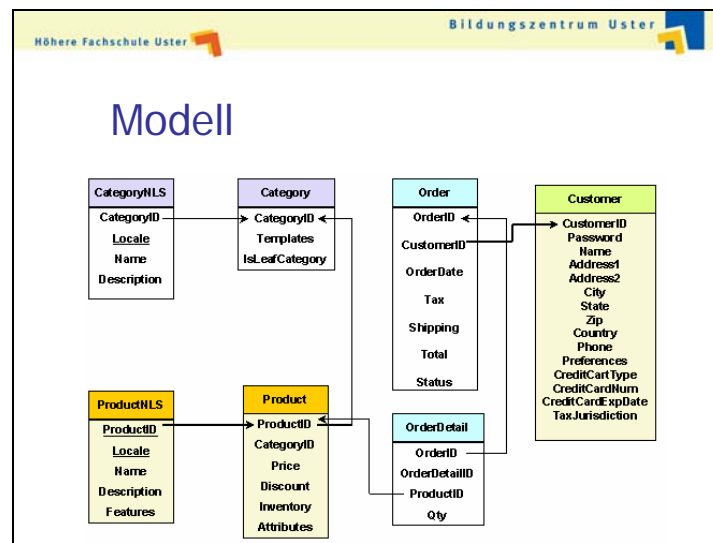


2 DATENMODELLIERUNG

2.1	Modell	2
2.2	3 – Schema - Architektur	4
2.2.a	Datenunabhängigkeit	5
2.2.b	Externes Schema.....	6
2.2.c	Konzeptionelles Schema	7
2.2.d	Internes Schema	8
2.3	Entity-Relationship-Model.....	9
2.3.a	Entität.....	10
2.3.b	Entitätsmenge.....	11
2.3.c	Relationship	12
2.3.c.1	Kardinalität.....	13
2.3.d	Attribut.....	15
2.3.d.1	NULL.....	15
2.3.d.2	Primary Key	16
2.3.d.3	Foreign Key.....	17
2.3.e	Verbundinstrumente.....	18
2.3.e.1	Rekursion	19
2.3.e.2	Spezialisierung / Generalisierung.....	20
2.3.e.3	Aggregation.....	21
2.4	Relationales DB-Modell.....	22
2.4.a	Kreuztabellen	23
2.4.b	Normalisieren.....	24
2.4.c	1. Normalform.....	25
2.4.d	2. Normalform.....	26
2.4.e	3. Normalform.....	27
2.4.f	4. Normalform.....	28
2.5	Übersicht Vorgehensweise	29
2.6	Varia	30
2.6.a	System-Analyse	30
2.6.b	Integrität.....	31
2.7	Übungen.....	33
2.7.a	3-Schema-Architektur.....	33
2.7.b	ERM Fussballmeisterschaft	34
2.7.c	Konzeptionelles Schema	35
2.7.d	Verbundinstrumente.....	37
2.7.e	Normalisierung	38
2.7.f	Datenmodellierung, Kontoverwaltungs-System	40
2.7.g	Normalisierung Verlag.....	41
2.7.h	Datenmodellierung, Mietwohnungen.....	42
2.7.i	Datenmodellierung, Liegenschaftsverwaltung.....	44
2.7.j	Datenmodellierung, Spital	45
2.7.k	Datenmodellierung, Adressen.....	46
2.7.l	Datenmodellierung, Krankenhaus.....	48
2.7.m	Datenmodellierung, Textilfärberei.....	50
2.7.n	Normalisierung Tonträger.....	51
2.7.o	Datenmodellierung Stundenplan.....	52
2.7.p	Datenmodellierung Bibliothek	53

2.1 Modell

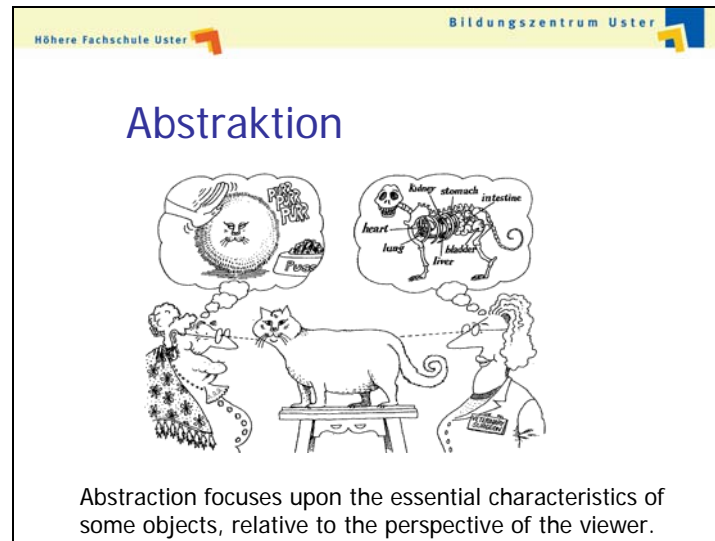


Datenmodell

- Das Datenmodell entspricht der Struktur der Daten. Die Daten selbst werden im Modell nicht festgehalten, lediglich deren Struktur.
- Mittels Datenmodell können Struktur, Inhalt und ergänzende semantische Aspekte des Datenbestandes entwickelt und festgehalten werden. Hierfür wurden in der Praxis unterschiedliche Methoden entwickelt. Eines der bekanntesten Modelle ist das **ERM** (Entity-Relationship-Model) .

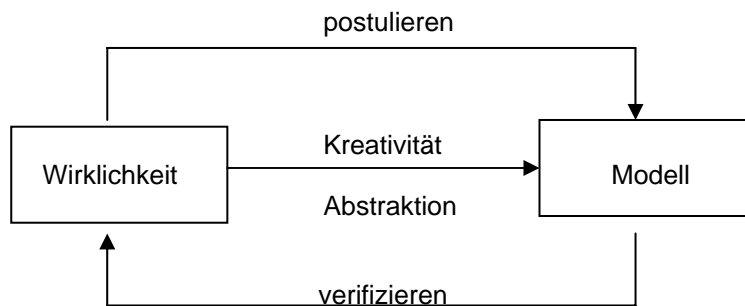
Charakteristiken von Modellen

- Ein Modell ist eine zweckorientierte, vereinfachte und strukturgleiche Abbildung der Wirklichkeit.
- Die Beziehung zwischen Modell und Wirklichkeit ist die Analogie.
- Ein Modell konzentriert sich auf das Wesentliche und reduziert so die Komplexität der Wirklichkeit.
- Ein Modell grenzt Unwesentliches aus -> Informationsverlust
- Ein Modell hat eine Systemgrenze. Da diese praktische nie gegeben ist, muss sie festgelegt werden. Wie die Systemgrenze gesetzt wird, ist eine Frage der Zweckmässigkeit.



Albert Einstein

Zur Aufstellung eines Modells genügt niemals das bloße Auflisten von Tatsachen und Erscheinungen. Es muss stets eine **freie Erfindung des menschlichen Geistes** hinzukommen, die dem Wesen der Dinge näher auf den Leib rückt. Man darf sich nicht begnügen mit der blossen Betrachtung, sondern muss **spekulativ** ein Modell postulieren und anschliessend verifizieren. Dieses Postulieren und Verifizieren ist ein iterativer Prozess.



Abstraktion

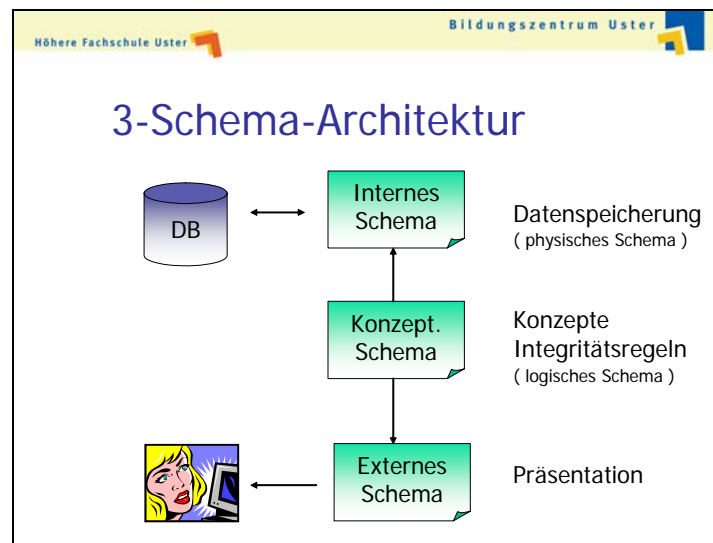
Zur Modellfindung gehört zwingend die Abstraktion.

Eine Abstraktion benennt die essentiellen Charakteristiken eines Objektes, welche dieses Objekt klar gegenüber allen anderen Arten von Objekten abgrenzt. Sie vermittelt scharf definierte Grenzen eines Konzepts, jeweils abhängig von der Perspektive des Betrachters.

Abstrahieren heisst:

- eine allgemein gültige Form finden
- gedanklich verallgemeinern
- eine Struktur für alle Anwendungsfälle definieren

2.2 3 – Schema - Architektur



Aufgrund der Komplexität, sowie der unterschiedlichen Sichtweisen der im Modellierungsprozess betroffenen Personen (Anwender, Entwickler, DBA) drängt sich eine den Sichtweisen der Personen angepasste Darstellung und Gliederung des Modells auf. Hierbei hat sich das 3-Schema-Modell (3-Schema-Konzept, 3-Schema-Architektur) des ANSI-SPARC-Komitees durchgesetzt.

Die 3 Ebenen heissen externe, konzeptionelle und interne Ebene. Sie haben klar voneinander getrennte Inhalte und Aufgaben, so dass man eine DB entwerfen kann, ohne sich ständig über das Gesamtproblem und die dazugehörige Komplexität den Kopf zerbrechen zu müssen.

Externes Schema

Das externe Modell beschreibt die reine Benutzersicht auf die Daten. Hier wird der für den Anwender sichtbare Teil der Daten aufbereitet, ein bestimmtes Stück des Kuchens. Dies ist sowohl datenschutztechnisch als auch organisatorisch sinnvoll. Ein Firmenmitarbeiter benötigt eventuell die Kontonummer eines Kollegen für die Überweisung, dessen Gehalt darf er jedoch nicht erfahren. Diese Sicht, auch View genannt, wird von SQL – einer normierten DB-Programmiersprache – unterstützt.

Internes Schema

Das interne Modell beschreibt die rein physischen Aspekte der DB. Hier sind die Zugriffspfade, die Such- und Sortiervverfahren etc. vermerkt, die einen wesentlichen Anteil an der Leistungsfähigkeit des gesamten DBS haben.

Konzeptionelles Schema

Das Verbindungselement, quasi die Pufferzone zwischen den beiden sehr gegensätzlichen Modellen ist das konzeptionelle Modell. Es stellt den logischen, von Benutzern und physischen Gegebenheiten unabhängigen Blick auf die DB dar. Dieses Schema wird unabhängig vom zu verwendenden DBS und der HW erstellt. Damit ist das konzeptionelle Modell frei von technischen Details und kann auch erstellt werden, wenn das Zielsystem noch nicht bekannt ist.

Das konzeptionelle Schema entspricht dem logischen Datenmodell und zeigt die Datenstruktur auf, enthält aber keine Daten.

2.2.a Datenunabhängigkeit



Einleitung

Ein wesentlicher Aspekt bei DB-Anwendungen ist die Unterstützung der Datenunabhängigkeit durch das DBS. Sowohl DB als auch Anwendungen haben in der Regel eine lange Lebensdauer, während derer die Datenstrukturen aus verschiedensten Gründen ständig modifiziert und erweitert werden. Das Konzept der Datenunabhängigkeit hat das Ziel, ein DBS von den notwendigen Änderungen der Anwendung abzukoppeln und umgekehrt.

Die Datenunabhängigkeit kann in 2 Aspekte aufgeteilt werden:

- Die **physische Datenunabhängigkeit** bedeutet, dass die konzeptionelle Sicht auf einen Datenbestand unabhängig von der für die Speicherung der Daten gewählten Datenstruktur besteht. Die technische Frage des Datenzugriffs – z.B. wie aus Datenfragmenten, die auf der Festplatte wild verstreut sind, eine Tabelle zusammengesetzt wird – spielt daher bei der Nutzung von Datenbanken so gut wie keine Rolle, und auch bei der Anwendungsentwicklung muss man sich kaum einmal damit beschäftigen.
- Die **logische Datenunabhängigkeit** hingegen koppelt die Datenbank von Änderungen und Erweiterungen der Anwendungsschnittstellen ab.


Zur Unterstützung der Datenunabhängigkeit wurde bereits in den 70er Jahren von einem ANSI-Gremium ein Drei-Ebenen-Schema-Architektur vorgeschlagen. Die dort vorgeschlagene Aufteilung in 3 Ebenen ist im Datenbankbereich inzwischen allgemein akzeptiert. Die drei Schemata oder Ebenen der Datenbankarchitektur können in gewissen Grenzen unabhängig voneinander gestaltet und betrachtet werden.

2.2.b Externes Schema

Höhere Fachschule Uster Bildungszentrum Uster

Externes Schema

Name	Titel	Jahr	ISBN
Meier	Datenbanken	1993	1-424-11
Schulze	Unix X	1998	3-452-12
Ibsen	Unix X	1998	3-452-12

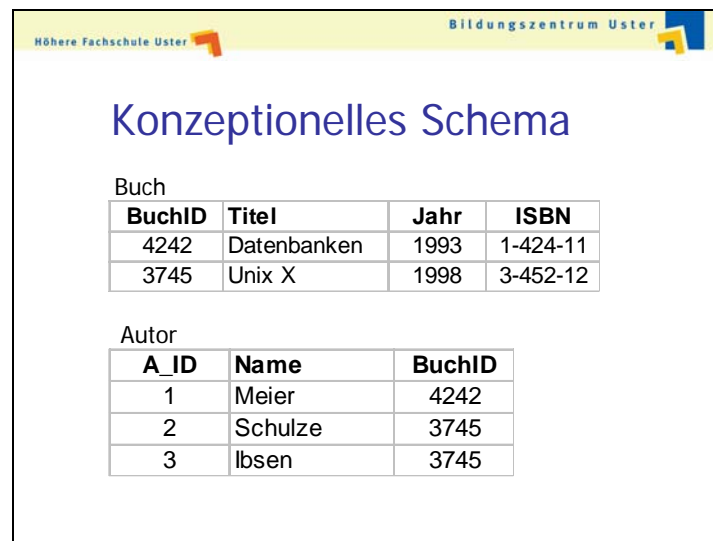


Wir wollen die Basisidee der Drei-Ebenen-Architektur im folgenden anhand einer kleinen Beispielmodellierung diskutieren. Dieses Beispiel enthält Daten über Bücher und Autoren.

- Ein Benutzer möchte sämtliche Autoren inklusive deren geschriebenen Werke aufgelistet haben.
- Aus Benutzersicht entspricht dies einer Tabelle.
- Diese externe Sicht kann in rel. DBS durch eine View definiert werden.

Für Datenbanken, die auf nicht vernetzten Einplatzrechnern installiert sind, gibt es häufig nur ein externes Schema, das mit dem konzeptionellen Schema identisch ist; mit andern Worten: Es gibt kein externes Schema.

2.2.c Konzeptionelles Schema



Konzeptionelles Schema

Buch

BuchID	Titel	Jahr	ISBN
4242	Datenbanken	1993	1-424-11
3745	Unix X	1998	3-452-12

Autor

A_ID	Name	BuchID
1	Meier	4242
2	Schulze	3745
3	Ibsen	3745

Die konzeptionelle Gesamtsicht erfolgt in relationaler Darstellung. Die Daten sind in zwei Tabellen gespeichert.

Das konzeptionelle Schema wird normalerweise mit einem -> **Entity-Relationship-Modell** beschrieben, das über die -> **Normalisierung** zum -> **logischen Datenbank-Schema** führt. Dieses Skript wird weiter hinten ausführlich auf diese drei Begriffe eingehen.

