen der Redundanzen), Lesbarkeit (gut für Benutzer), überlappende Entitäten, Einführen Generalisierung

Beispiel

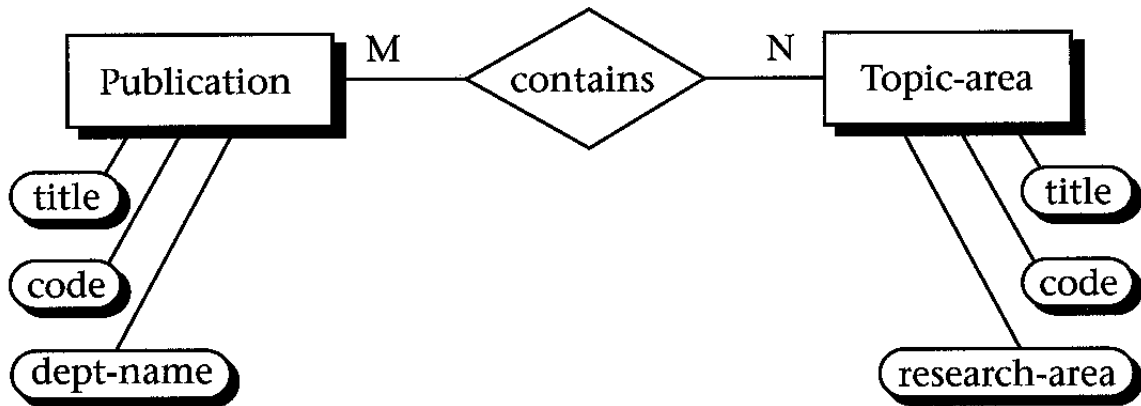


(a) Original schema 1, focused on reports

