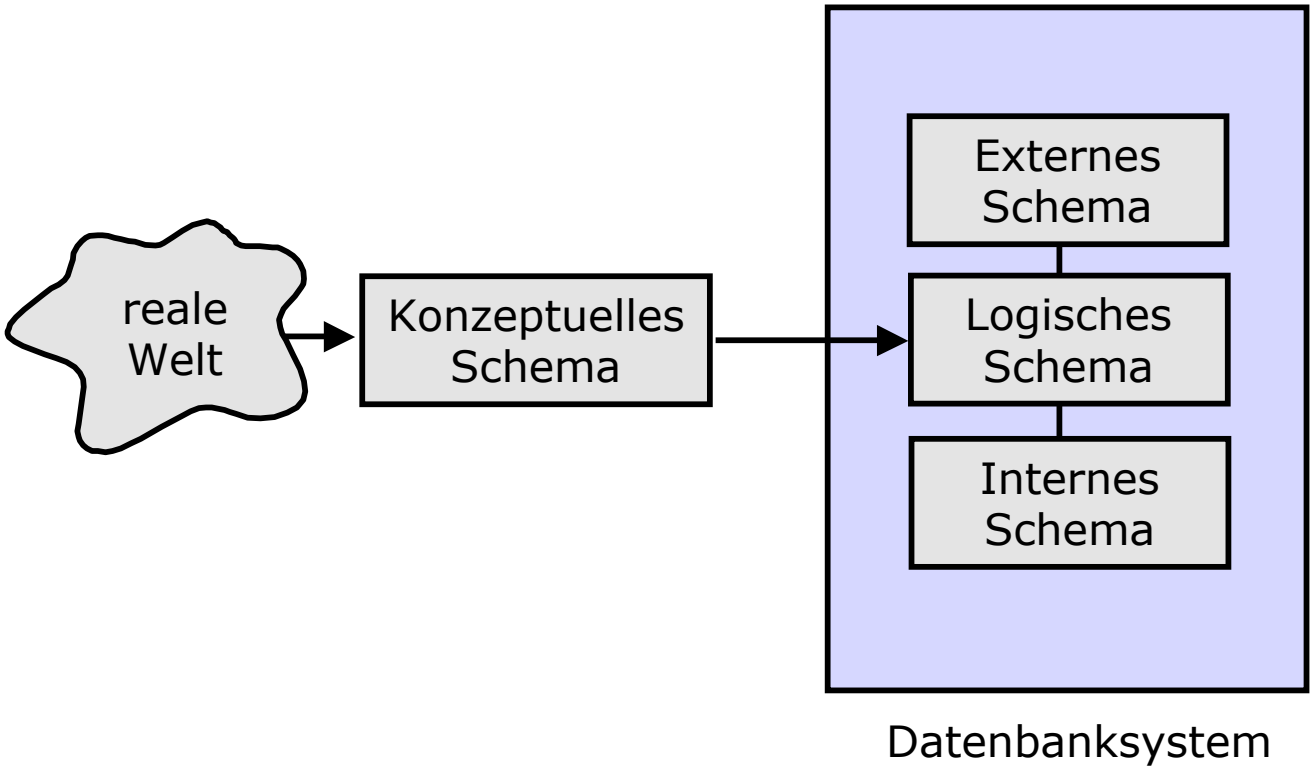
Datenbanksysteme SS 2007

Frank Köster
(Oliver Vornberger)

