



Landeshauptstadt
München

Referat für
Bildung und Sport

FIAE . DB

Datenbanktheorie

**Datenbankmodellierung und
Datenbankentwicklung**

**Städtische Berufsschule
für Informationstechnik**

Herausgeber	Städtische Berufsschule für Informationstechnik https://www.bsinfo.eu
Koordinator FIAE-DB	Michael Niedermair
Team	Betreuer: Pm Pm, Vk, Ra, Nm
Quellen, Material	Skipt Ba, Vk, Pm
Version	0.2 Alpha (2018-05-03)
Lizenz	closed version

Inhaltsverzeichnis

<u>Erste Überlegungen zur Datenbanktheorie</u>	2
<u>Datenbanksysteme</u>	4
<u>Aufgaben von Datenbanksystemen</u>	4
<u>Aufbau – ANSI Architekturmodell</u>	5
<u>Datenbankmodelle</u>	7
<u>Hierarchisches Datenbankmodell</u>	7
<u>Netzwerkbasiertes Datenbankmodell</u>	7
<u>Objektorientiertes Datenbankmodell</u>	8
<u>Relationales Datenbankmodell</u>	8
<u>Datenorganisation</u>	9
<u>Redundanz</u>	9
<u>Inkonsistenz</u>	10
<u>Anomalie</u>	10
<u>Phasen der Datenbankentwicklung</u>	12
<u>Anforderungsanalyse (Phase 1)</u>	13
<u>Konzeptionelle Phase – ER Modell (Phase 2)</u>	14
<u>Aufgaben ER-Modell</u>	18
<u>Logische Phase – relationales Datenbankmodell (Phase 3)</u>	21
<u>Grafische Darstellung</u>	23
<u>Aufgaben relationales DB-Modell (Teil 1)</u>	24
<u>Normalisierung</u>	26
<u>Referentielle Integrität</u>	29
<u>Aufgaben relationales DB-Modell (Teil 2)</u>	30
<u>Physische Implementierung (Phase 4)</u>	35
<u>Quellen</u>	36

Erste Überlegungen zur Datenbanktheorie

Aufgabe 1 „Personaldatenverwaltung“

Situation:

Susanne Reisinger, Personalsachbearbeiterin der Firma IT-Sys KG hat mit Hilfe eines Tabellenkalkulationsprogramms die Ihnen in Papierform vorliegende Datei erstellt und diese ihrem Chef als Lösung für eine Personaldatenverwaltung präsentiert.

P-Nr.	Nachname	Vorname	GebDatum	PLZ	Wohnort	Straße	Konto	Gehalt_Gruppe	Brutto	BLZ	Institut
10000	Obermotz	Heinrich	20.05.39	66950	Kaiserslautern	Friedrichstr. 7	241241234	E9	5.500,00 €	39901020	Commerzb. Bonn
10340	Kaiser	Hildegard	14.05.61	50342	Bonn	Goehtestr. 23	7024322	E5	2.649,32 €	39901020	Commerzb. Bonn
12125	Brösel	Wemer	27.06.59	74832	Heidelberg	Spinozastr. 56	8652341	E6	2.997,10 €	24520000	StSpk Germersheim
23497	Schubert	Ute	05.08.55	67948	Maxdorf	Westring 1	234934	E1	594,38 €	13298002	Raiba Maxdorf
29384	Walker	Jonny	19.04.52	64732	Walldorf	Koksgasse 17	907009234	E3	2.158,60 €	29050000	Citybank Bonn
35462	Dampf	Hannelore	12.09.67	67548	Nirgendwo	Inallengassen 1	70121235	E4	2.378,59 €	37811000	Vereinsbank Köln
36278	Stein	Frank N.	01.12.39	75843	Roßbach	Hintergasse 24	9274239	E4	2.378,59 €	29050000	Citybank Bonn
43521	Napf	Karl	15.12.56	63722	Hackstadt	Hauptstr. 17	721123	E7	3.254,80 €	39901020	Commerzb. Bonn
45985	Louis	King	01.03.57	66955	Pirmasens	Schloßstr. 3	65869324	E6	2.997,10 €	29050000	Citybank Bonn
46345	Kron	Maria	01.03.67	66748	Frankenthal	Ludwigstr. 134	75663442	E5	2.649,32 €	24520000	StSpk Germersheim
46372	Verdi	Michaela	03.01.61	67584	Worms	Kaiser-Franz-Str. 15	79234	E6	2.997,10 €	13298002	Raiba Maxdorf
46378	Bach	Johann	06.09.70	67453	Ludwigshafen	Bruchwiesenstr. 54	70093	E5	2.649,32 €	24520000	StSpk Germersheim
50023	Buckler	KarL	01.01.45	66543	Hannover	Koksgasse 7	77897234	E5	2.649,32 €	13298002	Raiba Maxdorf
56743	Koch	Robert	04.03.43	67849	Oggersheim	Schillerstr. 58	9342992	E6	2.997,10 €	24520000	StSpk Germersheim
64372	Schumann	Klara	18.11.54	67832	Eisenberg	Hauptstr. 78	8734221	E9	5.500,00 €	13298002	Raiba Maxdorf
69875	Stechlich	Willi B.	05.03.52	64732	Dürkheim	Landstr. 3	7532877	E4	2.378,59 €	70020000	StSpk Dürkheim
74830	Beam	Jim	17.05.62	57684	Eberbach	Waldstr. 5	12445	E7	3.254,80 €	24520000	StSpk Germersheim
77865	Fermi	Enrico	05.04.40	67483	Ludwigshafen	Oggersheimer Str. 88	73429874	E8	3.744,23 €	24520000	StSpk Germersheim
88323	Daniels	Jack	03.03.49	78432	Germersheim	Benzstr. 34	93823	E2	1.423,90 €	24520000	StSpk Germersheim
98230	Schönberg	Arnold	04.04.49	64732	Mannheim	E605	93473212	E7	3.254,80 €	24520000	StSpk Germersheim

Leider ist ihr Chef über die Lösung gar nicht glücklich. Er äußert folgende Kritikpunkte:

„Auf die Datei kann ja immer nur ein Mitarbeiter zugreifen. Es gibt aber doch eine klare Arbeitsteilung zwischen Ihnen, Frau Wagner (Sie ist zuständig für Personendaten), Frau Müller (Sie ist zuständig für Gehaltsinformationen) sowie Herrn Heimer (Er ist zuständig für die Verwaltung der Bankverbindungen und Informationen über Kreditinstitute). Deshalb soll es möglich sein soll, dass Sie alle drei gleichzeitig arbeiten können.“

Der Chef von Frau Reisinger, gibt ihr nun folgende Bedingungen vor:

Die Mitarbeiter sollen nur die Daten sehen und ändern dürfen, für die sie beauftragt sind (s. Kritikpunkte). Z.B. soll(en)

- Frau Wagner und Herr Heimer keine Informationen über die Gehälter der Mitarbeiter erhalten.
- Frau Müller zwar Gehaltsgruppen lt. Tarifverträgen anpassen und Mitarbeiter umgruppieren können, jedoch nur anonymisiert. „Mitarbeiter 46378 bekommt eine Gehaltserhöhung. Ab Monatsersten wird er in Gehaltsgruppe E5 eingestuft!“.
- Die Änderung mehrfach enthaltener Daten (z.B. Namen von Kreditinstituten) kann zu Fehlern führen. Daher sollen derartige Informationen nur noch einmalig vorkommen.
Schlüsselwerte können durchaus in mehreren Tabellen vorkommen.
- Es muss sichergestellt sein, dass auch mit der neu zu erarbeitenden Lösung noch alle Daten der einzelnen Mitarbeiter zugeordnet und berichtet werden können, z.B. um dem Chef die monatlichen Gehaltsüberweisungen zu ermöglichen.
- Die Tabellenanzahl der neuen Lösung kann beliebig sein.

Arbeitsauftrag:

Erarbeiten Sie in der Gruppe (3-4) einen Vorschlag, der die o.g. Bedingungen soweit wie möglich berücksichtigt. Der Vorschlag wird anschließend in der Klasse präsentiert.

Arbeitszeit:

30 Minuten

Arbeitsmittel je Gruppe:

- Schere
- Kleber
- Lineal
- Stift
- 3x Gehaltsliste DIN A 3
- Blanko DIN A 3 Blatt zum aufkleben

Datenbanksysteme

Das Beispiel aus der Aufgabe Personalverwaltung macht deutlich, dass eine Verwaltung von Daten mittels Datenbanksystem sinnvoll ist.

Üblicherweise arbeiten verschiedene Personengruppen an einem Datenbanksystem mit. Es gibt Administratoren der Datenbank sowie des Betriebssystems, Benutzer und ggf. andere Rollen. Alle haben eine andere Sicht auf das Datenbanksystem. Daher bietet sich an, eine Systemstruktur aus Komponenten, Ebenen, Schnittstellen etc. zu erschaffen.

Diese abstrahiert die Sichtweise und führt dadurch für die Benutzerrollen zu einer Verringerung der Komplexität.

Aufgaben von Datenbanksystemen

Um die komplexen und vielseitigen Anforderungen an eine konsistente und einheitliche Datenhaltung zu erfüllen, müssen Datenbanksysteme eine Reihe von Merkmalen aufweisen. Die Anforderungen an ein Datenbanksystem sind:¹ 1

- **Datenunabhängigkeit:** Anwendungsprogramm und Datenhaltung sollten unabhängig voneinander sein.
- **Effizienter Speicherzugriff:** Verwendung von Speichertechniken, um die Datenverarbeitung effizient zu gestalten (z. B. Indizes).
- **Paralleler Datenzugriff:** Mehrere Benutzer müssen gleichzeitig auf die Daten zugreifen können. Hierfür benötigt es Protokolle für Transaktionskontrollen.
- **Datenkonsistenz:** Eine Veränderung der Daten muss vollständig sein und zuvor definierte Regeln einhalten.
- **Gemeinsame Datenbasis:** Die Daten werden zentral abgelegt und verwaltet.
- **Datenintegrität:** Daten müssen vollständig und den Regeln entsprechend korrekt sein, ansonsten ist die Speicherung zu verhindern (z. B. Verweis auf einen nicht vorhandenen Fremdschlüssel).
- **Datensicherheit:** Der Zugriff darf nur durch autorisierte Benutzer gestattet werden.
- **Wiederherstellungsverfahren:** Im Fehlerfall muss ein Backup- oder Recovery-Verfahren zur Wiederherstellung einer konsistenten Datenbank vorhanden sein.
- **Abfragesprache:** Ein Datenbanksystem (DBS) muss eine Möglichkeit für die Datenmanipulation (Abfrage, Aktualisierung, Berichtsgenerierung, Veränderung) bereitstellen (bei relationalen DBS ist dies meist eine Form von SQL).
- **Keine / kontrollierte Redundanz:** Um Inkonsistenz und Integritätsverletzungen zu vermeiden, sollte die redundante (mehrfache) Speicherung von identischen Daten verhindert werden. In Ausnahmefällen kann Redundanz zur Performance-Optimierung verwendet werden, dies sollte jedoch überwacht werden.

¹ Vgl. Lexikon „Datenbanken verstehen“

Aufbau – ANSI Architekturmodell

Die ANSI-SPARC-Architektur ist ein abstraktes Design für ein Datenbanksystem (DBS) und wird auch als Drei-Schema-Architektur bzw. Drei-Schichten-Modell bezeichnet.

