

Datenbanken: Entity-Relationship-Model

Horst Gierhardt
Städtisches Gymnasium Bad Laasphe
horst@gierhardt.de

18.09.2014

Inhaltsverzeichnis

1	Entity-Relationship-Diagramm	2
1.1	Beispiel	2
1.2	Begriffserläuterungen	3
1.3	Alternative Tabellendarstellung	4
1.4	Übungsbeispiel 1:	4
1.5	Übungsbeispiel 2:	4
1.6	Übungsbeispiel 3:	5
1.7	Übung 4: Zur Kardinalität von Beziehungen	5
1.8	Übungsbeispiel 5:	5
2	Transformation vom ERD in Tabellen	6
2.1	Transformationsregeln	6
2.2	Transformationsbeispiel	8
2.3	Transformationsübung	9
3	Relationenalgebra	10
3.1	Das relationale Datenbankmodell	10
3.2	Durchschnitt	12
3.3	Vereinigung	12
3.4	Differenz	12
3.5	Produkt bzw. kartesisches Produkt	13
3.6	Selektion	13
3.7	Projektion	13
3.8	Join	14
	3.8.1 Left-Join	14
	3.8.2 Right-Join	14
3.9	Umbenennung	15
3.10	Aufgaben:	16
4	Quellen	18

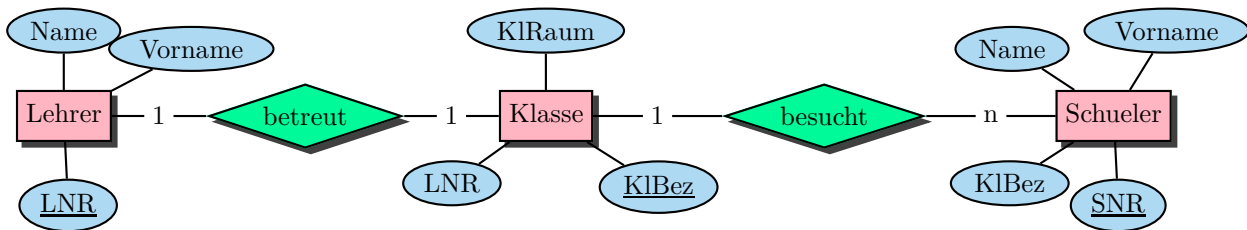
1 Entity-Relationship-Diagramm

1.1 Beispiel

Datenbanken bestehen meistens aus mehreren bzw. vielen Tabellen. Um den Überblick zu behalten, werden die Zusammenhänge meist in sogenannten *Entity-Relationship-Diagrammen*, kurz ERD (deutsch: Gegenstand-Beziehungs-Diagramm) dargestellt. Diese Diagramme sind wichtiger Bestandteil des Entity-Relationship-Modells (engl.: entity-relationship-model), kurz ERM. 1976 von PETER CHEN vorgeschlagen, ist es ein weithin anerkanntes Werkzeug zum Datenbankentwurf. ER-Modelle dienen dazu,

- Zusammenhänge in der realen Welt auf einer abstrakten Ebene zu modellieren.
- die inhaltlichen Bezüge der Tabellen einer Datenbank untereinander darzustellen.
- die Konzeption einer Datenbank zu erleichtern.
- die Kommunikation zwischen Auftraggeber und Entwickler einer Datenbank zu erleichtern.

Dies soll hier am Beispiel einer kleinen Datenbank mit Schülern, Lehrern und Klassen erläutert werden. Das ERD



gehört zu den folgenden Tabellen einer Datenbank:

Lehrer

<u>LNR</u>	Vorname	Name
1	Ludwig	Lempel
2	Beate	Bembel
3	Hans	Hempel

Klasse

<u>KlBez</u>	KlRaum	LNR
1a	R11	2
1b	R12	3
2a	R21	1

Schueler

<u>SNR</u>	KlBez	Name	Vorname
1	1b	Keegan	Kevin
2	2a	Messi	Michael
3	1a	Hempel	Luise

