Inhalt

Vom Dateisystem zum Datenbanksystem

Dateisystem

- Jedes Programm verwendet nur seine eigenen Daten
- Struktur der Daten ist im Programm festgelegt

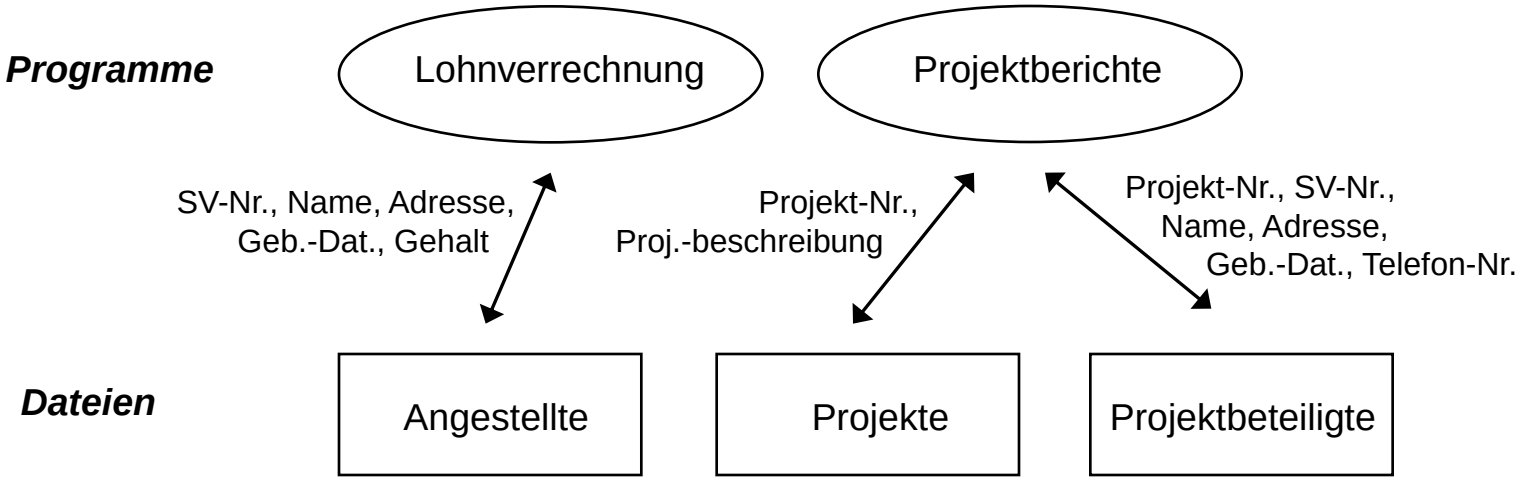
Nachteile:

- Abhängigkeit zwischen Programmlogik und physischer Datenstruktur (**physische Datenabhängigkeit**)
- Datenstrukturänderung erfordert Programmänderung
- Daten gleicher Bedeutung werden u.U. mehrfach gespeichert (**Redundanz**, Gefahr von **Inkonsistenz**)
- gleichzeitige Verwendung von Daten durch mehrere Programme unmöglich

Dateisystem - Beispiel

Programm Lohnverrechnung:
Datei Angestellte (SV-Nr., Name, Adresse, Geburtsdatum)

Programm Projektberichte:
Datei Projekte (Projekt-Nr., Projektbeschreibung)
Datei Projektbeteiligte (Projekt-Nr., SV-Nr., Name, Adr., Geb.dat.,Tel-Nr.)



Integrierte Datenverarbeitung

- Alle Daten werden an zentraler Stelle gesammelt.
- Datenkoordinator verwaltet den zentralen Dateienbestand mit Hilfe einer allgemein zugänglichen Referenzdatei, die Strukturbeschreibungen aller Dateien enthält (**Data Dictionary**).

Vorteile:

- Konsistenz wird verbessert
- Redundanz wird verringert

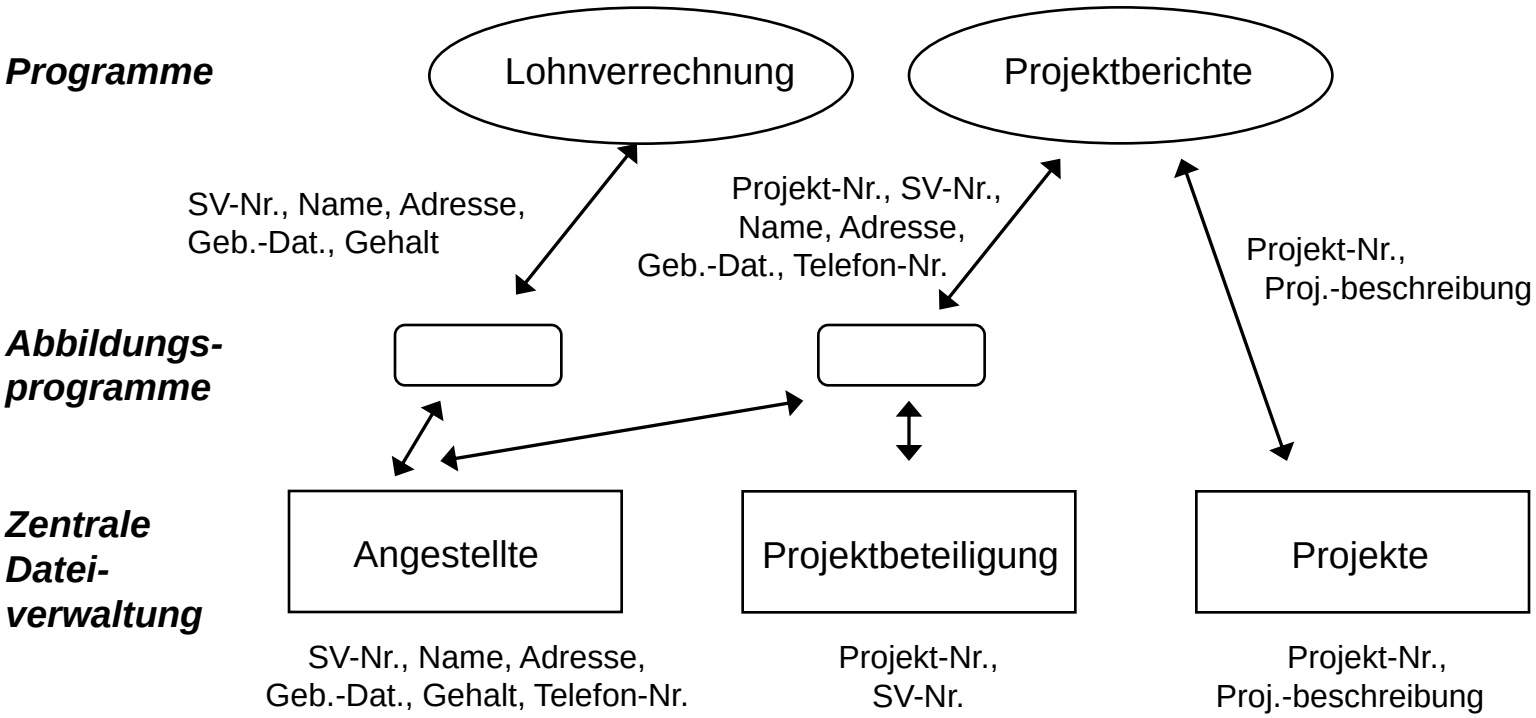
Nachteile:

- Änderung einer Datei erfordert Änderung von Programmen.
- Aus zentralen Dateien müssen für jedes Programm passende Dateiauszüge erzeugt werden (weiterhin physische Datenabhängigkeit).

Integrierte Datenverarbeitung - Beispiel

Zentrale Dateien:

Personal (SV-Nr., Name, Adresse, Geburtsdatum, Gehalt, Tel-Nr.)
Projekte (Projekt-Nr., Projektbeschreibung)
Projektbeteiligung (Projekt-Nr., SV-Nr.)



Motivation für den Einsatz eines DBMS [1]

Begriffe:

- Die gespeicherten Daten werden als **Datenbasis** oder Datenbank (**Database**) bezeichnet. Sie enthält die miteinander in Beziehung stehenden Informationseinheiten, die zur Kontrolle und Steuerung eines Aufgabenbereichs (evtl. eines ganzen Unternehmens) notwendig sind.
- Die Gesamtheit der Programme zum Zugriff auf die Datenbasis, zur Kontrolle der Konsistenz und zur Modifikation der Daten wird als Datenbankverwaltungssystem (**Database Management System = DBMS**) bezeichnet.
- Datenbasis und Datenbankverwaltungssystem bilden dann ein **Datenbanksystem**.

Leider werden diese Komponenten oft auch weniger scharf getrennt.

Motivation für den Einsatz eines DBMS [2]

Typische Probleme bei Informationsverarbeitung ohne DBMS :

- Redundanz und Inkonsistenz
- beschränkte Zugriffsmöglichkeiten
- Probleme beim Mehrbenutzerbetrieb
- Verlust von Daten
- Integritätsverletzung
- Sicherheitsprobleme
- hohe Entwicklungskosten für Anwendungsprogramme

Datenunabhängigkeit

- **Physische Datenunabhängigkeit:** Die Modifikation physischer Speicherstrukturen hat keine Änderung der auf die Daten zugreifenden Programme zur Folge (z.B. nachträgliches Anlegen eines Indexes oder Speicherung der Daten auf einem anderen Laufwerk).
- **Logische Datenunabhängigkeit:** Änderungen in der logischen Sicht der Daten (in der ‚Logischen Ebene', siehe später) haben keine Änderungen der Programme zur Folge, die diese Änderung nicht für ihre Funktionalität benötigen.

Abstraktionsebenen eines Datenbanksystems [1]