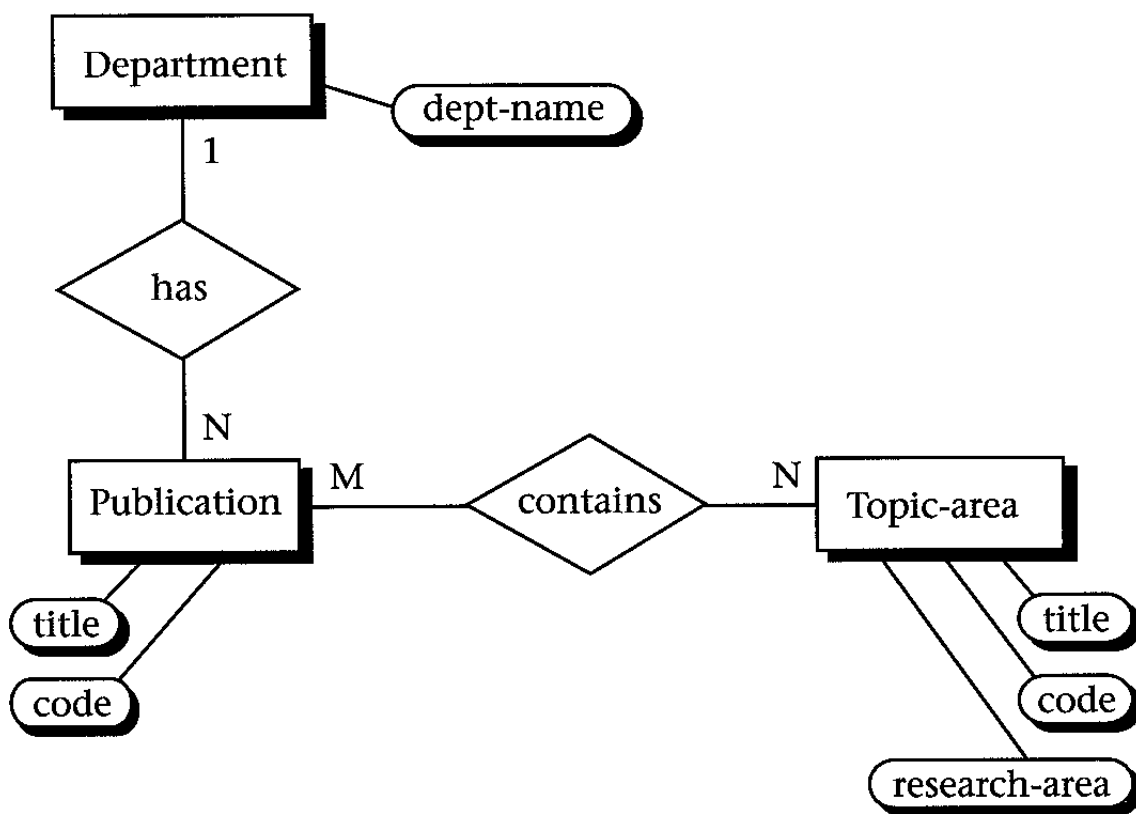


(b) Original schema 2, focused on publications

Beispiel

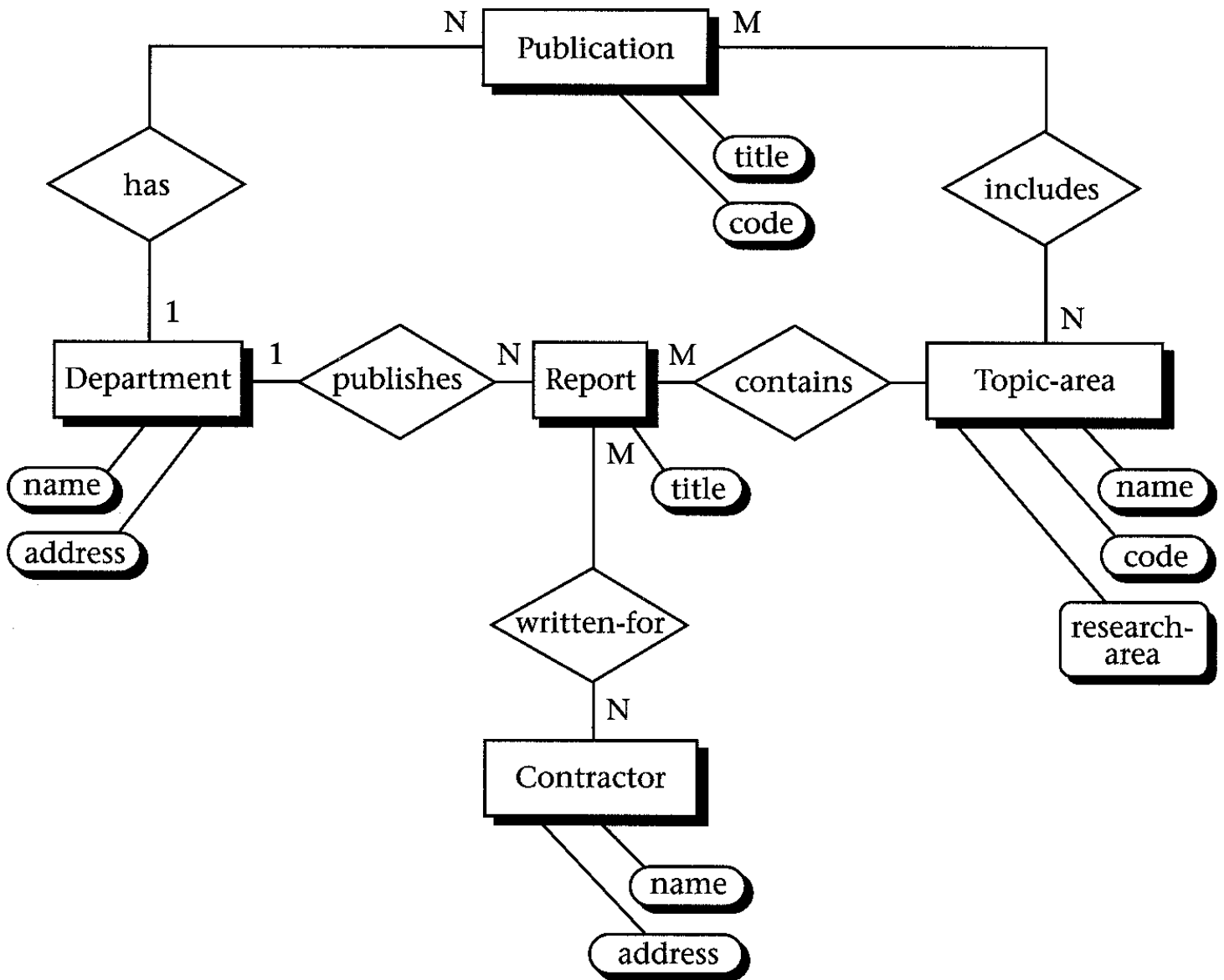