Institut für Informatik
Universität Osnabrück

Kapitel 3:
Logische Datenmodelle

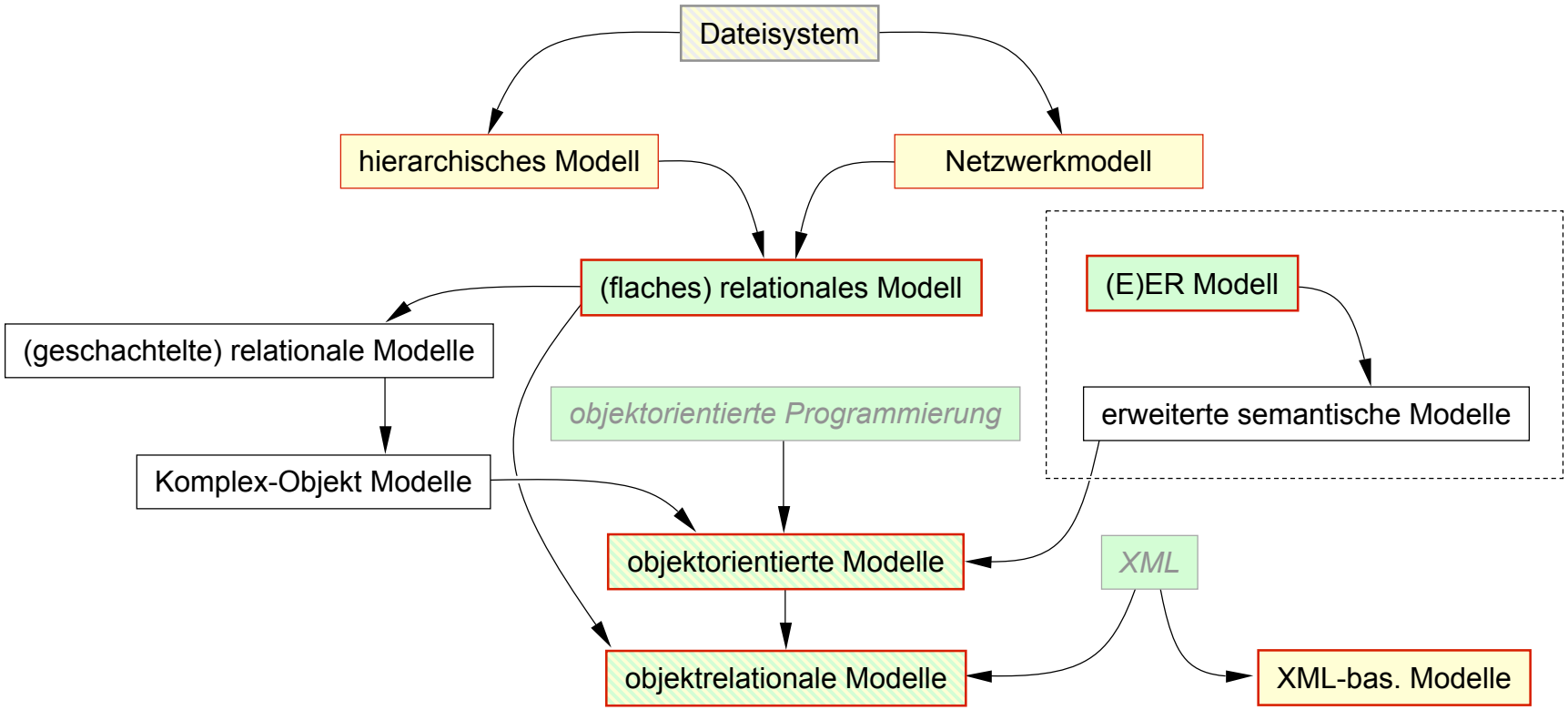
Modellierungskonzepte



Logische/konzeptuelle Datenmodelle

- ER-Modell
- hierarchische Modell
- Netzwerkmodell
- (flache) relationale Modell
- geschachtelte relationale Modell
- Komplex-Objekt Modell
- objektorientierte Modell
- objektrelationale Modell
- XML-basierte Modell

Logische/konzeptuelle Datenmodelle



Hierarchisches Modell

Das hierarchische Modell gilt als das älteste klassische Datenmodell.

- Ein Datensatz und alle hierarchisch von ihm abhängigen Datensätze werden darin als eine Einheit betrachtet, wobei die Gliederung eines Dokuments in Kapitel, Unterkapitel und Abschnitte ein anschauliches Beispiel hierfür ist.
- Bei der praktischen Anwendung können natürliche Hierarchien i.d.R. direkt abgebildet werden:

  Assignment (Course → Offering → Student)

- Als künstliche Hierarchien werden solche hierarchischen Zusammenhänge bezeichnet, die keine natürliche Entsprechung in der Miniwelt haben:

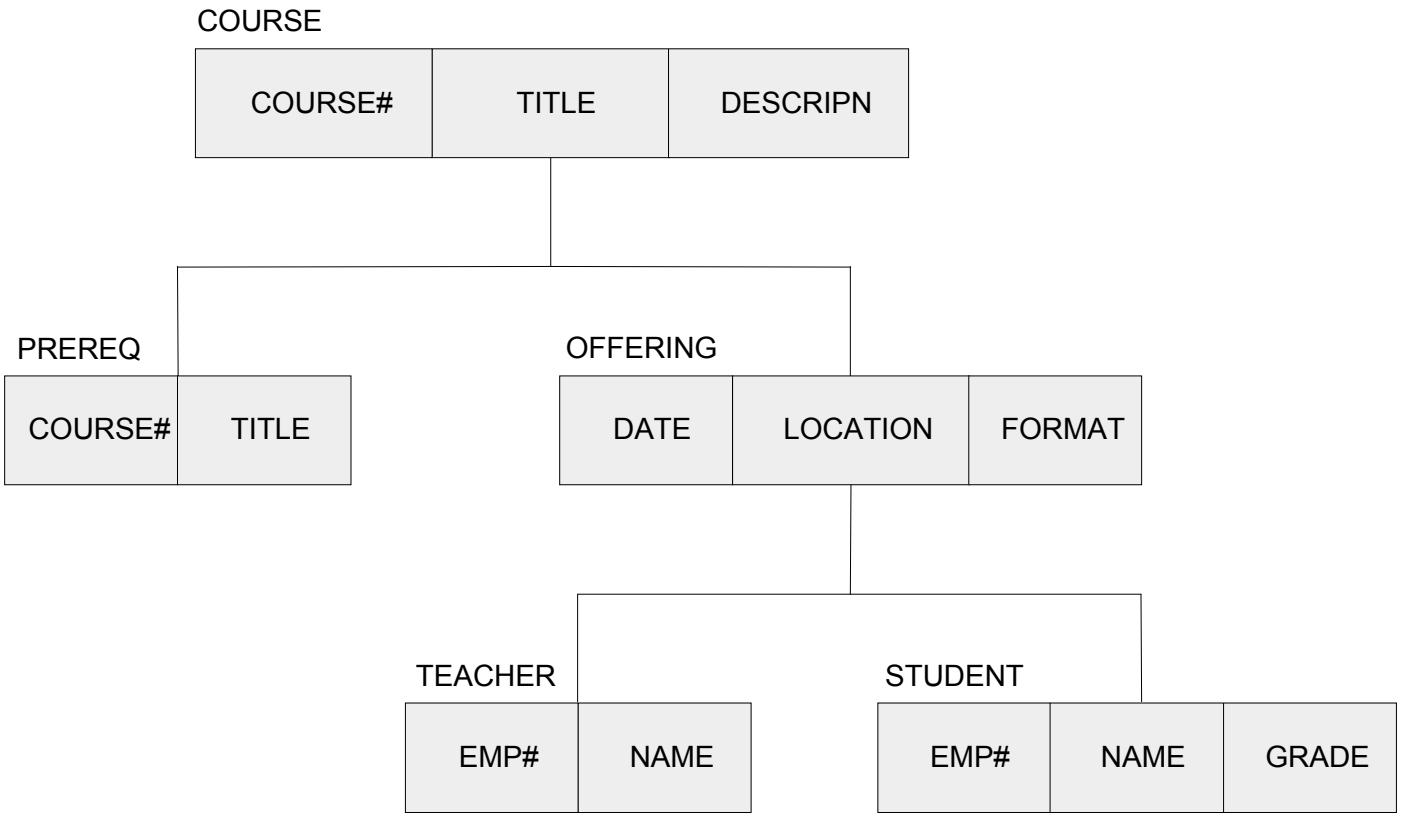
  Supplier (Product → Source → Address)