Zum konzeptionelles Schema gehören auch Integritätsregeln:

- Nur in der Tabelle Buch erfasste Bücher können bestellt werden.
- Eine BuchID darf nicht doppelt vergeben sein.
- Der Verkaufspreis muss grösser als der Einkaufspreis sein – ausser bei Sonderangeboten.
- Nicht immatrikulierte Studenten können keine Bücher ausleihen.

In der nicht sehr fernen Vergangenheit konnten solche Integritätsbedingungen nur dadurch gewährleistet werden, dass sie in den Applikationen programmiert wurden. Damit wurden natürlich viele Regeln mehrfach in den verschiedensten Source-Codes programmiert. Die Folge waren Widersprüche in verschiedenen Applikationen. Schlimmer noch: Das Problem der Integrität ist sehr schwer in den Griff zu bekommen, wenn es jedem einzelnen Programmierer überlassen bleibt. Diesem sind entweder nicht alle Regeln bekannt oder sie werden nicht konsequent befolgt. Das Problem wächst mit zunehmender Verfügbarkeit an Client-Software, die Konsequenzen für die gesamte DB sind nicht mehr überschaubar.

Die Forderung lautet daher: Alle Integritätsbedingungen müssen als Eigenschaften der Daten selbst behandelt werden. Folglich müssen sie in der DB selbst niedergelegt werden, also: **Integrität in die Datenbank.**

2.2.d Internes Schema

Höhere Fachschule Uster

Bildungszentrum Uster

Internes Schema

Page 386

Offset

0	4		16	20	24		36
1	Meier		4242	2	Schulze		3745

Page 497

Offset

0	4		16	20	24		36
3	Ibsen		3745				

Page

Alle Daten einer Datenbank sind auf einer Harddisk persistent gespeichert. Da aber der Prozessor, der einen Datenbankbefehl ausführt, immer nur auf Daten, die im RAM liegen, zugreifen kann, müssen die Daten zunächst von der Festplatte ins RAM geladen werden. Pro Zugriff wird immer genau eine Page (manchmal auch Sektor genannt, in der Regel 1, 2, 4 oder 8 kByte) von der Platte gelesen bzw. auf die Platte geschrieben.

I/O Cache

Im günstigsten Fall ist ein vom DBMS angeforderter Datensatz auf der gleichen, im RAM zwischengepufferten Page wie der letzte angeforderte Satz. So ist keine physische I/O-Operation notwendig. Um dies möglichst oft nutzen zu können, werden in der Regel viele solcher I/O-Pages im RAM gehalten. Dieser RAM-Bereich wird I/O-Cache genannt.

Organisationsformen

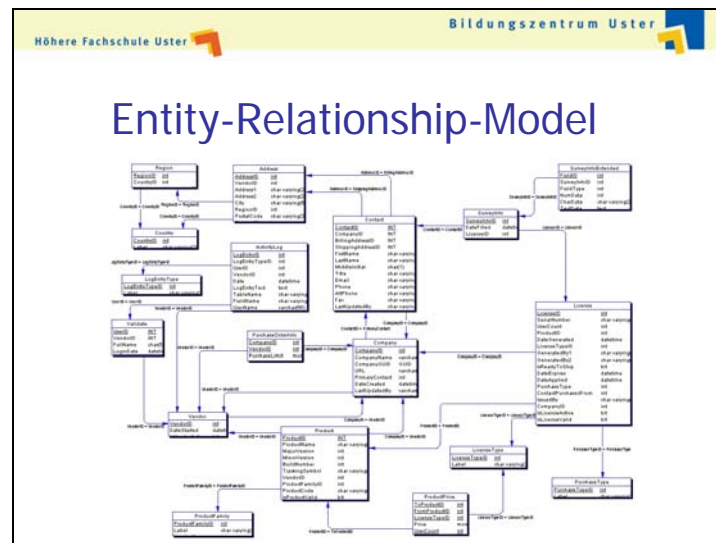
Eine Page beinhaltet ein oder mehrere Datensätze. Die physische Reihenfolge von Pages entspricht, speziell nach Reorganisationen und Wachstum, nicht mehr der Erfassungs-Reihenfolge. Eine logische Verkettung der Pages ist somit notwendig. Eine Tabelle kann intern (physisch) wie folgt abgespeichert werden:

- sequentiell sortiert
 - hash-organisiert
 - baumartig
- geordnet nach einem bestimmten Attribut
die Adressberechnung der Pages folgt einem Hash-Algorithmus
die Pages sind in einer hierarchischen Baumstruktur organisiert

Interne Ebene ist stark herstellerabhängig

Die DB-Hersteller können das interne Schema (Synonym: physisches Schema) auf recht unterschiedliche Weise realisieren. Auch bei Produkten eines Herstellers kann es von Version zu Version erhebliche Unterschiede in der Art der Speicherstrukturen geben, ohne dass dies bestehende Anwendungen tangieren darf.

2.3 Entity-Relationship-Model



Die Datenmodellierung legt fest, wie die Daten einer Anwendung konzeptionell strukturiert sind. In diesem Vorgang müssen verschiedene, zum Teil widersprüchliche Zielsetzungen und Bedürfnisse befriedigt werden, z.B.

- Das Datenmodell muss die notwendigen Informationen der Anwendung vollständig darstellen können, dabei ist die Bestimmung der Systemgrenze wichtig.
- Mit den gespeicherten Informationen im Datenmodell müssen sämtliche Geschäftsprozesse der Anwendung ausführbar sein. Eine Modellierung ohne jegliche Kenntnis der grundsätzlich gewünschten Funktionalität der Anwendung kann daher kein zweckmässiges Datenmodell liefern.
- Das Modell soll derart gebildet werden, dass auch zukünftige Bedürfnisse befriedigt werden können.

Das Erstellen eines Datenmodells kann daher kein fest vorgegebener, streng mathematischer Ablauf sein. Es ist viel mehr ein kreativer Prozess, in welchem die Abstraktion eine wichtige Rolle spielt, in welchem immer und immer wieder die Vor- und Nachteile unterschiedlicher Lösungsansätze verglichen werden. Der Datenmodellierer muss daher über Kreativität, Abstraktionsvermögen, Ausdauer und Erfahrung verfügen.

Aus den vorhergehenden Erläuterungen geht auch hervor, dass es kein Standardmodell geben kann, welches die Bedürfnisse einer bestimmten Branche unternehmensspezifisch abdeckt.

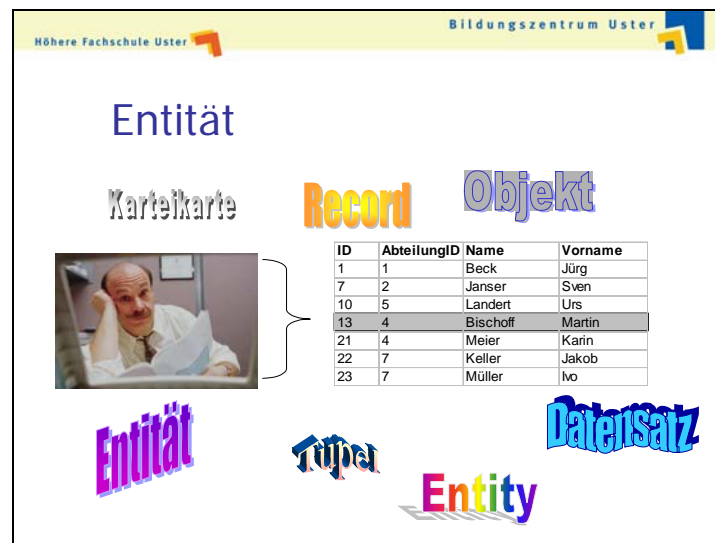
Entity-Relationship-Model (dt. Entitäten-Beziehungs-Modell)

- Das ERM ist speziell gut geeignet um Sachverhalte für Datenbankanwendungen zu modellieren.
- Das ERM gehört zum konzeptionellen Schema.
- Der Begriff ERD wird auch häufig verwendet. D steht für Diagramm. Gemeint ist dasselbe wie ERM.
- Das ERM wird oft auch als semantischen Modell bezeichnet.

Das ERM besteht auf folgenden Komponenten:

- Entität mit Attributen
- Entitätsmenge
- Beziehung (Relationship)

2.3.a Entität



Definitionen

- Eine Entität ist ein individuelles und identifizierbares Exemplar einer Sache, einer Person oder eines Begriffs aus der realen oder gedachten Vorstellungswelt.
- Eine Entität ist eine eigenständige Einheit, die im Rahmen des zu betrachteten Modells eindeutig identifiziert werden kann. Dieses Identifizierungsmerkmal wird als Schlüssel (engl. Key) bezeichnet.
- Eine Entität ist ein Objekt der realen oder der Vorstellungswelt, über das Informationen zu speichern sind.

Eine Entität kann folgendes sein:

- ein Gegenstand, z.B. eine Auto
- eine Person, z.B. ein Mitarbeiter einer Firma
- ein Ereignis, z.B. ein Fussballmatch
- eine abstrakte Grösse, etc.

Merke

- Eine Entität wird durch eine Menge von Eigenschaften (Attributen) beschrieben.
- Eine Eigenschaft hat einen Bezeichner und einen Wert.
- Die Eigenschaften einer Entität können geändert werden.

Da Entitäten jeweils nur einen Ausschnitt der Wirklichkeit wiedergeben – nämlich den Ausschnitt, für den wir uns in einem bestimmten Anwendungsbereich interessieren – ist es sehr wichtig, festzulegen, durch welche Attribute wir die Entitäten beschreiben. So sind die Attribute, die eine Person als Patienten in einer Arztpraxis beschreiben, andere als die Attribute, die dieselbe Person als Mitarbeiter in einem Unternehmen beschreiben. Daneben gibt es eine Unzahl von Attributen, für die sich glücklicherweise niemand interessiert.

Synonyme

Tupel
Datensatz
Karteikarte
Record
Objekt
Entität
Zeile, Reihe (row)

Ursprung

Relationen Modell
Programmiersprachen
aus Grossvaters Zeiten
Programmiersprachen, Pascal
reale Welt, OO-Programmiersprachen
ERM
SQL

2.3.b Entitätsmenge

The diagram illustrates the concept of an Entity Set. It features a central photo of a group of people, with a bracket connecting it to a table of employee data. The table is titled 'Entitätsmenge' and 'Relation'. The table columns are ID, AbteilungID, Name, and Vorname. The table data is as follows:

ID	AbteilungID	Name	Vorname
1	1	Beck	Jürg
7	2	Janser	Sven
10	5	Landert	Urs
13	4	Bischoff	Martin
21	4	Meier	Karin
22	7	Keller	Jakob
23	7	Müller	Ivo

Bei der Modellierung werden nicht die einzelnen Entitäten selbst dargestellt, sondern es werden Mengen aus Entitäten gleicher Art gebildet, sogenannte Entitätsmengen (engl. entity set).

Definitionen

- Eine Entitätsmenge ist eine eindeutig benannte Kollektion von Entitäten gleichen Typs.
- Eine Entitätsmenge entspricht einer zweidimensionalen Tabelle mit einem Primary Key.

Beispiele

- Die Menge aller zu einem festen Zeitpunkt in einem Unternehmen angestellten Mitarbeiter.
- Die Menge aller Studenten an einer Schule bilden eine Entitätsmenge.

In der Praxis werden die Begriffe Entität und Entitätsmenge leider häufig als Synonyme verwendet. Eine klare Trennung der Begriffe ist aber zweckdienlich, da z.B. eine Schulklasse eine einzelne Entität sein kann, aber auch eine Entitätsmenge, nämlich eine Menge von Schülern. Im zweiten Fall ist der Schüler die Entität.

Entitäten und Entitätsmengen können zusätzlich als unabhängige oder abhängige Entitäten, bzw. Entitätsmengen klassifiziert werden. Eine abhängige Entität ist von der Existenz einer andern Entität abhängig, kann also nur dann existieren, wenn diese referenzierte Entität auch existiert. Als Beispiel für eine abhängige Entität kann z.B. ein Aktiendepot dienen. Es darf nur dann existieren, wenn diesem ein Kunde zugeordnet ist (Depots ohne Eigentümer gibt es nicht).

Merke

- Die Anzahl der Elemente einer Entitätsmenge ist zu jedem Zeitpunkt durch die tatsächlich vorhandenen Entitäten gegeben – diese Menge kann sich zu jedem Zeitpunkt ändern.
- Die Reihenfolge der Entitäten innerhalb der Entitätsmenge ist irrelevant.
- Als Symbol für eine Entitätsmenge wird in den meisten Notationen ein Rechteck verwendet.
- Eine Entitätsmenge ist eine **Kernentitätsmenge**, wenn es möglich ist, Entitäten hinzuzufügen, ohne dass auf andere Entitätsmengen geachtet werden muss, d.h. die Entitätsmenge darf keinen Fremdschlüssel enthalten.

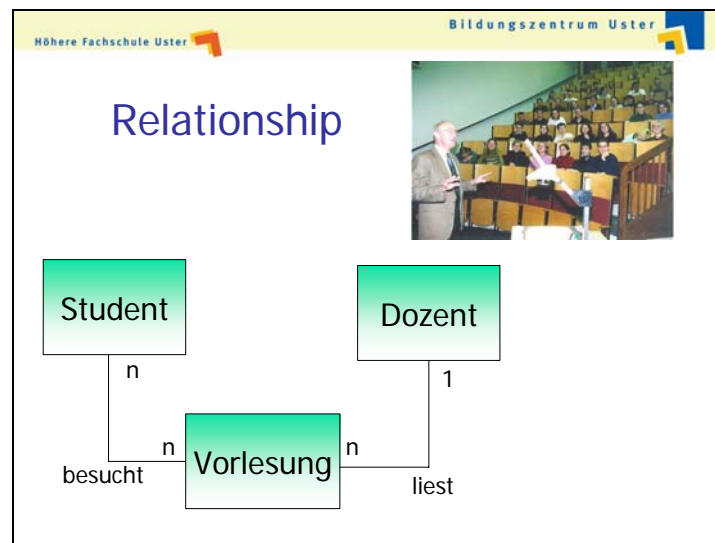
Synonyme

Tabelle
Kartei
Relation
Entitätsmenge
Objektmenge
Entity set

Ursprung

logischen DB-Schema
aus Grossvaters Zeiten
Relationen Modell
ERM
reale Welt
engl.

2.3.c Relationship



Die einzelnen Entitätsmengen einer Datenbasis dürfen nicht isoliert betrachtet werden, da zwischen ihnen diverse Beziehungen bestehen können. Die Anzahl der möglichen Beziehungstypen ist beschränkt und ergibt sich aus der Kombination der möglichen Assoziationstypen.

Definition

Eine Beziehung (engl. relationship) assoziiert wechselseitig mindestens zwei Entitäten.

Merke

- Eine Beziehung wird durch eine Linie dargestellt.
- Auf der Linie ist die Bezeichnung der Beziehung enthalten. Auf diese Benennung der Beziehung kann verzichtet werden, falls sie aus dem Zusammenhang eindeutig ersichtlich ist, muss aber immer dann erfolgen, wenn diese nicht a priori klar ist.

Assoziation

Assoziation bedeutet, dass eine Entität eine andere Entität kennt und mit ihr in Wechselwirkung steht. Für jede Beziehung wird angegeben, in welchem Mengenverhältnis die Entitätsmengen zueinander stehen.