(a) Schema 2.1, in which Keyword has changed to Topic-area



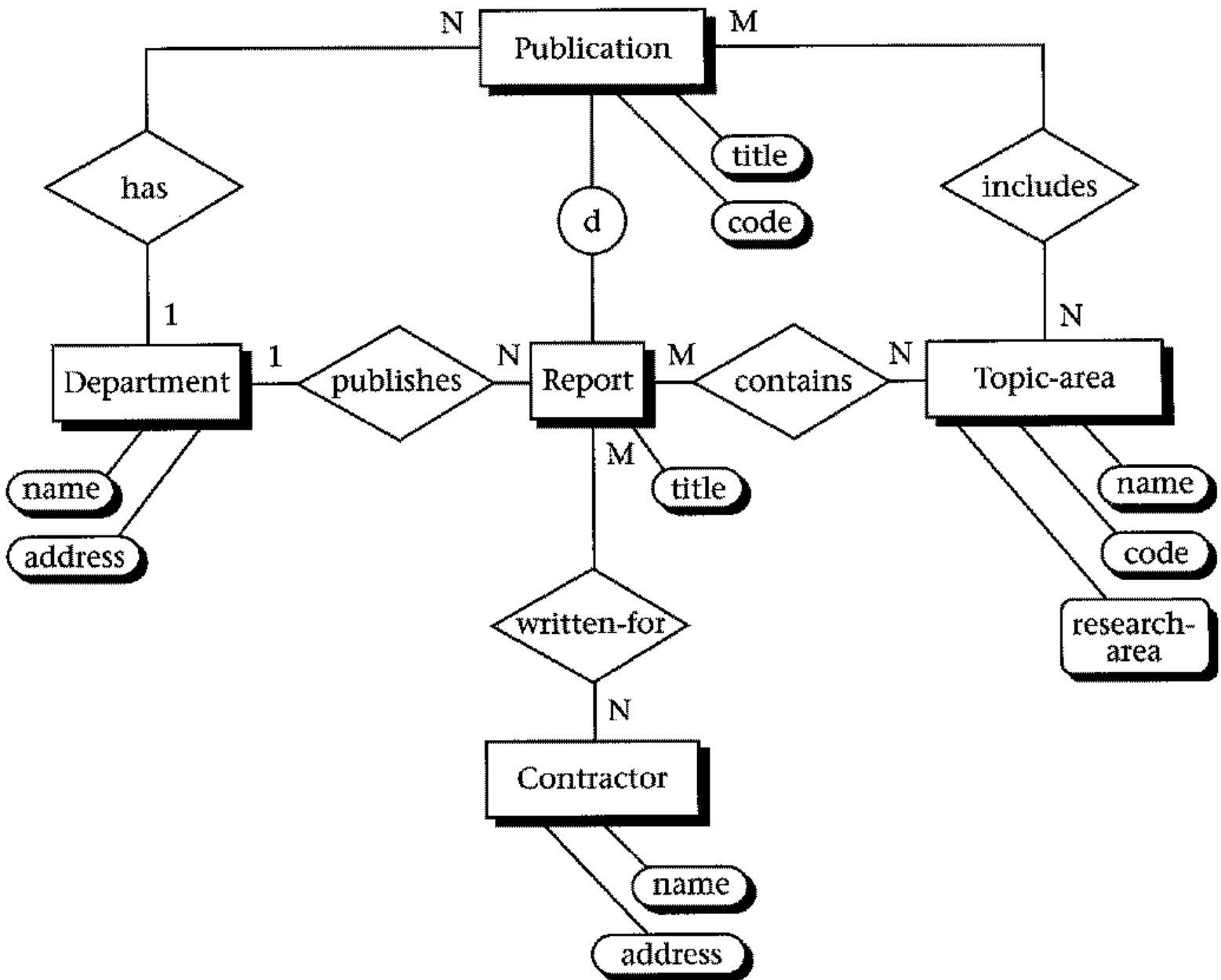
(b) Schema 2.2, in which the attribute dept-name has changed to an attribute and an entity