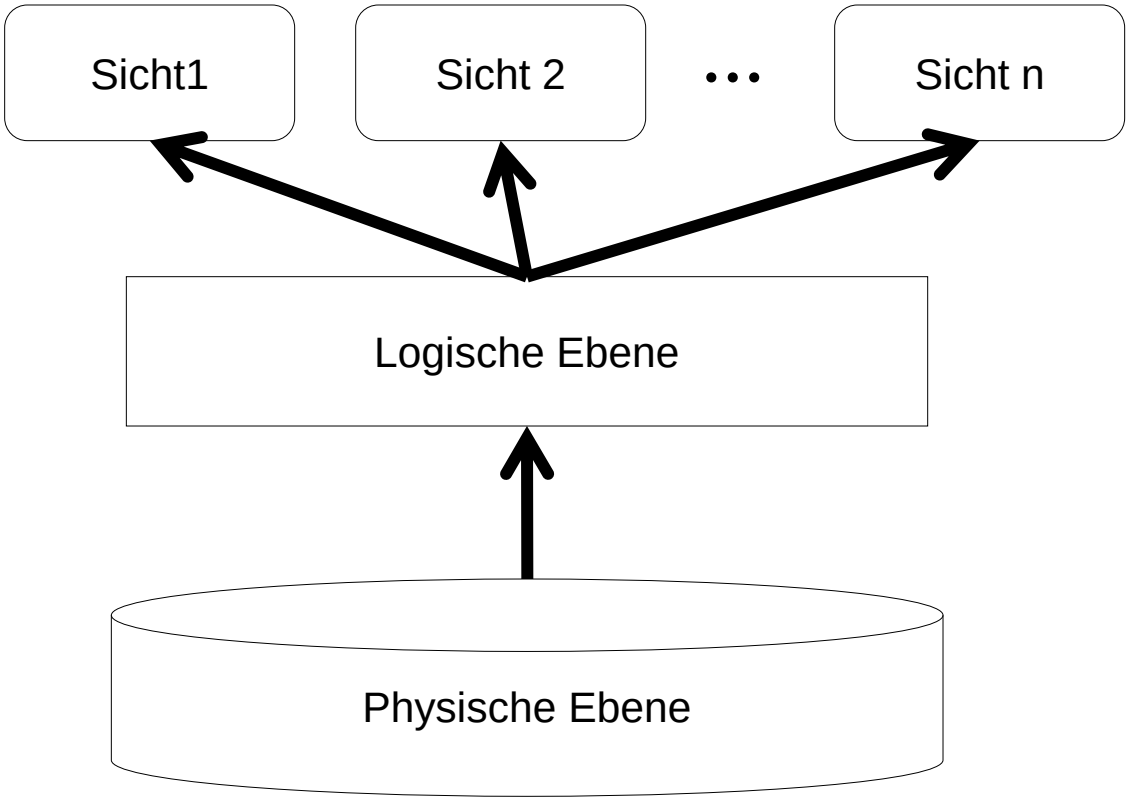


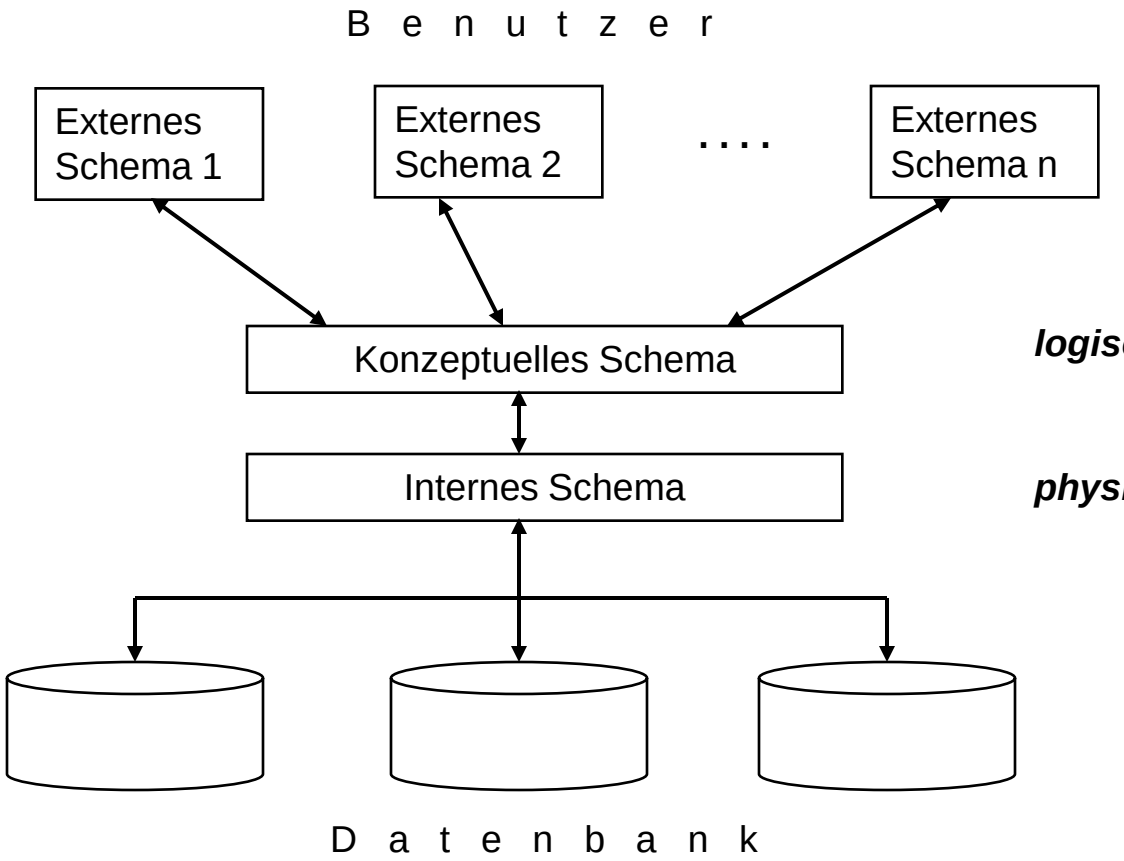
Abb.: Drei Absraktionsebenen eines Datenbanksystems
(entnommen aus [1])

Abstraktionsebenen eines Datenbanksystems [2]

- **Physische Ebene:** Auf dieser Ebene wird festgelegt, wie die Daten gespeichert sind. Im Allgemeinen sind die Daten auf dem Hintergrundspeicher (meistens auf einer Festplatte) abgelegt. Auch die Festlegung der physischen Speicherstruktur (z.b. in Blöcken, oder in Sätzen, mit oder ohne Index, ...) fällt in diese Ebene.
- **Logische Ebene:** Auf der logischen Ebene wird in einem sogenannten Datenbankschema festgelegt, welche Daten abgespeichert sind.
- **Sichten (Externe Ebene):** Während das Datenbankschema der logischen Ebene ein integriertes Modell der gesamten Information darstellt, werden in den Sichten Teilmengen der Information bereitgestellt. Die Sichten sind auf die Bedürfnisse der jeweiligen Benutzergruppen zugeschnitten.

Abstraktionsebenen eines Datenbanksystems [3]

Drei-Schichten-Architektur von Datenbanksystemen (ANSI/SPARC 1978)



logische Datenunabhängigkeit

physische Datenunabhängigkeit

Entwicklungsgeschichte von DBMS-Systemen

- Einzeldateien ohne Integration (bis 1974)
- Physische Dateiintegration (bis 1975)
- Zwei-Schichten-Architektur (ab 1975)
- Drei-Schichten-Architektur (ab 1978)