1.2 Begriffserläuterungen

- Eine **Entität** ist ein bestimmtes Objekt der realen Welt bzw. der Vorstellung (z.B. eine Person, ein Gegenstand, ein Ereignis), eine Entität ist sozusagen *Irgendetwas*. In einer Tabelle ist eine Zeile eine Entität. Eine Zeile bezeichnet man auch als **Tupel**.
- Entitäten mit gleichen Eigenschaften werden zu **Entitätstypen** (manchmal auch als **Entitätsmengen** bezeichnet) zusammengefasst. Im oben dargestellten Beispiel ist Ludwig Lempel eine Entität, die Menge aller Lehrer aber ein Entitätstyp. Entitätstypen werden durch **Rechtecke** veranschaulicht. Im Sinne der Objektorientierung: Entitäten sind Objekte und Entitätstypen die entsprechenden Klassen. In einer Tabelle ist der Tabellename der Entitätstyp.
- Die Eigenschaften aller Entitäten und Beziehungen eines Entitätstyps bzw. eines Beziehungstyps werden mit Hilfe von **Attributen** erfasst. Attribute werden durch **Ovale** dargestellt. In einer Tabelle sind die Attribute die einzelnen Spaltenüberschriften. Auch Rauten (Beziehungen) können Ovale (Attribute) besitzen.
- **Beziehungen** zwischen den Entitätstypen werden durch **Rauten** dargestellt.
- Ein Beziehungstyp wird durch die **Kardinalität** genauer bezeichnet. Es gibt die Kardinalitäten (1:1), (1:n), und (n:m). In einer Klasse sind viele (n) Schüler. Ein Lehrer betreut eine bestimmte Klasse als Klassenlehrer. Man spricht hier von 1-zu-n-Beziehungen bzw. 1-zu-1-Beziehungen.

Andere Sprechweise: Eine Beziehung besteht aus zwei **Assoziationen** in den beiden Richtungen.

- Der **Primärschlüssel** (primary key) dient zur eindeutigen Identifikation einer Entität. Als Primärschlüssel kann
 - ein natürlicher Schlüssel (z.B. die Fahrgestellnummer eines Autos) oder
 - ein künstlicher Schlüssel (z.B. die Schülernummer in einer Schuldatenbank oder auch
 - eine Kombination aus mehreren Attributen (z.B. Name, Vorname, Geburtstag), wenn diese eindeutig eine Entität beschreiben, sein.

Unterstrichene Attribute sind **Primärschlüssel**.

- Ein **Fremdschlüssel** ist ein Attribut einer Tabelle, das in einer anderen Tabelle ein Primärschlüssel ist. Auf diese Art werden Beziehungen zwischen Entitätstypen hergestellt. Die oben angegebenen Entitätstypen können kurz auch so dargestellt werden:
 - Lehrer(LNR, Vorname, Name)
 - Klasse(KlBez, KlRaum, LNR)
 - Schueler(SNR, KlBez, Name, Vorname)

LNR, KlBez und SNR sind Primärschlüssel. Die Klassenbezeichnung KlBez in der Tabelle *Schueler* ist ein Fremdschlüssel.

1.3 Alternative Tabellendarstellung

Statt eines ERDs kann auch eine Tabelle gute Dienste leisten:

Entitätstyp	Beziehung	Attribut
Lehrer		<u>LNR</u> Name Vorname
	betreut	
Klasse		<u>KlBez</u> LNr KlRaum
	besucht	
Schueler		<u>SNR</u> KlBez Name Vorname

1.4 Übungsbeispiel 1:

Sie sind damit beauftragt worden, die Verwaltung der Universität zu organisieren – und zwar mit Hilfe einer Datenbank. Folgendes wissen Sie über den Universitätsbetrieb: Ein Professor beschäftigt mehrere Assistenten, die für ihn arbeiten. Dabei kann jeder Assistent jedoch nur ein einziges Fachgebiet haben. Neben dem Namen verfügt jeder Mitarbeiter einer Universität über eine Personalnummer, auch die Professoren. Die Professoren sind jedoch immer einem festen Raum zugeteilt und besitzen einen festen Rang (zb.: C4). Jeder Professor liest mehrere Vorlesungen im Semester. Eine Vorlesung geht über eine gewisse Anzahl von Semesterwochenstunden (SWS) und kann entweder über ihre Nummer oder über ihren Namen von anderen Veranstaltungen unterschieden werden. Ein Student, der neben seiner Matrikelnummer auch seinen Namen und seine Semesteranzahl angeben muss, kann mehrere Vorlesungen hören.

Stellen Sie diesen Sachverhalt in einem ER-Diagramm dar!