Beispiel

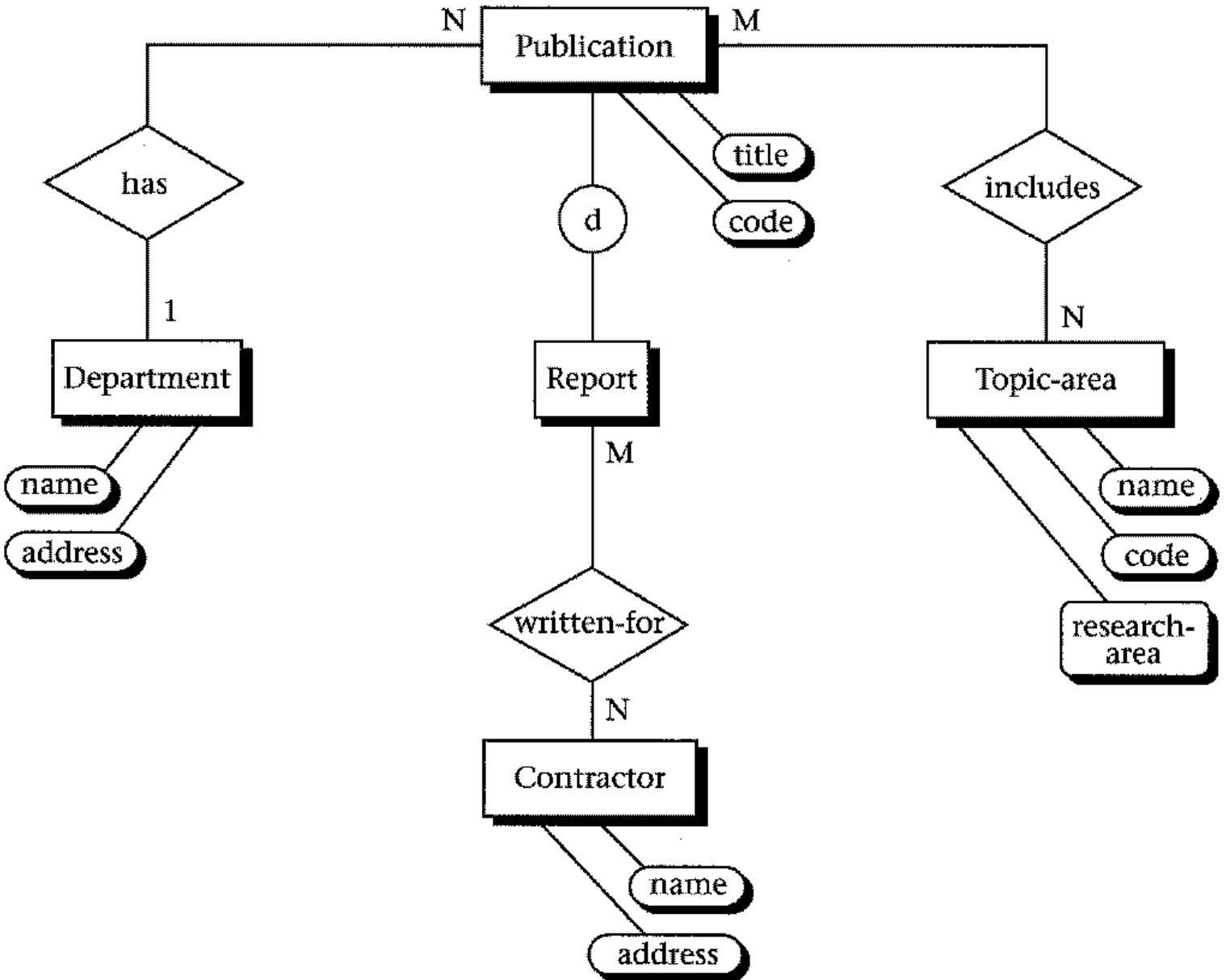


(a) Schema 3, the result of merging schemas 1 and 2.2

Beispiel



(b) Schema 3.1: creation of a generalization relationship

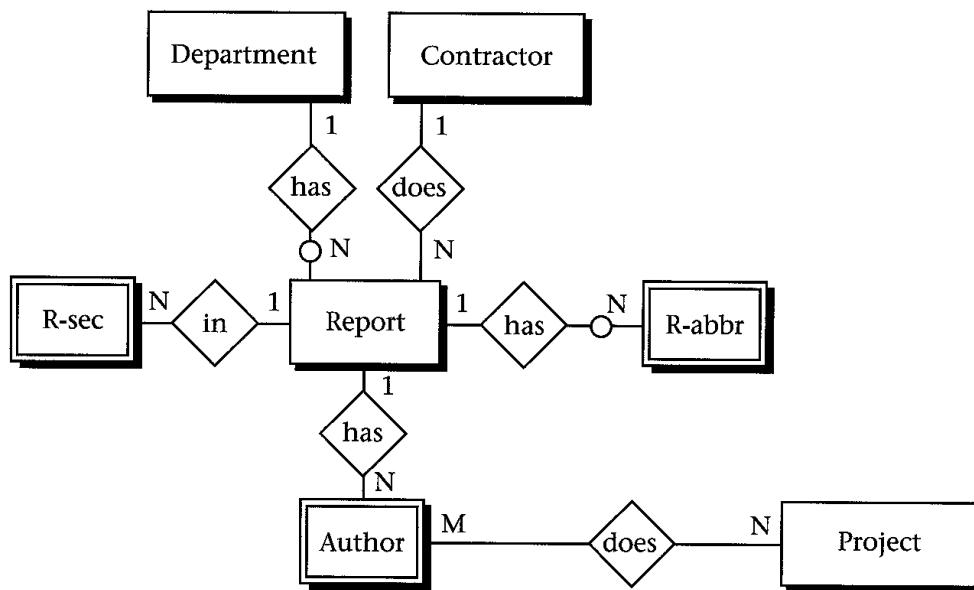


(c) Schema 3.2: elimination of redundancy

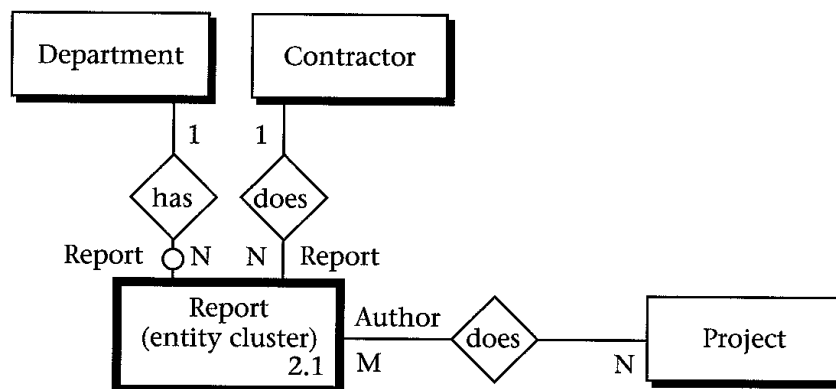
Entity Clustering

Ziel: Mehr Abstraktion → Übersichtlichkeit durch Gruppierung von Entitäten und Beziehungen