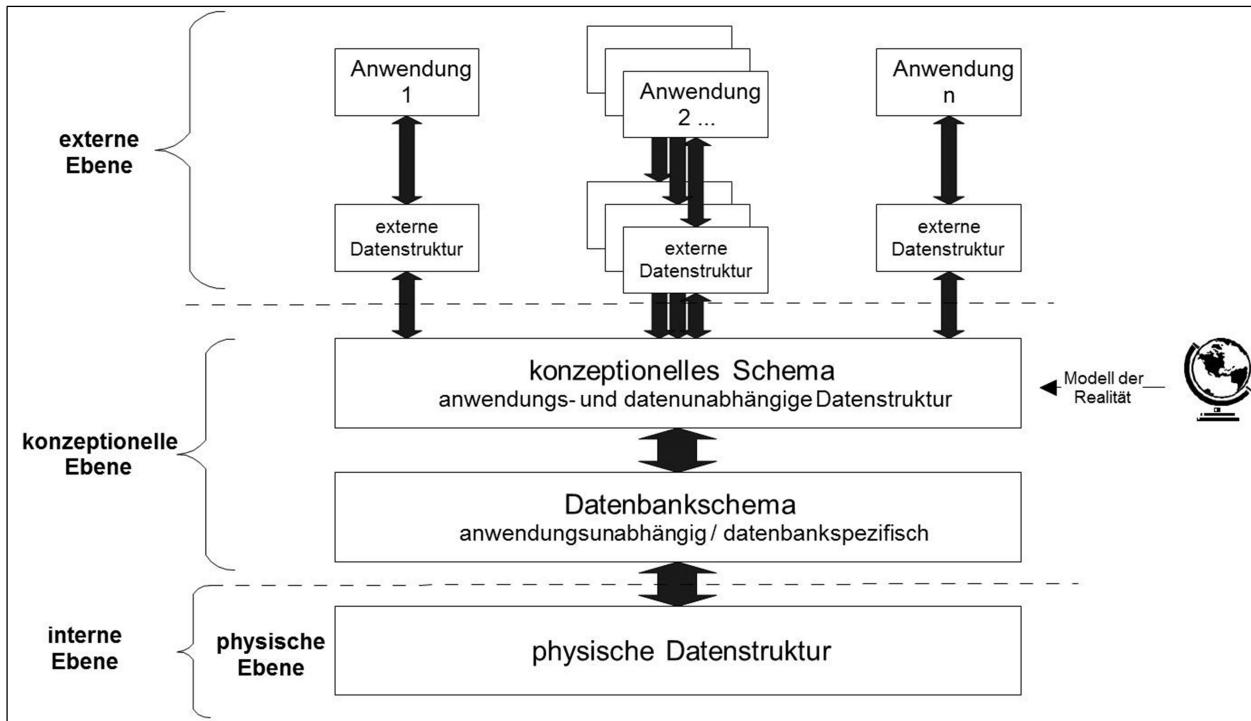


Abb.: ANSI-Architekturmodell²

- Die externe Ebene stellt Benutzern individuelle Benutzersichten, wie Formulare, Listen etc. bereit.
- Die konzeptionelle Ebene beschreibt, welche Daten gespeichert sind und wie deren Beziehung untereinander ist. Designziel ist hier eine vollständige und redundanzfreie Darstellung aller zu speichernden Informationen. (z.B.: ER-Modell)
- Die interne Ebene beschreibt, wie und wo die Daten gespeichert werden. Dies ist die physische Sicht der Datenbank auf den Computer. Das Designziel ist ein effizienter Zugriff auf die Informationen.

Der Vorteil dieser Architektur liegt in der physischen wie logischen Datenunabhängigkeit. Durch die Trennung der Ebenen kann beispielsweise das Speichermedium gewechselt werden, ohne dass dies Einfluss auf die konzeptionelle oder externe Ebene hätte. Weiterhin wirken sich Änderungen im Layout der Datenbank nicht auf Formulare oder andere Schnittstellen aus.³

² Vgl. Wiki FH Köln

³ Vgl. Küspert: Datenbanksysteme 1

Aufgabe 2 „ANSI-Architekturmodell“

Ordnen Sie die folgenden Begriffe den Ebenen im ANSI-Architekturmodell zu!

- a) DBMS
- b) Administrator
- c) Formular
- d) MySQL
- e) ER-Modell
- f) Gespeicherte Daten z.B. C:\Daten\Produkte
- g) GUI
- h) Sachbearbeiter

Ebene	Begriff
externe Ebene	
Konzeptionelle Ebene	
interne Ebene	

Datenbankmodelle

Ein Datenbankmodell ist die theoretische Grundlage für eine Datenbank und legt fest, auf welche Art und Weise die Daten im Datenbanksystem gespeichert und bearbeitet werden können.

Ein Datenbankmodell von heute kann sehr komplex werden, da immer mehr Informationen in einem Datenbankmodell abgebildet und modelliert werden müssen.⁴ Das weitverbreitetste Datenbankmodell ist das relationale Datenbankmodell.

Hierarchisches Datenbankmodell

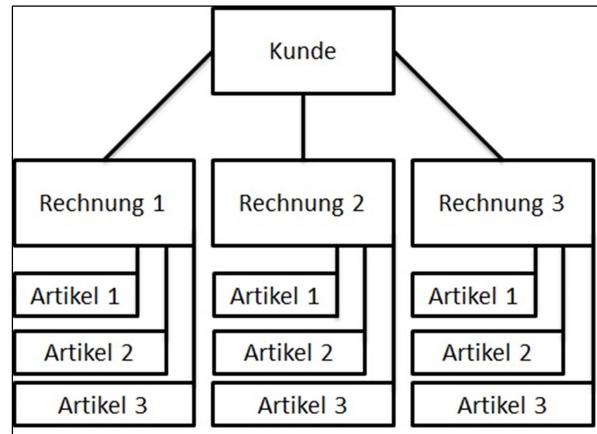
Das hierarchische Datenbankmodell bildet die reale Welt durch eine hierarchisch aufgebaute Baumstruktur ab.

Viele Dateisysteme von Betriebssystemen benutzen dieses Modell, um Ihre Daten abzubilden. Heute ist dieses Datenbankmodell weitestgehend von anderen Modellen abgelöst worden.

Jeder Datensatz hat über Eltern-Kind-Beziehungen genau einen Vorgänger, mit Ausnahme genau eines Datensatzes, der nämlich die Wurzel der Baumstruktur bildet.

Ein gutes Beispiel für ein Hierarchisches Datenbankmodell ist ein Modell, das eine Beziehung zwischen einem Kunden, seinen Rechnungen und den darin enthaltenen Artikeln widerspiegelt:

Abb.: hierarchisches Datenbankmodell⁵

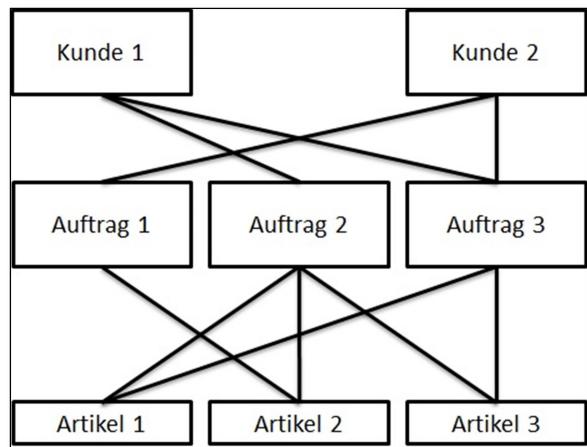


Netzwerkbasiertes Datenbankmodell

Das Netzwerk-Datenbankmodell besitzt eine hohe Beschreibungsmächtigkeit und ist daher näher am E/R-Modell als die hierarchische Variante.

Es besitzt keine strenge Hierarchie. Ein Datenfeld besteht aus einem Namen und einem Wert. Durch das netzwerkartige Modell existieren meist unterschiedliche Suchwege, um einen bestimmten Datensatz zu ermitteln.

Abb.: netzwerkbasiertes Datenbankmodell⁶



⁴ Vgl. Lexikon „Datenbanken verstehen“

⁵ Vgl. Lexikon „Datenbanken verstehen“

⁶ Vgl. Lexikon „Datenbanken verstehen“

Objektorientiertes Datenbankmodell

Ein objektorientiertes Datenbankmodell verfolgt den Ansatz, Daten zusammen mit ihren Funktionen in einem Objekt zu speichern. Das Datenmodell basiert auf den Paradigmen der objektorientierten Programmierung und im Einzelnen der jeweiligen Programmiersprache, die zum Einsatz kommt (z.B. C++, Java oder C#).

Die Abfrage der Objektinformationen kann über die objekteigenen Funktionen oder die Abfragesprache Object Query Language erfolgen. Dadurch steigt die Performance von Abfragen, da aufwendige Verknüpfungen von Relationen wegfallen.

Im Vergleich zu relationalen Datenbanksystemen nimmt die Performance von Transaktionen mit der Zunahme an zu verarbeitenden Daten sehr schnell ab.

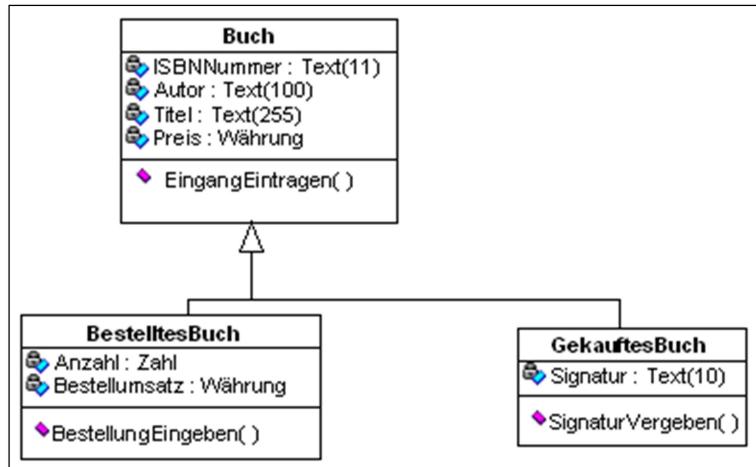


Abb.: objektorientiertes DB-Modell⁷

Relationales Datenbankmodell

Die Darstellung der Daten erfolgt mittels Relationen (=Tabellen), die miteinander verknüpft sind.

Das relationale Datenbankmodell wird in diesem Skript ([siehe Phase 3 der Datenbankentwicklung](#)) noch detailliert beschrieben.

⁷ Vgl. Hans Ott

Datenorganisation

Szenario - "Ein Beispiel, das zu denken gibt!"

An dem folgenden Beispiel eines Sportvereins wollen wir zunächst auf die Problematik der Informationserfassung und -verarbeitung eingehen, um später an Hand dieses Informationsmaterials eine saubere Datenbanklösung erarbeiten.



Ein Sportverein hat auf der Jahresversammlung beschlossen, in Zukunft die Vereinsverwaltung rechnerunterstützt vorzunehmen. Als Datenbankentwickler werden wir beauftragt, alle nötigen Arbeiten zu erledigen. Zunächst erhalten wir einige Dokumente mit den die bisher gepflegten Informationen. Selbstverständlich haben reale Sportvereine eine viel reichere und tiefere Semantik. Zur Zeit werden mittels Excel-Tabelle die Mitglieder und deren Beteiligung an Sportarten geführt:

M_Nr	Vorname	Nachname	m/w	Strasse	PLZ	Ort	Sportart	Beitrag
1	Ulrich	Becker	m	Maxweg 14	85408	Gauting	Gymnastik	80,00
2	Ulrich	Becker	m	Maxweg 41	85408	Gauting	Handball	55,00
3	Julia	Berger	w	Fischerweg 22	85395	Attenham	Handball	55,00
4	Manuela	Fiedmann	w	Badstr. 23	85406	Zolling	Powerwalking	55,00
5	Otto	Fischer	m	Karlweg 12	85375	Neufarn	Fußball	75,00
..								
..								
29	Georg	Frohmann	m	Meierweg 99	85408	Daberg	Fußball	75,00
30	Hans	Huber	m	Poststr. 12	85368	Bruck	Gymnastik	80,00
31	Hans	Huber	m	Poststr. 12	85368	Bruck	Fußball	75,00
..								

Bevor wir weitere Informationsquellen begutachten, wollen wir diese Tabelle analysieren und dabei entdecken, welche "Ungereimtheiten", Fehler und Probleme aufzufinden sind:

Redundanz

Redundanzen sind doppelte Informationen in einer Datenbank bzw. Datenbank-Tabelle. Redundanzen in Datenbanken sind ein Zeichen für ein schlechtes Datenbankdesign.

