

Kapitel 4: Referentielle DB-Modelle - HDM

	Relationales Datenmodell (RDM)	Netzwerk- und Hierarchisches Datenmodell (NDM, HDM)	Objekt-orientierte Datenmodelle (OODM)	Objekt-relationale Datenmodelle
Überblick über die Konzepte	3.1	4.1 4.2	5.1	6.1
Darstellung von Assoziationen				
Datendefinition				
Anfragen				
Aktualisierungsoperationen				
Spezifika	3.2 SQL		5.2 ODMG	6.2, 6.3

Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.1

HDM: Überblick über die Konzepte (1)

Analogie zum HDM: Hierarchische Strukturierungen und Benennungsschemata in vielen Bereichen (z.B. Organigramme, Klassifikationsschemata, ...)

► Ausrichtung des Hierarchischen Datenmodells speziell auf hierarchisch organisierte Informationsstrukturen

- In vielen Aspekten bestehen Ähnlichkeiten zum Netzwerkdatenmodell, aber keine „multiple Subsumption“.

Grundstruktur des HDM:

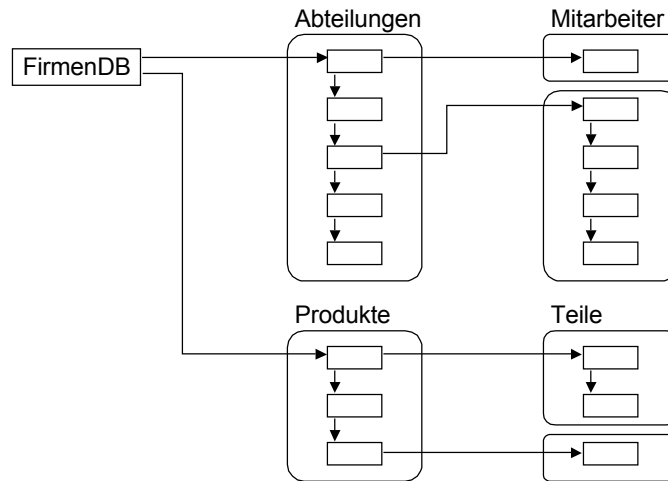
- HDM-Bäume*: Rekords, die durch Referenzen miteinander verknüpft sind (hier: nur *Baumstrukturen*, also hierarchische Strukturen, zulässig)
- Eine hierarchische Datenbank besteht aus einer Menge von *HDM-Bäumen*.

Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.2

HDM: Überblick über die Konzepte (2)

Beispiel: Hierarchische Datenbank *FirmenDB* („Listenstruktur“)

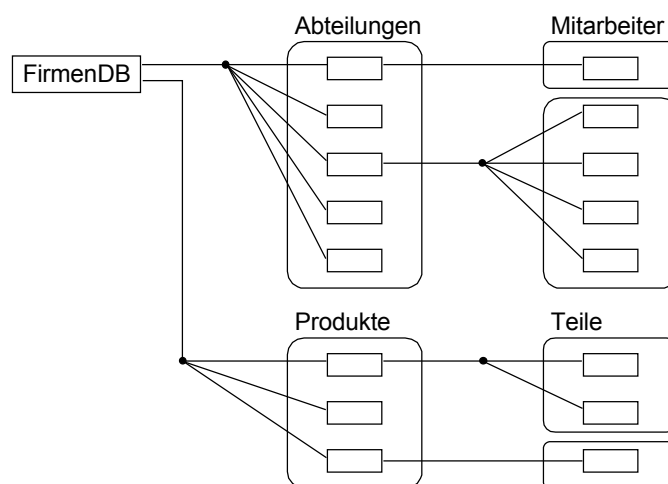


Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.3

HDM: Überblick über die Konzepte (2 - Alternative)

Beispiel: Hierarchische Datenbank *FirmenDB* („Baumstruktur“)