Architekturübersicht eines DBMS

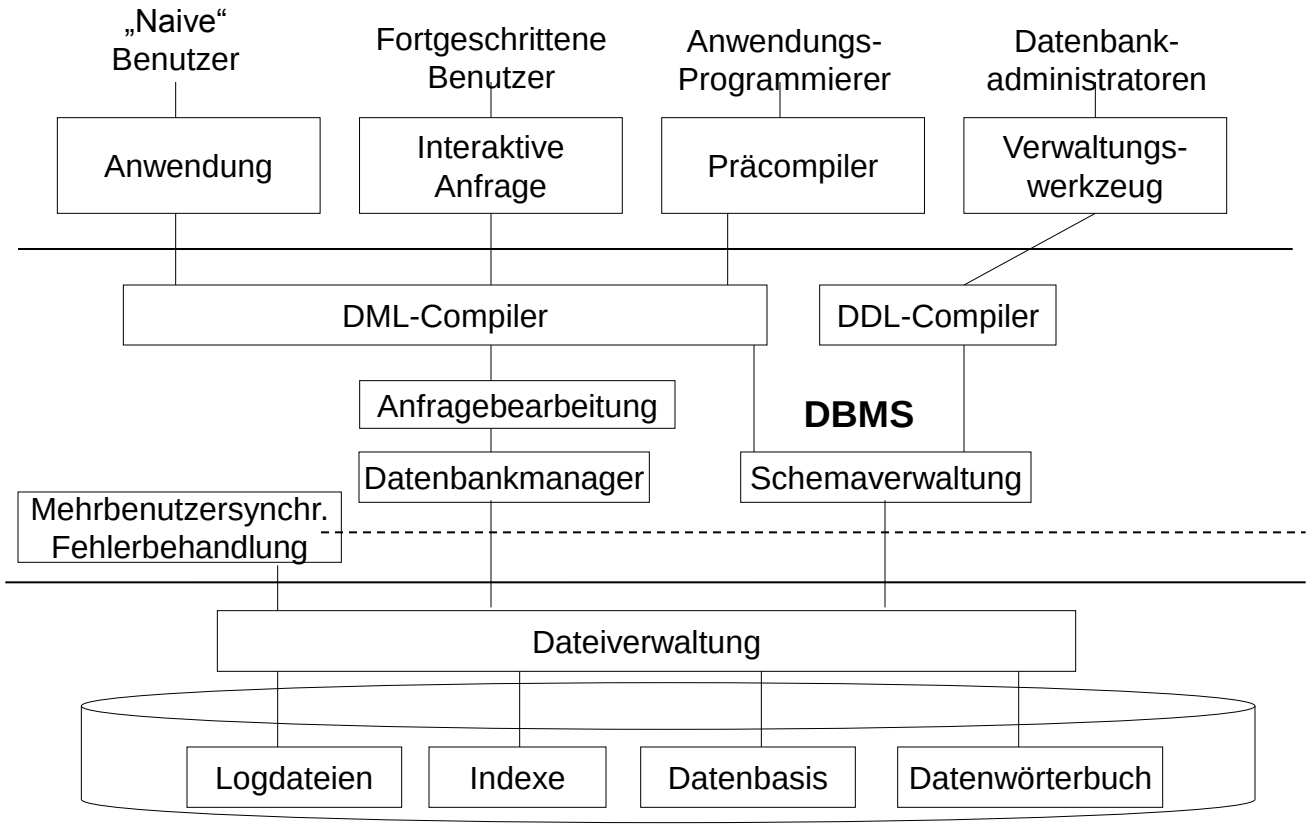


Abb.: Architekturübersicht eines DBMS
(entnommen aus [1])

Allgemeines, Definitionen [1]

Datenbanksysteme basieren auf einem **Datenmodell**, das sozusagen die ‚Infrastruktur' für die Modellierung der realen Welt zur Verfügung stellt. Das Datenmodell legt die Modellierungskonstrukte fest, mittels derer man ein computerisiertes Informationsabbild der realen Welt (bzw. des relevanten Ausschnitts) generieren kann. Es beinhaltet die Möglichkeit zur

- Beschreibung der Datenobjekte und zur
- Festlegung der anwendbaren Operatoren und deren Wirkung.

Das Datenmodell ist somit analog zu einer Programmiersprache. Es legt die generischen Strukturen und Operationen fest (z.B. Typkonstruktoren und Sprachkonstrukte), mit deren Hilfe man dann die Anwendungsprogramme realisiert.

Allgemeines, Definitionen [2]

Ein Datenmodell besteht aus zwei Teilsprachen

- Datendefinitionssprache (**Data Definition Language, DDL**)
- Datenmanipulationssprache (**Data Manipulation Language, DML**)

Die DDL wird benutzt, um die Struktur der abzuspeichernden Datenobjekte zur beschreiben. Dabei werden gleichartige Datenobjekte durch ein gemeinsames **Schema** beschrieben (analog zu einem Datentyp in Programmiersprachen).

Die Strukturbeschreibung aller Datenobjekte des betrachteten Anwendungsbereichs nennt man das **Datenbankschema**.

Allgemeines, Definitionen [3]

Die Datenmanipulationssprache DML besteht aus

- der Abfragesprache (**Query Language**) und
- der ‚eigentlichen' Datenmanipulationssprache
  - Änderung von gespeicherten Datenobjekten
  - Einfügen von Daten
  - Löschen von Daten

Die DML kann in zwei Arten genutzt werden:

- interaktiv (DML-Kommandos werden direkt am Arbeitsplatzrechner eingegeben)
- in einem Programm einer höheren Programmiersprache, das ‚eingebettete' DML-Kommandos enthält.

Allgemeines, Definitionen [4]

Datenbankschema und Ausprägung:

Das Datenbankschema legt die Struktur (bei einem objektorientierten Datenmodell auch das Verhalten) der abgespeicherten Datenobjekte fest. Es sagt also nichts über die individuellen Datenobjekte. Deshalb kann man ein Datenbankschema auch als Metadaten - also Daten über Daten - verstehen.

Die **Datenbankausprägung** ist demgegenüber der momentan gültige Zustand der Datenbasis. Sie muss den im Schema festgelegten Strukturbeschreibungen gehorchen.

Man spricht auch von der **intensionalen Ebene** (Schema) und der **extensionalen Ebene** (Ausprägung) einer Datenbank.

Änderungen am Schema werden als **Schemaevolution** bezeichnet.

Allgemeines, Definitionen [5]

Weitere Definitionen von ‚Datenmodell':

Dittrich: Ein Datenmodell ist ein Satz von abgestimmten Konzepten zur Beschreibung der Repräsentation von Information durch Daten.

Vossen: Ein Datenmodell ist das zentrale Hilfsmittel der ‚Datenbanktechnologie' zur Herstellung einer Abstraktion eines gegebenen ‚Realwelt-Ausschnitts' und gleichzeitig von den Einzelheiten der physischen Speicherung. Es umfasst im Allgemeinen eine gewisse Menge von Konzepten, mit welchen sich die Struktur einer Datenbank beschreiben lässt. Mit ‚Struktur' sind dabei Datentypen, Beziehungen und Bedingungen gemeint, welche von den Daten erfüllt werden müssen.