Beispiel: Wenn ein Mitglied in mehreren Abteilungen aktiv ist, wird die Information über das Mitglied (vollständiger Name und Adresse) mehrfach geführt. So hat ein Mitglied entsprechend mehrere Mitgliedsnummern (Welche identifiziert denn nun eindeutig?).

Redundante Daten sind meist der Grund für Dateninkonsistenz.

Inkonsistenz

Inkonsistenz bezeichnet einen Zustand, in dem zwei Dinge, die beide als gültig angesehen werden sollen, nicht miteinander vereinbar sind. Es liegt also ein Widerspruch vor.

Beispiel: Die Straße von Ulrich Becker: Maxweg 14 und Maxweg 41 ist offensichtlich ein Zahlendreher. Es ist nicht erkennbar, welches die richtigen Daten sind. Somit eine Inkonsistenz, resultiert aus der Redundanz – doppelte Daten (Mitglied Ulrich Becker).

Anomalie

Anomalien in relationalen Datenbanken bezeichnen Fehlverhalten der Datenbank.

Man unterscheidet:

Änderungsanomalie

Auch Update-Anomalie genannt. Gleiche Attribute eines Datensatzes werden in einer Transaktion nicht automatisch geändert. So entsteht eine Dateninkonsistenz.

Beispiel: Beim Ändern von Personendaten muss überprüft werden, in wie vielen Abteilungen das betreffende Mitglied aktiv ist, so dass die Änderungen durchgeführt werden können, da andernfalls unkorrekte Datensätze existieren.

Entsprechend bekommt man Probleme, wenn der Beitrag einer Abteilung geändert werden soll. So muss dies bei allen Mitgliedern dieser Abteilung durchgeführt werden. Wird einer übersehen, ist die Datenbank nicht mehr konsistent.

Einfügeanomalie

Daten werden beim Einfügen in eine Datenbank nicht richtig übernommen.

Beispiel: Wenn eine neue Abteilung aufgemacht werden soll, kann dies nur in Verbindung mit zugehörigen Mitgliedern geschehen, da andernfalls der Datensatz nicht in diese eine Tabelle eingegeben werden kann.

Löschanomalie

Hier kann es passieren, dass beim Löschen von Informationen indirekt andere zusammenhängende Informationen parallel mitlöscht werden.

Beispiel: Unter Umständen wird eine ganze Abteilung gelöscht, falls alle momentan in der Abteilung aktiven Mitglieder gelöscht werden (Löschen von Mitglied Nr. 4/Fiedmann => Powerwalking als Sportart wird auch gelöscht).

Aufgabe 3 „Datenorganisation“

Situation "Getränkequelle"

Ein Getränkehändler, der nur Wiederverkäufer und Großverbraucher beliefert, unterstützt die Auftragsbearbeitung mit Hilfe einer einfachen Tabellenkalkulation. Die Datenhaltung mit diesem System weist hohe Redundanzen auf, die zu dieser aufwändigen und fehleranfälligen Datenpflege gehören. Die Informationen sind unübersichtlich angeordnet, Auswertungen lassen sich nur umständlich erstellen. Für Aktualisierungen, etwa eine Sortimentserweiterung, ist das System zu unflexibel.

Zunächst ein Ausschnitt aus dem vorhandenen Datenmaterial:

AUFNR	AUFDAT	KNR	KNAME	PLZ	ORT	STRASSE	TELEFON	GNR	GART	EINHEIT	PREIS	MENGE
104	10.01.04	K3	Uni-Mensa	95448	Bayreuth	Uniweg 3	0921/123	W1	Stilles Wasser	Kasten	5,00 €	45
104	10.01.04	K3	Uni-Mensa	95448	Bayreuth	Uniweg 30	0921/123	W2	Spritziges Wasser	Kasten	7,50 €	50
204	10.01.04	K1	Panorama-Hotel	95444	Bayreuth	Badstr. 20	0921/543	C20	Afri-Cola	10l Container	15,00 €	30
								B12	Bionade	Kasten	15,80 €	
304	10.01.04	K4	M-Getränke	95463	Bindlach	Hofstr. 2	09208/894	W2	Spritziges Wasser	Kasten	7,50 €	60
...
704	15.02.04	K3	Uni-Mensa	95448	Bayreuth	Uniweg 3	0921/123	W1	Stilles Wasser	Kasten	5,00 €	45
704	15.02.04	K3	Uni-Mensa	95448	Bayreuth	Uniweg 3	0921/123	W2	Spritziges Wasser	Kasten	7,50 €	50
904	10.01.04	K1	Hotel Rabat	91233	Hof	Bergweg 2	0921/543	C21	Schweppes	Kasten	13,50 €	5
...

Analysieren Sie die obige Tabelle hinsichtlich möglicher Mängel, Fehler, Redundanzen, Inkonsistenzen und Anomalien!

Phasen der Datenbankentwicklung

Die Entwicklung einer Datenbankanwendung, mit der die gewünschten und benötigten Informationen erfasst, gepflegt und bereitgestellt werden können, kann in unterschiedlich viele aufeinander folgende Phasen eingeteilt werden.

Die bekannten Vorgehensweisen unterscheiden sich darin, wie detailliert die einzelnen notwendigen Aktionen untergliedert werden. In der Praxis hat erfolgt die Datenbankentwicklung meist in vier Phasen.

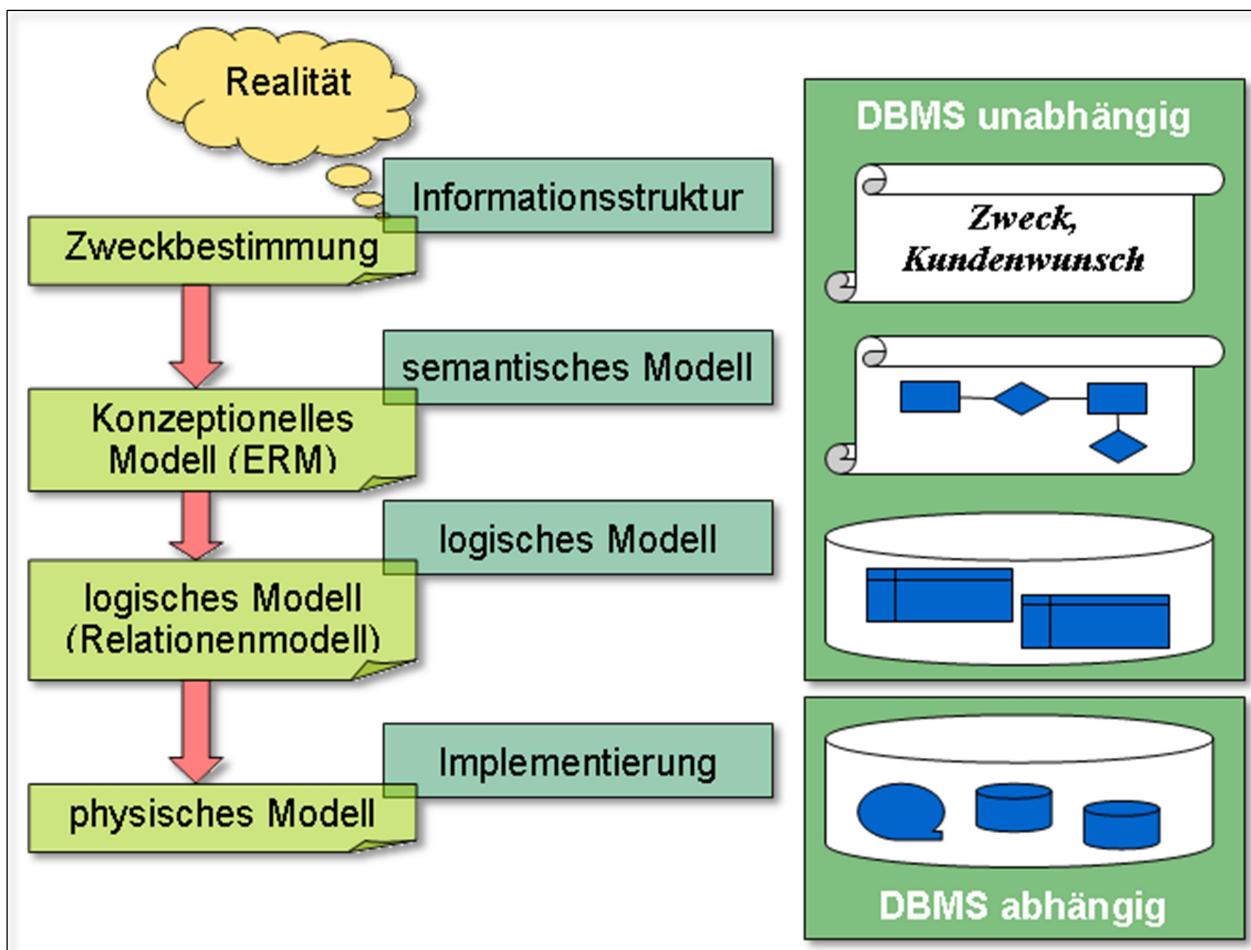


Abb.: Phasen Datenbankentwicklung⁸

Phase 1: Anforderungsanalyse -> Pflichtenheft

Phase 2: Konzeptionelle Phase -> ER-Modell/Semantisches Modell/Klassendiagramm

Phase 3: Logische Phase -> Verarbeitungsorientiertes Modell/relationales Modell

Phase 4: Physische Implementierung mittels DBMS, z.B. MySQL, DB2, MaxDB,...

⁸ Vgl. Gymnasium Fulda

Anforderungsanalyse (Phase 1)

In der ersten Phase wird der Informationsbedarf der Benutzer ermittelt. Dazu dienen die Verfahren der Ist-Aufnahme mit anschließender Analyse der gewonnenen Informationen. Hier fließen auch die Zielvorstellungen des Auftraggebers mit ein.

Für die Durchführung der Ist-Aufnahme können unterschiedliche Techniken eingesetzt werden:

- Auswertung vorhandener Unterlagen
- Befragung
- Beobachtung
- Selbstaufschreibung
- Sonstige Techniken (aktive Mitarbeit, Schätzungen, usw.)

Es muss analysiert werden, welche Informationen das Informationssystem bereitstellen soll (Output) und welche Informationen dafür benötigt werden (Input). Daraus ergibt sich die Informationsstruktur. Der Input bildet später die Datenbasis der Datenbank (z. B. Objekte wie Mitglieder, Trainer und Sportarten), der Output die zu erzielenden Ergebnisse, die Benutzersichten (views), z. B. in Form von Berichten und Formularen (Darstellungsobjekte wie Mitglieder-Erfassungsmasken oder Listen).

Das Ergebnis der Anforderungsanalyse wird häufig in Form eines **Pflichtenhefts** festgehalten.

Bei der Darstellung der Informationsstruktur lassen sich u.a. folgende zwei Ansätze unterscheiden:

- **Top-down-Ansatz** (globales/unternehmensweites Datenmodell)

Die Informationsanforderungen aller späteren Datenbanknutzer (nicht die einzelne Anwendung) bestimmen die Informationsstruktur.

Beispiel 1:

Im Rahmen der Entwicklung einer Datenbank für die Verwaltung einer Schule werden *alle* relevanten Objekte (Schüler, Lehrer, Klassen, Fächer, Eltern, Räume, Ausstattung usw.) erfasst.

Beispiel 2:

Das System SAP/R3 basiert auf einem *globalen*, unternehmensweiten Datenmodell zur Abbildung industrieller Geschäftsprozesse (ARIS-Modell, "Architektur rechnergestützter Informationssysteme").

Es wird zunächst eine grobes Datenmodell entworfen und dann schrittweise verfeinert, so dass einzelne Anwendungen (Applikationen) entstehen.

- **Bottom-up-Ansatz** (anwendungsorientiertes Datenmodell)

Ein spezielles Problem ist Ausgangspunkt für die Datenbankentwicklung. Für die Lösung des Problems wird eine Anwendung entwickelt.

Beispiel 3

Um Zeit zu sparen, sollen nur die Zeugnisse und Schulbesuchsberechtigungen über eine EDV-Anlage ausgefertigt werden.

Die Integration der einzelnen und auch später noch zu entwickelnden Anwendungen kann dann zu einem umfassenderen Datenmodell führen.