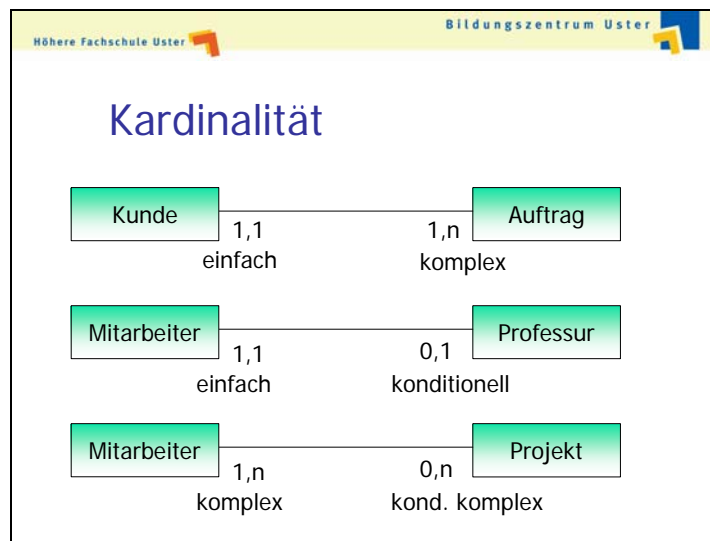
Beispiel (Abbildung oben)

- **Ein** Student besucht **mehrere** Vorlesungen (natürlich nicht gleichzeitig).
- **Ein** Dozent liest **mehrere** Vorlesungen.
- **Eine** Vorlesung wird von genau **einem** Dozenten gehalten, aber von **mehreren** Studenten besucht.

Beziehung

Jede Assoziation (Tabelle A zu Tabelle B) besitzt auch eine Gegenassoziation (Tabelle B zu Tabelle A). Kombiniert man diese zwei Assoziationen miteinander, so spricht man von einer Beziehung.

2.3.c.1 Kardinalität

**Definition**

- Die Kardinalität dient einer genaueren Bestimmung der Mengenverhältnisse zwischen 2 Entitätsmengen.
- Sie legt für jede Richtung fest, wie viele Entitäten an einer Beziehung teilnehmen (müssen).
- Dies geschieht über die Angabe eines Minimums (0 oder 1) und eines Maximums(1 oder n).

Beispiel

- Leserichtung Auftrag -> Kunde: **Ein** Auftrag hat genau einen Kunden (min = 1, max = 1)
- Leserichtung Kunde -> Auftrag: **Ein** Kunde kann beliebig viele Aufträge haben (min = 1, max = n)





Aus den möglichen Kardinalitäten entstehen grundsätzlich vier Arten von Assoziationen:

- einfach (genau eine)
- konditionelle (keine oder eine)
- komplexe (eine oder mehrere)
- konditionell-komplexe (keine, eine oder mehrere)

Ausgehend von diesen vier verschiedenen Assoziationstypen sind also zwischen 2 Tabellen $4 \times 4 = 16$ Beziehungstypen möglich:

	1,1	0,1	1,n	0,n	Beziehungstyp
1,1	1,1 : 1,1	0,1 : 1,1	1,n : 1,1	0,n : 1,1	<- hierarchisch
0,1	1,1 : 0,1	0,1 : 0,1	1,n : 0,1	0,n : 0,1	<- konditionell
1,n	1,1 : 1,n	0,1 : 1,n	1,n : 1,n	0,n : 1,n	<- netzwerkförmig
0,n	1,1 : 0,n	0,1 : 0,n	1,n : 0,n	0,n : 0,n	(n : n)

Bereits hier sei gesagt, dass $n : n$ Beziehungen im relationalen Datenmodell nicht erlaubt sind und umgewandelt werden müssen.

Kardinalität, Notationen				
Bereich	ERM	Chen	Zehnder	Bedeutung
0,1		1	c	konditionell
0,n		N	mc	konditionell-komplex
1,1		1	1	einfach
1,n		N	m	komplex

Im Bereich der Darstellung von logischen Datenstrukturen gibt es keine einheitlichen Notationen und schon gar keine verankerte internationale Norm. Die oben stehende Tabelle ist daher nur eine Zusammenstellung der gebräuchlichsten Notationen.

Bei einer **einfachen Assoziation** steht jedes Element der Entitätsmenge A jederzeit mit genau einem Element der Entitätsmenge B in Beziehung.

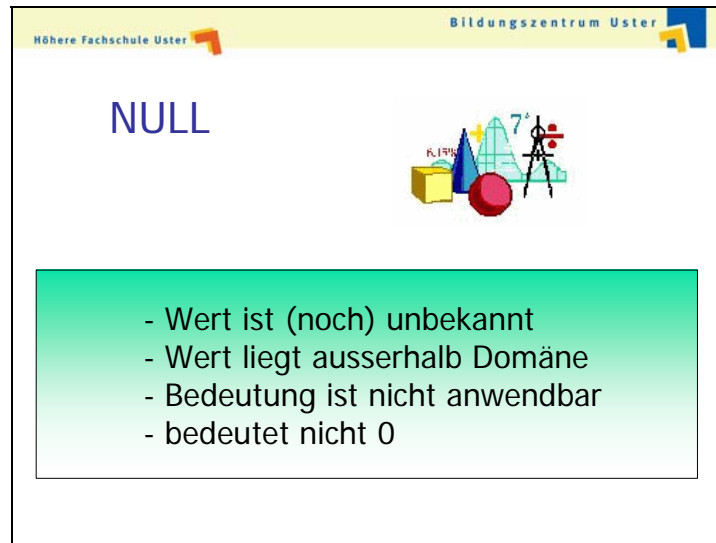
Bei einer **konditionellen Assoziation** steht ein Element der Entitätsmenge A jederzeit mit höchstens einem Element der Entitätsmenge B in Beziehung.

Bei einer **komplexen Assoziation** steht jedes Element der Entitätsmenge A jederzeit mit beliebig vielen Elementen der Entitätsmenge B in Beziehung.

Bei einer komplexen Assoziation muss zusätzlich unterschieden werden, ob auch zulässig ist, dass zu einem Zeitpunkt keine Assoziation besteht, das entspricht dann einer **konditionell-komplexen Assoziation**.

2.3.d Attribut

2.3.d.1 NULL



Attribut

Ein Attribut beschreibt eine bestimmte Eigenschaft (Merkmal), die für jede Entität einer Entitätsmenge einen bestimmten Wert aufweist. Das Attribut ist die eigentliche Speicherzelle, in welcher Informationen gespeichert sind. Für alle Attribute ist ein geeigneter Datentyp (Integer, Real, Zeichenkette, ...) festzulegen.

Wie bei einer Variablen ist bei einem Attribut zwischen seinem Namen und seinem Wert zu unterscheiden. Während der Name ein Attribut benennt und identifiziert, gibt der Wert die konkrete Ausprägung des Attributes für eine bestimmte Entität an. Welche diskreten Werte ein Attribut annehmen kann, legt sein Wertebereich (domain) fest.

Wert für ein Attribut fehlt

In relationalen DBS ist es möglich, dass für einen Datensatz der Wert für ein Attribut fehlt. Wir sagen: „Das Attribut ist NULL“, oder „Das Attribut hat den Wert NULL“.


Bedeutung

- NULL kann für ein Attribut stehen, für das in der Realität kein Wert existiert.
z.B. Telefonnummer für Personen ohne Telefon
- NULL kann für ein Attribut stehen, das in der Realität einen Wert hat, der aber nicht bekannt ist.
z.B. Energiewert eines Lebensmittels, dessen Kalorien-Gehalt nicht bekannt ist.
- NULL kann für ein Attribut stehen, dessen Wert ausserhalb des Attribut-Wertebereichs ist.
z.B. Geschlecht eines Zitters, Wertebereich lässt nur männlich oder weiblich zu
- Daneben sind weitere semantische Interpretationen von fehlenden Werten möglich.
z.B. das fehlende Lieferdatum als Information, dass ein Auftrag noch nicht abgeschlossen ist
z.B. das fehlende Todesdatum als Information, dass die entsprechende Person noch lebt
- NULL ist nicht identisch mit dem numerischen Wert 0 und auch nicht mit dem Leerzeichen.


2.3.d.2 Primary Key

Höhere Fachschule Uster Bildungszentrum Uster

Primary Key



Nr	Name	BuchID
1	Meier	4242
2	Schulze	3745
3	Ibsen	3745



BuchID	Titel	Jahr
4242	Datenbanken	1993
3745	Unix X	1998

Primary- und Foreign Keys sind auch Attribute, die aber noch zusätzliche Aufgaben wahrnehmen:

Ausgangslage

Zum Verarbeiten und zur Erstellung der Beziehungen zwischen den Daten muss sichergestellt sein, dass jede Entität eindeutig identifiziert werden kann. Es muss daher festgelegt sein, welches Attribut eine einzelne Entität eindeutig identifiziert. Dieses Attribut wird Primary Key genannt. In einer Entitätsmenge können sich durchaus mehrere Attribute oder Attributskombinationen als PK eignen. In der Regel sollte versucht werden, einen Schlüssel mit möglichst wenig Attributen zu wählen, welcher zudem durch den Anwender möglichst leicht gehandhabt werden kann (Verständlichkeit). Es kann aber auch der Fall sein, dass eine Entitätsmenge kein für einen PK geeignetes Attribut aufweist. In diesem Fall muss ein künstlicher PK eingeführt werden, d.h. es wird ein weiteres, geeignetes Attribut aufgenommen.

Synonyme

Identifikationsschlüssel, Entitätenschlüssel, Schlüsselattribut, Primärschlüssel, PK

Zusammengesetzter Primärschlüssel

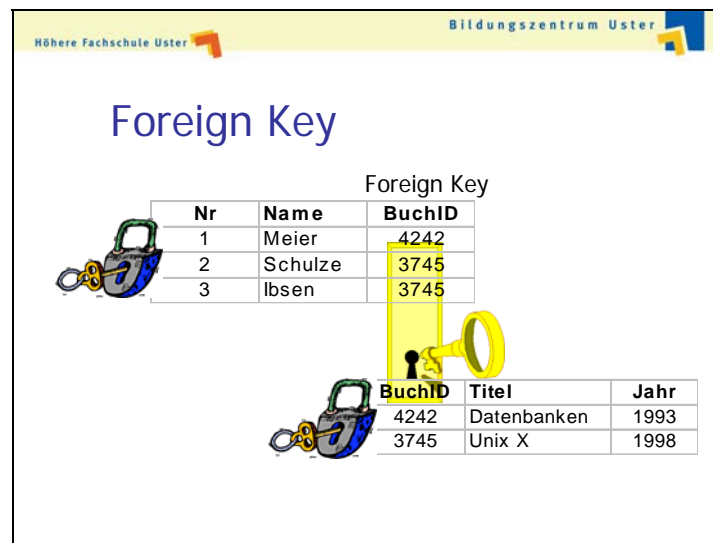
Unter Umständen lässt sich die eindeutige Identifikation nur mit einem zusammengesetzten Schlüssel erzielen, der aus mehreren Attributen besteht.

Merkmale

- einfach, kurz, möglichst nur ein Attribut
- eindeutig
- Wert darf nie NULL sein
- Integer ist besser als Zeichenfolge
- unveränderbar

Wichtig Im Datenbanksystem muss der PK explizit definiert sein, damit das DBS sicherstellen kann, dass nur eineindeutige Werte für diese Schlüsselattribut zugewiesen werden (Sicherstellung der Entitätsintegrität).

2.3.d.3 Foreign Key

**Ausgangslage**

Im ERM erfolgt die Verknüpfung zwischen Entitätsmengen über Werte. Es gibt nun aber keine vordefinierten Verknüpfungspfade zwischen Entitätsmengen. Diese Verknüpfungspfade können und müssen vom Datenmodellierer definiert werden.

Idee

- Im ERM wird der Foreign Key zur Erstellung der Beziehungen zwischen den Entitätsmengen verwendet.
- Der Fremdschlüssel verbindet seine Entitätsmenge mit einer andern Entitätsmenge über dessen Primary Key.
- Diejenigen Datensätze mit exakt identischen Werten in Foreign Key und Primary Key gehören logisch zusammen. Im obigen Beispiel gehört Meier zu den Datenbanken und Schulze/Ibsen gehören zu Unix X.

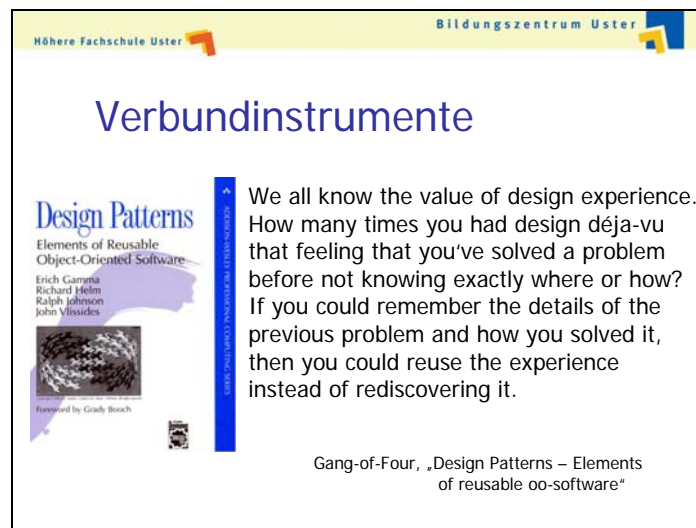
Einschränkungen

- Der Aufbau (Anzahl Attribute, Datentyp und Domäne) des Fremdschlüssels muss dem Aufbau des Primärschlüssels der referenzierten Entitätsmenge entsprechen.
- Die Namensgebung des Fremdschlüsselattributes ist zwar frei, dennoch empfiehlt es sich nach Möglichkeit den selben Attributnamen wie im Entitätenschlüssel zu verwenden.
- Im Gegensatz zum Primary Key können Foreign Keys in ihren Attributen den Wert NULL annehmen. In diesem Fall ist der Entität mit dem Fremdschlüssel keine Entität der referenzierten Entitätsmenge zugeordnet.

Referentielle Integrität

Für jeden von NULL verschiedenen Fremdschlüsselwert muss ein entsprechender Entitätenschlüsselwert der referenzierten Tabelle existieren. Das DBS überwacht diese Integrität.

2.3.e Verbundinstrumente



Idee

Experten gehen nicht jedes Problem von Grund auf neu an. Sie verstehen es, bereits gefundene Lösungen, die sie oder andere zuvor erfolgreich eingesetzt haben, wieder zu verwenden. Haben sie einmal eine gute Lösung gefunden, verwenden sie diese wieder und wieder. Solche Erfahrungen sind Teil dessen, was sie zu Experten macht.

Ein Verbundinstrument ...

- ist eine generische Lösung für eine Gruppe von ähnlichen Problemen.
- ist ein Zusammenspiel von mehreren Entitätsmengen (Ausnahme: Rekursion).
- ist vergleichbar mit den Design Pattern im SW-Engineering.
- Ist eine mentale Werkzeugkiste zur Entwicklung von SW.
- hilft bei der Bewältigung der Komplexität.
- ist kein Programmiertrick, kein Algorithmus und keine Programmiertechnik.
- beschreibt ein in unserer Umwelt beständig wiederkehrendes Problem und erläutert den Kern der Lösung für dieses Problem, so dass die Lösung beliebig oft angewendet werden kann.

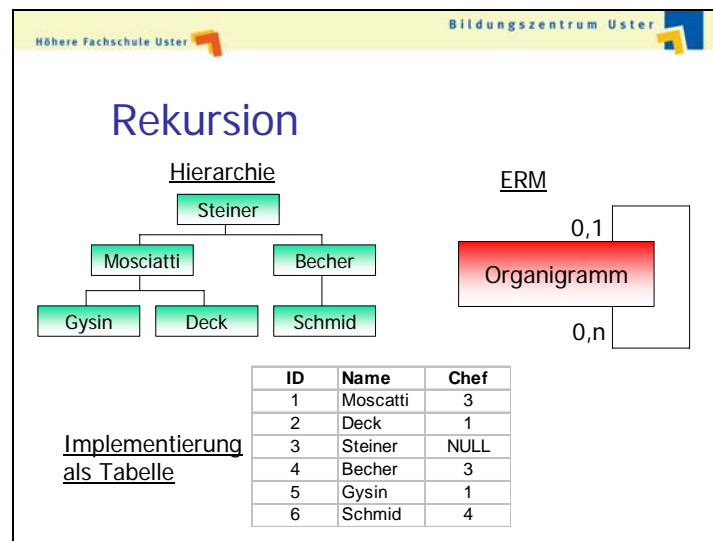
Benefit

- Mit Verbundinstrumenten profitieren wir von der langjährigen Erfahrung von andern Informatikern.
- Es geht darum, das Rad nicht immer wieder neu zu erfinden.
- Ein Verbundinstrument hat sich in einem bestimmten Kontext bewährt, und es könnte sich in einem ähnlichen Kontext auch bewähren.
- Verbundinstrumente helfen uns, Erfahrungen beim SW-Entwurf festzuhalten, so dass sie von andern verwendet werden können.