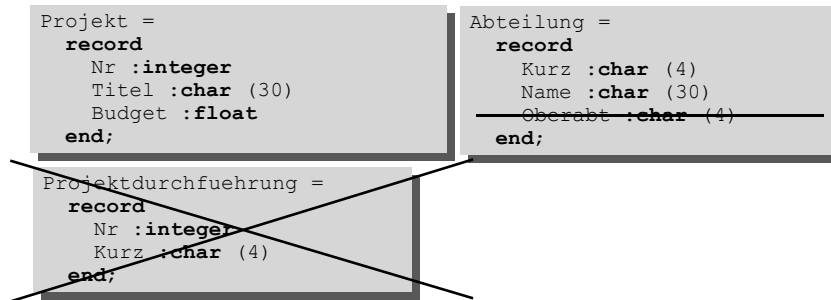
Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.4

HDM: Grundstruktur (1)

Rekords:

- Durch Referenzen miteinander verbunden (vgl. Strukturierungsprimitive des NDM)
- Flache Struktur; keine assoziative Identifikation



- Hierarchie in DDL festgelegt, z.B. *Abteilung* unterhalb von *Projekt*

HDM: Grundstruktur (2)

HDM-Bäume:

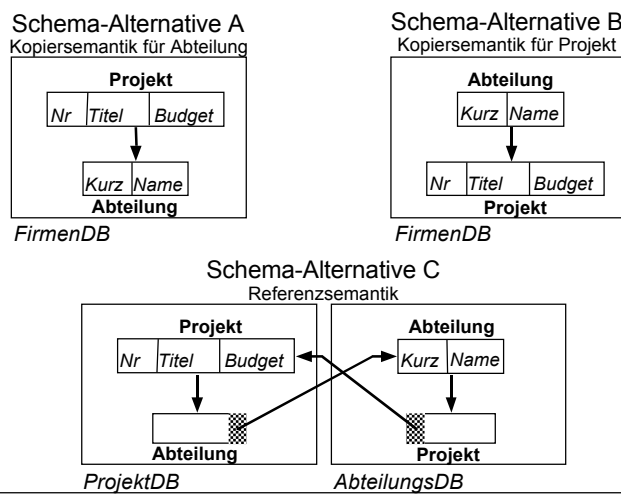
- Inhomogene Struktur*: Auf jeder Ebene treten neue Rekordtypen für die Knoten auf.
- Im Rahmen der Schemadefinition wird ein *HDM-Baumtyp* spezifiziert, von dem alle Bäume, aus denen sich eine Datenbank zusammensetzt, Ausprägungen darstellen.
- Beginnend mit der Wurzel legt der HDM-Baumtyp die Rekordtypen für die Knoten jeder Ebene fest.
- Es ist nicht zulässig, den gleichen Rekordtyp auf mehreren Ebenen zu verwenden (rekursive Beziehungen nicht direkt modellierbar).
- Eine Datenbank besteht aus einer benannten Kollektion von Rekords, den Wurzelementen der Bäume.
- Jeder dieser Rekords besitzt Verweise auf eine oder mehrere Kollektionen von Rekords (für jeden Nachfolgerknotentyp eine Kollektion), die die Nachfolgerknoten bilden usw..
- Kollektionen sind in der Regel Listen, so daß eine Reihenfolge auf den Elementen der Kollektion definiert ist (vgl. Folie 4.2.3)

HDM: Assoziationen (1)

- ❑ HDM nur für die Modellierung von Assoziationen für binäre 1:1 und 1:N-Beziehungen geeignet.
- ❑ Direkte Unterstützung von binären 1:N-Beziehungen, da in den Baumstrukturen N Nachfolgerknoten einem Elternknoten zugeordnet werden.
- ❑ Binäre 1:1-Beziehungen lassen sich durch Beschränkung auf einen Nachfolgerknoten darstellen.
- ❑ Schwierigkeiten treten bei der Modellierung von N:M-Beziehungen, von allgemeinen n-ären Beziehungen und dann auf, wenn ein Knotentyp als Nachfolgerknoten verschiedener Knotentypen auftritt.
- ❑ Für die Darstellung einer N-M-Beziehung muß zwischen alternativen Modellierungsvarianten gewählt werden, die sich aus einer relativen Anordnung der beteiligten Rekordtypen in der Baumstruktur ergeben.
- ❑ Anwendungsabhängige Wahl der Modellierungsvariante unter erforderlicher Kenntnis der bevorzugten Zugriffsarten. (s. nächste Folie)