Konzeptionelle Phase – ER Modell (Phase 2)

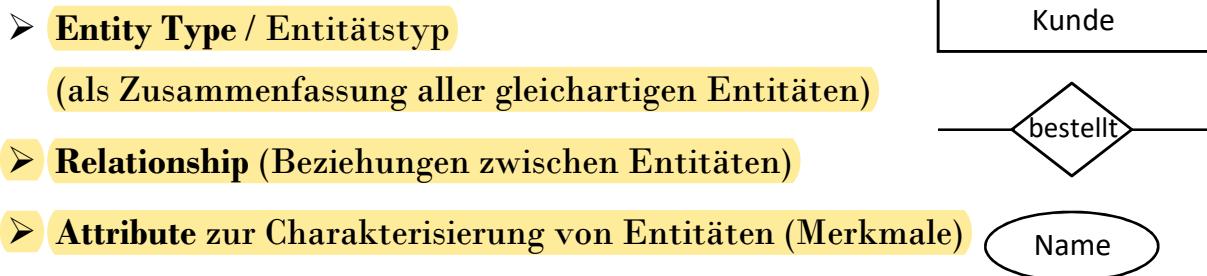
Nach der Festlegung des Informationsbedarfs werden die Ergebnisse der ermittelten Informationsstruktur formalisiert. Dazu werden alle relevanten Objekte und Beziehungen zwischen den Objekten in einem semantischen Datenmodell erfasst und beschrieben. Das zu entwickelnde Modell dient zum einen in der konzeptionellen Phase der Anwendungsentwicklung der Verständigung zwischen Anwendern und Entwicklern, zum anderen in der Implementierungsphase als Grundlage für das Design der Datenbank.

In der konzeptionellen Phase werden zwei Strukturierungs- und Darstellungsverfahren verwendet: das Klassenmodell (UML – Unified Modelling Language) und das Entity-Relationship-Modell (ERM).

Entity-Relationship-Modell (ERM) bzw. semantisches Modell

Alle benötigten Informationen werden auf einer logischen, für den Anwender nachvollziehbaren Ebene dargestellt.

Die Daten werden also in einer völlig technikneutralen Ebene, mit großer Nähe zur Realität mit Hilfe von drei Darstellungselementen abgebildet:



Entity Type

Eine **Entität** (entity) ist eine eindeutig identifizierbare Einheit. Zum Beispiel ein Datensatz, also eine Zeile in einer Tabelle. Man spricht auch von einer Instanz der zu Grunde liegenden Entity Type.

Eine **Entity Type** fasst alle Entitäten zusammen, die durch gleiche Merkmale charakterisiert sind.

Zur grafischen Darstellung von Entity Types werden Rechtecke verwendet. Die Bezeichnung wird im Singular formuliert.

Relationship

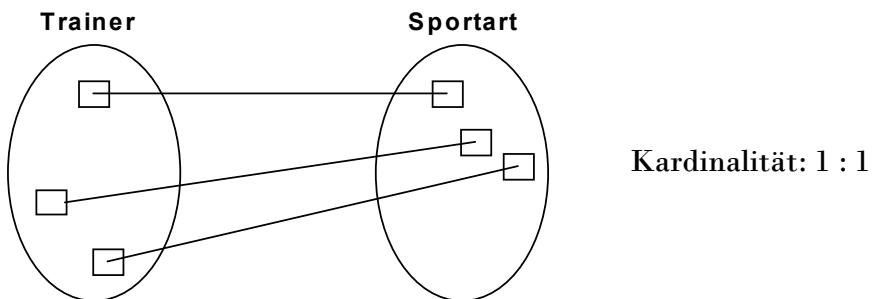
Eine **Beziehung** (Relationship/Assoziation) kann wechselseitig zwischen zwei (oder mehreren) Entitäten auftreten. Dabei bedeutet Assoziation, dass eine Entität eine andere Entität kennt und mit ihr in Wechselwirkung steht.

Es gibt viele Arten von Beziehungen (Kardinalitäten). Wir beschränken uns hier auf drei Unterscheidungen:

Die Kardinalität einer Assoziation $a(E1, E2)$ gibt an, wieviel Entitäten der Entity Type $E2$ einer beliebigen Entität der Entity Type $E1$ zugeordnet sein können.

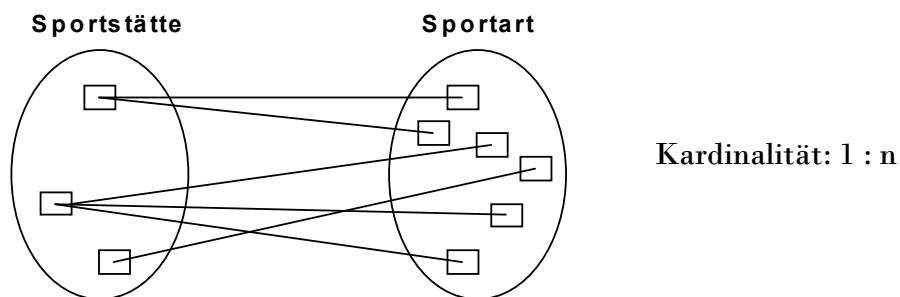
- Konstellation 1: 1 : 1 - Beziehung

Ein Trainer ist nur in einer Sportart Übungsleiter. Dies ist sicher eine vereinfachende Einschränkung, wir wollen aber mit ihr modellieren.



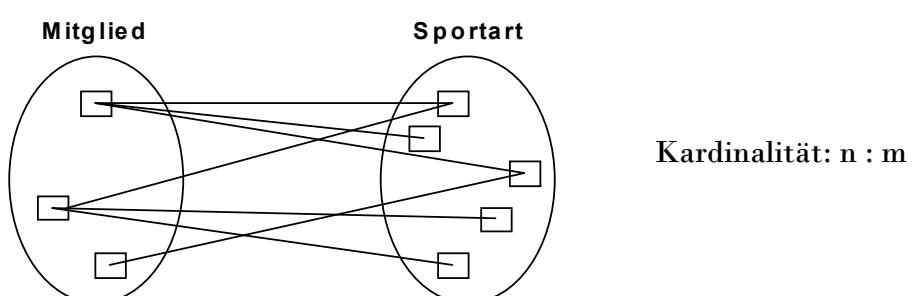
- Konstellation 2: 1 : n - Beziehung

Wir wollen so modellieren, dass eine Sportart nur an einer Sportstätte ausgeübt werden kann. Allerdings bietet eine Sportstätte schon Übungsgelegenheiten für unterschiedliche Betätigungen.

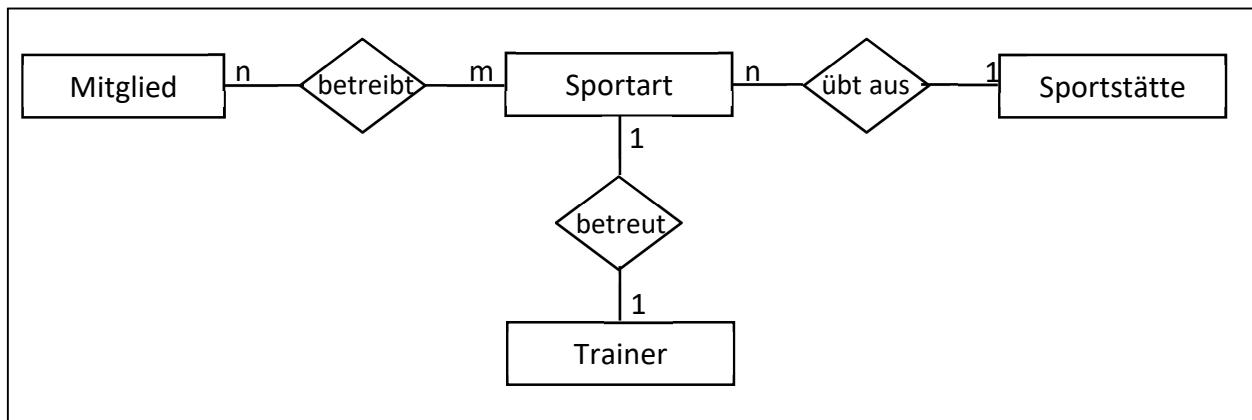


- Konstellation 3: n : m - Beziehung

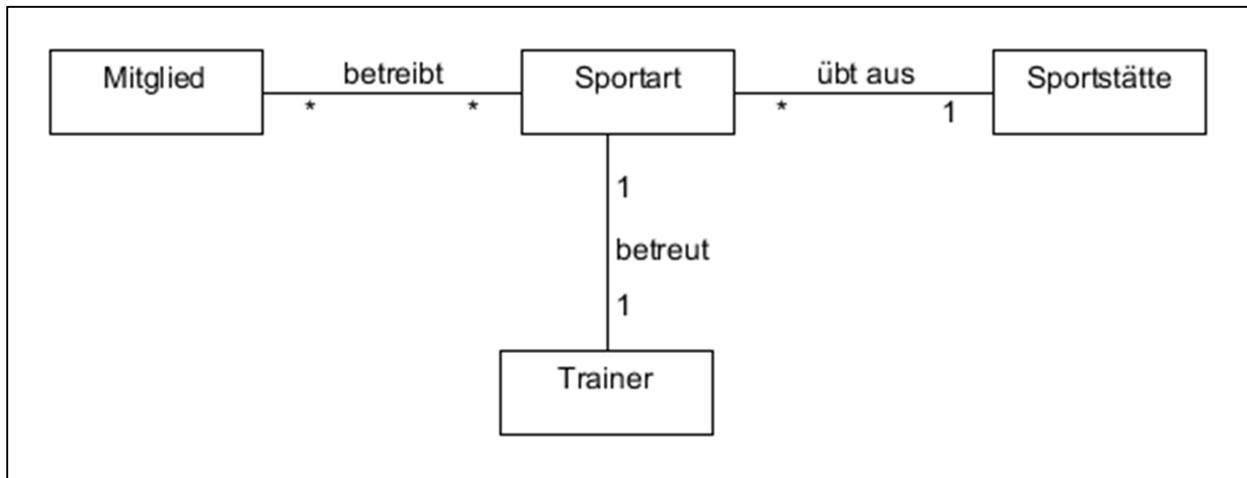
Ein Mitglied kann möglicherweise mehrere Sportarten ausüben. Und eine Sportart kann auch von mehreren Mitgliedern betrieben werden.