Allgemeines, Definitionen [6]

Einordnung der Datenmodelle [1]

Übersicht der Datenmodellierung

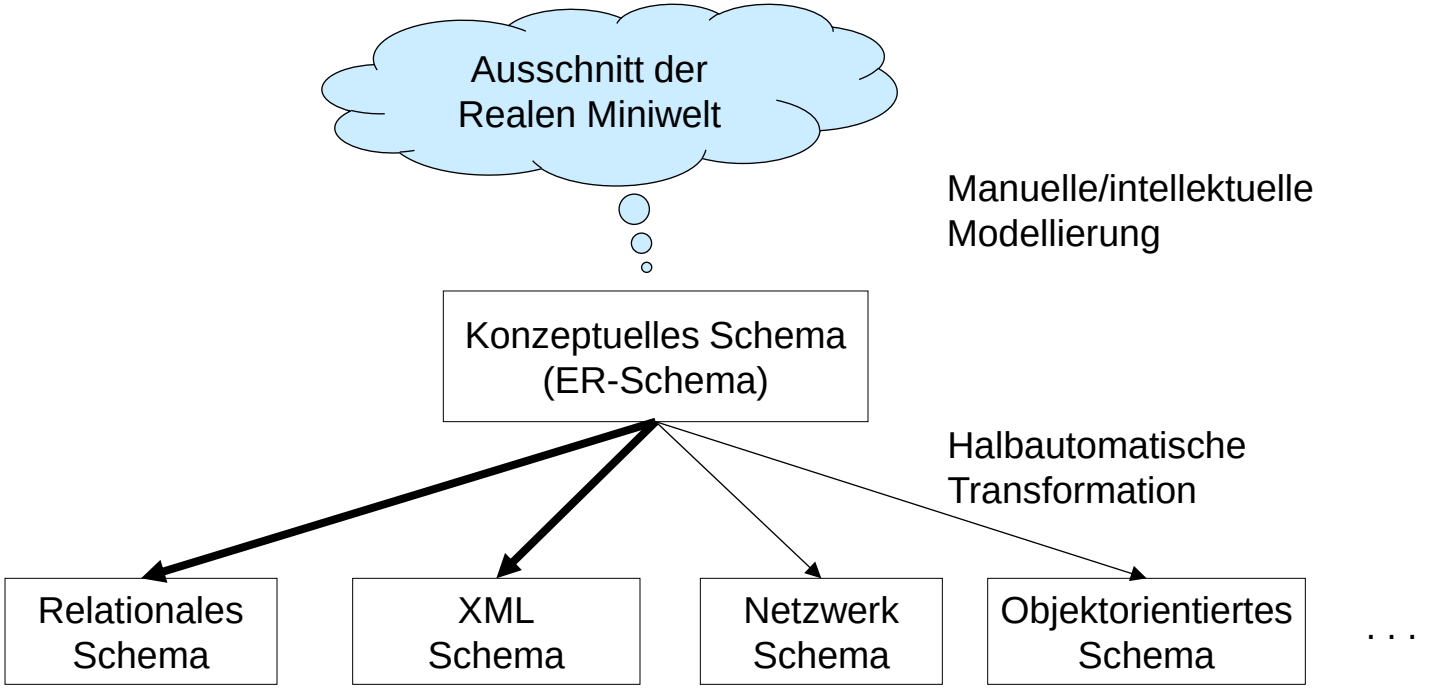


Abb.: Übersicht der Datenmodellierung
(entnommen aus [1])

Allgemeines, Definitionen [7]

Einordnung der Datenmodelle [2]

Modelle des konzeptuellen Entwurfs

- Entity-Relationship-Modell (ER-Modell)
- semantisches Datenmodell
- objektorientierte Entwurfsmodelle

Diese Modelle verfügen im allgemeinen nur über eine DDL. Sie verzichten auf die Abbildung individueller Datenobjekte, benötigen also keine DML.

Logische (Implementations-) Datenmodelle

- Netzwerkmodell
- hierarchisches Datenmodell
- relationales Datenmodell
- objektorientiertes Datenmodell
- deduktives Datenmodell

Allgemeines, Definitionen [8]

Mini-Beispiel für Datenmodellierung [1]

Konzeptuelle Modellierung (Anwendung des ER-Modells)

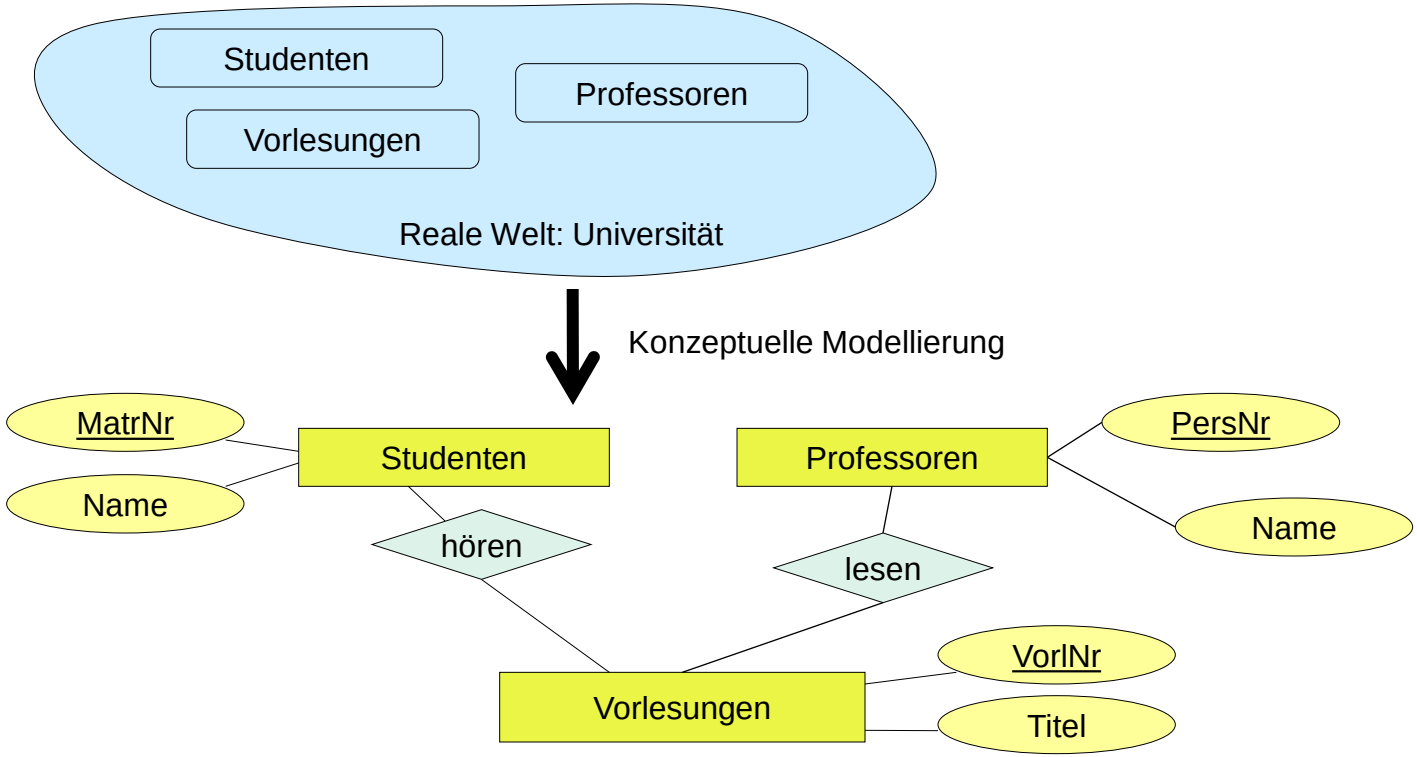


Abb.: Konzeptuelle Modellierung
einer (sehr kleinen) Beispielsanwendung
(entnommen aus [1])

Allgemeines, Definitionen [9]

Mini-Beispiel für Datenmodellierung [1]

Logisches Datenmodell (in diesem Fall ein relationales Datenmodell)

Studenten	
MatrNr	Name
26120	Fichte
25403	Jonas
...	...

hören	
MatrNr	VorlNr
25403	5022
26120	5001
...	...