Hierarchisches Modell

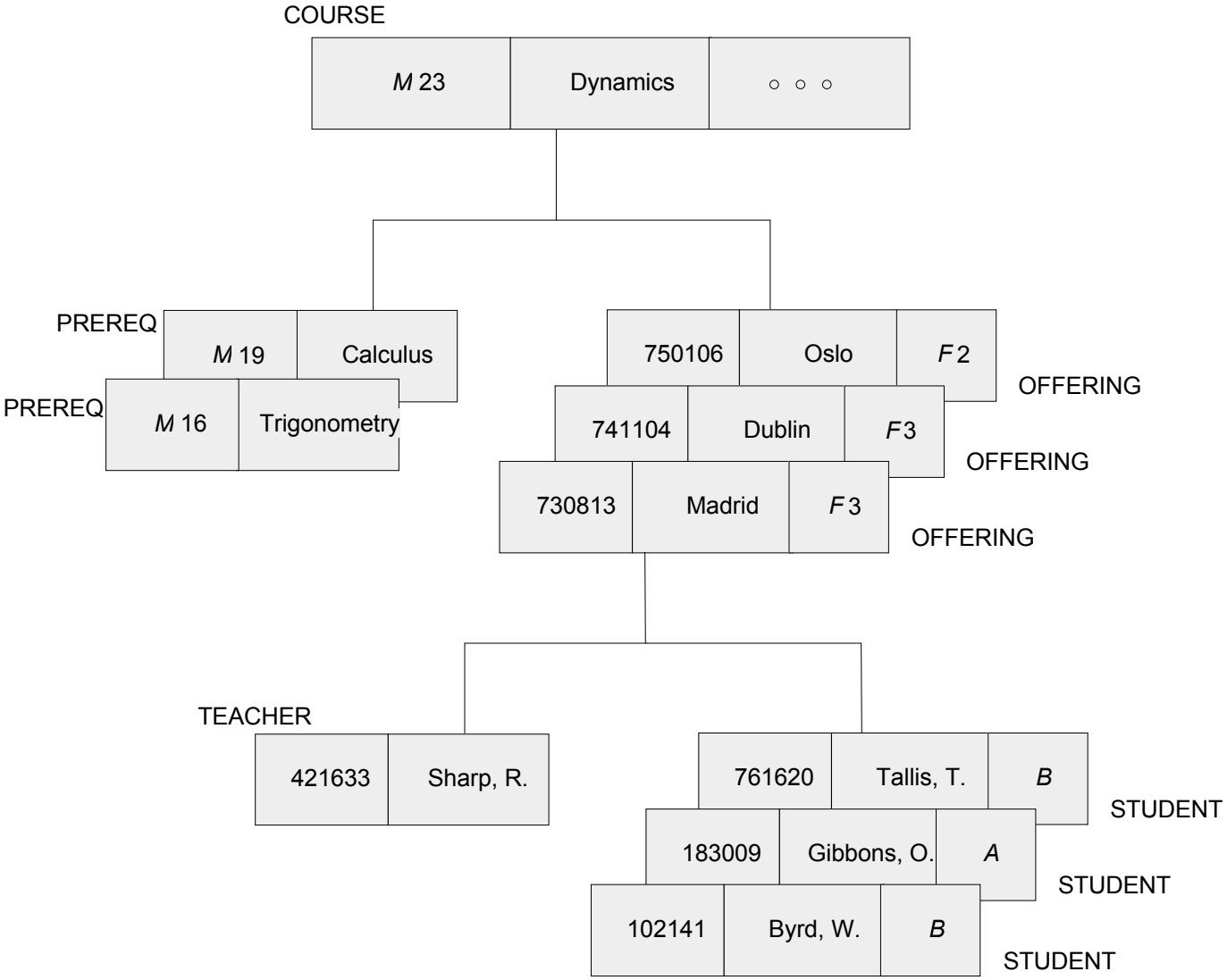
Bzgl. der Darstellung von Beziehungen ist das hierarchische Datenmodell limitiert, da folgende Randbedingungen gelten:

- In einstufigen Hierarchien werden genau einem Elternelement ein oder mehrere Kinder zugeordnet.
- In mehrstufigen Hierarchien werden mehrere Gruppen (dies sind selbst Hierarchien) in Beziehung gesetzt, wobei
  - jedes Element nur in einer Gruppe Kindelement ist;
  - das Wurzelelement selbst nicht Kindelement ist – keine Kreisbezüge.
- Folgendes kann deshalb für Beziehungen im hierarchischen Modell festgestellt werden:
  - 1 : 1 - Elternelement wird Kindelement zuordnet
  - 1 : n - Elternelement werden mehrere Kindelemente zugeordnet
  - m : n - nicht darstellbar.