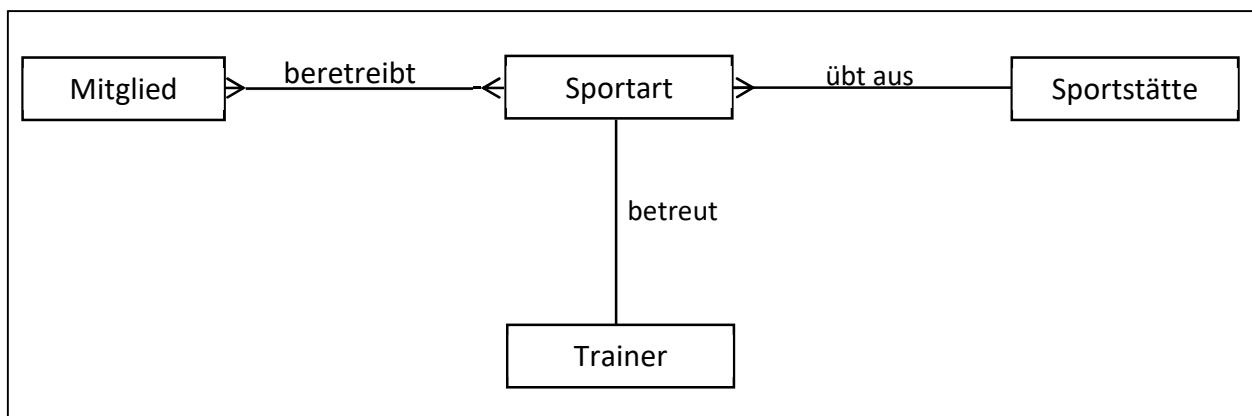
Für unser Beispiel „Sportverein“ lässt sich nun folgendes ER-Modell (Chen-Notation) darstellen:



Eine andere Darstellungsmöglichkeit wäre das UML-Klassendiagramm:



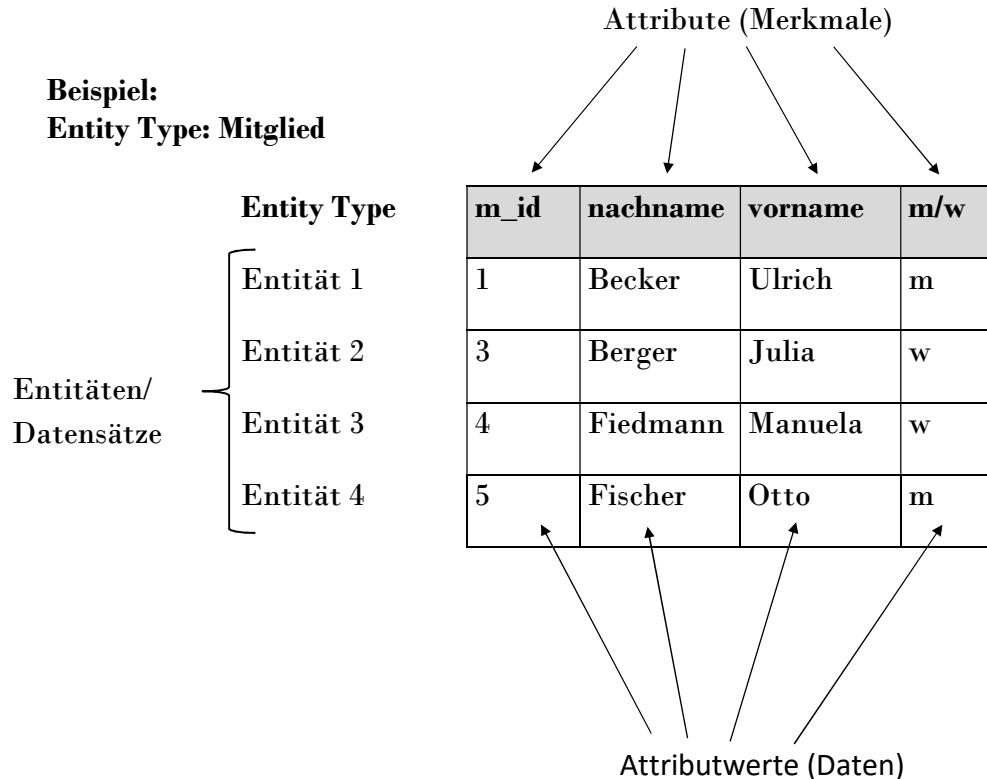
Eine weitere Variante der Darstellung: ER-Modell (Martin-Notation / „Krähenfuß“):



Die meist verbreitete Variante ist die Darstellung nach Chen. Sie wird jedoch mehr und mehr durch die UML-Darstellung ersetzt.

Attribute

Die im semantischen Modell abgebildeten Entity Types und Beziehungen können präzisiert werden, wenn man Attribute (Merkmale) erfasst, die man diesen Entity Types und Beziehungen zuordnet.



Ein **Attribut** (engl. attribute) beschreibt eine bestimmte Eigenschaft, die sämtliche Entitäten einer Entity Type aufweisen. Es wird im ER-Modell als Oval dargestellt.

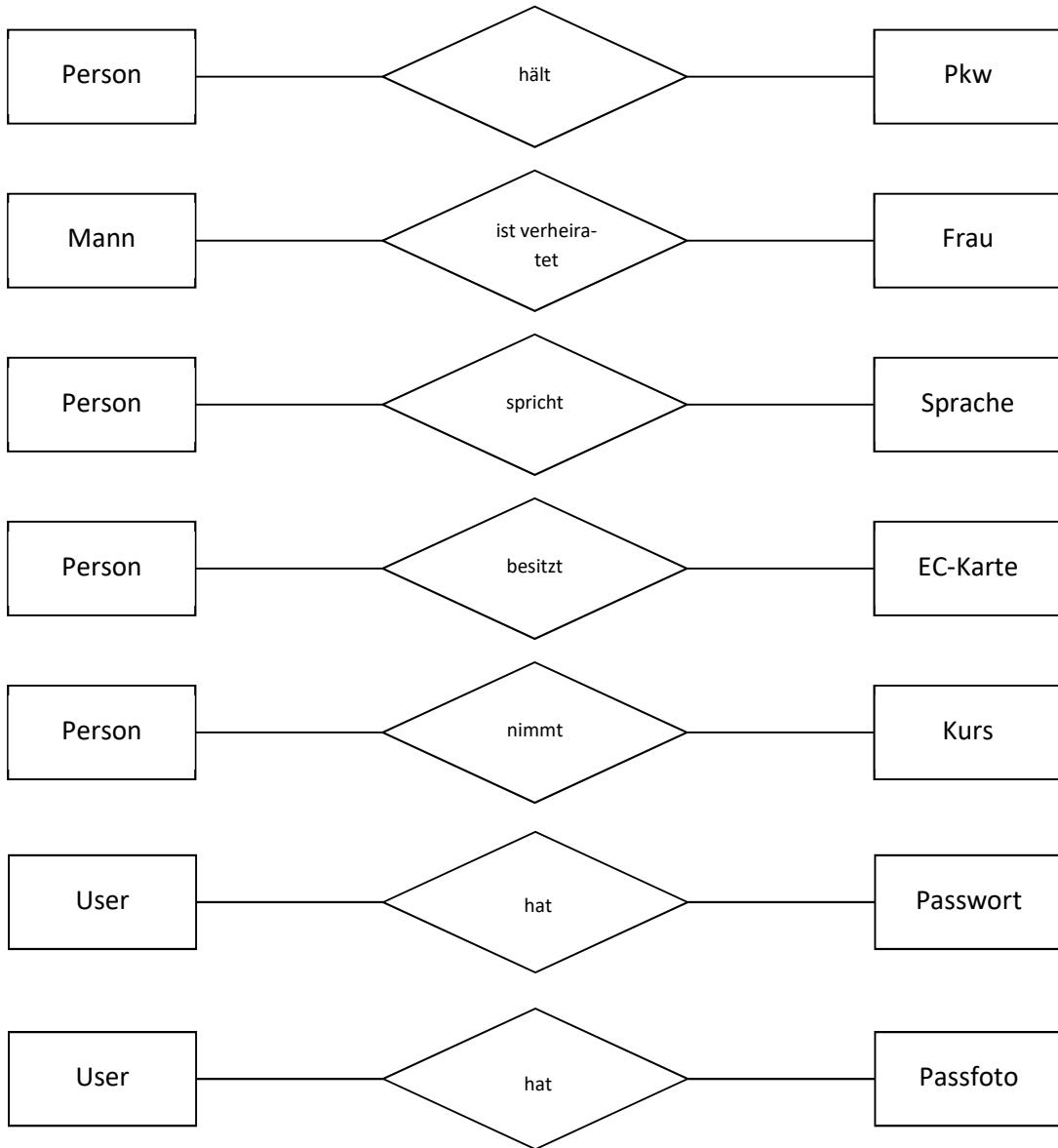


Die Attribute sind im optional. Um die ER-Modelle übersichtlich zu halten, werden Attribute in den meisten Fällen nicht dargestellt. Im Tabellen-Modell (Phase 3) trägt man die Attribute ein.

Aufgaben „ER-Modell“

Aufgabe 4:

Ermitteln Sie die Kardinalitäten der Beziehungen!



Aufgabe 5: Projekte

Folgender kurze Ausschnitt aus einem Entwurf zur Modellierung der Informationsstruktur eines Unternehmens soll modelliert werden:

- Eine Abteilung besteht aus mindestens einem Mitarbeiter. Jeder Mitarbeiter ist einer Abteilung zugeordnet.
- Es gibt die unterschiedlichsten Projekte. In den Projekten können Mitarbeiter aus verschiedenen Abteilungen eingesetzt werden.

Erstellen Sie das semantische Modell ohne Attribute!

Aufgabe 6: Schulverwaltung

Für ein Schulverwaltungssystem liegen folgende Informationen vor:

- In einer Klasse sitzen bis zu 32 Schüler.
- Jede Klasse wird von einigen Lehrern unterrichtet.
- Von den Schülern und Lehrern sollen zunächst nur Name und Vorname bekannt sein.
- Bei den Klassen interessiert lediglich die Klassenbezeichnung. Die Unterrichtsfächer werden aus Gründen der Vereinfachung nicht berücksichtigt.
- In jeder Klasse wird ein Schüler zum Klassensprecher gewählt.
- Außerdem wird ein Lehrer, der in der Klasse unterrichtet, zum Klassenleiter ernannt."

Erstellen Sie das semantische Modell ohne Attribute!

Aufgabe 7: Kinoprogramm

Auf www.kino.de können Sie das Kinoprogramm abrufen. Für München bspw. sieht das so aus (Ausschnitt):

Cadillac & Veranda 81925 München, Rosenkavalierplatz 12 Stadtteil: Bogenhausen	<ul style="list-style-type: none">▶ Merida - Legende der Highlands 16:00▶ Schutzengel 17:15 20:00▶ Vatertage - Opa über Nacht 20:00▶ Wer's glaubt wird selig 18:00
Cincinnati  81549 München, Cincinnatistr. 31 Stadtteil: Obergiesing	<ul style="list-style-type: none">▶ Vatertage - Opa über Nacht 20:30▶ Wer's glaubt wird selig 18:30
Cinema  80335 München, Nymphenburger Str. 31 Stadtteil: Maxvorstadt	<ul style="list-style-type: none">▶ Resident Evil: Retribution (OV)  21:45▶ To Rome with Love (OV) 16:45▶ Wie beim ersten Mal (OV) 19:20
CinemaxX München  80331 München, Isartorplatz 8 Stadtteil: Altstadt	

Erstellen Sie das semantische Modell ohne Attribute, welches der Kino.de-Datenbank zugrunde liegen könnte!

Aufgabe 8: Keks-Produktion

Die MAMUT AG produziert Kekse. Um einen reibungslosen Ablauf zu gewährleisten, sollen Produktion und Vertrieb mit einem Datenbanksystem verwaltet werden.

Hierzu hat man folgende Hinweise zur Informationsstruktur:



- Die Kekse werden in verschiedenen Sortimenten (Packungen) zusammengestellt. In jeder Packung können alle Kekse in verschiedener Stückzahl vorkommen.
- Die Kekse werden aus mehreren Zutaten gebacken. Dabei soll auch vermerkt werden, in welcher Menge und Mengeneinheit die einzelnen Zutaten zur Herstellung der jeweiligen Kekse verwendet werden.
- Die Zutaten (Rohstoffe) werden in eigenen Regalen gelagert. Dabei ist ein Rohstoff nur in einem bestimmten Regal zu finden, in einem Regal können allerdings verschiedene Rohstoffe gelagert werden.
- Da für einige wenige Rohstoffe spezielle Lagerbedingungen gelten, sollen diese in einer eigenen Liste verwaltet werden. Für einen Rohstoff existiert nur ein Eintrag in dieser Liste. Jeder Eintrag (z.B. kühl, trocken,...) kann jedoch mehreren Rohstoffen zugeordnet werden.
- Einzelne Zutaten können von verschiedenen Lieferanten bezogen werden. Ein Lieferant kann aber auch unterschiedliche Zutaten liefern. EK-Preis und Lieferbedingung hängen sowohl vom Lieferanten als auch vom Rohstoff ab.

Erstellen Sie das semantische Modell ohne Attribute!

Aufgabe 9: Wald

Für die Verwaltung der Wälder einer Region liegen leider nur sehr oberflächliche Angaben vor. Versuchen Sie trotzdem sich ein Bild von der zu modellierenden Miniwelt zu machen!