Vorlesungen	
VorlNr	Titel
5001	Grundzüge
5022	Glaube und Wissen
...	...

```
Select  Name
From    Studenten, hören, Vorlesungen
Where   Studenten.MatrNr = hören.MatrNr  and
        hören.VorlNr = Vorlesungen.VorlNr  and
        Vorlesungen.Titel = `Grundzüge´;
```

Beisp. für Query Language

```
update     Vorlesungen
set        Titel = `Grundzüge der Logik´
where      VorlNr = 5001;
```

Beisp. für ‚eigentliche' DML

Modelle des Konzeptuellen Entwurfs

ER-Modell

Das Entity-Relationship-Modell (ER-Modell) ist das nahezu ausschließlich verwendete Modell im Konzeptuellen Datenbankentwurf. Es wird im Rahmen dieser Lehrveranstaltung in einem eigenen Kapitel gesondert behandelt.

Semantisches Datenmodell

Für den konzeptuellen Datenbankentwurf wurden mehrere andere Datenmodelle konzipiert, die aber bei weitem nicht die gleiche praktische Bedeutung erlangt haben wie das ER-Modell.

Sie werden oft als semantische Datenmodelle bezeichnet, weil sie die Bedeutung (Semantik) der Anwendungsobjekte besser erlauben zu modellieren, als dies beim ER-Modell der Fall ist.

Objektorientierte Entwurfsmodelle

Sie können durchaus als Erweiterungen des ER-Modells gesehen werden.

Neben der Bestimmung von Kandidaten für Objekttypen (vergl. Entitytypen) und Beziehungen zwischen ihnen wird aber auch die Modellierung des Verhaltens der Objekte in den Entwurf integriert.

Hilfsmittel dazu sind unter anderem die sogenannten ‚use cases' (speziell für Anforderungsanalyse), die ‚Objektszenarien' und ‚Interaktionsdiagramme' (mit beiden kann Objektverhalten graphisch dargestellt werden)

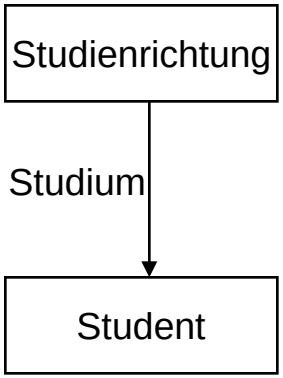
Objektorientierte Entwurfsmodelle sind nicht Gegenstand dieser Lehrveranstaltung. Sie werden in eigenen Lehrveranstaltungen genauer behandelt.

Hierarchisches Modell [1]

Die ersten Datenbanksysteme basierten auf diesem Modell (wichtigste Implementierung: IMS von IBM). Es wurde zur Modellierung hierarchischer Strukturen entwickelt.

Beispiel 1:

Assoziationsgraph (Typebene)

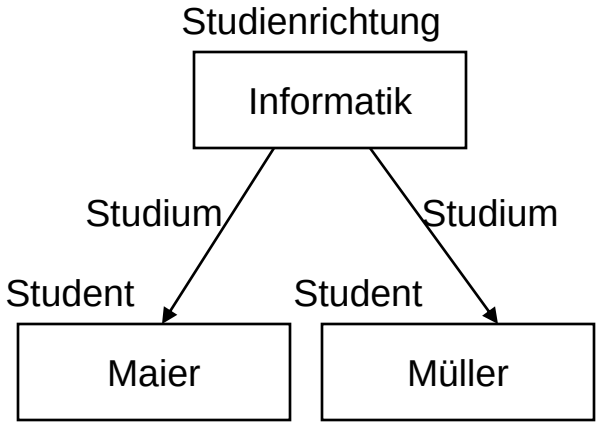


Übergeordneter Objekttyp

1:n - Assoziationstyp

untergeordneter Objekttyp

Realisierungsgraph (Exemplarebene)



1 übergeordnetes Objekt

1 Assoziations-exemplar

2 untergeordnete Objekte

Hierarchisches Modell [2]

Im Assoziationsgraph gilt:

- Jeder Objekttyp hat höchstens einen übergeordneten Objekttyp.
- Zyklen sind nicht erlaubt (Ein Objekttyp kann nicht sich selbst übergeordnet sein).
- d.h., der Assoziationsgraph besteht aus Bäumen

Im Realisierungsgraph gilt:

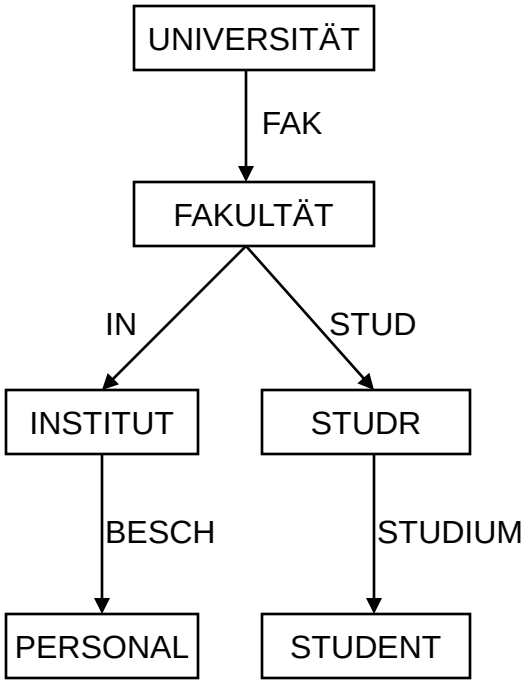
- Jedes Objekt darf mit höchstens einem Objekt eines übergeordneten Objekttyps verbunden sein.