Hierarchisches Modell – Schema

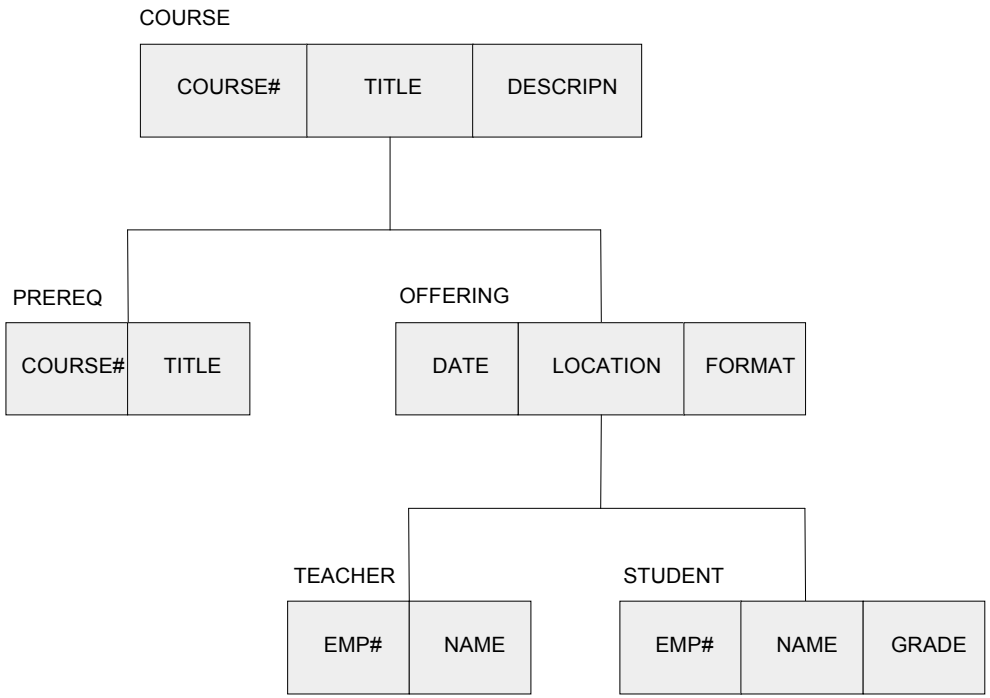


Hierarchisches Modell – Extension



Hierarchisches Modell – Operationen

Welche Studenten sind im Kurs M23 am 13.08.1973 ?




```
GU COURSE(COURSE#='M23')
   OFFERING (DATE='730813')
if gefunden then
begin
    GNP STUDENT
    while gefunden do
    begin
      write(STUDENT.NAME)
      GNP STUDENT
    end
end;
```

Hierarchisches Modell

Augenfällige Vorteile und Nachteile des hierarchischen Modells:

- Das hierarchische Modell ist ein einfaches und effizientes Datenmodell, welches in speziellen Anendungsfällen auch heute noch eine Relevanz hat. Es kommt bspw. in Directory- und Web-Servern wie auch im Kontext der Verwaltung des Index in mySQL-DBen zur Anwendung.
- Die Nachteile des hierarchischen Modells liegen in seiner mangelnden Flexibilität begründet. Da eine hierarchische Struktur für viele Anwendungen zu starr ist und mit dem reinen hierarchischen Modell keine n:m-Beziehung Modelliert werden können.

Netzwerkmodell

Das Netzwerkmodell schreibt das hierarchische Modell fort:

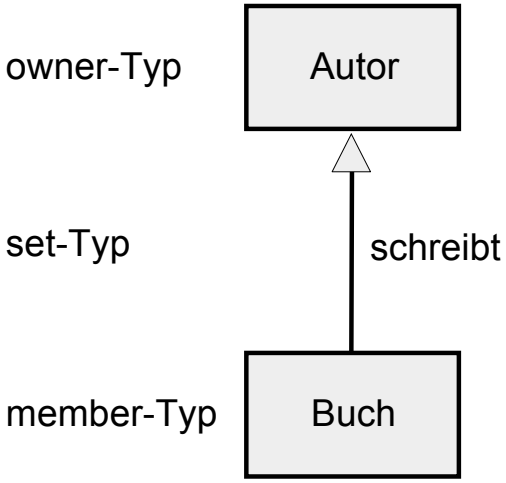
- Insbesondere ist die Restriktion entfernt, dass ein Element nur einer Gruppe zugeordnet sein kann.
- Zudem sind im Netzwerkmodell mehrere Wurzelelemente möglich.
- Hierdurch wird das Netzwerkmodell mächtiger als hierarchies Modell, so dass m:n-Beziehungen beschrieben werden können.

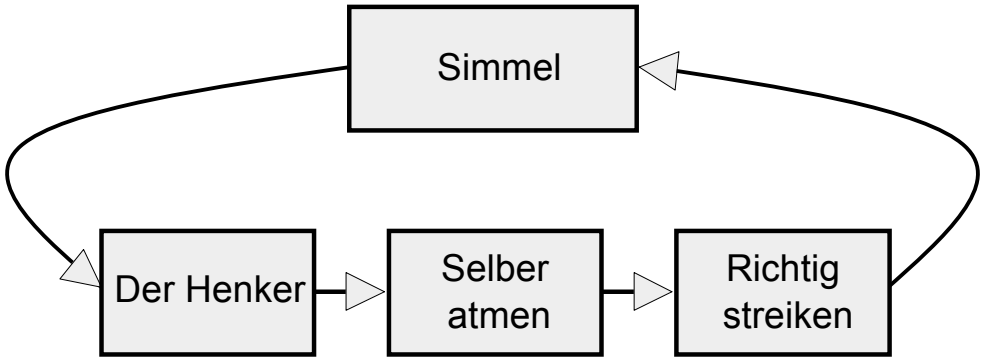
  Sie werden allerdings nicht direkt modelliert, sondern über zwei 1:m-Beziehungen umgesetzt:

  Kursbelegung (Student ← Belegung → Kurs)

- Nachteilig ist ein gewisser Verlust von Einfachheit und Übersichtlichkeit. Zudem ist das sequenzielle Auslesen komplizierter bzw. ineffizienter und demzufolge langsamer.