1.5 Übungsbeispiel 2:

In einem Restaurant kehren verschiedene Gäste ein. Ein Gast kann entweder alleine an einem Tisch sitzen oder aber auch mit mehreren Gästen einen Tisch belegen. Auf jeden Fall gibt jeder Tisch seine Bestellungen gesammelt ab. Dabei besteht jede Bestellung aus den vollständigen Angaben: Vorspeise, Hauptgericht, Dessert und Getränk. Selbstverständlich könnte ein Gast auch nur ein Hauptgericht bestellen. Um eine Bestellung aufzugeben, werden eine Speisekarte und eine Getränkekarte zu Rate gezogen, aus denen jeweils der Name des Gerichts/Getränks, die Beschreibung und der Preis ersichtlich werden. Rechnungen werden immer pro Tisch erstellt. Damit sich die Kellner nicht in die Quere kommen, ist jedem Tisch ein fester Kellner zugeordnet, der alle Bestellungen eines Tisches entgegennimmt.

1.6 Übungsbeispiel 3:

Die Stadtbücherei wird von mehreren Bibliothekarinnen geführt, die alle eine eindeutige Personalnummer besitzen und in festen Schichten (Schichten: 1,2,3) arbeiten. Von den Kunden der Bücherei müssten – natürlich neben den Namen – auch die Adresse und das Geburtsdatum bekannt sein. Nur so ist sichergestellt, dass man im Schadensfall auch den richtigen Ansprechpartner findet. Die Bücherei selbst besteht aus großen Regalen, die durch Kürzel wie If oder La gekennzeichnet sind. Außerdem sollte man aus logistischen Gründen wissen, wie viele Bücher in einem Regal stehen und wie viel Raum noch für andere Bücher zur Verfügung steht. Bei der Ausleihe eines Buches wird neben der Signatur (If3e oder La5j) und dem Titel auch festgehalten, von welcher Bibliothekarin das Buch verliehen und von welchem Kunden das Buch ausgeliehen wurde. Aufgabe der Bibliothekarinnen sollte es außerdem sein, die Bücher richtig einzusortieren, damit sie auch schnell wieder gefunden werden können. Dafür wird in jeder Schicht eine der Bibliothekarinnen für diese Aufgabe abgestellt.

1.7 Übung 4: Zur Kardinalität von Beziehungen

Die folgende Tabelle zeigt jeweils zwei Entitätstypen und den zugehörigen Beziehungstyp. Geben Sie für jede Beziehung die Kardinalität an.

Schüler	hat	Tutor.
Schüler	bekommt heute	Zeugnis.
Schüler	darf arbeiten an	Computer.
Schüler	hat ausgeliehen	Buch.
Schüler	besucht	Kurs.
Schüler	ist befreundet mit	Schüler.

1.8 Übungsbeispiel 5:

Die folgende Musterrechnung dokumentiert eine Miniwelt *Rechnungschreiben* in einer Firma.

Herrn Horst Staniczek Birnbaum 3 65510 Hünstetten					
Rechnungsnummer: R123		Rechnungsdatum: 14.10.2013			
Kundennummer: K002		Rechnungsbetrag: 1397,00			
Position	Artikelnummer	Bezeichnung	Anzahl	Einzelpreis	Gesamtpreis
1	A3257	Laptop	2	499,00	998,00
2	A4210	Laserdrucker	1	399,00	399,00

1. Stellen Sie fest, welche Entitätstypen und Beziehungen sich daraus ableiten lassen.
2. Skizzieren Sie ein Modell der Miniwelt.
3. Formulieren Sie Geschäftsregeln für die Miniwelt.

2 Transformation vom ERD in Tabellen

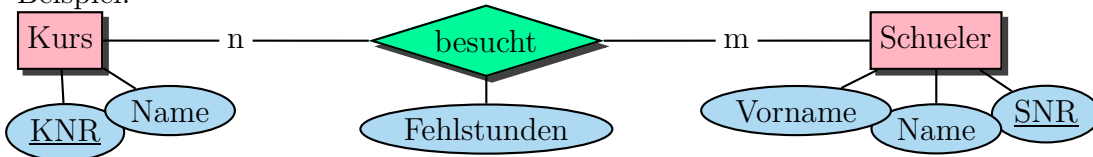
Nach der Modellierung einer Datenbank mit Hilfe eines ER-Diagramms erfolgt die Erstellung der entsprechenden Tabellen. Dies ist ein technischer Prozess, der nach festen Regeln abläuft und der heutzutage auch schon von Softwarewerkzeugen übernommen werden kann.