Problem: Darstellung von m:n-Beziehungen sind ohne komplexe Hilfskonstruktionen nicht möglich.

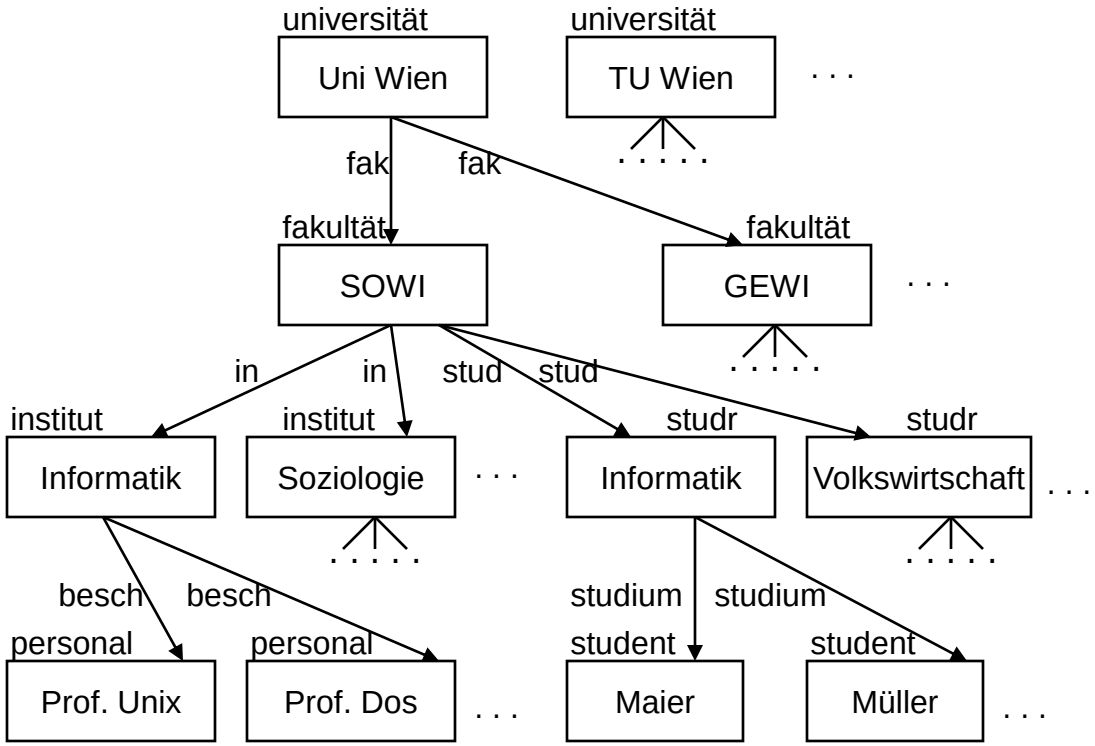
Hierarchisches Modell [3]

Beispiel:

Assoziationsgraph:



Realisierungsgraph:



Hierarchisches Modell [4]

Auffinden einer gesuchten Information:

- Erreichen des Objekts durch Navigation. Startobjekt → Weg durch den Realisierungsgraphen zum gewünschten Objekt (logischer Zugriffspfad).

Zugriffspfade - mögliche Schritte:

A: übergeordnetes Objekt → Assoziation → erstes Objekt des untergeordneten Typs

B: Objekt eines Typs → nächstes Objekt desselben Typs mit demselben übergeordneten Objekt

C: untergeordnetes Objekt Assoziation übergeordnetes Objekt

Abfragen können also nur entlang der vordefinierten Zugriffspfade abgearbeitet werden (Zugriffspfadabhängige Abfragebearbeitung).

Netzwerkmodell [1]

- Ein Objekttyp kann mehrere übergeordneten Objekttypen besitzen (wird mit Hilfe eines **Verkettungstyp**s realisiert.
- Der Assoziationsgraph besteht aus einem gerichteten Graphen (vergl. hierarchisches Datenmodell, Baum)

Der Verkettungstyp ermöglicht die Darstellung von:

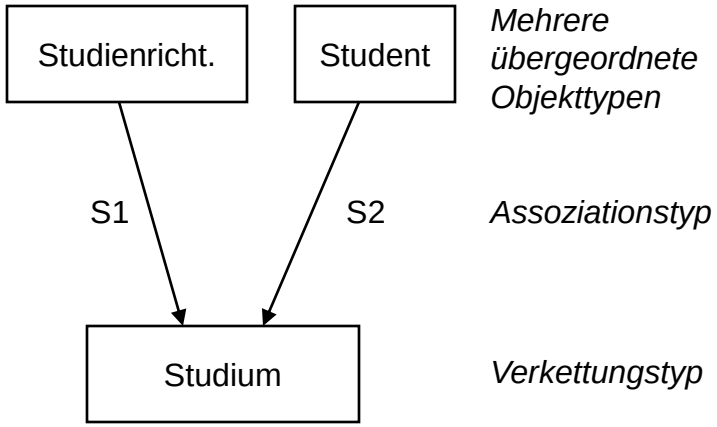
- m:n-Beziehungen
- Rekursionen

Das Auffinden gesuchter Information geschieht analog zum hierarchischen Modell entlang der Zugriffspfade im Realisierungsgraph.

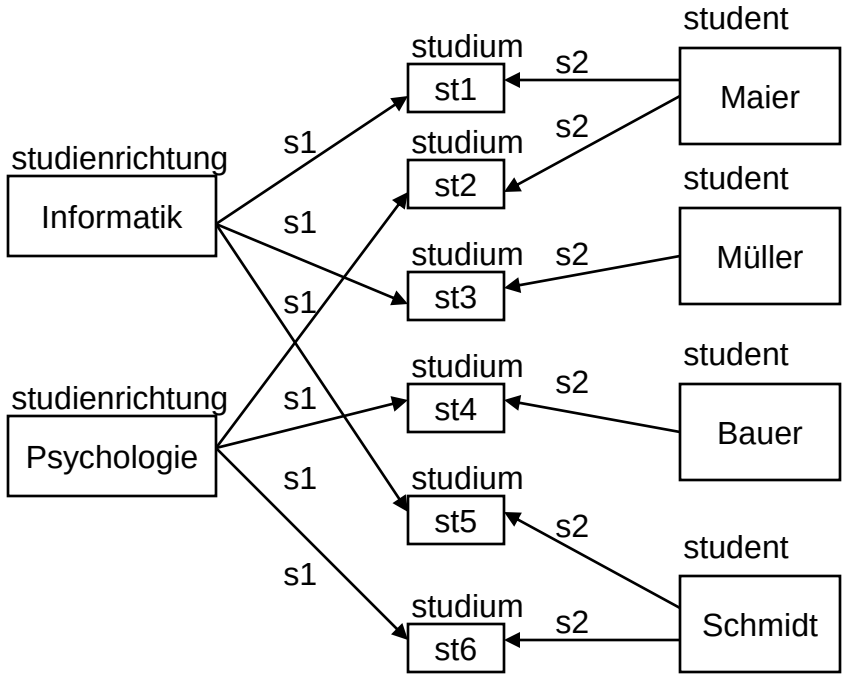
Netzwerkmodell [2]