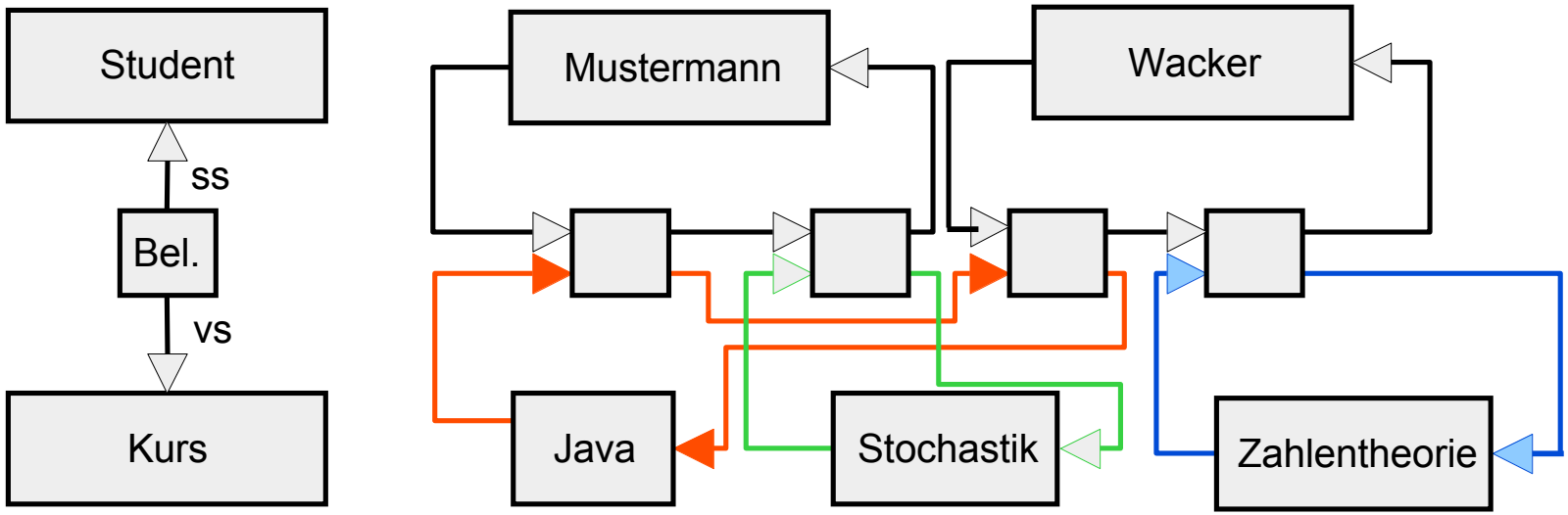
Netzwerkmodell – Schema / Extension





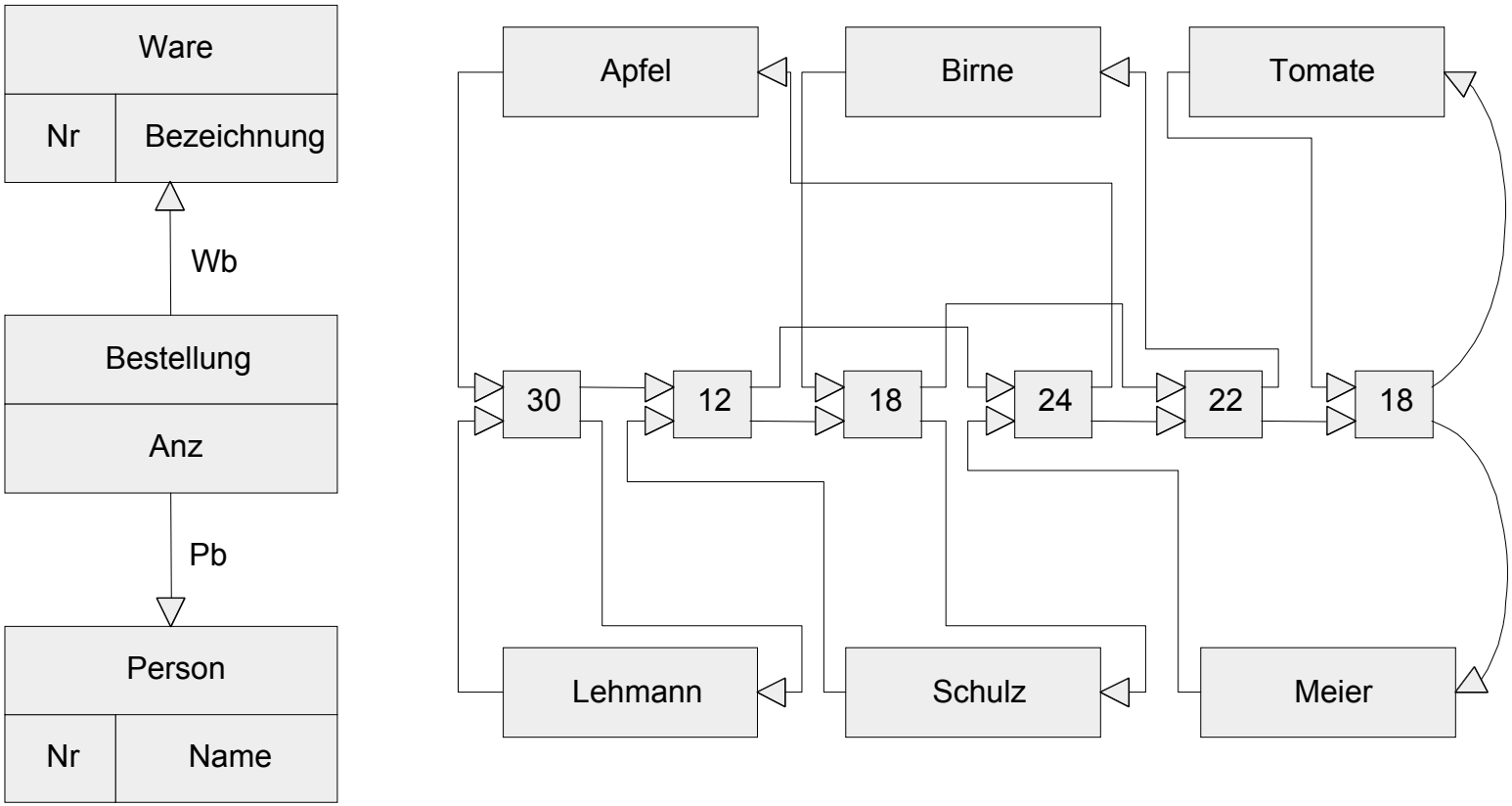
Netzwerkmodell – Kettrecord

(M:N-Beziehung)



Netzwerkmodell – Kettrecord mit Attribut

(M:N-Beziehung)