Synonyme

Verbundinstrumente, Design Pattern, Entwurfsmuster

2.3.e.1 Rekursion

**Idee**

Bei der Rekursion erstellt eine Entitätsmenge mittels einer 0,1 : 0,1 oder 0,1 : 0,n Beziehung via Fremdschlüssel eine Referenz auf sich selbst. Dabei sind Minimumkardinalitäten von 1 fast immer ungeeignet, da sonst eine Entität immer eine Referenz aufweisen muss, wobei zwangsläufig Zyklen, Kreise entstehen.

Paarige Verknüpfung

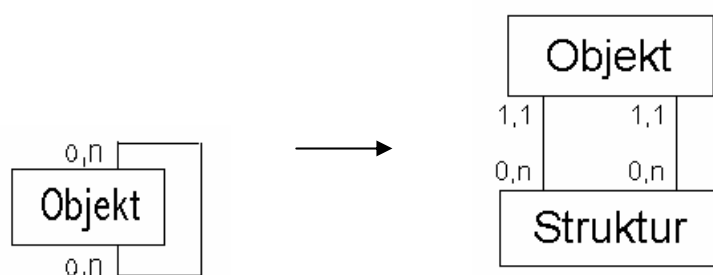
Eine rekursive 0,1 : 0,1 Beziehung lässt eine paarige Verknüpfung zu. So liesse sich z.B in einer Entitätsmenge Kunde festhalten, welches der Lebenspartner – welcher auch Kunde sein kann – des Kunden ist.

Hierarchie

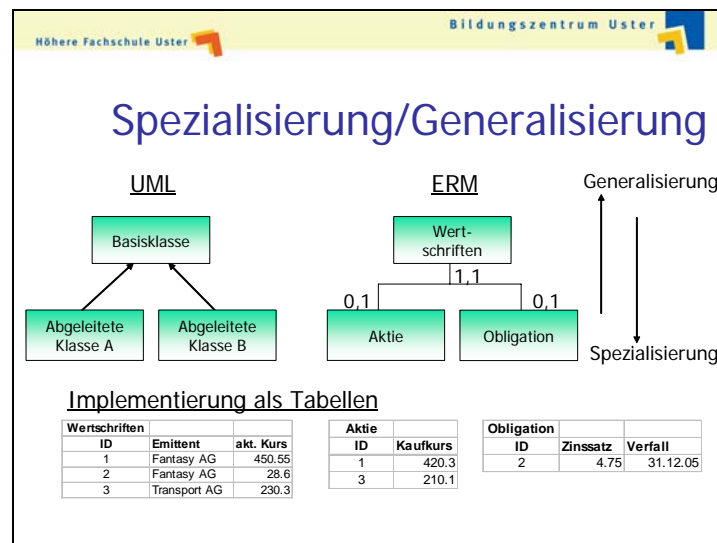
Mit einer rekursiven 0,1 : 0,n Beziehung wie oben abgebildet kann eine beliebige Hierarchie recht einfach abgebildet werden. Bei einer hierarchischen Grundstruktur hat eine Entität maximal eine übergeordnete Entität. Nur die Entität der obersten Hierarchiestufe hat keine übergeordnete Entität mehr.

Stücklistenproblematik

Die Stücklistenproblematik ist nicht eine Erfindung der Datenmodellierung. Sie ist ein Thema, das v.a. in der Fertigung auftaucht. Eine Stückliste beschreibt die Zusammensetzung eines Objekts, das aus mehreren Elementen besteht. Jedes Element kann wieder aus Unterelementen bestehen und / oder aber selber wieder ein Objekt sein. Ausserdem kann jedes Element in mehreren Objekten auftreten. Diese Stücklistenproblematik kann man mit einer rekursiven 0,n : 0,n Beziehung darstellen. Solch eine n : n Beziehung muss aber mit einer Hilfstabelle (Struktur) aufgelöst sein.



2.3.e.2 Spezialisierung / Generalisierung

**Anwendung**

'Ähnliche' Tabellen mit identischen Attributen sollten diese identischen Attribute in eine gemeinsame, übergeordnete Entitätsmenge auslagern.

Prinzip

Die Spezialisierung/Generalisierung besteht aus einer Superentitätsmenge und mehreren Subentitätsmengen. Die Superentitätsmenge enthält alle gemeinsamen Eigenschaften (Attribute), während die Subentitätsmengen nur die für sie spezifischen Eigenschaften enthalten. Die gesamten Eigenschaften einer Entität sind dabei nur bekannt, falls die Attribute beider Entitäten zusammen betrachtet werden.

Analogie

In der objekt-orientierten SW-Entwicklung entspricht diese Spezialisierung/Generalisierung dem Konzept der **Vererbung** mit einer Basisklasse und einer oder mehreren abgeleiteten Klassen.

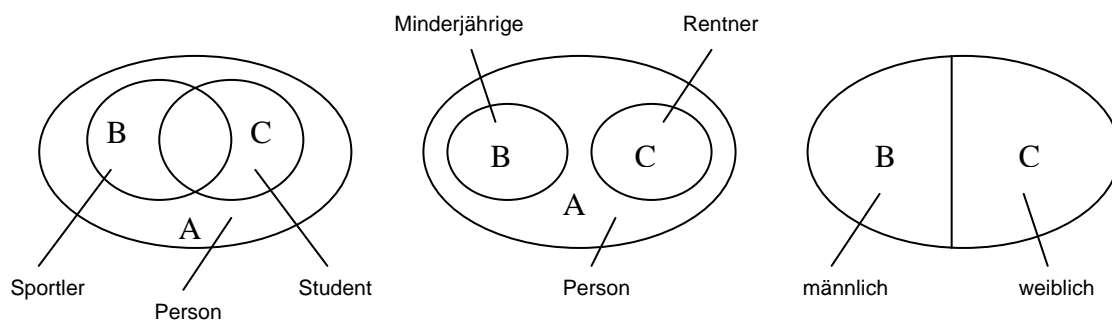
Hinweis

Viele NULL-Werte in einer Entitätsmenge sind häufig ein Hinweis darauf, dass die betroffene Entitätsmenge mittels einer Spezialisierung in eine oder mehrere Subentitätsmengen zerlegt werden könnte.

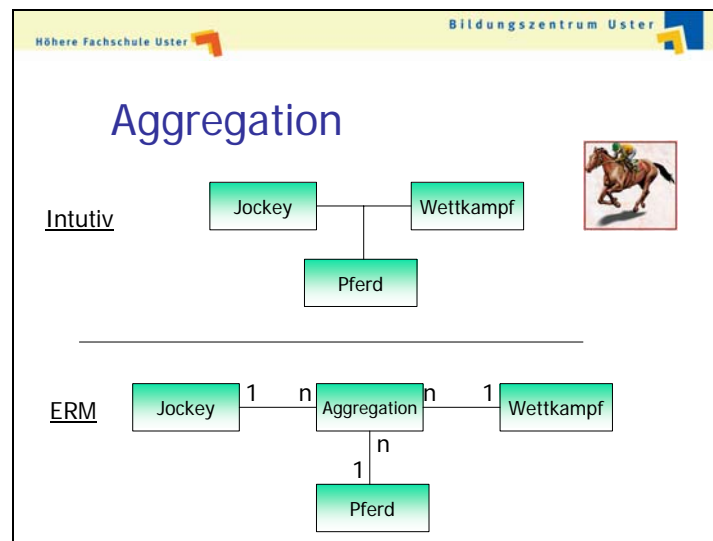
Überlappung

Diese Unter- und Obermengenbeziehungen lassen sich in drei verschiedene Fälle gruppieren:

- Subentitätsmengen mit zugelassener Überlappung
- Subentitätsmengen ohne Überlappung
- Superentitätsmengen mit vollständiger Überdeckung



2.3.e.3 Aggregation

**Idee**

Es gibt relativ viele Situationen, bei denen rein intuitiv ein Knotenpunkt zwischen den Beziehungen entsteht. Dies ist aber gemäss ERM nicht erlaubt. Solche Situationen können mit einer Aggregation gelöst werden. Eine Aggregation entspricht einer zusätzlichen Entitätsmenge – auch Beziehungsmenge genannt – im Knotenpunkt.

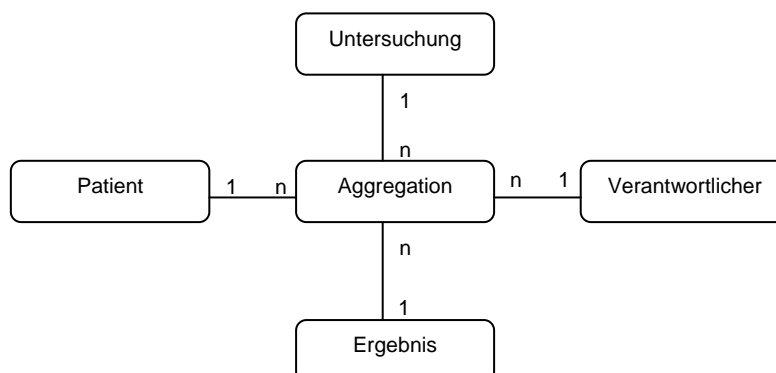
Mit diesem Verbundmuster kann jede beliebige Jockey – Pferd – Wettkampf Kombination erzeugt werden.

Netzstrukturen

Eine solche Aggregations-Entitätsmenge kann beliebig viele andere Entitätsmengen via Foreign Key referenzieren. Dadurch können alle denkbaren Netzstrukturen gebildet werden.

Jeder-mit-jedem Kombination

Ein Spital bietet Hunderte von Untersuchungen (HIV, Bilirubin, Hepatitis, etc.) an. Ausführende (Verantwortliche) sind Ärzte, Laborantinnen, MTA, etc. Es soll möglich sein, dass theoretisch mit jedem Patient jede Untersuchung durch jeden Verantwortlichen gemacht wird. Auch hier bietet sich die Aggregation als Verbundinstrument an.



2.4 Relationales DB-Modell



Idee

- Ein gutes Konzept zum Entwurf einer Datenbank besteht in der Verbindung der Methoden des Entity-Relationship-Modells und des relationalen Datenbankmodells.
- Das **ERM** eignet sich vor allem zur Bildung eines Datenmodells aus der Realität.
- Mit den Methoden des **relationalen Datenbankmodells** können die aus dem ERM gewonnenen Strukturen auf Redundanz untersucht werden.

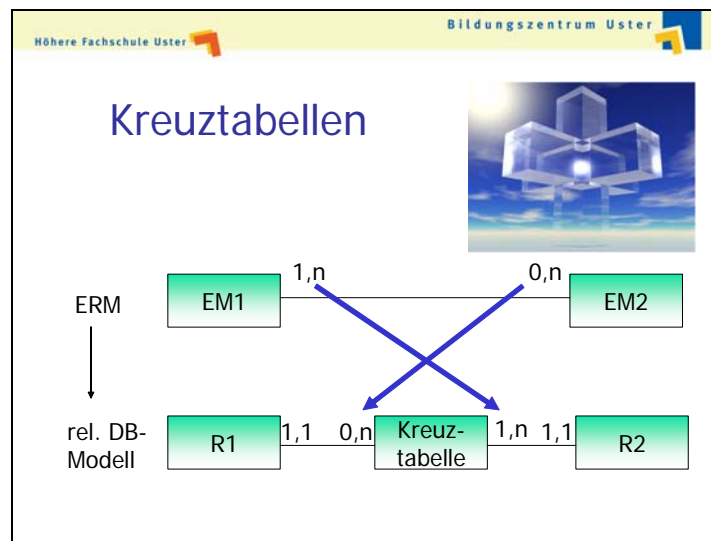
Vorgehensweise

1. Definition des Pflichtenheftes
2. Erstellen eines ERM's mit allen Entitätsmengen und deren Attributen, sowie den Beziehungen zwischen den Entitätsmengen und deren Kardinalitäten.
3. Überführung des ERM's in das relationale Datenbankmodell.
4. Entfernen der Redundanzen via Normalisierung (wird weiter hinten erklärt)

Ziel

Das Ergebnis ist ein redundanzfreies relationales Datenbankmodell. Dieses wird dann mit Hilfe von SQL auf dem Ziel-DBS implementiert.

2.4.a Kreuztabellen



Das ERM kann sehr einfach ins relationale DB-Modell überführt werden. Es gelten folgende Regeln:

Regeln

1. Der Name der Entitätsmenge wird zum Namen der Relation.
2. Attribute der Entitätsmenge werden zu Attributen der Relation.
3. Die Beziehungen und Kardinalitäten zwischen den Relationen werden aus dem ERM übernommen.
4. 1 : 1 Beziehungen können in eine Relation überführt werden.
5. n : n Beziehungen erhalten eine Zwischentabelle, die sogenannte Kreuztabelle.
6. Die Kreuztabelle erhält die Kardinalitäten über's Kreuz, die zwei referenzierten Relationen erhalten die Kardinalität 1,1.

n : n Beziehung

damit sind gemeint:

- 0,n : 0,n
- 0,n : 1,n
- 1,n : 0,n
- 1,n : 1,n

2.4.b Normalisieren

Höhere Fachschule Uster
Bildungszentrum Uster

Normalisieren



Unnormalisierte Form

*stud_nr	name	fach	dozent
1011	Müller	Rt, Ct, If	Ga, Sc, Ff
1209	Meier	Ct, Eo	Sc, Hi
981	Kramer	Eo, Nt, If	Hi, Su, Ff

Normalisieren von Relationen

Unter der Normalisierung versteht man ein systematisches Untersuchen einer Relation mit dem Zweck, eine qualitativ hochwertige Relation zu erhalten. Eine Relation ist dann normalisiert, wenn sie folgende Eigenschaften aufweist:

- Redundanzfreiheit
- keine Inkonsistenzen bei Einfüge-, Veränderungs- und Löschoperationen

Wissenswertes

- Beim Normalisieren steigt gleichzeitig die Verständlichkeit der Datenstruktur.
- Die Normalisierung findet auf der konzeptionellen Ebene statt.
- In der Normalisierung sind mehrere Normalformen bekannt. Jede Normalform stellt sicher, dass die Daten bestimmte Bedingungen einhalten. Am bekanntesten sind die 1. 2. und 3. Normalform (NF).
- In der Regel treten beim Erstellen des konzeptionellen Modells durch einen erfahrenen Modellierer gar keine Redundanzen auf.
- Verletzungen der Normalformen treten nur auf, falls inhaltlich unabhängige Entitäten in eine gemeinsame Entitätsmenge gepackt werden.

Definition: Unnormalisierte Relation

Eine Relation ist dann unnormalisiert, wenn am Kreuzungspunkt einer Spalte und einer Zeile kein einzelner Wert steht, sondern eine Gruppe oder Liste mehrerer Werte.

Diese Form ist schlecht zu handhaben und in den meisten DBS gar nicht verarbeitbar. Da solch eine Relation Redundanz enthält, ist sie auch anfällig auf Anomalien beim Verändern von Datensätzen.