HDM: Assoziationen (2)

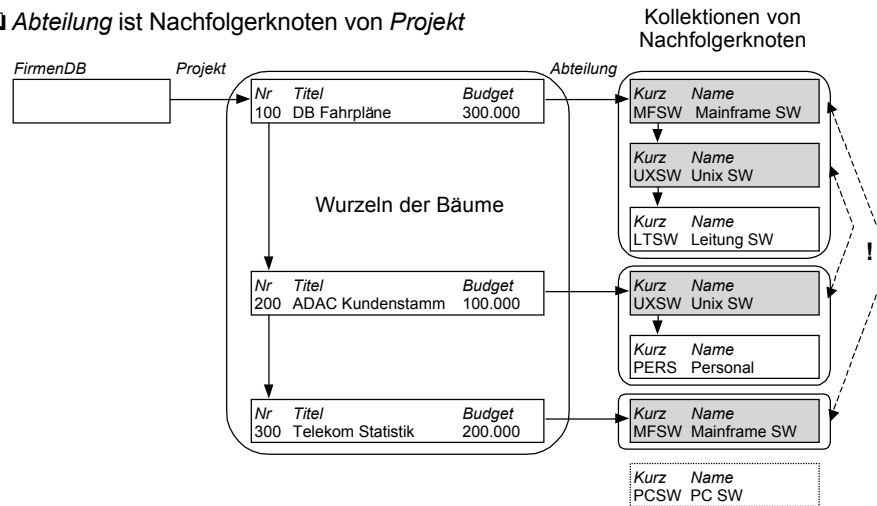
Schemadiagramme für drei Modellierungsvarianten einer N:M-Beziehung zwischen Projekten und Abteilungen:



HDM: Assoziationen (3)

Schemaalternative A (Kopiersemantik für Abteilungen):

- Abteilung ist Nachfolgerknoten von Projekt



Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.9

HDM: Assoziationen (4)

Schemaalternative A: Fortsetzung

- Einfacher Zugriff auf alle Abteilungen, die an einem Projekt arbeiten
- Beim Zugriff auf alle Projekte, an denen eine Abteilung arbeitet, muß die gesamte Bauebene (alle Projekte mit den dazugehörigen Abteilungen) durchsucht werden.
- Die Darstellung von N:M-Beziehungen erfordert die *Duplizierung von Information*, da jeder Rekord nur höchstens einen Elternknoten besitzen darf (im Beispiel auf der vorigen Folie kommen die Informationen zu den Abteilungen *Unix SW* und *Mainframe SW* jeweils doppelt in der Datenbank vor, da jede dieser Abteilungen an zwei Projekten arbeitet).
- Problem: Die Abteilung *PC Software*, die aktuell nicht an einem Projekt arbeitet, kann nicht in der Datenbank dargestellt werden, obwohl sie in der Anwendungsdomäne weiterhin existiert.

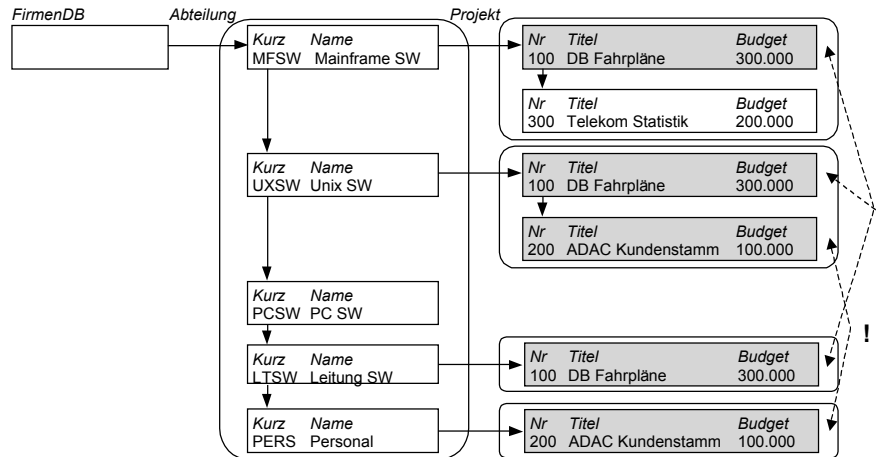
Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.10