Operationen im Netzwerkmodell

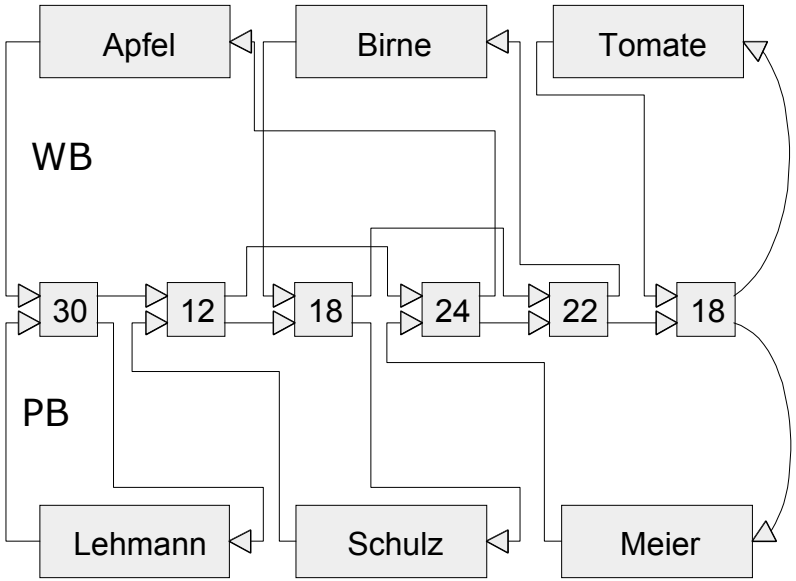
Typische Operation:
Navigation durch die verzeigerten Entities ...

```
FIND ANY Person

FIND NEXT Bestellung WITHIN PB

FIND OWNER WITHIN WB
```

Operationen im Netzwerkmodell




```
PERSON.NAME  := 'SCHULZ';
FIND ANY PERSON USING NAME;
IF GEFUNDEN THEN
BEGIN
    FIND FIRST BESTELLUNG WITHIN PB;
    WHILE GEFUNDEN DO
    BEGIN
        FIND OWNER WITHIN WB;
        GET WARE;
        WRITE(WARE.BEZEICHNUNG);
        FIND NEXT BESTELLUNG WITHIN PB;
    END
END;
```

Relationales Datenmodell

1970: Edgar Codd: "A relational model for large shared data banks“ Communications of the ACM.

1977: Lawrence Ellison gründet Oracle

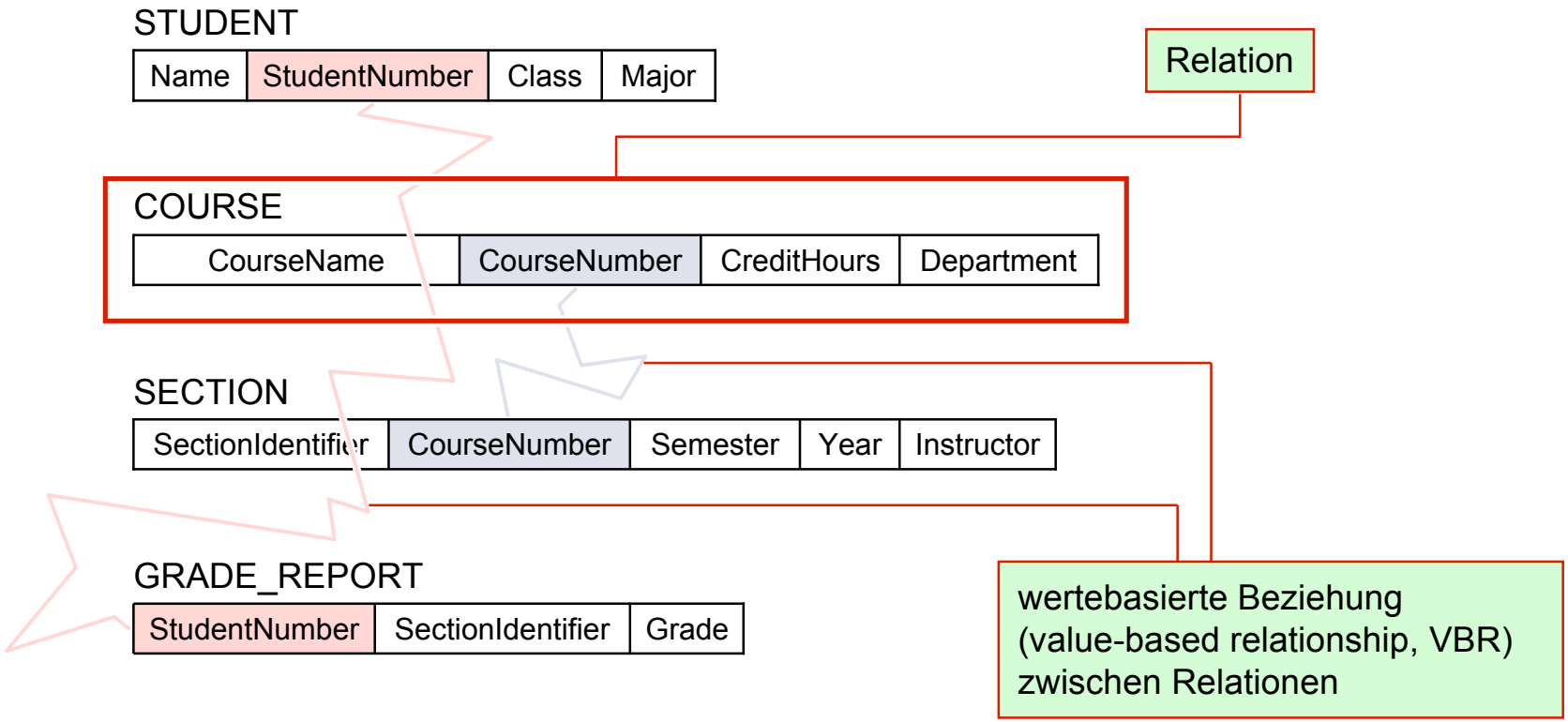
1981: Turing Award an Edgar Codd

Das relationale Datenmodell ist das für die heutige Praxis wichtigste Datenmodell.