Beispiel

Eine unnormalisierte Form ist nicht a priori schlecht, sie ist einfach in relationalen DBS nicht verarbeitbar. Aber der Mensch kann sie in geeigneter Darstellung recht gut lesen, das beste Beispiel dafür ist das Telefonbuch.

2.4.c 1. Normalform

Höhere Fachschule Uster

Bildungszentrum Uster

1. Normalform

*ID	stud_nr	name	fach	dozent
1	1011	Müller	Rt	Ga
2	1011	Müller	Ct	Sc
3	1011	Müller	If	Ff
4	1209	Meier	Ct	Sc
5	1209	Meier	Eo	Hi
6	981	Kramer	Eo	Hi
7	981	Kramer	Nt	Su
8	981	Kramer	If	Ff

Definition

Eine Relation ist dann in der ersten Normalform (1NF), wenn sie an den Kreuzungspunkten der Tupel und der Attribute jederzeit höchstens einen Wert aufweist.

Vorgehen

1. Eine Relation wird in die 1NF gebracht, indem jedes Tupel, das in einem Attribut eine Werteliste hat, auf mehrere Tupel verteilt wird.
2. Nach dieser Mehrfacheintragung ist allerdings der bisherige Primary Key in aller Regel nicht mehr eindeutig, daher muss dieser um geeignete Attribute erweitert werden:

Beispiel oben: neuer Primary Key *ID

Variante unten: zusammengesetzter Primary Key *stud_nr, *fach

Folgerung

Eine Relation in der 1NF ist immer noch für Defekte anfällig, weil sie Redundanzen aufweisen kann.

Variante

*stud_nr	name	*fach	dozent
1011	Müller	Rt	Ga
1011	Müller	Ct	Sc
1011	Müller	If	Ff
1209	Meier	Ct	Sc
1209	Meier	Eo	Hi
981	Kramer	Eo	Hi
981	Kramer	Nt	Su
981	Kramer	If	Ff

2.4.d 2. Normalform

Höhere Fachschule Uster

Bildungszentrum Uster

2. Normalform

*ID	stud_nr	name	fach	dozent
1	1011	Müller	Rt	Ga
2	1011	Müller	Ct	Sc
3	1011	Müller	If	Ff
4	1209	Meier	Ct	Sc
5	1209	Meier	Eo	Hi
6	981	Kramer	Eo	Hi
7	981	Kramer	Nt	Su
8	981	Kramer	If	Ff

Definition

Eine Relation befindet sich dann in der 2NF, wenn sie in der 1NF ist und zudem jedes Nichtschlüsselattribut vom gesamten Primärschlüssel abhängig ist, nicht aber von Schlüsselteilen.

Die 2NF kann daher nur verletzt sein, falls sich der Primärschlüssel aus mehreren Attributen zusammensetzt und der Datensatz Nichtschlüsselattribute enthält.

Idee

Sind Nichtschlüsselattribute vorhanden, die nur von einem Teil des zusammengesetzten Schlüssels bestimmt sind, so müssen diese in eine neue Relation ausgelagert werden. In unserer Variante ist „name“ (Nichtschlüsselattribut) nur von „stud_nr“ abhängig. „Dozent“ ist nur von „fach“ abhängig. In den meisten Fällen ist die Teilung der Entitätsmengen intuitiv klar und einfach.

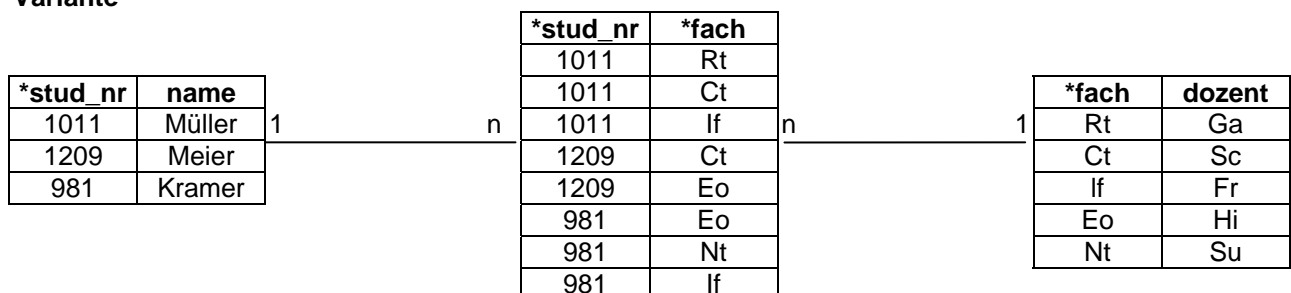
Zerlegungsprozess zur Bildung der 2NF

1. Kennzeichnung des Teilschlüssels (stud_nr) und der bereits davon abhängigen Nichtschlüsselattribute (name).
2. Entfernen der gekennzeichneten Nichtschlüsselattribute aus der Ausgangsrelation.
3. Bildung einer neuen Relation aus den entfernten Nichtschlüsselattributen und den gekennzeichneten Teilschlüssel aus der Ausgangsrelation.

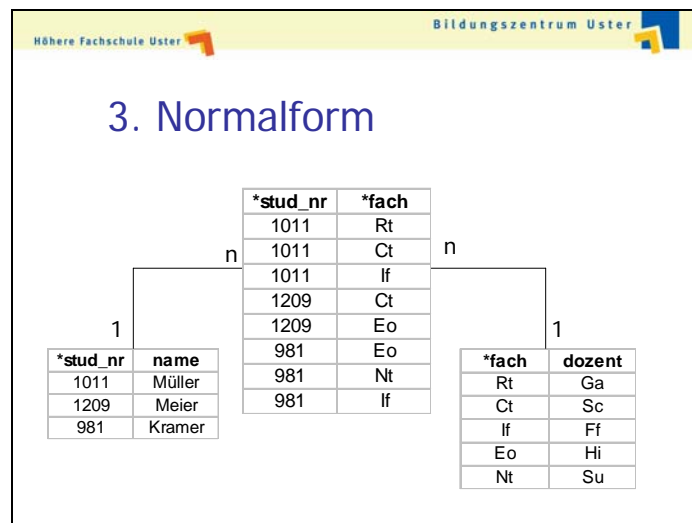
Folgerung

Auch ein Relation in der zweiten Normalform ist nicht unbedingt frei von Redundanz und somit anfällig für Defekte. Solche Redundanzen in der zweiten Normalform entstehen, wenn Nichtschlüsselattribute der Relation von anderen Nichtschlüsselattributen anhängig sind. Diese Art der Abhängigkeit nennt man **transitive Abhängigkeit**. Daher ist es notwendig eine weitere Normalform einzuführen, die 3NF.

Variante



2.4.e 3. Normalform



Idee

In der 3NF werden nun störende Abhängigkeiten zwischen Nichtschlüsselattributen gesucht und eliminiert.

Definition

Eine Relation befindet sich in der 3NF, wenn sie in der 2NF ist und keine transitiven Abhängigkeiten aufweist (keine Abhängigkeiten zwischen Nichtschlüsselattributen).

Beispiel

Im unserem Beispiel der 2NF ist das Nichtschlüsselattribut `dozent` vom Nichtschlüsselattribut `fach` abhängig und wird daher in eine separate Relation ausgelagert. Ebenso ist `name` nur von `stud_nr` abhängig.

Feststellungen

- Die Anzahl Datensätze in der ursprünglichen Relation bleibt immer erhalten. Nur in den „neuen“ Relationen kann/sollte die Anzahl Datensätze reduziert werden.
- Wenn eine Relation keinen zusammengesetzten Schlüssel hat und in der 1NF ist, ist sie automatisch auch in der 2NF. Nichtschlüsselattribute können ja nicht von einem Teilschlüssel abhängig sein, wenn gar kein Teilschlüssel existiert.
- Wenn eine Relation in der 2NF ist und weniger als 2 Nichtschlüsselattribute aufweist, ist sie automatisch auch in der 3NF. Nichtschlüsselattribute können ja nicht voneinander abhängig sein, wenn es nur 1 Nichtschlüsselattribut gibt.

Wichtig

- In jedem Fall kann die ursprüngliche Relation immer mit SQL aus den normalisierten Formen gewonnen werden.
- Information darf beim Normalisieren keine verloren gehen.

Variante

ist identisch

2.4.f 4. Normalform



Höhere Normalformen

Um die höheren Normalformen zu erklären, müssen zwei neue Begriffe eingeführt werden:

- 1) Als **lokale Attribute** werden alle Attribute bezeichnet, welche nur innerhalb einer einzigen Tabelle vorkommen und nicht Bestandteil eines PK oder FK sind.
- 2) Als **globale Attribute** werden alle Attribute bezeichnet, welche Bestandteil des PK oder FK sind.

Beispiel

Es können nun innerhalb einer Datenbasis Attribute existieren, welche weder lokalen noch globalen Charakter haben. Als Beispiel sollen folgende zwei Tabellen dienen:

Segelflugzeuge (SFNr, Fluggerät, Alter, Plätze, Spannweite)
 Motorflugzeuge (MFNr, Fluggerät, Alter, Plätze, Antriebsart)

Beide Tabellen befinden sich in der 3. Normalform und die gespeicherten Informationen sind redundanzfrei, solange niemand auf die Idee kommt, die Daten eines Motorseglers aufnehmen zu wollen. Ein Motorsegler müsste nämlich in beiden Tabellen erscheinen und bekäme zwei verschiedene PK's. Man sieht auch, dass in den Attributen Fahrzeugtyp, Alter und Plätze Redundanzen auftreten, weil für einen Motorsegler die entsprechenden Attributwerte zweimal vorkommen müssten. Gemäss Definition sind eben diese Attribute weder global noch lokal.

Lösung

Die Lösung dieses Problems liegt in der Erschaffung einer Superentitätsmenge:

Flugapparate (FNr, Fluggerät, Alter, Plätze)
 Segelflugzeuge (FNr, Spannweite)
 Motorflugzeuge (FNr, Antriebsart)

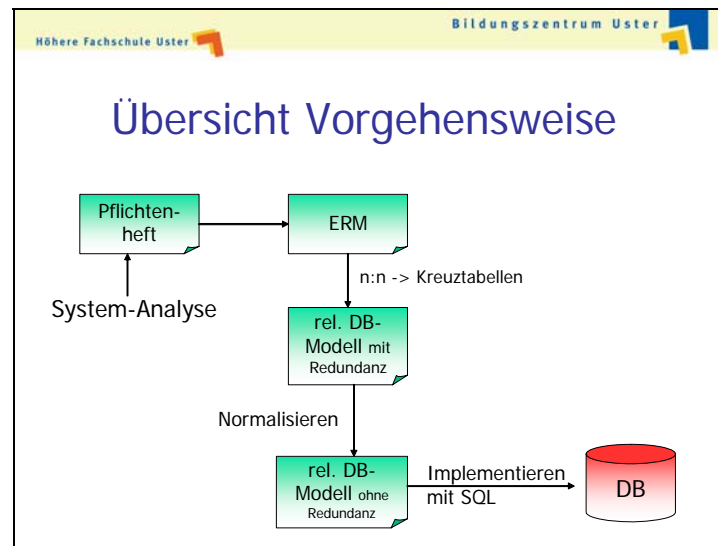
In dieser Datenbasis sind nur noch lokale und globale Attribute vorhanden und die Datenbasis befindet sich in der 4. Normalform.

Definition: Eine Datenbasis befindet sich in der 4. Normalform, wenn sich alle Tabellen in der 3. Normalform befinden und nur noch lokale und globale Attribute existieren.

Denormalisierung

Der Normalisierungsprozess scheint alle Datenmodellierungsprobleme zu lösen. Leider ist dem nicht so. Eine Datenbasis in der 4. Normalform kann sich in Sachen Performance durchaus als ineffizient erweisen. Dies liegt daran, dass mit steigendem Normalisierungsgrad immer mehr Tabellen entstehen. Dadurch werden Abfragen zunehmend komplexer und damit langsamer. Man kann dem entgegen wirken, indem man Tabellen wieder zusammenführt und so bewusst Redundanzen in Kauf nimmt. Solche Redundanzen müssen dann zwingend per Software überwacht werden.

2.5 Übersicht Vorgehensweise



Varianten

In der Literatur gibt es viele verschiedene Varianten bezüglich Vorgehensweise, diese hier ist nur eine davon. Sie ist an der Praxis orientiert.

Andere Vorgehensweisen unterscheiden nicht zwischen ERM und rel. DB-Modell. Es gibt Modelle, wo die Kreuztabellen im 2. Normalisierungsschritt eingeführt werden, wieder andere führen sie ganz am Schluss ein. Erfahrene Datenmodellierer können vom Pflichtenheft direkt auf das relationale DB-Modell schliessen, ERM und Normalisierung werden übersprungen. Was aber in jedem Fall gemacht werden muss, ist die Prüfung auf Redundanz, sei es nun intuitiv oder formal via Normalisierungsregeln.

Synonyme für relationales DB-Modell

DB-Schema, log. Schema, log. DB-Design, Datenmodell, Datenstruktur

Einordnung

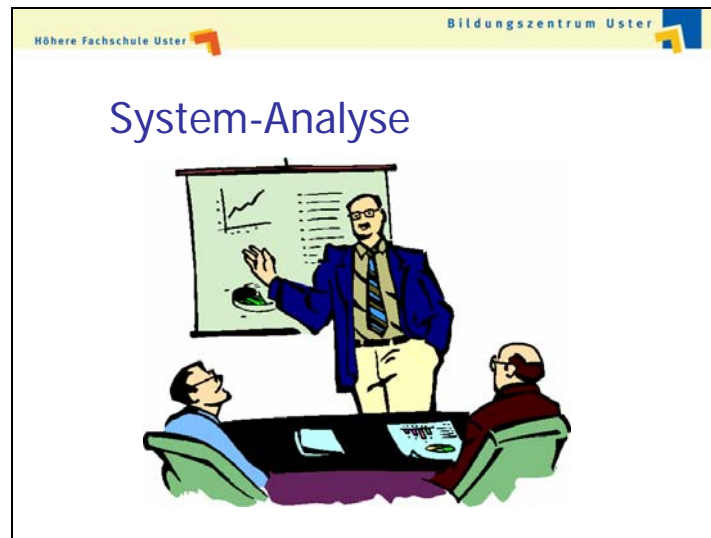
Das relationale DB-Modell gehört immer noch zum konzeptionellen Schema.

Integritätsregeln

Zum rel. DB-Modell gehören auch Integritätsregeln. Diese werden in Klartext festgehalten und dann direkt mit SQL implementiert (siehe Kapitel SQL).

2.6 Varia

2.6.a System-Analyse



Ziel

Die erste Phase ist die Systemanalyse, auch Anforderungsdefinition oder Pflichtenheft-Spezifikation genannt. Die Benutzerwünsche sind Ausgangspunkt der Systemanalyse. Ziel und Resultat ist die exakte, für Fachleute und Nichtfachleute gleichermaßen verständliche Formulierung der Forderungen an das zu erstellende Software-System.

Vorgehen

Der erste Schritt ist die Informationsbeschaffung, geht es doch darum, die Entitäten, die in der Realität existent sind, und deren Beziehungen unabhängig von sw- und hw-spezifischen Eigenschaften zu eruieren. Es handelt sich hier um einen länger dauernden Prozess grösster Wichtigkeit. Die beste Lösung nützt nichts, wenn nicht das richtige System entwickelt wird. Die Definition des richtigen Ziels ist wichtiger als die Auswahl der besten Lösung. Es ist daher unerlässlich, die Betroffenen (Auftraggeber, Benutzervertreter, Kunde ...) laufend mit einzubeziehen.

Voraussetzungen

Analysen setzen Fachwissen, Situationskenntnisse, Methodik, Psychologie und Planung voraus.