- Jeder Wald hat einen Namen und eine eindeutige Nummer
- Ein Wald ist einem Ort(Gemeinde/Stadt) zugeordnet. Es gibt Wälder, die auf dem Gebiet mehrerer Gemeinden liegen.
- Ein Wald hat einen Eigentümer.
- Es gibt diverse Arten von Bäumen. Die Baumart hat eine Bezeichnung (z. B. Tanne, Fichte), einen Typ (z.B. Nadelbaum oder Laubbbaum) und eine Maximalhöhe.
- In einem Wald können mehrere Arten vorkommen.
- Einige Wälder haben konkrete Versuchsbäume für Untersuchungen über den Einfluss der Luftverschmutzung. Die Versuchsbäume sind mit einer eindeutigen Nummer gekennzeichnet. Das Pflanzdatum des Versuchsbiums ist ebenfalls gespeichert.
- An den Versuchsbäumen werden Messungen durchgeführt, die mit Datum, Nummer und Messergebnis vermerkt werden.

Erstellen Sie das semantische Modell mit Attributen, welches alle genannten Punkte abbildet!

Logische Phase (Phase 3): Entwurf des verarbeitungsorientierten Modells/ relationales Datenbankmodell

Im dritten Schritt findet der Übergang statt vom semantischen Modell (ER-Modell) zum relationalen Modell, das als Darstellungselemente Tabellen benutzt. Nach Erstellung des relationalen Modells werden die Tabellen auf Redundanzfreiheit geprüft: Normalisierung.

Schlüssel

Jeder Datensatz einer Tabelle muss eindeutig identifizierbar sein. Dies kann durch ein Attribut oder eine Kombination von Attributen gewährleistet werden, wodurch die jeweiligen Entitäten einer Entity Type eindeutig identifiziert werden.

Bei einer Entity Type *Schüler* könnte man z. B. die Attributkombination aus *Name*, *Vorname* und *Geburtsdatum* wählen. Diese Kombination (zusammengesetzter Schlüssel) würde wohl nur einmal vorkommen und den einzelnen Schüler eindeutig identifizieren. Allerdings ist dies 1. in der Realität nicht garantiert ("Maier, Josef, 01.01.1997" kann doch zweimal vorkommen!) und 2. ist ein zusammengesetzter Schlüssel in Bezug auf Performanz langsamer.

Ein Schlüssel (key) ist ein Bestandteil eines Datensatzes, der aus einem oder mehreren Datenfeldern besteht und über den dieser Datensatz aufgefunden werden kann (entspricht oft einem Ordnungsbegriff, z.B. Kundennummer). Allerdings ist es günstiger, einen künstlichen Schlüssel (Nummer der Entität) einzurichten.

Man unterscheidet:

- **Primärschlüssel** (primary key, PK, Hauptordnungsbegriff)

Ein Primärschlüssel, auch Identifikationsschlüssel (identification key) genannt, besteht aus einem Attribut oder aus einer Kombination von Attributen, welche jede Entität einer Entity Type eindeutig (unique) identifizieren. Manchmal ist er eine laufende Nummer, die vom DBMS automatisch vergeben wird.

- **Fremdschlüssel** (foreign key, FK)

Eine feste Beziehung (Verknüpfung) zwischen den Relationen (Tabellen) wird über Fremdschlüssel hergestellt. Der Fremdschlüssel muss in der Ursprungstabelle ein Primärschlüssel (Identifikationsschlüssel) sein.

- **Sekundärschlüssel** (secondary key, Nebenordnungsbegriff)

Der Sekundärschlüssel dient nicht zur Identifikation einer Entität sondern ist bei der schnelleren Suche behilflich. (Beispiel: Sekundärschlüssel auf dem Nachnamen in der Tabelle *personal*: Suche über Nachnamen, Matchcode). Ein Sekundärschlüssel kann dabei ein einzelnes Attribut sein (z. B. der *nachname*) oder auch eine Kombination von Attributen (etwa *nachname* und *wohnort*).

- **Künstlicher Schlüssel** (artificial key)

Es empfiehlt sich, einen eigenen, künstlichen Schlüssel (integer, unique) zu definieren, um garantiert immer nur einen eindeutigen Kennzeichner für ein Tupel zu haben.

- **Zusammengesetzter Schlüssel** (concatenated key)

Dieser Schlüssel besteht, wie es der Name schon sagt, aus mindestens zwei verketteten Attributen.

Abbildungsregeln

Die Umsetzung des semantischen Modells in das verarbeitungsorientierte Modell (logisches Modell) wird am Beispiel des Sportvereins vorgenommen:

Regel 1: Jede Entity Type wird als eigenständige Tabelle abgebildet.

In jeder Tabelle wird ein Attribut festgelegt, das zur eindeutigen Identifizierung einer Entität (Tupel) taugt. -> Primärschlüssel! Dieses Attribut kann einmal ein natürliches Attribut der Tabelle sein (vielleicht die Bezeichnung der Sportstätte). Generell ist es jedoch sinnvoll, ein "künstliches" Attribut als Schlüssel zu wählen (sp_id). Damit ist dann die Eindeutigkeit gewährleistet und zufällige Namensgleichheiten bleiben konsequenzlos.

Die weiteren Attribute werden zu Spalten dieser Tabelle. Daraus ergeben sich die Tabellen Mitglied, Sportart, Trainer und Sportstätte.

Regel 2: Eine 1:1-Beziehung wird dadurch abgebildet, dass der Primärschlüssel einer der beiden Tabellen als zusätzliche Spalte in die andere Tabelle als Fremdschlüssel aufgenommen wird. In welcher Tabelle dies geschieht, ist von der Sachlage abhängig. In unserem Beispiel ist es wahrscheinlich sinnvoll, den Primärschlüssel der Tabelle "Trainer" (t_id) in Tabelle "Sportart" als Fremdschlüssel einzufügen.

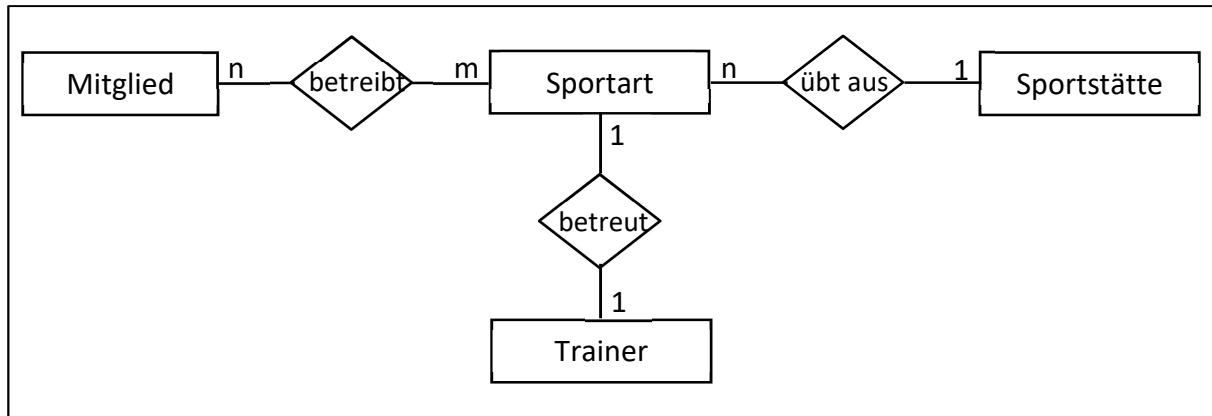
Regel 3: Eine 1:n-Beziehung wird dadurch abgebildet, dass der Primärschlüssel der "1"-Beziehungsseite als zusätzliche Spalte in die "n"-Seite als Fremdschlüssel eingefügt wird. Da gibt es keine Wahlfreiheit! In unserem Beispiel wird der Primärschlüssel der Tabelle "Sportstätte" (sp_id) in die Tabelle "Sportart" eingefügt.

Regel 4: Eine m:n-Beziehung wird dadurch abgebildet, dass eine neue Tabelle erzeugt wird, die diese Beziehung repräsentiert. In unserem Beispiel wird die m:n-Beziehung Mitglieder üben Sportart aus in eine neue Beziehungstabelle ZT_Mitglied_Sportart aufgelöst. Wir erhalten dadurch eine neue 1:n- und eine n:1-Beziehung.

Die beiden Fremdschlüsselattribute (m_id und s_id) ergeben zusammen den Primärschlüssel.

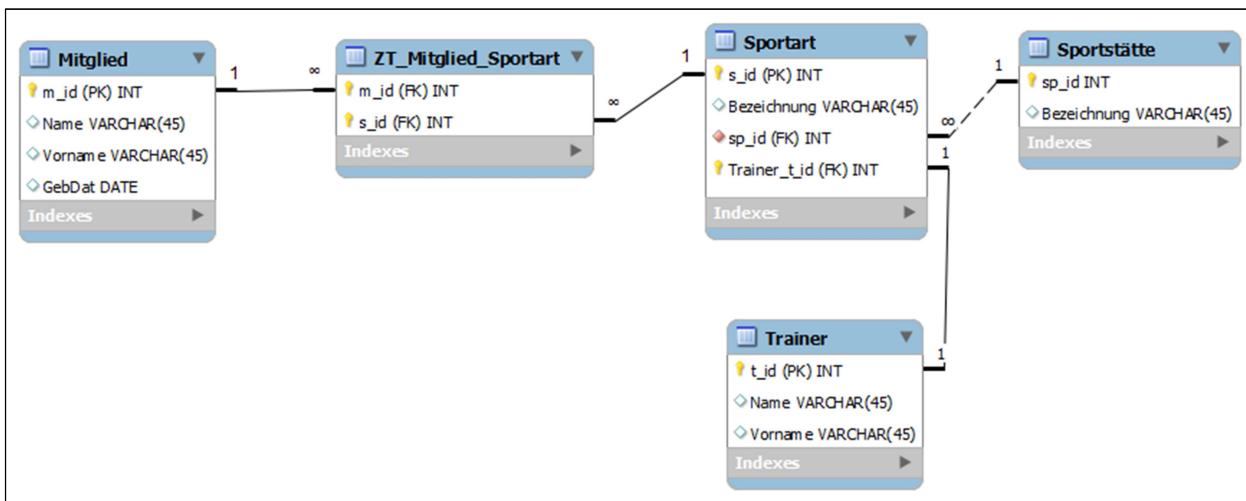
Grafische Darstellung

Für unser Beispiel „Sportverein“ hatten wir dieses **ER-Modell** entwickelt:



Mit Hilfe der Abbildungsregeln ergibt sich folgendes

Relationales Modell



Tipp: Mit Hilfe des MySQL Workbench lassen sich sehr einfache relationale Datenbankmodelle erstellen.



Verwenden Sie dazu das Tool Data Modelling.

Die Zwischentabelle wird beim Einfügen einer n:m-Beziehung automatisch erzeugt.

Die erzeugten Modelle können später auch als SQL CREATE SCRIPT exportiert werden.

Aufgaben „relationales Datenbankmodell“ (Teil 1)

Aufgabe 10: Zertifikat

Das Unternehmen DWA (<http://de.dwa.de>) bietet verschiedene Seminarveranstaltungen an. Am Ende eines Kurses wird den Teilnehmern ein Zertifikat ausgestellt.

Das kann dann zum Beispiel so aussehen:



- Das gleiche Seminar(Kurs mit bestimmten Inhalten) wird zu verschiedenen Zeiten/Zeiträumen angeboten. -> Seminartermin.
- An einem Seminartermin können mehrere Studenten teilnehmen.
- Jeder Seminartermin hat einen speziellen Kursleiter. Wobei das gleiche Seminar an unterschiedlichen Terminen auch unterschiedliche Kursleiter haben kann.

Erstellen Sie zunächst ein ER-Modell (Skizze).

Wandeln Sie nun das ER-Modell mittels Abbildungsregeln in ein relationales DB-Modell um!

Aufgabe 11: Rechnung

Die Fakturierung der Software-Direct KG soll mit Hilfe eines relationalen Datenbanksystems erfolgen.