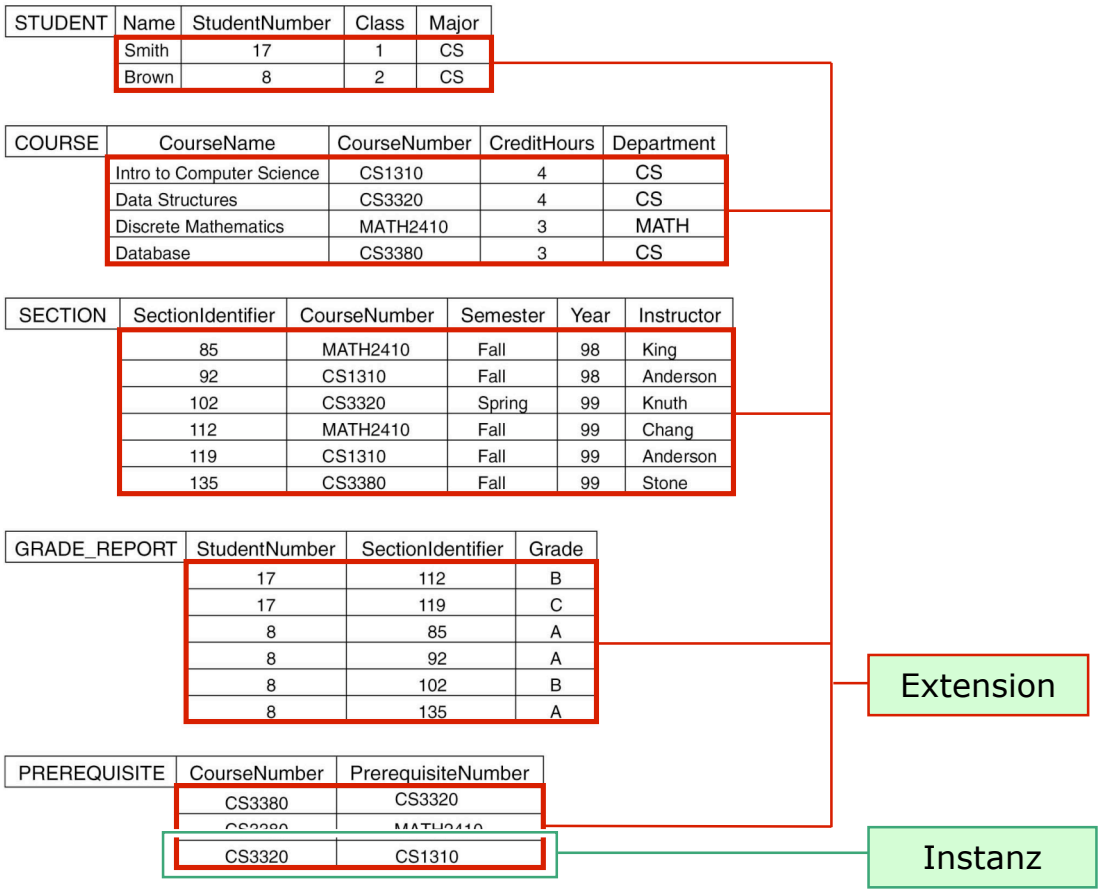
Relationales Datenmodell – Elemente

- Tabellen bzw. Relationen: Bilden Objekte bzw. Konzepte der (realen) Welt ab – z.B. Student, Kurs, Angestellter oder Aufgabe.
  - Ihre Eigenschaften werden durch Spalten der Tabellen festgelegt – z.B. Name, Adresse, Thema, Gehalt oder Qualifikation.
  - Die Konzepte selbst sind durch konkrete Wertebelegungen in den Zeilen der Tabellen wiedergegeben.
- Beziehungen: Logische Verknüpfungen einzelner Fakten sind durch ihre Gruppierung in Tabellen sowie wertebasierte Zusammenhangsbeschreibungen umsetzbar – z.B. Kursbelegung, Projektmitarbeiter.

Relationales Datenmodell – Anschauung



Relationales Datenmodell – Anschauung



STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3320	CS1310

→ University-DB

- → Studenten
- → Kurse
- → Voraussetzungen
- → Arbeitsgruppen
- → Noten

... mit ihren jeweiligen Eigenschaften.

→ PLUS

- → Datentypen (Wertebereiche)
- → Beziehungen (Konsistenz/Integrität)
- → Einschränkungen (von Werten/über Werte)

Relationales Datenmodell – Operationen

- Selektion: Suche alle Tupel einer Relation mit gewissen Attributeigenschaften
- Projektion: filtere gewisse Spalten heraus
- Verbund: Finde Tupel in mehreren Relationen, die bzgl. gewisser Spalten übereinstimmen.

Anfrage (=Query):

Welche Vorlesungen belegte der Student *Smith*?

```
select  COURSE.CourseName
from    STUDENT, COURSE, GRADE_REPORT
where   STUDENT.Name = 'Smith'
  and   STUDENT.StudentNumber = GRADE_REPORT.StudentNumber
  and   GRADE_REPORT.SectionIdentifier = SECTION.SectionIdentifier
  and   SECTION.CourseNumber = COURSE.CourseNumber
```

Objektorientiertes Datenmodell

Objektorientierte Modelle (oo-Modelle) begründen sich gleichermaßen im wachsenden Anspruch bzgl. der Handhabung komplexer Objekte in DBen und dem Ziel der Verringerung der Kluft zwischen objektorientierten Programmiersprachen und DBen.

- Wesentliche Motive
  - Natürlichere / flexiblere Modellierung durch Konzepte der objektorientierten Modellierung bzw. Programmierung – Abbildung komplexer Objekte inklusive Ihres Verhaltes (Methoden).
  - Vermeidung von "Brüchen" aufgrund von Paradigmenwechseln zwischen Anwendungsprogrammen und DBen.
  - Freier navigierender Zugriff in DBen.

- Mögliche Zugänge zur praktischen Umsetzung von oo-DBen
  - Anreicherung von Programmiersprachen um Persistenz und andere DB-Eigenschaften (persistente Programmiersprachen).
  - Entwicklung von DBen unter Rückgriff auf oo-Konzepte.