HDM: Assoziationen (5)

Schemaalternative B (Kopiersemantik für Projekte):

- ❑ *Projekt* ist Nachfolgerknoten von *Abteilung*



Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.11

HDM: Assoziationen (6)

Probleme bei den Modellierungsalternativen A und B:

- ❑ Duplizierung von Information in der Datenbank erhöht sowohl den Speicherbedarf als auch die Gefahr von Inkonsistenzen
- ❑ Bei Aktualisierungsoperationen ist sicherzustellen, daß mit einem Datenobjekt zugleich auch alle evtl. existierenden Kopien geändert werden.
- ❑ Willkürliche Asymmetrie

Einführung in Datenbanksysteme

Das hierarchische Datenmodell 4.2.12

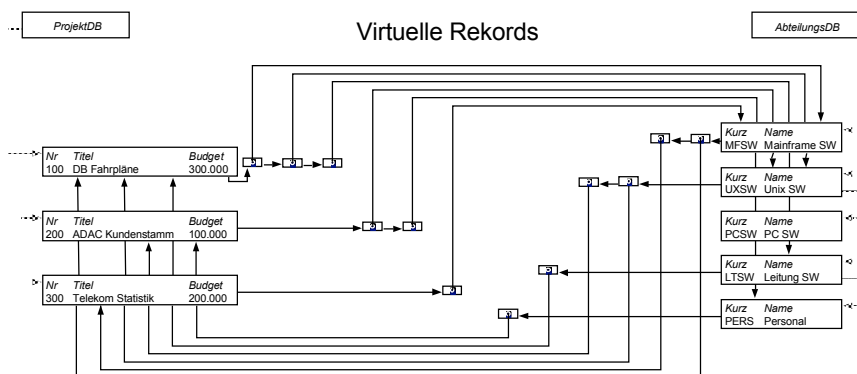
HDM: Assoziationen (7)

Schemaalternative C (Konzept des *Virtuellen Rekords*, Referenzsemantik):

- Ziel: Umgehung der Probleme der Informationsduplizierung und der Asymmetrie bei der Darstellung von Beziehungen.
- Mehr als ein hierarchisches Schema für eine Datenbank wird unterstützt: d.h., daß der gleiche Datenbestand scheinbar gleichzeitig gemäß verschiedener HDM-Baumtypen organisiert wird (s. nächste Folie: Baumtypen *ProjektDB* und *AbteilungsDB* in den Modellierungsvarianten A und B *innerhalb* der Modellierungsvariante C).
- **Zusätzliche Indirektion:**
 - Anstatt einen Rekord *r* zu duplizieren wird ein *virtueller Rekord* eingeführt, der einen Verweis auf den Rekord *r* enthält.
 - Der tatsächliche Rekord tritt nur einmal in einem Baum auf. An den anderen Stellen werden nur die virtuellen Rekords verwendet, von denen auch mehrere auf dasselbe Datenobjekt verweisen können.
 - Jeder virtuelle Rekord kommt wiederum nur einmal in der Datenbank vor.

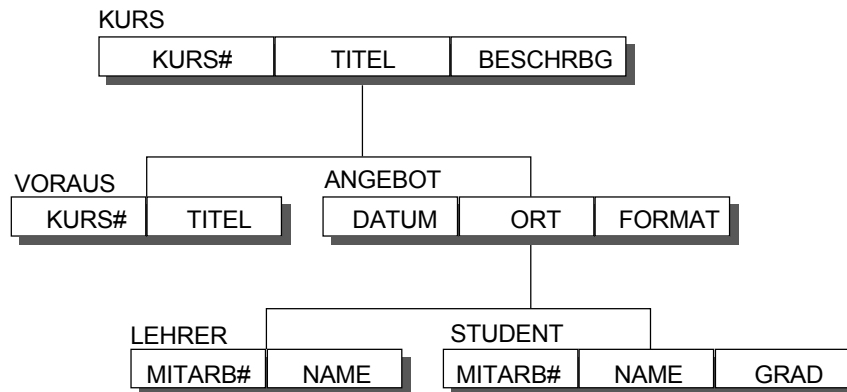
HDM: Assoziationen (8)