Bottom-up-Abstraktion



(a) ER model before clustering



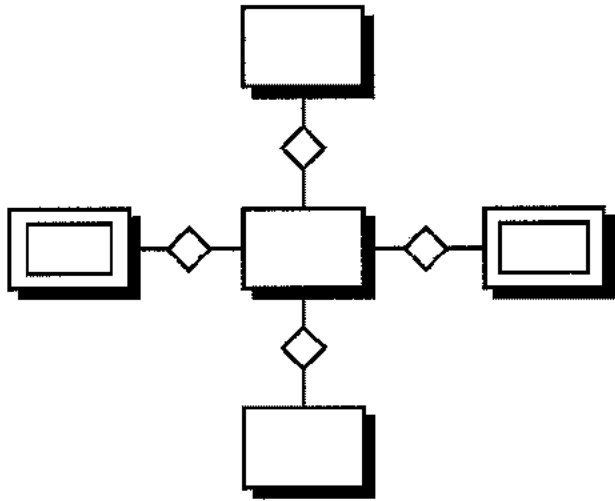
(b) ER model after clustering

Name des Clusters: dominante Entität

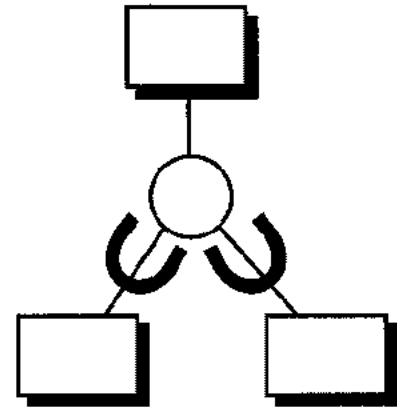
Gruppierungsabstraktion

- Dominanzgruppierung: dominantes Objekt mit weniger dominanten Objekten zusammenfassen, z. B. schwache Entitäten
- Abstraktionsgruppierung: Zusammenfassen von Generalisierungshierarchien
- Constraintgruppierung: Existenzbeziehungen (NIAM)
- Beziehungsgruppierung: drei- oder mehrstellige Beziehungen mögliche Gruppierungen