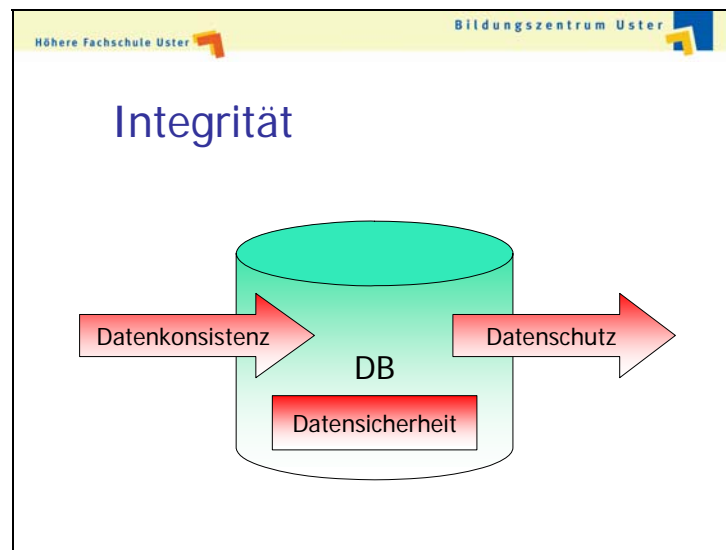
Experte vs Informatiker

- Die Anforderungsanalyse erhält die Berechtigung auch dadurch, dass SW-Entwickler normalerweise keine Experten in der Domäne (Anwendungsfach) sind.
- Anwender (Experten) und SW-Entwickler müssen sich gegenseitig kompetent machen.
- Während der Analyse treten eine Vielzahl von Begriffen aus der Domäne auf, die unterschiedlich interpretiert und nur in den wenigsten Fällen von den Informatikern vollständig verstanden, geschweige denn im Gedächtnis behalten werden. Solche Begriffe müssen erkannt und aufgelöst werden.

Aktivitäten

- Fragen stellen -> Info festhalten, strukturieren
- alle Anwendungsfälle, priorisieren
- Ist-Zustand -> Soll-Zustand
- Systemgrenze -> Umfeld beschreiben
- Akteure identifizieren, wer macht was
- Informationsquellen, Ansprechpartner
- Lösungsvorschläge, Alternativen
- PM-Aufgaben: Ressourcen, Zeitplan, Finanzen

2.6.b Integrität



Der Begriff Integrität wurde bisher mehrfach verwendet, allerdings unsystematisch und fallweise. Wir versuchen hier eine umfassende Betrachtung aufzuzeigen. In der Praxis werden die Begriffe Konsistenz und Integrität häufig als Synonyme verwendet. Die Integrität kann in drei Komponenten aufgeteilt werden:

1) Datenkonsistenz (logische Betrachtung)

Unter Integrität/Konsistenz versteht man die Widerspruchsfreiheit von Daten. Sämtliche Daten sind korrekt erfasst und geben den gewünschten Informationsgehalt wieder. Mittels Konsistenzregeln (Constraints in SQL) werden gültige Datenzustände definiert, ungültige werden verboten. Hierdurch wird die Qualität der Daten selbst gewährleistet. Die Datenkonsistenz kann wiederum in 3 Komponenten zerlegt werden:

1. **Entitätsintegrität:** Das DBS stellt sicher, dass der Wert des Primärykeys nur einmal pro Tabelle auftritt.
2. **Referentielle Integrität:** Das DBS stellt sicher, dass jeder Fremdschlüssel immer einen korrekten Wert enthält.
3. **Benutzerdefinierte Konsistenz:** Der Benutzer definiert Regeln (Constraints, Prozeduren, Triggers in SQL), die das DBS überprüft und durchsetzt, z.B. der Verkaufspreis muss grösser sein als der Einkaufspreis.

Siehe mehr zu diesem Thema im Kapitel SQL, Datenintegrität.

2) Datensicherheit (physische Betrachtung)

Datensicherheit ist der Schutz der Daten vor Verfälschung, Zerstörung oder Verlust. Datensicherheit wird mittels technischer und organisatorischer Massnahmen sichergestellt und hat daher grundsätzlich nichts mit Datenmodellierung zu tun. Siehe mehr zu diesem Thema weiter hinten im Kapitel Datensicherungskonzept.

3) Datenschutz (juristische und ethische Betrachtung)

Datenschutz ist der Schutz von Daten (und damit auch die von den Daten Betroffenen) vor unberechtigtem Zugriff und Gebrauch. Man unterscheidet zwei Levels:

- Der Zugang zur DB ist durch allgemeine Zugangsprivilegien geregelt. Der Standard geht davon aus, dass jeder Benutzer des DBS mit einer UserID und einem Passwort ausgestattet ist und dass das DBS Benutzern, die keine derartige Autorisierung vorweisen können, jeglichen Zugriff auf alle Funktionen verweigert.
- Die Rechte zur Manipulation einzelner DB-Objekte (Tabellen, Views, etc.) werden für jeden Benutzer mit objektbezogenen Privilegien verwaltet (GRANT und REVOKE in SQL). Siehe mehr zu diesem Thema im Kapitel SQL, Datenschutz, Grundsätze.

**Definition**

Datensicherheit ist der Schutz der Daten vor Verfälschung, Zerstörung oder Verlust und ist nicht zu verwechseln mit Datenschutz.

Datensicherheit hat nichts mit Datenmodellierung zu tun, kann aber im weitesten Sinn im Kapitel Integrität betrachtet werden.

Fragen, die zu klären sind

- Was welche Datenbestände
- Wann online/offline, Tag/Nacht
- Wie oft täglich / wöchentlich, ..
- Wie viele Generationen
- Wer ist verantwortlich
- Wie welches Medium und welche SW
- Wo wird der Datenträger aufbewahrt

Grundsätze

- Es gibt keine 100%-ige Sicherheit, da auch hier die Kosten in Richtung Perfektion exponentiell gegen unendlich steigen.
- Jedes Konzept muss nicht nur einmal installiert werden, sondern dauernd überwacht, hinterfragt und gegebenenfalls angepasst werden.
- Den Schadenfall beherrscht nur, wer ihn getestet und eingeübt hat.
- Ein Ziel-Konflikt besteht zwischen Datensicherheit und Datenverfügbarkeit, weil die Datensicherheit den Zugriff auf die Daten einschränkt, aber die Datenverfügbarkeit möglichst keine Einschränkungen anstrebt.

2.7 Übungen

2.7.a 3-Schema-Architektur

1. Die Ebene des 3-Schema-Konzepts des ANSI-SPARC-Komitees, in welcher die Performance der DB-Anwendung wesentlich beeinflusst wird, ist ...
 - a. die externe Ebene
 - b. die konzeptionelle Ebene
 - c. die interne Ebene
 - d. keine der Ebenen
 - e. alle Ebenen

2. Für das konzeptionelle Datenmodell sind leistungsbestimmende Überlegungen nicht erwünscht, *weil* idealerweise der Benutzer/Auftraggeber bei der Entwicklung des konzeptionellen Modells einbezogen werden sollte.
 - a. + weil +
 - b. + / +
 - c. + / -
 - d. - / +
 - e. - / -

3. Die externe Ebene wird auch Benutzersicht (View) genannt, weil der Benutzer die Daten sieht, wie sie physisch gespeichert sind.
 - a. + weil +
 - b. + / +
 - c. + / -
 - d. - / +
 - e. - / -

4. Die 3 Ebenen haben voneinander getrennte Ziele, weil sie die Sichtweise von Personen mit unterschiedlichen Aufgaben repräsentieren.
 - a. + weil +
 - b. + / +
 - c. + / -
 - d. - / +
 - e. - / -

5. Zählen sie die Merkmale des konzeptionellen Datenmodells auf:

ja	nein	
<input type="checkbox"/>	<input type="checkbox"/>	berücksichtigt Hardwareüberlegungen
<input type="checkbox"/>	<input type="checkbox"/>	Integritätsregeln gehören auch dazu
<input type="checkbox"/>	<input type="checkbox"/>	erfolgt in relationaler Darstellung
<input type="checkbox"/>	<input type="checkbox"/>	repräsentiert die Sicht des Benutzers

2.7.b ERM Fussballmeisterschaft

Erstellen Sie aus folgenden Informationen ein ERM für ein Meisterschaftsprogramm. Folgende Informationen sollen erfasst werden können:

- Ein Klub besteht aus mind. einer Mannschaft.
- Eine Mannschaft spielt gegen andere Mannschaften, eine solche Begegnung kann in mehreren Spielen stattfinden.
- Jedes Spiel findet in einem bestimmten Stadion statt.
- Personen können verschiedene Rollen einnehmen:
 - Fans unterstützen einen Klub und können Spiele besuchen.
 - Trainer trainieren mind. eine Mannschaft, welche mehrere Trainer haben kann.
 - Spieler spielen in genau einer Mannschaft, welche aus mind. einem Spieler bestehen muss.
 - Ein Klub-Präsident führt den Klub.

2.7.c Konzeptionelles Schema**1. Ein Primärschlüssel ...**

- a) ist auch immer ein Fremdschlüssel.
- b) identifiziert jede Entität und darf nie NULL sein.
- c) ist meistens eindeutig.
- d) ist höchstens aus zwei Attributen zusammengesetzt.

2. NULL-Werte ...

- a) sind Blanks.
- b) sind Nullen.
- c) stehen für `kein Wert`.
- d) sind Hex-0 Werte.
- e) bedeuten keine Anzeige.

3. Die Kardinalität ...

- a) ist eine Menge von verschiedenen Datenwerten.
- b) zwischen 2 Entitätsmengen legt fest, wie viele Entitäten aus Entitätsmenge 2 einer Entität aus Entitätsmenge 1 zugeordnet sind und umgekehrt.
- c) ist eine Beziehungsform zwischen zwei Entitäten.

4. Was bedeutet referentielle Integrität ?

- a) Der Primärschlüssel darf nie den Wert NULL haben.
- b) Jeder Primärschlüssel muss auch als Fremdschlüssel in einer andern Entitätsmenge auftreten.
- c) Stellt sicher, dass Fremdschlüssel keine NULL-Werte haben.
- d) Für jeden von NULL verschiedenen Fremdschlüsselwert muss ein entsprechender Primärschlüssel existieren.
- e) Verbietet NULL-Werte in Attributen.

5. Für welchen Begriff gilt der unten beschriebene Sachverhalt ?

- a) Entität
 - b) Entitätsschlüssel
 - c) Fremdschlüssel
 - d) Nullwerte
- 1) Tupel im relationalen Modell
- a) b) c) d)
- 2) Verhindert, dass doppelte Einträge in einer Tabelle auftreten
- a) b) c) d)
- 3) Basiert auf Werten, die in einer andern Entitätsmenge auftreten
- a) b) c) d)

6. Für welchen Begriff gilt der unten beschriebene Sachverhalt ?

- a) Entitätsmenge
 - b) Entitätsschlüssel
 - c) Domäne
 - d) Attribut
- 1) Eine definierte Menge von verschiedenen Datenwerten

-
- a) b) c) d)
2) Beschreibung einer bestimmten Eigenschaft einer Entität
a) b) c) d)
3) Menge von Tupeln
a) b) c) d)
4) Identifizierendes Attribut
a) b) c) d)
6. Der Fremdschlüssel ist in seinem Aufbau (Anzahl Attribute, Datentyp) identisch zum Entitätenschlüssel der referenzierten Entitätsmenge, weil der Fremdschlüssel die Entität der referenzierten Entitätsmenge eindeutig identifizieren muss.
- a. + weil + b. + / + c. + / - d. - / + e. - / -
7. Der Primary Key sollte so gewählt werden, dass dieser
- a) einfach und wenn möglich eindeutig ist.
b) möglichst kurz und immer eindeutig innerhalb einer Entitätsmenge ist.
c) numerisch ist.
8. Welche Aussagen gelten für den Fremdschlüssel ?
- ja nein
☐ ☐ Die referentielle Integrität sichert den korrekten Zustand aller Fremdschlüssel.
☐ ☐ Die Foreign Keys müssen im DBS definiert werden, damit dieses deren Zustand kontrollieren kann.

2.7.d Verbundinstrumente

1. Bei der Rekursion ...
 - a) werden zwei Entitätsmengen mittels Fremdschlüssel rekursiv verknüpft.
 - b) wird eine 0,n : 0,n Beziehung aufgelöst.
 - c) wird eine Entitätsmenge mit sich selbst via Fremdschlüssel verknüpft.
 - d) wird kein Fremdschlüssel benötigt.

2. Bei der Spezialisierung ...
 - a) werden Entitäten einer einzelnen Entitätsmenge mittels Attribut spezifiziert.
 - b) entsteht eine Super- und mehrere Subentitätsmengen.
 - c) entsteht eine selbstbezügliche Beziehungsmenge.
 - d) entstehen mehrere 1,1:0,n, bzw. 1,1:1,n Beziehungen.
 - e) sind Entitäten hierarchisch geordnet.

3. Ist eine Hierarchie als Rekursion realisiert, kann einer untergeordneten Entität nur genau eine Entität der übergeordneten Entitätsmenge zugeordnet werden, weil der Fremdschlüssel der Entität in der untergeordneten Entitätsmenge nur genau eine Entität referenzieren kann.

a. + weil +	b. + / +	c. + / -	d. - / +	e. - / -
-------------	----------	----------	----------	----------

4. Bei der Spezialisierung, bzw. Generalisierung ...
ja nein

<input type="checkbox"/>	<input type="checkbox"/>	handelt es sich um die selben, aber umgekehrten Vorgänge.
<input type="checkbox"/>	<input type="checkbox"/>	werden Entitätsmengen eliminiert.
<input type="checkbox"/>	<input type="checkbox"/>	werden Super- und Subentitätsmengen gebildet.
<input type="checkbox"/>	<input type="checkbox"/>	werden Entitäten zyklisch verknüpft.

5. Bei der Aggregation ...
ja nein

<input type="checkbox"/>	<input type="checkbox"/>	wird höchstens 1 Foreign Key benötigt.
<input type="checkbox"/>	<input type="checkbox"/>	können alle Entitätsmengen Attribute enthalten.
<input type="checkbox"/>	<input type="checkbox"/>	kann die Aggregations-Entitätsmenge ausschliesslich aus Foreign-Keys bestehen.
<input type="checkbox"/>	<input type="checkbox"/>	handelt es sich um einen Spezialfall der Generalisierung.

6. Subentitäten ...
ja nein

<input type="checkbox"/>	<input type="checkbox"/>	enthalten den Entitätenschlüssel der Superentität als Fremdschlüssel.
<input type="checkbox"/>	<input type="checkbox"/>	sind Spezialisierungen der Superentität.
<input type="checkbox"/>	<input type="checkbox"/>	haben Referenzen auf die Attribute der Superentität.
<input type="checkbox"/>	<input type="checkbox"/>	dürfen nicht konditional mit der Superentität assoziiert sein.

2.7.e Normalisierung

1. Folgende Relation befindet sich in

ja nein

- ☐ ☐ unnormalisierter Form
☐ ☐ 1NF
☐ ☐ 2NF
☐ ☐ 3NF
☐ ☐ 3NF aber nicht in 2NF

*PC-NR	Typ	RAM	HD
PC11	HP	128	2
PC15	Compaq	256	20
PC17	HP	512	40
PC28	Compaq	128	9

2. Die Normalisierung ...

- a) wird auf der internen, physischen Ebene ausgeführt.
 b) hat das Ziel, Redundanzen innerhalb einer Relation zu minimieren.
 c) ist eine umfassende Entwurfsmethodik für konzeptionelle Datenmodellierung.
 d) zerlegt Relationen und fügt diese in einem geordneten Prozess wieder zusammen.

3. Eines der Ziele der Normalisierung ist ...

- a) die übersichtliche Gestaltung der Relationen.
 b) das Erzeugen neuer Relationen.
 c) die Minimierung der Redundanz.
 d) die Gewährleistung der Benutzerfreundlichkeit.
 e) das Erkennen der funktionalen Abhängigkeiten.