Software-Direct KG
Software aus dem Internet zu sagenhaften Preisen

Software-Direct KG, Postfach 23 45, 18055 Rostock

Jürgen Schuster
Hauptstraße 123

34266 Niestetal

Kundennummer 8847
Bestellung vom 16.04.04
Rechnungsnummer 4711

Sehr geehrter Herr Schuster,

wir fakturieren für unsere Lieferung vom 17.04.2004:

Pos	Art-Nr	Artikel	Menge	Einzelpreis	Gesamt
1	187	3D Graphic Power	1	98,00 €	98,00 €
2	243	Super Games	2	34,20 €	68,40 €
				Nettobetrag	166,40 €
				Umsatzsteuer 16 %	26,62 €
				Bruttobetrag	<u>193,02 €</u>

Geschäftsadresse
Ernst-Reuter-Platz 1-3
18055 Rostock

Bankverbindung
Hanse Bank
BLZ 100 200 00
Kto.Nr. 0116836

Amtsgericht
Rostock
HRA 390822

UStd. IdNr. DE 5826984258, Steuernummer 108/5155/1453215

Erstellen Sie zunächst ein ER-Modell (Skizze), welches den Vorgang „Kunde erhält Rechnung über bestellte Produkte“ abbildet.

Wandeln Sie nun das ER-Modell mittels Abbildungsregeln in ein relationales DB-Modell um!



Verwenden Sie dazu das MySQL Workbench!

Normalisierung

Im relationalen Datenmodell ist die Normalisierung ein wichtiges Konzept, um einen guten logischen Entwurf zu erhalten. Mit Hilfe der Normalisierung werden sogenannte Anomalien vermieden, die bei schlecht entworfenen Relationenschemata auftreten können.

Durch die Normalisierung werden die vorhandenen Schemata verfeinert, d.h. es erfolgt eine Zerlegung der Relationen. Dieser Prozess ist, wie schon betont, dann erforderlich, wenn nicht zusammenpassende Relationen modelliert wurden. Die Normalisierung ist demnach das formale Konzept, mit dem die unzulänglichen Entwürfe korrigiert werden können.

Die Normalisierung erfolgt in mehreren Teilschritten, die Normalformen genannt werden. Im Folgenden werden die ersten drei Normalformen behandelt. Es gibt noch weitere Normalformen, die jedoch nicht die Bedeutung wie die ersten drei Normalformen haben.

In jedem Falle muss die ursprüngliche Relation nach dem Normalisierungsprozess wieder werden rekonstruierbar sein. Man sagt, es gilt für die Normalisierung das *Dualitätsprinzip*.

Ausgangsbeispiel

PersNr	Mitarbeiter	AbtNr	Abteilung	ProjektNr	Beschreibung	Zeit
1	Lorenz Sophia	1	Personal	2	Verkaufspromotion	83
2	Hohl Tatjana	2	Einkauf	3	Konkurrenzanalyse	29
3	Wild Theodor	1	Personal	1, 2, 3	Kundenumfrage, Verkaufspromotion, Konkurrenzanalyse	140, 92, 110
4	Richter Hans-Otto	3	Verkauf	2	Verkaufspromotion	67
5	Wieland Brunhilde	2	Einkauf	1	Kundenumfrage	160

Erste Normalform (1. NF)

Die **erste Normalform** verlangt, dass in der Relation nur **atomare Werte** vorhanden sind. Alle Werte, die zusammen mit einem anderen Wert in der Domäne stehen, können später nicht einzeln abgefragt werden.

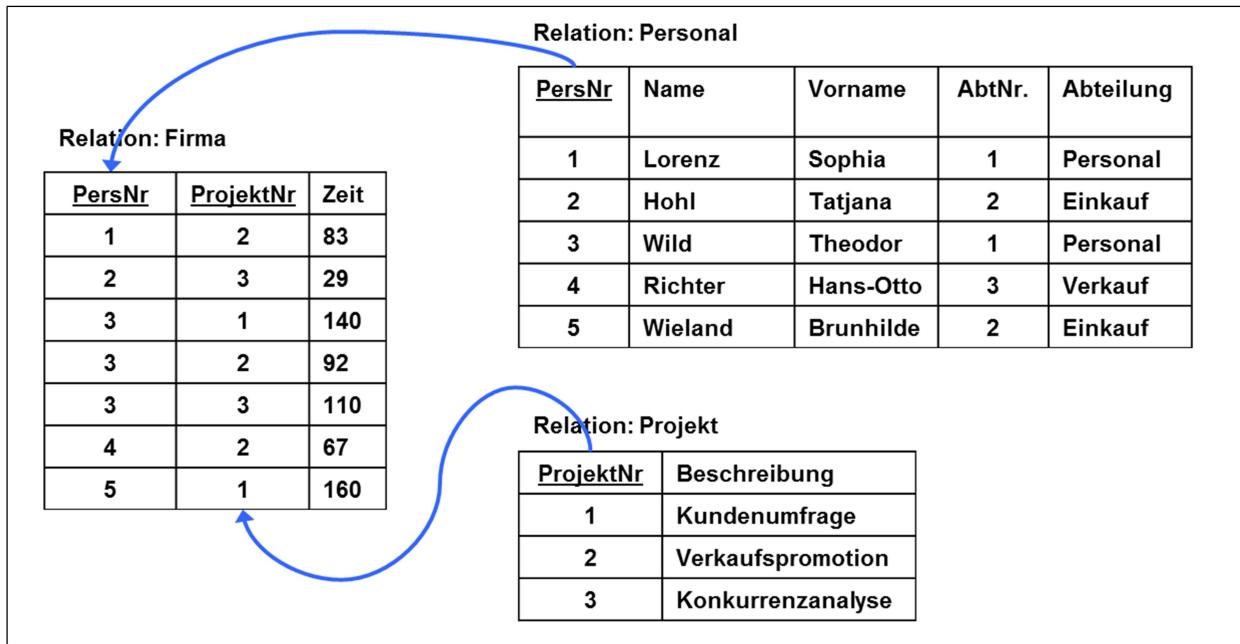
D.h. für unser Beispiel: **Mitarbeiter** muss in zwei Spalten Vorname und Nachname, aufgeteilt werden. Außerdem muss die Spalte **Beschreibungen** geteilt werden.

PersNr	Name	Vorname	AbtNr	Abteilung	ProjektNr	Beschreibung	Zeit
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion	83
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse	29
3	Wild	Theodor	1	Personal	1	Kundenumfrage	140
3	Wild	Theodor	1	Personal	2	Verkaufspromotion	92
3	Wild	Theodor	1	Personal	3	Konkurrenzanalyse	110
4	Richter	Hans-Otto	3	Verkauf	2	Verkaufspromotion	67
5	Wieland	Brunhilde	2	Einkauf	1	Kundenumfrage	160

Zweite Normalform (2. NF)

Die **zweite Normalform** verlangt über die erste Normalform hinaus, dass die Attribute einer Relation vom gesamten **Primärschlüssel** (und nicht nur von einem Teil des Schlüssels) **abhängig** sind.

D.h. für unser Beispiel: Die Tabelle muss in mehrere Tabellen zerlegt werden.



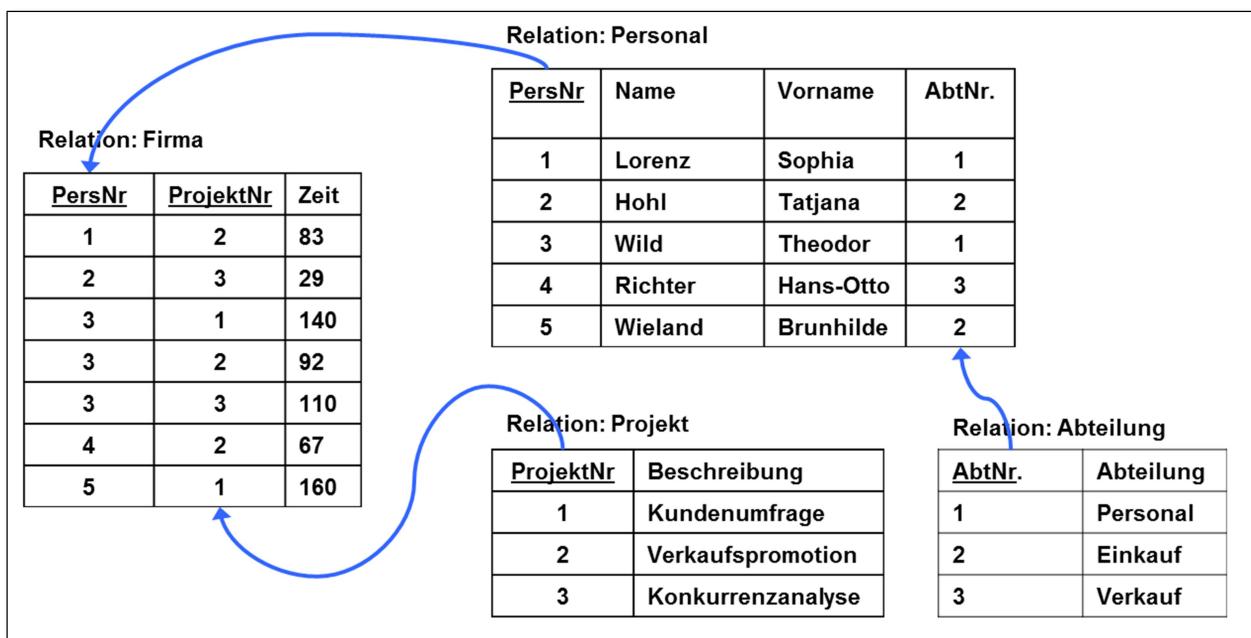
Die zweite Normalform ist nur relevant, wenn sich der Primärschlüssel aus mehreren Attributen zusammensetzt.

Dritte Normalform (3. NF)

Die **dritte Normalform** besagt – wiederum als Erweiterung der Forderungen der zweiten Normalform, dass alle Attribute, die nicht zum Primärschlüssel gehören, nicht untereinander abhängig (**nicht transitiv abhängig**) sein dürfen.

Ziel: Beseitigung von Abhängigkeiten zwischen Nichtschlüsselattributen

D.h. für unser Beispiel: In der 2. NF gibt es transitive Abhängigkeiten. Die Abteilung in der Tabelle „Personal“ ist auch vom Nichtschlüsselattribut AbtNr. abhängig. Somit ist eine weitere Zerlegung notwendig.



Die 3. NF ist nun erfüllt.

Aufgaben „Normalisierung“

- Stellen Sie fest welche Normalform vorliegt!
- Welcher Schritt ist zur Überführung in die nächste NF notwendig?

Aufgabe 12:

<u>ID</u>	<u>Name</u>	<u>Wohnort</u>
1	Müller, Peter	Ringelweg 12, 12345 Testhausen
...		

Aufgabe 13:

<u>MID</u>	<u>SportID</u>	<u>Sport</u>
1	1	Fußball
2	1	Fußball
1	2	Handball
3	3	Schwimmen
...		

Aufgabe 14:

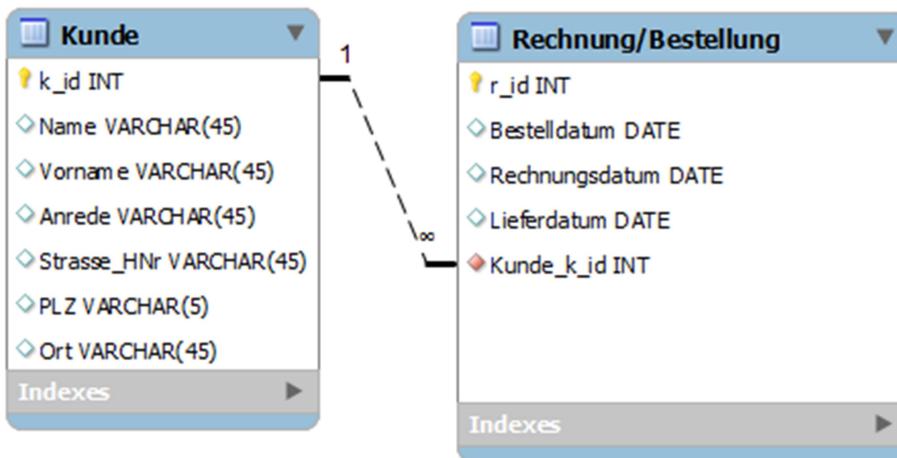
<u>ID</u>	<u>Name</u>	<u>Geburtsdatum</u>	<u>Alter</u>
1	Peter	1990-10-23	27
2	Uschi	1999-08-21	19
...			