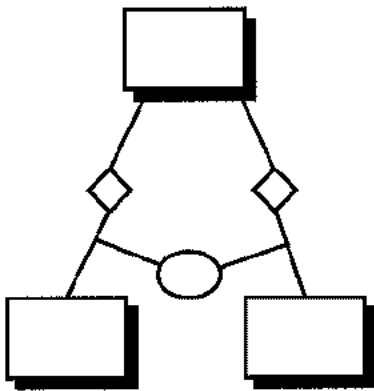
Bsp: Gruppierungsabstraktionen



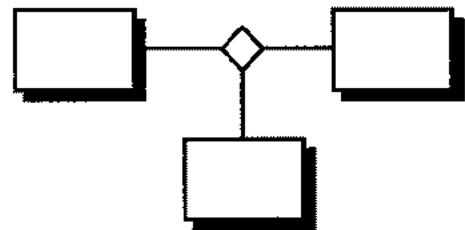
(a) Dominance grouping



(b) Abstraction grouping



(c) Constraint grouping

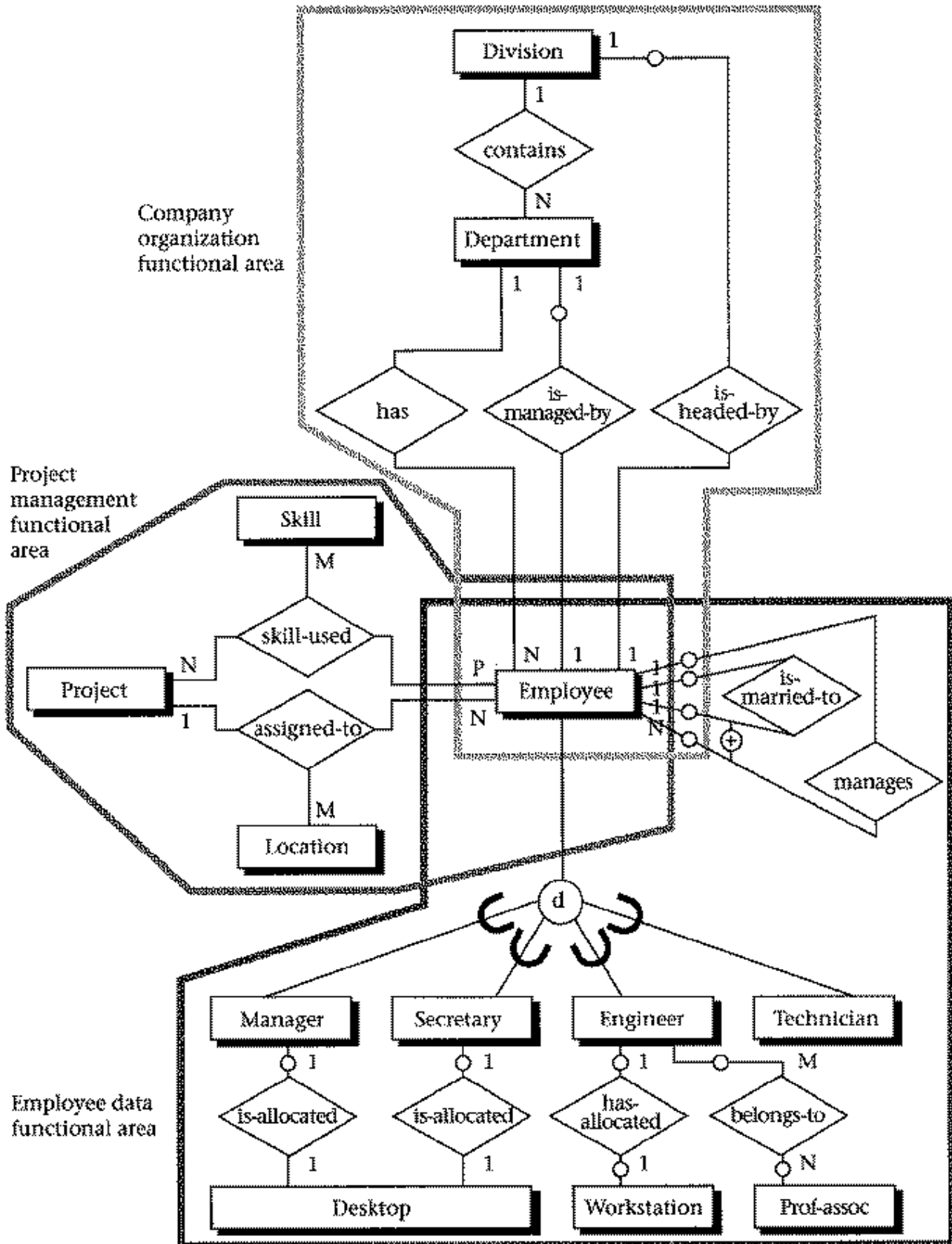


(d) Relationship grouping

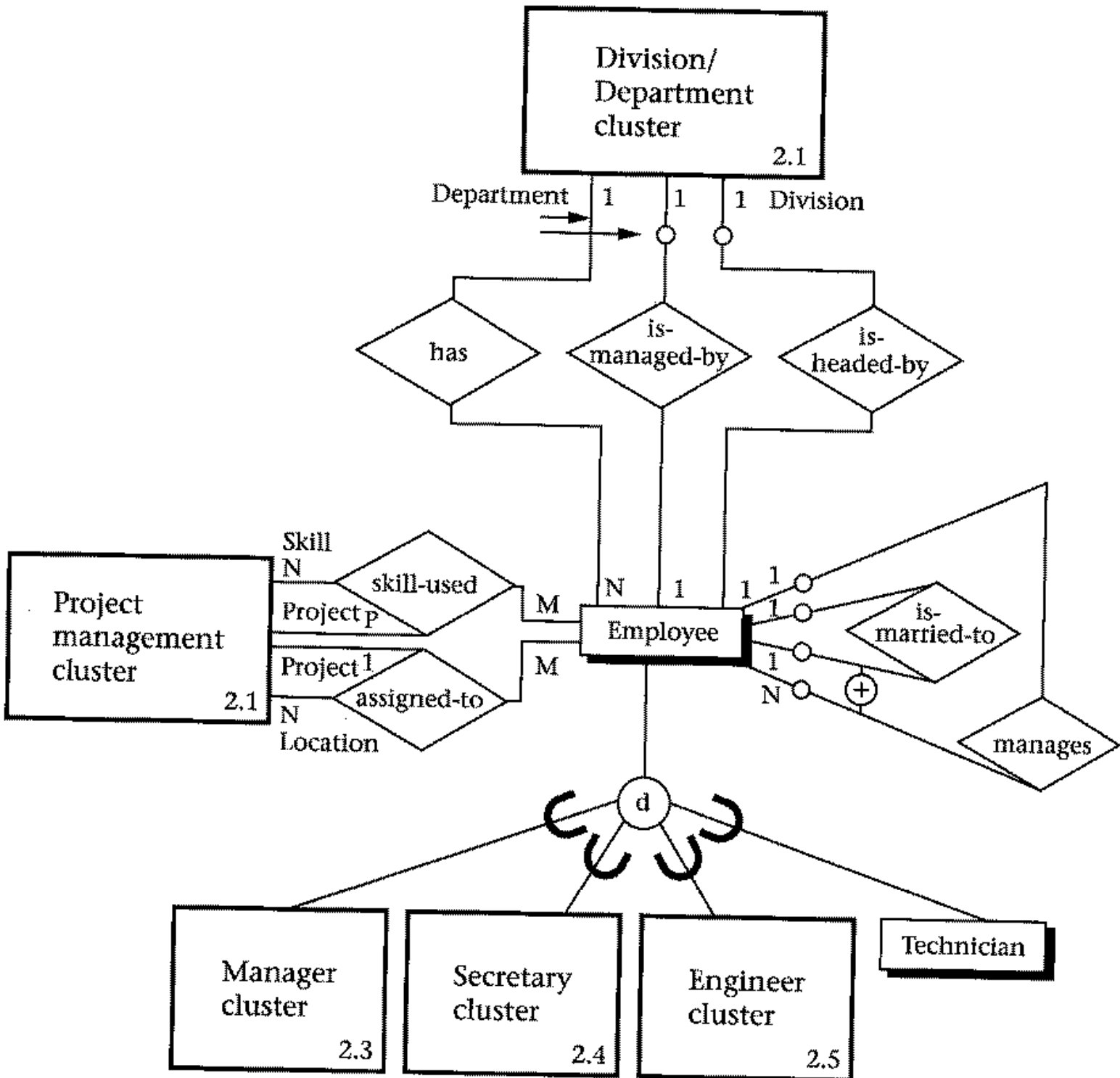
Clustering-Technik

1. Bestimmung funktionaler Gebiete
2. Gruppierung funktionaler Gebiete
3. Über mehrere Ebenen
4. Validierung

Globales ER-Schema mit funktionalen Einheiten



Entity Cluster der 2. Ebene



Vorteile ER

- Kommunikationshilfe
- relativ formal
- enthält viele Informationen
- stellt wichtige semantische Konstrukte zur Verfügung
- es gibt semi-automatisches Verfahren, um aus ER-Modell relationales Schema zu gewinnen