4. Welche Begriffe lassen sich einander zuordnen ?

- a) Normalisierung
 b) Denormalisierung
 c) Referentielle Integrität
 d) Schlüsselkandidaten
 e) Restriktion
- 1) Performance – Verbesserung
 a) b) c) d) e)
 2) Redundanzminderung
 a) b) c) d) e)
 3) Voraussetzung für Fremdschlüssel-Entitätsschlüsselbeziehung
 a) b) c) d) e)
 4) Verminderung von Speicheranomalien
 a) b) c) d) e)

5. Durch die Normalisierung der Relationen werden Anomalien bei Speicheroperationen vermieden, weil die Normalisierung eine umfassende Entwurfsmethodik für eine konzeptionelle Datenbank ist.

- a. + weil + b. + / + c. + / - d. - / + e. - / -

6. Entitätsmengen in der 1NF ...

- a) lassen am Schnittpunkt Attribut/Datensatz auch mehrere Werte zu.
- b) haben keine Nichtschlüsselattribute, die vom Gesamtschlüssel abhängig sind.
- c) haben grössere Redundanzen als voll normalisierte Relationen.

7. Was sind die Ziele und Eigenschaften der Normalisierung ?

ja nein

- ☐ ☐ Redundanzen innerhalb einer Relation zu minimieren.
- ☐ ☐ Sorgt für eine zweckmässige Strukturierung der Daten, auch für die interne Ebene.
- ☐ ☐ Dient der Verminderung von Anomalien bei Speicheroperationen.

2.7.f Datenmodellierung, Kontoverwaltungs-System

Die Bank 'SuperSerios AG' möchte zur Verwaltung der Konten ein Kontoverwaltungssystem erstellen, um das manuelle Karteisystem abzulösen. Zur Zeit bestehen zwei Karteien für Kunden und Konten. Ein Kunde kann mehrere Konten haben, ein Konto kann von einem oder mehreren Kunden eröffnet werden. Zusätzlich müssen für jedes Konto die einzelnen Kontobewegungen (Datum, Buchungsbetrag, Begünstigter bei Belastungen bzw. Auftraggeber bei Vergütungen) festgehalten werden. Die Kontobewegungen sind genau einem Konto zugeordnet (sie sind daher nicht mit dem Buchungssatz der Buchhaltung zu vergleichen, der immer zwei Konten angibt).

1. Modellieren Sie diesen Sachverhalt mit einem geeigneten ERM (ohne Attribute, aber mit Beziehungen und Kardinalitäten).
2. Stellen Sie ein rel. DB-Modell auf, basierend auf diesem ERM, inkl. Attributen. Kennzeichnen Sie *Primary Key und Foreign Key. Fügen Sie Beispieldatensätze derart zu, dass ein Beispiel durchgehend nachvollzogen werden kann.

2.7.g Normalisierung Verlag

Ein Verlag erteilt Ihrer Firma den Auftrag ein Informationssystem für die Verwaltung der verlegten Bücher zu erstellen. Im Verlaufe des Projekts erhalten Sie als Projektleiter einen Vorschlag, wie die Daten in Tabellen einer relationalen Datenbank abgelegt werden sollen.

Bücher								
*Buch_Nr	Titel	Untertitel	Seiten	Auftraggeber_Nr	Auftraggeber_Na	*Kapitel	Autor_Nr	Autor_Name
14	Client/Server	Survival Guide	243	12	Heierli	1	1	Smith
14	Client/Server	Survival Guide	243	12	Heierli	2	1	Smith
14	Client/Server	Survival Guide	243	12	Heierli	3	3	Kelley
15	Business Rules	Concept. Model	265	29	Müller	1	5	Fisher
16	CH in der EU	100 Pro's	13	15	Meier	1	3	Kelley

Anmerkung: Auftraggeber ist eine Person, die dem Verlag den Auftrag zum Herausgeben eines Buches erteilt. Jedes Kapitel eines Buches kann unterschiedliche Autoren haben.

Aufgaben

1. Markieren Sie Verletzungen der 1., 2. und 3. NF
2. In welcher Normalform befindet sich die Tabelle ?
3. Falls sich die Tabelle nicht in der 3.NF befindet: Überführen Sie die Tabelle in die dritte Normalform.

2.7.h Datenmodellierung, Mietwohnungen

Gegeben ist das Relationenmodell Mietwohnungen im Anhang.

1. Erstellen Sie das zugehörige ERM (ohne Attribute), mit min. und max. Kardinalitäten.

2. Es kann angegeben werden, ob zu einer Wohnung auch eine Garage gemietet wurde. Entsprechend kann auch die gesamte Miete für Wohnung und Garage berechnet werden.

ja nein
☐ ☐

3. Eine Garage gehört maximal zu einer Wohnung.

ja nein
☐ ☐

4. Die Struktur verhindert, dass eine Garage, die zu einer Liegenschaft x gehört, einer Wohnung einer anderen Liegenschaft (y) vermietet werden kann.

ja nein
☐ ☐

5. Das Datenmodell verlangt pro Liegenschaft zwingend genau einen Hauswart.

ja nein
☐ ☐

6. Einer Liegenschaft können mehrere Wohnungen zugewiesen werden.

ja nein
☐ ☐

Anhang

Wohnung:

ID	Bezeichnung	Groesse	Preis	Hauswart	Vermietet	Liegen- schaftID	Wasch- kuecheID
1	8/1	3.5	1450	No	Yes	1	4
2	8/2	5.5	1900	No	Yes	1	4
3	8/3	4.5	1600	No	Yes	1	4
4	8/4	4.5	1800	No	Yes	1	4
5	8/5	2.5	600	Yes	Yes	1	4
6	8/6	5.5	2200	No	No	1	4
7	6/1	4.5	1450	No	No	1	3
8	6/2	4.5	1500	No	Yes	1	3
9	6/3	2.5	1250	No	Yes	1	3
10	6/4	4.5	1800	No	Yes	1	3
11	6/5	2.5	1400	No	No	1	3
12	6/6	3.5	1600	No	Yes	1	3
13	8/1	3.5	1450	No	Yes	2	8
14	8/2	5.5	1900	No	Yes	2	8
15	8/3	4.5	1600	No	No	2	8
16	8/4	4.5	1800	No	Yes	2	8
17	8/5	2.5	1400	No	Yes	2	8
18	8/6	5.5	2200	No	No	2	8
19	6/1	4.5	1450	No	Yes	2	7
20	6/2	4.5	1500	No	Yes	2	7
21	6/3	2.5	1250	No	No	2	7
22	6/4	4.5	1200	Yes	Yes	2	7
23	6/5	2.5	1400	No	No	2	7
24	6/6	3.5	1600	No	Yes	2	7

Waschkueche

ID	Standort	Anzahl- Trocknungs- raeume	Tumbler	LiegenschaftID
1	U1	2	Yes	1
2	U2	1	Yes	1
3	U3	2	Yes	1
4	U4	2	No	1
6	U1	1	Yes	2
7	U2	1	Yes	2
8	U3	1	Yes	2

Liegenschaft:

ID	Bezeichnung	Ortschaft
1	Bungertweg	Niederhasli
2	Puntweg	Niederhasli

Garage

ID	Parkplatz	Mietpreis	LiegenschaftID	WohnungID
1	20	100	1	
2	21	100	1	16
3	22	100	1	1
4	23	100	1	4
5	24	100	1	3
6	25	100	1	24
7	26	100	1	13
8	27	100	1	5
9	28	100	1	12
10	29	100	1	19
11	20	100	2	9
12	20b	70	2	9
13	21	100	2	
14	22	100	2	
15	23	100	2	
16	23b	70	2	
17	24	100	2	17
18	25	100	2	

2.7.i Datenmodellierung, Liegenschaftsverwaltung

Sie leiten ein Projekt, in welchem eine EDV-gestützte Mietwohnungs-Verwaltung entwickelt werden soll. Ihr Spezialist für Datenmodellierung hat auf ihren Auftrag untenstehendes Datenmodell entwickelt. Der End-Benutzer des zukünftigen Systems hatte zuvor Aussagen gemacht, welche durch das Datenmodell abgedeckt werden soll. Ihre Aufgabe ist es nun, das fertige Datenmodell zu überprüfen.

Aufgabe: Geben Sie an, welche der folgenden fünf Aussagen nicht korrekt abgebildet sind. Gefundene Fehler sind im Datenmodell zu korrigieren und fehlende Modell-Elemente sind zu ergänzen.

Korrekt ?

1. Eine Person kann mehrere Wohnungen besitzen.

ja nein
☐ ☐

2. Eine Wohnung kann an mehrere Personen vermietet werden; sie muss aber nicht zwingend vermietet sein.

ja nein
☐ ☐

3. Eine Wohnung befindet sich in genau einem Gebäude, welches immer Wohnungen umfasst.

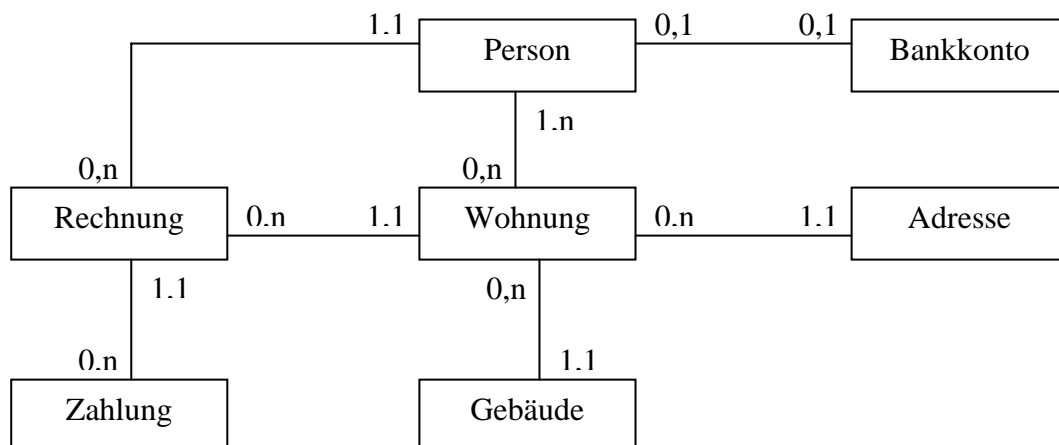
ja nein
☐ ☐

4. Jede Person hat genau eine Wohnadresse (zur Rechnungsstellung).

ja nein
☐ ☐

5. Alle Wohnungen eines Gebäudes befinden sich an derselben Adresse.

ja nein
☐ ☐



2.7.j Datenmodellierung, Spital

Ein Spital ist dabei, eine SW für die Administration seiner Patienten zu entwickeln. Für diese Aufgabe wird ein Ausschnitt aus diesem Informationssystem betrachtet. Das Spital stellt folgende Anforderungen an das Datenmodell:

- Ein Spital bietet die verschiedensten Therapien zu definierten Kosten an.
 - Jedem Patienten stehen alle Therapien offen.
 - Ein Patient kann keine oder mehrere Behandlungen (Therapien) haben, wobei jede Behandlung eindeutig einem Patienten zugeordnet werden muss.
 - Für die Behandlung ist genau ein Arzt verantwortlich. Ein Arzt ist für Behandlungen an unterschiedlichsten Patienten zuständig. Es gibt aber auch nur administrativ tätige Ärzte, die keinen Patientenkontakt haben.
 - Ein Patient ist während seiner Behandlung genau einem (stationär) oder keinem (ambulant) Bett zugeordnet. Ein Bett wird von vielen Patienten benutzt.
1. Modellieren Sie diesen Sachverhalt mit einem geeigneten ERM (ohne Attribute, aber mit Beziehungen und Kardinalitäten).
 2. Stellen Sie ein rel. DB-Modell auf, basierend auf diesem ERM, inkl. Attributen. Kennzeichnen Sie *Primary Key und Foreign Key. Fügen Sie Beispieldatensätze derart zu, dass ein Beispiel durchgehend nachvollzogen werden kann.

2.7.k Datenmodellierung, Adressen

Gegeben ist das Relationenmodell Adressen im Anhang.

1. Erstellen Sie das zugehörige ERM (ohne Attribute), mit min. und max. Kardinalitäten.

Firma

Telefon

ID	Nummer	Type	FirmaID
1	01'804'66'00	Telefon	1
2	01'304'67'11	Telefon	2
3	01'930'59'50	Natel	3
4	01'701'21'39	Telefon	4
5	031'781'20'53	Telefon	5
8	01'804'67'00	Fax	1
9	042'312'66'23	Telefon	12
11	01'920'61'81	Fax	3
12	AlfaLaval@cyberlink.ch	email	1
13	062'351'22'79	Telefon	7
14	01'41'92'02	Telefon	6
15	042'344'23'89	Fax	12
16	01'341'34'46	Telefon	8
17	052'633'18'87	Telefon	0
18	01'741'08'63	Telefon	9
19	01'341'69'23	Telefon	10
20	01'791'10'83	Fax	3
21	056'839'18'10	Telefon	
22	077'244'84'41	Natel	
23	077'352'29'69	Natel	
24	032'947'13'42	Telefon	
25	01'462'59'43	Telefon	11
26	071'245'90'80	Fax	
27	055'740'18'44	Telefon	
28	01'371'05'52	Telefon	13
29	032'640'39'50	Telefon	
30	058'640'84'58	Fax	
31	081'325'16'28	Telefon	
32	056'51'24'64	Fax	
33	056'422'93'78	Telefon	
34	Admin@abb.com	email	4
42	032'41'59'33	Telefon	

ID	Name	Segment	Adressel D
1	Alfa Laval	Automation	25
2	Tetra Pak	Verpackung	25
3	SKA	Bank	26
4	ABB	Maschinenbau	27
5	Ascom Hasler	Rüstung	28
6	Sulzer	Maschinenbau	29
7	Holderbank	Zement	30
8	Messerli	Druckerei	31
9	SBG	Bank	36
10	Elektrowatt	Ingenieurbüro	32
11	Locher	Bau	33
12	Landis & Gyr	Gebäudetechnik	34
13	Kuoni	Reisebüro	35

Beziehung

ID	Beziehungstyp	Person1ID	Person2ID	ID	FirmaID	PersonID
1	Ehepaar	1	4	1	1	5
2	Elternteil	1	19	2	1	20
3	Elternteil	4	19	3	1	1
4	Paar	15	22	4	5	20
5	Elternteil	10	1	5	5	16
6	Elternteil	29	16	6	5	4

PersonFirma

Person

ID	Anrede	Name	Vorname	Geburtsjahr	Geschlecht
1	Frau	Baumann	Renate	65	f
3	Herr	Brandenberg	Lukas	93	m
4	Herr	Baumann	Christoph	63	m
5	Herr	Frischknecht	Markus	69	m
6	Herr	Geiser	Christof	67	m
7	Herr	Gisi	Stefan	42	m
8	Herr	Guglielmetti	Didier	65	m
9	Herr	Gysler	René	58	m
10	Herr	Huber	Rene	39	m
11	Herr	Imhof	Felix	62	m
12	Herr	Irniger	Klaus	61	m
13	Herr	Kissling	Urs	64	m
14	Herr	Kohler	Manuel	67	m
15	Herr	Krucker	Beat	72	m
16	Frau	Meier	Esther	69	f
17	Herr	Nydegger	HansJürg	67	m
18	Herr	Peterka	Boris	67	m
19	Herr	Baumann	Roman	85	m
20	Herr	Rüsch	Markus	35	m
21	Herr	Scherrer	Fabian	55	m
22	Frau	Signer	Nicole	69	f
29	Herr	Vollmer	Patrick	31	m