Objektorientiertes Datenmodell

The Object-Oriented Database System Manifesto (1989)

Atkinson, Bancilhon, DeWitt, Dittrich, Maier und Zdonik (1989).
The Object-Oriented Database System Manifesto. In Proceedings of the First International Conference on Deductive and Object-Oriented Databases, pages 223-240, Kyoto (Japan).

- Eine oo-DB muss die folgenden zwei Kriterien erfüllen
  - sie muss eine DB sein
  - sie muss ein objektorientiertes System sein
- genauer ...

essentielle OO-Aspekte

- Objektidentität
- komplexe Objekte
- Kapselung
- Typ-/Klassenhierarchie, Vererbung
- Überladen etc.
- spätes Binden
- operationale Vollständigkeit
- Erweiterbarkeit

DBMS-Aspekte

- Persistenz
- Externspeicherverwaltung
- Concurrency
- Recovery
- ad-hoc-Anfragesprache

optionale OO-Aspekte

- Mehrfachvererbung
- Versionen
- Design-Transaktionen ...

open choices

- ...

Objektorientiertes Datenmodell – z.B.

- Eine Klasse repräsentiert einen Entity-Typ, charakterisiert durch Struktur und Verhalten.
- Struktur und Verhalten können an eine Unterklasse vererbt werden.
- Binäre Beziehungen werden durch mengenwertige Attribute modelliert.

Objektorientiertes Modell – Klassen

```
class Studenten {
  attribute long Matrnr;
  attribute String Name;
  relationship set <Vorlesungen> hoert
                   inverse Vorlesungen::Hoerer;
}
```

```
class Professoren {
  attribute long PersNr;
  attribute String Name;
  relationship set <Vorlesungen> liest
                   inverse Vorlesungen::gelesenVon;
}
```

```
class Vorlesungen {
  attribute long VorlNr;
  attribute String Titel;
  relationship Professoren gelesenVon
                   inverse Professoren::liest;
  relationship set <Studenten> Hoerer
                   inverse Studenten::hoert;
}
```

Objektorientiertes Modell – Query

Welche Studenten besuchen Vorlesungen von *Sokrates*?

```
select s.Name
from s in AlleStudenten, v in s.hoert
where v.gelesenVon.Name = "Sokrates"
```

```
select Studenten.Name
from Studenten, hoeren, Vorlesungen, lesen, Professoren
where Studenten.MatrNr    = hoeren.MatrNr
  and hoeren.VorlNr       = Vorlesungen.VorlNr
  and Vorlesungen.VorlNr  = lesen.VorlNr
  and lesen.PersNr        = Professoren.PersNr
  and Professoren.Name    = "Sokrates"
```

Objektrelationales Datenmodell

Objektrelationale Modelle (or-Modelle) sind vergleichbar zu oo-Modellen motiviert, weisen allerdings einen strikten Bezug zum relationalen Modell auf – relationale Strukturen bilden die Basis zur Speicherung komplexer Objekte.

- Wesentliche Motive
  - Natürlichere / flexiblere Modellierung und Vermeidung von "Brüchen" aufgrund von Paradigmenwechseln.
  - or-DBen schreiben die Entwicklung relationaler DBen evolutionär fort.
  - "relationale DB-Entwickler" können ihre Erfahrungen auf or-DBen übertragen.
  - Effiziente Implementierung möglich.
- Mögliche Zugänge zur praktischen Umsetzung von or-DBen
  - Anreicherung von Programmiersprachen um Persistenz und andere Datenbankeigenschaften wie Integrität, Zuverlässigkeit etc. (persistente Programmiersprachen).
  - Weiterentwicklung relationaler DBen durch integration von oo-Konzepten.

Objektrelationales Datenmodell

The Third Generation Database System Manifesto (1990)

The Committee for Advanced DBMS Function (M. Stonebraker, L.A. Rowe, B. Lindsay, J. Gray, M. Carey, M. Brodie, P. Bernstein, and D. Beech (1990). *Third-Generation Database System Manifesto*.

To achieve broad exposure this paper is being published in the United States in SIGMOD RECORD and in Europe in the Proceedings of the IFIP TC2 Conference on Object Oriented Databases.

Lehr- / Glaubenssätze zur Gestaltung von DBMSe der dritten Generation

- T1: Neben klassichen DB-Konzepten sollen DBMSe der dritten Generation Objekte verarbeiten und Regeln (insbesondere aktive DB-Mechanismen) anbieten.
- T2: DBMSe der dritten Generation umfassen DBMSe der zweiten Generation (relationale DBMSe).
- T3: DBMSe der dritten Generation sind offene Systeme.

XML-basierte Datenmodelle

XML-basierte Modelle sind bspw. geeignet, wenn die in einer DB zu handhabenden Konzepte einer Miniwelt bzgl. ihrer Eigenschaften bzw. Daten nicht präzise zu fassen sind.

Dies ist bspw. in folgenden Situationen der Fall:

- Die zu verarbeitenden Daten besitzen eine geringe Dichte.
- Die Struktur der Daten ist unbekannt, unpräzise oder über die Zeit einem stärkeren Wandel unterworfen.
- Die Daten bechreiben Kapselungshierarchien, die auch rekursiv sein können.
- Die Daten weisen eine inhärente Reihenfolge auf.
- Abfragen oder Aktualisierungen der Daten sollen ihre Struktur berücksichtigen.

XML-basierte Datenmodelle

Ferner bieten sich XML-basierte Modelle an, wenn die Daten durch unterschiedliche Anwendungen (auf verschiedenen Plattformen) verarbeitet werden müssen, wobei ihre Transformation in jeweils spezielle Formate zwingend notwendig ist.

Bei der Integration von XML und DB-Technologien werden native Ansätze von Ansätzen zur Integration von XML in relationale bzw. objektrelationale DBen unterschieden.

Query (Nutzung von XPath):

```
select MessageIndex,
       extractValue(Message,'/Adresse/Name')
from MyTab
where MessageIndex >= 11011969;
```

Datenbanksysteme SS 2007

Ende von Kapiel 3: Logische Datenmodelle