Referentielle Integrität

Die **referentielle Integrität** wird bei relationalen Datenbanken dazu verwendet, um die Konsistenz und die Integrität der Daten sicherzustellen.

Bei der **referentiellen Integrität** (auch Beziehungsintegrität) können Datensätze die einen Fremdschlüssel aufweisen **nur dann gespeichert werden**, wenn der Wert des **Fremdschlüssels einmalig in der referenzierten Tabelle existiert**. Im Falle, dass ein referenzierter Wert nicht vorhanden ist, kann der Datensatz nicht gespeichert werden.

Beispiel:



Im Fall der referentiellen Integrität können in der Tabelle Rechnung/Bestellung nur Kundennummern (Kunde_k_id) gespeichert werden, wenn diese in der Tabelle Kunde als Kundennummer (k_id) auch tatsächlich vorhanden sind.

Erweiterungen:

- Änderungsweitergabe

Wenn der Wert des Primärschlüssels (k_id in der Tabelle Kunde) geändert wird, ändert sich automatisch der Wert des Referenzschlüssels (Kunde_k_id in der Bestellung).

- Löschweitergabe

Wird der Datensatz mit dem Primärschlüssel (z.B. ein Kunde) gelöscht, werden automatisch alle Datensätze mit diesem Referenzschlüssel gelöscht (z.B. alle Bestellungen dieses Kunden)

Diese Funktionen können je nach DBMS optional gesetzt werden. Die referentielle Integrität ist dafür die Grundlage.

Aufgaben „relationales Datenbankmodell“ (Teil 2)

Aufgabe 15: James Bond



Die Filmfans wissen, dass es eine ganze Reihe von James-Bond-Filmen gibt. Der erste Film "James Bond 007 jagt Dr. No" entstand bereits 1963. Es folgten weitere 23 Kassenschlager.

Exemplarisch liegen uns 6 Filme vor:

Film	Jahr	Haupt-darsteller	Gegenspieler	Automobil	Spezialeffekte
Goldfinger	1964	Sean Connery	"Auric Goldfinger" (Gerd Fröbe) "Oddjob" (Harold Sakata)	Aston Martin DB5 ("Silver Bird")	Schild als Kugelfang Nebelwerfer Drehbares Nummernschild ("BMT-216-A") Maschinengewehr Schleudersitz
Feuerball	1965	Sean Connery	"Emilio Largo" (Adolfo Celi)	Aston Martin DB5	Maschinengewehr (30er Browning) Navigation (Radar) Nummernschild
Golden Eye	1995	Pierce Brosnan	"Alec Trevelyan" (Sean Bean) "General Ouromow" (Gottfried John")	BMW Z3	Bremsfallschirm Raketen (2 Stinger) Navigation (Satellit)
...
Ein Quantum Trost	2008	Craig, Daniel	Dominic Greene (Mathieu Amalric)	Aston Martin DBS V12	Handschuhfach mit integrierter Walther P99 Antidot-Ampullarium Automatischer Externer Defibrillator (AED)
Skyfall	2012	Craig, Daniel	Raoul Silva/Tiago Rodriguez (Javier Bardem)	Aston Martin DB5	Maschinengewehr Schleudersitz
Spectre	2015	Craig, Daniel	Ernst Stavro Blofeld/Franz Oberhauser (Christoph Waltz)	Aston Martin DB10	Flammenwerfer doppelläufige Pistole Schleudersitz

Erstellen Sie zunächst ein ER-Modell (Skizze), welches die vorliegenden Informationen abbildet. Auch wenn im Laufe eines Films das Auto ausgetauscht werden muss, gibt es offiziell nur ein Auto pro Film.

- a) Ohne Spezialeffekte.
- b) Mit Spezialeffekten. Zur Vereinfachung: Die Effekte sind direkt dem Auto zugeordnet.

Wandeln Sie dann das ER-Modell in ein relationales DB-Modell um! Kennzeichnen Sie alle Primär- und Fremdschlüssel und prüfen Sie, ob sich die Datenbank in 3. Normalform befindet.

Aufgabe 16: Baufirma

Die Firma BAURENT GmbH vermietet Baumaschinen an Geschäftskunden. Für den Kerngeschäftsprozess "Baumaschinenvermietung" soll die Struktur einer relationalen Datenbank entworfen werden.

Folgende Informationen liegen vor:

- Ein Kunde wird erfasst, wenn er erstmals eine oder mehrere Baumaschinen mietet.
- Für die ordnungsgemäße Geschäftsabwicklung und Kundeninformation sind folgende Kundendaten notwendig: Kundennummer, Firma, Ansprechpartner und Adresse mit Telefon- und Faxnummer
- Über die Baumaschinen sollen folgende Daten gespeichert werden: Maschinennummer, Bezeichnung (z. B. "Raupenbagger"), Hersteller (z. B. "Wacker"), Typbezeichnung (z. B. "RL 500"), Preis pro Tag.
- Es gibt Kunden, die schon mehrfach verschiedene Baumaschinen angemietet haben. Es kommt vor, dass eine Maschine öfters von ein und demselben Kunden angemietet wird. Die kürzeste Mietdauer beträgt einen Tag.
- Ein Mietvertrag kann mehrere Baumaschinen beinhalten, die für einen gemeinsamen Zeitraum gemietet werden.
- Um Doppelvermietungen zu vermeiden, zur Rechnungsstellung und Abwicklung von Schadensfällen ist es notwendig, dass für jeden Mietvertrag die Mietvertragsnummer und das Abschlussdatum, und für jedes Mietobjekt Mietbeginn, Mietende, Kautionsbetrag und Abschluss einer Maschinenbruchversicherung aufgenommen werden.

Erstellen Sie zunächst ein ER-Modell (Skizze), welches die vorliegenden Informationen abbildet.

Wandeln Sie dann das ER-Modell in ein relationales DB-Modell um!

Kennzeichnen Sie alle Primär- und Fremdschlüssel und prüfen Sie, ob sich die Datenbank in 3. Normalform befindet.

Aufgabe 17: Eisenbahn-Gesellschaft

Eine Eisenbahngesellschaft möchte die Abwicklung ihres Zugverkehrs mit Hilfe einer relationalen Datenbank verwalten.

Informationsstruktur:

- Jeder Zug hat einen Ausgangsort, kann aber mehrere Zielorte besitzen. Von jedem Ausgangsort können mehrere Züge abfahren, an jedem Zielort können mehrere Züge ankommen.
- Es gibt verschiedene Zugkategorien; ein Zug gehört zu genau einer Kategorie (z. B.: Güterzug, ICE, EC, Regionalexpress, etc.).
- Ein Zug kann von mehreren Lokomotiven gezogen werden, eine Lok kann mehrere Züge ziehen.
- Das Personal wird den Zügen zugeordnet.
- Einige Mitglieder des Personals sind Lokführer mit besonderen Eigenschaften, diese sollen in einer besonderen Liste verwaltet werden. Für jeden Lokführer existiert nur ein Eintrag in dieser Liste. Jeder Eintrag kann genau einem Lokführer zugeordnet werden.
- Die Waggons können verschiedenen Zügen zugeordnet werden; ein Zug besteht aus mehreren Waggons.
- Die Waggons können einem Wagentyp zugeordnet werden. Diese Zuordnung ist eindeutig für jeden Waggon.

Erstellen Sie zunächst ein ER-Modell (Skizze), welches die vorliegenden Informationen abbildet.

Wandeln Sie dann das ER-Modell in ein relationales DB-Modell um!

Kennzeichnen Sie alle Primär- und Fremdschlüssel und prüfen Sie, ob sich die Datenbank in 3. Normalform befindet.

Aufgabe 18: Notenbuch

Ein Wirtschaftslehrer macht in seinem Notenbuch Aufzeichnungen über die schriftlichen und mündlichen Leistungen seiner Schüler. Dabei stellt er fest, dass sein "System" zwar alle wichtigen Informationen enthält, aber doch sehr unübersichtlich ist und manche Auswertungen sehr umständlich - manuell! - herzustellen sind.

Nun übergibt er Ihnen folgenden Auszug aus seinem Notenbuch mit dem Auftrag, ein übersichtliches und systematisches Notenblatt zu entwerfen.

Bevor man daran denken kann, eine benutzerfreundliche Oberfläche anzubieten, machen Sie sich an die Arbeit und schlagen vor, zur effizienten und ausbaufähigen Verwaltung seiner Daten einen relationalen Datenbankentwurf zu liefern

Noten Jahr 20.../..									
Name	Vorname	Klasse	Ausbil-dung	BW			RC		
				schriftlich	mündlich	Gesamt	schriftlich	mündlich	Gesamt
Alb	Fred	IK33D	IK	2	1 (17.09.)		1, 2	1 (22.09.) 3 (13.01.)	
Eder	Hans	IK33D	IK	4	4 (17.09.)		2, 2	2 (22.09.) 1 (13.01.)	
Elger	Rainer	IK33D	IK	1	1 (17.09.)		1, 1	1 (22.09.)	
Glotz	Gabi	IK33D	IK	1	1 (17.09.)		2, 1	3 (22.09.) 1 (13.01.)	
...						
Forst	Georg	SK33K	SK	4	2 (21.09.) 3 (30.11.)				
Klock	Stefan	SK33K	SK	1	1 (21.09.) 2 (30.11.)				
...						
Zauner	Regina	SK33K	SK	3	3 (21.09.) 1 (30.11.)				

Erstellen Sie zunächst ein ER-Modell (Skizze), welches die vorliegenden Informationen abbildet.

Wandeln Sie dann das ER-Modell in ein relationales DB-Modell um!

Kennzeichnen Sie alle Primär- und Fremdschlüssel und prüfen Sie, ob sich die Datenbank in 3. Normalform befindet.

Aufgabe 19: Prüfungsamt

Folgende Beschreibung über die (stark reduzierte) Informationsstruktur eines Prüfungsamts der Studierenden an einer Fachhochschule liegt Ihnen vor:

- Von den Studenten interessieren Name, Vorname, Studiengang und Geburtsdatum.
- Im Verlaufe ihres Grundstudiums müssen die Studenten an vier Pflichtklausuren teilnehmen: (Einführung in die OOP, Java – Grundlagen der Programmiersprache, Java für Fortgeschrittene, Datenbanktechnologien).
- Jedes Klausurenthema wird von (genau!) einem Hochschullehrer (Dozent) betreut. Allerdings werden mehrere Termine für ein Thema angeboten, aus denen der Student einen auswählt.
- Am Ende des Grundstudiums schreiben die Studenten eine Hausarbeit. Dabei werden sie von (nur!) einem Dozenten betreut.
- Schließlich will das Prüfungsamt noch wissen, aus welchen Orten die Studenten und die Dozenten kommen.

Erstellen Sie zunächst ein ER-Modell (Skizze), welches die vorliegenden Informationen abbildet.

Wandeln Sie dann das ER-Modell in ein relationales DB-Modell um!

Kennzeichnen Sie alle Primär- und Fremdschlüssel und prüfen Sie, ob sich die Datenbank in 3. Normalform befindet.

Physische Implementierung (Phase 4)

In der letzten Phase der Datenbankentwicklung wird die entworfene Struktur implementiert. D.h. die Datenbank wird entsprechend des relationalen Modells angelegt.

Dies erfolgt mit Hilfe einer Datensprache, wie z.B. SQL.

Quellen:

- Lexikon „Datenbanken verstehen“:
<http://www.datenbanken-verstehen.de/lexikon/datenbanksystem/>
- Wiki FH Köln:
http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/ANSI-3-Ebenenmodell
- Küspert: Datenbanksysteme 1:
<http://uni-skripte.lug-jena.de/kuespert-dbs1.pdf>
- Gymnasium Fulda - Website:
<http://www.info-wsf.de/index.php/Datenbankentwurf>
- Hans Ott: Objektorientierte Datenbanksysteme
<http://script.hans-ott.de/45oodbs.htm>