Adresse

ID	Adresse	PLZ	Ort
5	Scherzerstr. 7	5116	Schinznach-Bad
6	Perronweg 8	8855	Wangen
7	Im Grüntal 13	8405	Winterthur
8	Uttenberg	8934	Knouau
9	Klausfeld 3	6037	Root
10	Oberhardstr. 28	5413	Birmenstorf
11	Kaltbreiterstrasse 99	8003	Zürich
12	Sonnenbergstr. 2	4127	Birsfelden
13	Gerenstr. 40	9202	Gossau
14	Landstr. 56	4452	Ittingen
15	Möhlstr. 91	8006	Zürich
16	Rütistrasse 226	4703	Kestenholz
17	Mülethal 6	3270	Aarberg
18	äussere Stammerau 13	8500	Frauenfeld
19	Zürcherstr. 19	8154	Oberglatt
20	Via San Gian 26	7500	St. Moritz
21	Spinnereistr. 29	8640	Rapperswil
22	Alte Stationsstrasse 22	8154	Oberglatt
23	Landstr. 73	8450	Andelfingen
25	Oberfeldstrasse 20	8302	Kloten
26	Albisgüetli	8000	Zürich
27	Brown-Boweristr. 1	8000	Zürich
28	Dufourstr.	3001	Bern
29	Industrie 21/b	8040	Altstetten
30	Zentralquartier	5030	Aarau
31	Bahnhofstr. 28	8153	Glattbrugg
32	Tiefenbrunnen 12	8057	Zürich
33	Brunau	8091	Zürich
34	Baarerstr. 112	6000	Zug
35	Zentrum Glatt	8150	Wallisellen
36	Paradeplatz	8000	Zürich

PersonTelefon

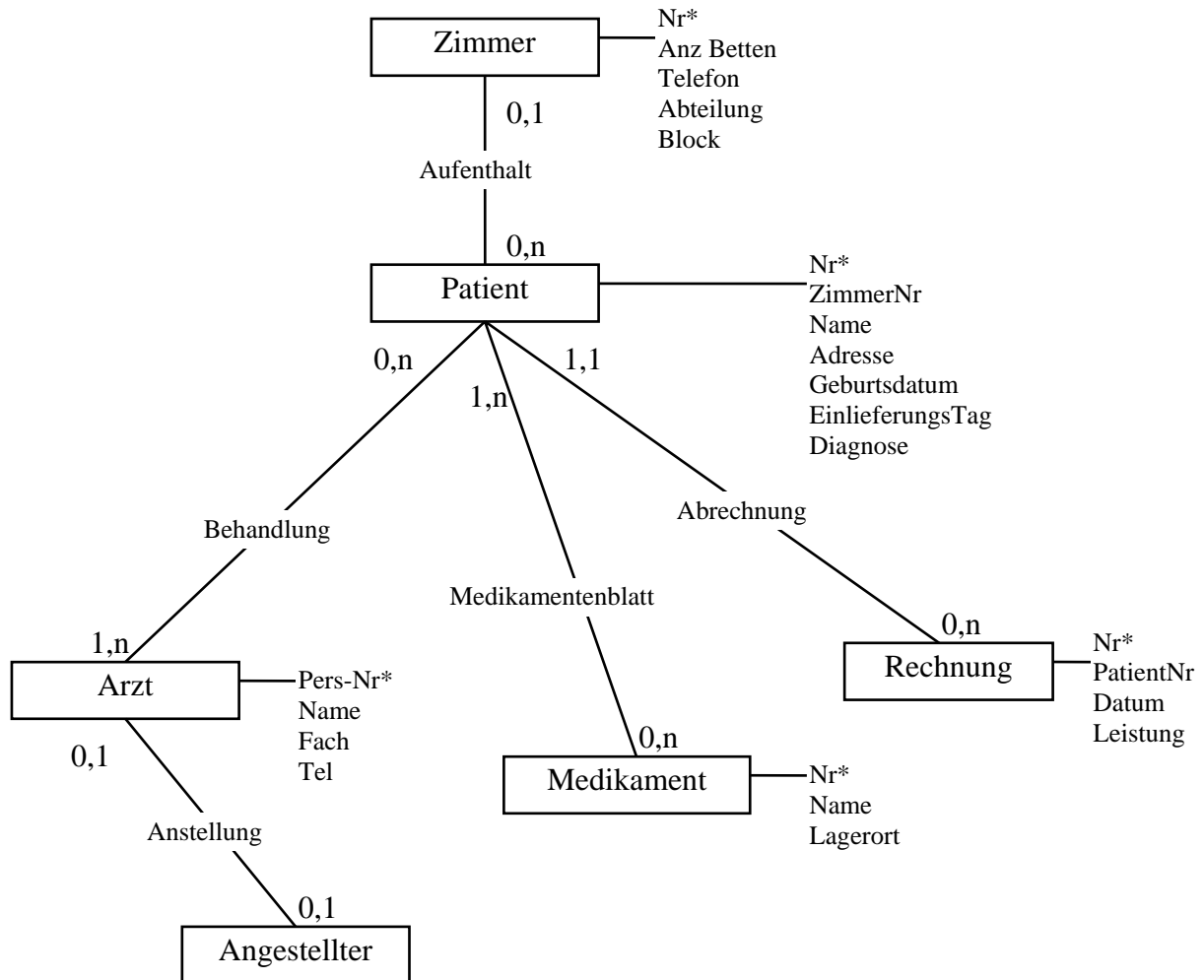
ID	PersonID	TelefonID
1	1	24
2	4	24
3	19	24
4	4	5
5	5	17
6	6	21
7	14	21
8	17	21
9	6	4
10	14	4
11	17	18
12	7	22
13	8	23
14	9	27
15	21	31
16	9	16
17	21	16
18	20	42
19	21	33
20	22	31
21	21	32
22	29	29

PersonAdresse

ID	PersonID	AdresseID
1	1	16
2	3	18
3	4	16
4	5	18
5	6	10
6	7	11
7	7	7
8	8	23
9	9	21
10	10	8
11	11	19
12	12	22
13	12	9
14	13	5
15	14	10
16	15	20
17	16	6
18	17	10
19	18	15
20	19	16
21	20	17
22	21	21
23	22	20
24	29	6

2.7.1 Datenmodellierung, Krankenhaus

Gegeben sei das folgende ERM, welches ein System „Krankenhaus“ beschreibt.



Geben Sie an, ob die folgenden Aussagen, abgeleitet aus diesem ERM, richtig oder falsch sind.

1. In einem Zimmer können mehrere Patienten liegen.

ja nein nicht definiert
☐ ☐ ☐

2. Nur ein Patient, der auf einem Zimmer liegt, kann eine Rechnung erhalten.

ja nein nicht definiert
☐ ☐ ☐

3. Der behandelnde Arzt stellt die Rechnung.

ja nein nicht definiert
☐ ☐ ☐

4. Ein Patient erhält genau eine Rechnung.

ja nein nicht definiert
☐ ☐ ☐

-
5. Alle Patienten mit derselben Diagnose liegen im selben Block.
ja nein nicht definiert
☐ ☐ ☐
6. Die Anzahl freier Betten in einem Zimmer lässt sich berechnen.
ja nein nicht definiert
☐ ☐ ☐
7. Der Name eines Medikamentes steht zwingend auf einer Rechnung.
ja nein nicht definiert
☐ ☐ ☐
8. Alle Zimmer derselben Abteilung liegen im selben Block.
ja nein nicht definiert
☐ ☐ ☐
9. Ein Arzt behandelt verschiedene Patienten, aber alle mit derselben Diagnose.
ja nein nicht definiert
☐ ☐ ☐
10. Ein Medikament für Patient X kann ins richtige Zimmer geliefert werden.
ja nein nicht definiert
☐ ☐ ☐

2.7.m Datenmodellierung, Textilfärberei

Die Textilfärberei Coulur-It GmbH hält in ihrem EDV-System unterschiedliche Stammrezepte (pro Farbe ein Stammrezept) fest. Das Stammrezept setzt sich dabei aus unterschiedlichen Chemikalien zusammen. Im Stammrezept wird festgehalten, wie viel (in Gramm) Chemikalie pro Liter Färbebrühe benötigt wird. Pro Auftrag wird ein Stammrezept ausgeführt. Im Auftrag wird zusätzlich das Volumen der Färbebrühe festgehalten. Daraus berechnet sich dieses Rezeptverwaltungssystem die benötigten Mengen an Chemikalien selbständig.

Aufgabe: Modellieren Sie diesen Sachverhalt mit einem geeigneten Relationenmodell, inkl. Beziehungen, Kardinalitäten, Tabellennamen und Attributen (kennzeichnen Sie *Primary- und Foreign-Keys). Fügen Sie Beispieldatensätze derart zu, dass ein Auftrag durchgehend nachvollzogen werden kann.

2.7.n Normalisierung Tonträger

Bringen Sie folgende unnormalisierte Tabelle in die 3. Normalform.

*Titel:	Träger:	Ort:	Stil:	Interpret:	Musikstück:	Dauer:	Bewertung:	Note:
i tell this night	cd	cd-ständer	pop	stephan eicher	i tell this night two people in a room tu tournes mon coeur	04:52 04:08 04:43	5 5.5 4.5	5
pop95	mc	kasten	pop	roxette die prinzen ace of base	joyride gabi und klaus happy nation	04:30 03:23 04:00	5.5 5.5 5.5	5.5
das leben ist grausam	cd	cd-ständer	accapella	die prinzen	gabi und klaus mein fahrrad die vögel	03:23 02:25 04:52	5.5 5.5 3.5	5
wir wollen nur deine seele	cd	cd-ständer	kabarett rock	nimm zwei	mr. pharao slimer wie ein sturm	05:46 03:48 05:35	6 6 5	5.5

2.7.o Datenmodellierung Stundenplan

Aufgabe

Erstellen Sie das Datenmodell für die Stundenpläne einer Schule.

Dem folgenden Ausschnitt eines Stundenplanes sind die relevanten Daten zu entnehmen:

Sommersemester 2005:

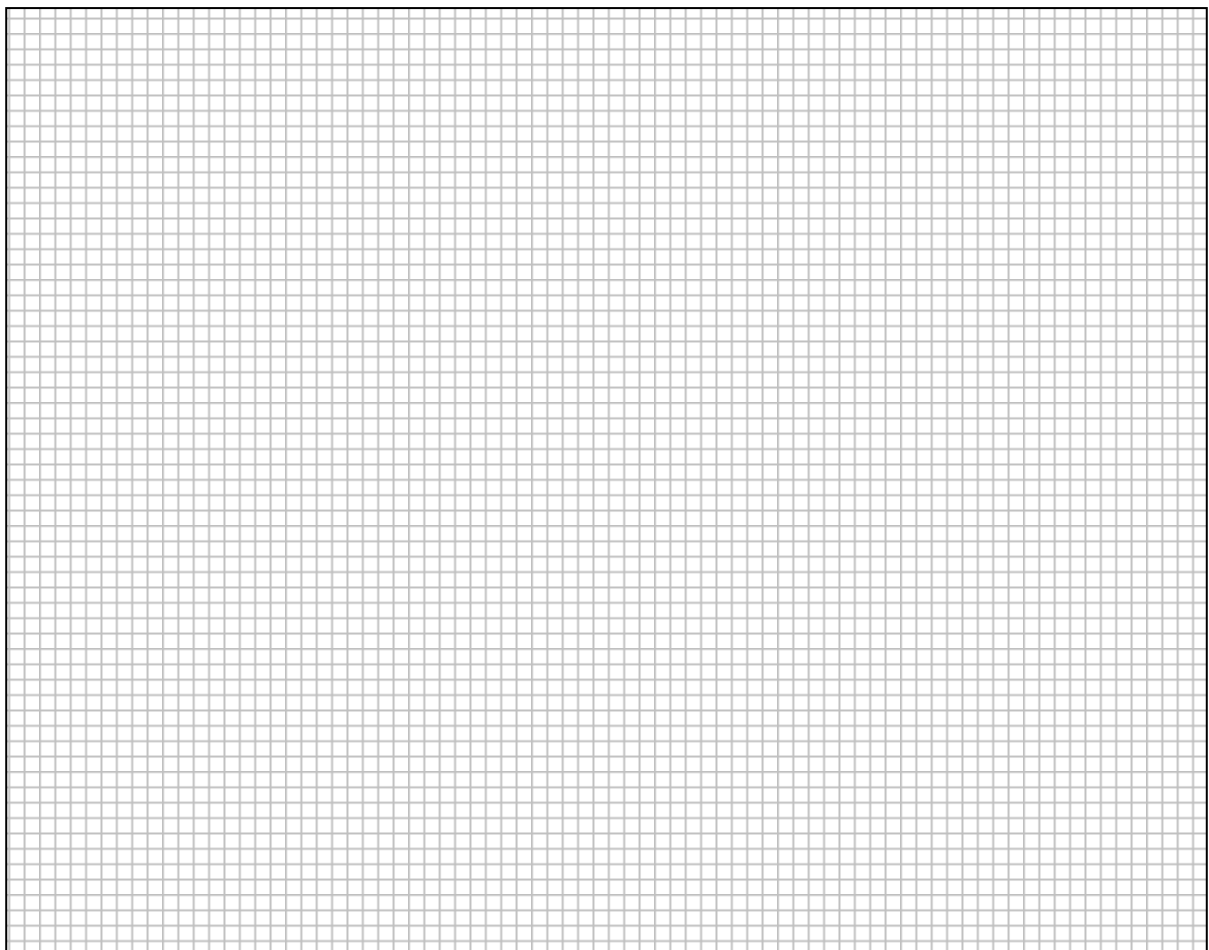
Tag	Zeit	Lehrer	Fach	Klasse	Zimmer
Montag	07:55 – 08:40	Grob	DB	01 NDW	325
Montag	08:45 – 09:30	Grob	DB	01 NDW	325
Dienstag	09:50 – 10:35	Kuhn	OS	99 NDI	U18
Mittwoch	10:40 – 11:25	Kuhn	OS	00 Inf	U18

Randbedingungen

Zusätzlich zu den Stundenplandaten sollen noch weitere Informationen verwaltet werden können:

- Jedes Zimmer hat einen verantwortlichen Lehrer.
- Die inventarisierten Geräte, wie Hellraumprojektor etc., sind einem Zimmer zuzuordnen.
- Nicht jedes Zimmer eignet sich für jedes Fach
- Die Stundenzeiten sind fix und an jedem Tag gleich.
- Die Planung soll sich über mehrere Semester erstrecken.
- Ein Lehrer kann mehrere Fächer unterrichten.
- Ein Fach kann von verschiedenen Lehrern unterrichtet werden.

Modellieren Sie diesen Sachverhalt mit einem geeigneten Relationenmodell, inkl. Beziehungen, Kardinalitäten, Tabellennamen und Attributen. Kennzeichnen Sie Primary- und Foreign-Keys.



2.7.p Datenmodellierung Bibliothek

Aufgabe

Entwickeln Sie das Datenmodell für eine Bibliothek.

Datenumfang

- folgende Ausleihen eines Buches müssen erfasst werden können:
 - die aktuellen
 - die vergangenen
 - die zukünftigen (Reservationen)
- die wichtigsten Personalien für Mitglieder und Angestellte, Angestellte können auch Bücher ausleihen
- die Personen sind ihren Personentypen zuzuordnen. (z.B. Angestellter, Mitglied etc)
- eine Person kann über mehrere Adressen verfügen
- die Autoren der Bücher, Autoren pflegen mehrere Bücher zuschreiben, ein Buch kann mehrere Autoren haben
- jedes Buch ist einem Buchtyp zuzuordnen. (z.B. Krimi, Lexikon, etc.)
- die Lieferanten der Bücher, erfolgreiche Bücher können im Laufe der Zeit von unterschiedlichen Lieferanten feilgeboten werden
- Kaufdatum des Buches
- Preis des Buches
- der Buchverlag
- über jedes Buch müssen Stichworte erfasst werden können, identische Stichwörter können mehreren Büchern zugeordnet sein.

Modellieren Sie diesen Sachverhalt mit einem geeigneten Relationenmodell, inkl. Beziehungen, Kardinalitäten, Tabellennamen und Attributen. Kennzeichnen Sie Primary- und Foreign-Keys.