Beispiel: Darstellung einer m:n-Beziehung

Assoziationsgraph



Realisierungsgraph



Relationales Modell

Die Besonderheit dieses Modells liegt in der **mengenorientierte**n **Verarbeitung** im Gegensatz zu den bis dahin vorherrschenden satzorientierten Verarbeitung (hierarchisches Modell, Netzwerkmodell).

Es ist sehr einfach strukturiert. Es gibt im wesentlichen nur Tabellen (Relationen) in denen die Zeilen den Datenobjekten entsprechen. Diese Daten werden durch entsprechende Operationen ausschließlich mengenorientiert verknüpft und verarbeitet.

Die relationale Datenbanktechnologie nimmt heute eine marktdominierende Stellung ein.

Dieses Modell wird im Rahmen dieser Lehrveranstaltung in eigenen Kapitel detailliert behandelt.

Objektorientiertes Datenmodell [1]

Die objektorientierte Datenmodellierung **integriert Struktur- und Verhaltens-Beschreibung** in einer einheitlichen Objekttyp-Definition.

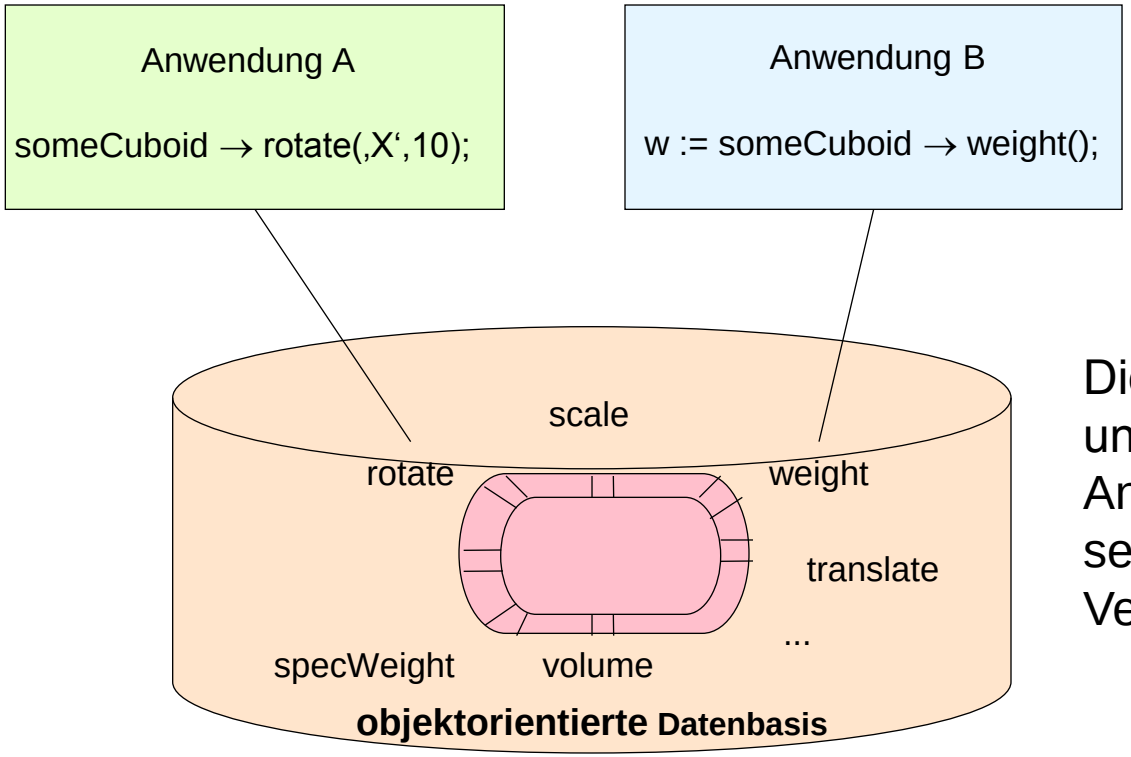
Oft spricht man von Objektkapselung, da die interne Struktur eines Objektes bleibt den Benutzern verborgen. Sie sehen nur die dem Objekttyp zugeordneten Operationen.

Dies ist besonders bei komplexen Objekten von Vorteil, die im Relationalen Modell nur mittels vieler Relationen modelliert werden können und dann in relationalen Datenbanken ausschließlich mit Insert-, Update und Delete-Operationen auf Einzelrelationen bearbeitet werden können.

Die Object Database Management Group (ODMG) hat einen de-facto-Standard formuliert, das ODMG-Modell.

Objektorientiertes Datenmodell [2]

Visualisierung der Vorteile objektorientierter Datenmodellierung



Die Operationen ‚rotate‘ und ;weight‘ stehen allen Anwendungen datenbank-seitig bereits zur Verfügung.

Abb.: Visualisierung der Vorteile objektorientierter Datenmodellierung
(entnommen aus [1])

Deduktives Datenmodell

Das relationale Datenmodell wurde um eine Dedukionskomponente erweitert. Diese basiert auf dem Prädikatenkalkül - also der Logik erster Stufe.

Die drei grundlegenden Komponenten eines deduktiven Datenbanksystems:

- Extensionale Datenbasis (EDB): Manchmal auch Faktenbasis, entspricht einer ‚ganz normalen' relationalen Datenbank
- Deduktionskomponenten: Menge von (Herleitungs-) Regeln. Die Regelsprache heißt **Datalog**.
- Intensionale Datenbasis (IDB): Menge von hergeleiteten Relationen und Ausprägungen; wird durch Auswertung von Datalog-Regeln generiert.