Zustand der Firmendatenbank (HDM-Modellierungsvariante C, Referenzsemantik):



HDM: Datendefinition (1)

Beispiel: Ausbildungsdatenbank



HDM: Datendefinition (2)

Datenbankbeschreibung (DBD) der Ausbildungsdatenbank (AUSBPDB):

❑ Hier: "Konzeptuelle DDL" für IBM's Information Management System (IMS)

❑ Vgl. C.J. Date. An Introduction to Database Systems. 3. Auflage. Addison-Wesley, Reading, Massachusetts, 1981.

1	DBD	NAME=AUSBPDB
2	SEGM	NAME=KURS, BYTES=256
3	FIELD	NAME=(KURS#, SEQ), BYTES=3, START=1
4	FIELD	NAME=TITEL, BYTES=33, START=4
5	FIELD	NAME=BESCHRBG, BYTES=220, START=37
6	SEGM	NAME=VORAUSS, PARENT=KURS, BYTES=36
7	FIELD	NAME=(KURS#, SEQ), BYTES=3, START=1
8	FIELD	NAME=TITEL, BYTES=33, START=4
9	SEGM	NAME=ANGEBOT, PARENT=KURS, BYTES=20
10	FIELD	NAME=(DATE, SEQ, M), BYTES=6, START=1
11	FIELD	NAME=ORT, BYTES=12, START=7
12	FIELD	NAME=FORMAT, BYTES=2, START=19
13	SEGM	NAME=LEHRER, PARENT=ANGEBOT, BYTES=24
14	FIELD	NAME=(MITARB#, SEQ), BYTES=6, START=1
15	FIELD	NAME=NAME, BYTES=18, START=7
16	SEGM	NAME=STUDENT, PARENT=ANGEBOT, BYTES=25
17	FIELD	NAME=(MITARB#, SEQ), BYTES=6, START=1
18	FIELD	NAME=NAME, BYTES=18, START=7
19	FIELD	NAME=GRAD, BYTES=1, START=25

HDM: Anfragen

- Die Formulierung von Anfragen geschieht im HDM satzorientiert und prozedural wie im NDM.
- Die Operationen werden in einer Gastsprache eingebettet.
- Die Kommunikation zwischen der Datenbank und einer Applikation findet ebenfalls wie im Netzwerkdatenmodell über eine *UWA (User Work Area)* statt.
- Die NDM-Operationen *find* und *get* sind in der *get*-Funktion im HDM zusammengefaßt.
- Operationen zum Durchsuchen von Teilbäumen werden unterstützt.
- Suchoperationen können Prädikate als Parameter enthalten, die den Suchraum einschränken.
- Navigation immer von der Wurzel aus abwärts

HDM: Aktualisierungsoperationen

- Operationen zum Einfügen, Löschen und Modifizieren der Datenbank
- Neue Bäume werden durch Einfügen eines neuen Rekords des Wurzelknotentyps erzeugt.
- Beim Einfügen eines neuen Rekords in eine Hierarchie wird dessen Position durch Umsetzen der entsprechenden Aktualitätszeiger festgelegt.
- Das Löschen und Modifizieren von Elementen der Datenbank geschieht in zwei Schritten:
 1. Lokalisierung des betreffenden Rekords: Benutzung einer Variante der o.g. *get*-Operation
 2. Ausführung der Lösch- oder Modifikationsoperation auf dem aufgefundenen Rekord: Gelöscht wird dabei der gesamte Teilbaum, dessen Wurzel dieses Rekord ist. Die Modifikation von Attributwerten geschieht wie beim NDM im Kommunikationsrekord durch eine geeignete Zuweisungsoperation der Gastsprache und nachfolgendes Einbringen der Änderung in die Datenbank.

HDM: Bewertung

- Das HDM übernimmt alle Nachteile des NDM
- Zusätzliche Probleme bei der Modellierung von Zusammenhängen, die von einer streng hierarchischen Ordnung abweichen
- Sehr effizient bei eindeutig hierarchischer Problemstellung