2.1 Transformationsregeln

Regel 1: Jeder Entitätstyp wird als Tabelle dargestellt. Jede Tabelle benötigt einen Primärschlüssel.

Regel 2: Jede n:m-Beziehung wird durch eine eigene Tabelle dargestellt.

Beispiel:



Kurs

<u>KNR</u>	Name
1	if2
2	M3
3	ph1

besucht

<u>KNR</u>	<u>SNR</u>	Fehlstunden
2	2	0
1	3	3
3	1	1

Schueler

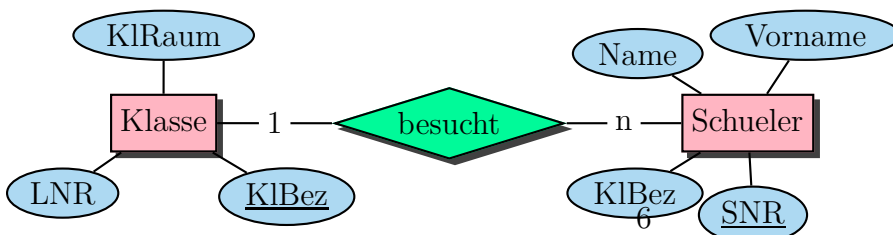
<u>SNR</u>	Name	Vorname
1	Keegan	Kevin
2	Messi	Michael
3	Hempel	Luise

Am Beispiel erkennt man zudem, dass die neue Tabelle die Primärschlüssel der verknüpften Entitätstypen und die eigenen Attribute enthält.

Regel 3: Jede 1:n- und 1:1-Beziehung **mit** eigenen Attributen wird wie bei Regel 2 durch eine eigene Tabelle dargestellt.

Regel 4a: Jede 1:n-Beziehung **ohne** eigene Attribute wird so dargestellt, dass der Primärschlüssel des 1-Entitätstyps Fremdschlüssel des n-Entitätstyps wird.

Beispiel:



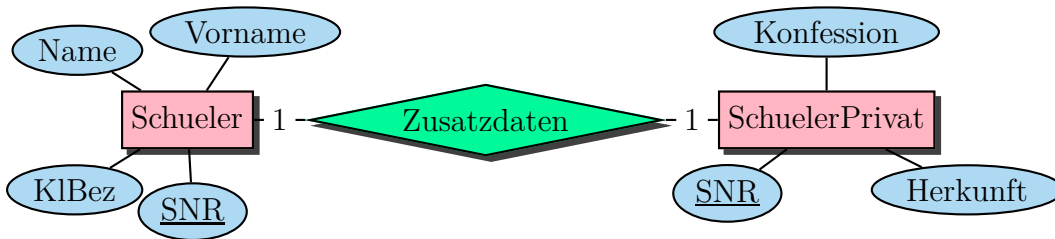
Klasse

<u>KlBez</u>	KlRaum	LNR
1a	R11	2
1b	R12	3
2a	R21	1

Schueler

<u>SNR</u>	<u>KlBez</u>	Name	Vorname
1	1b	Keegan	Kevin
2	2a	Messi	Michael
3	1a	Hempel	Luise

Regel 4b: Jede 1:1-Beziehung **ohne** eigene Attribute wird so dargestellt, dass der Primärschlüssel des ersten Entitätstyps beim zweiten Entitätstyp Primär- und Fremdschlüssel zugleich wird.



In Kurzschreibweise:

Schueler(SNR, Name, Vorname, KlBez)

SchuelerPrivat(SNR, Konfession, Herkunft)

Regel 4c: Sind Regel 4a oder 4b nicht anwendbar, dann wird für die Beziehung eine gesonderte Tabelle angelegt.

Beispiel:



Scheinbar liegt hier eine 1:1-Beziehung nach abendländischem Recht vor. Bei einer **zwingenden** Beziehung muss Heiratspflicht bestehen und es müssten gleichgroße Entitätsmengen vorhanden sein. Wenn ein Partner stirbt, tritt eine Löschanomalie in der Datenbank auf. Die gleiche Problematik liegt bei 1:n-Beziehungen vor („muslimische Ehe“).

Falsche Lösung:

Name	Ehefrau
Adam	Eva
Udo	Sabine
Detlef	?
Ike	Tina

Name	Ehemann
Berta	?
Eva	Adam
Tina	Ike

Da nicht jeder verheiratet sein muss, also eine freie Beziehung besteht, gibt es in der Tabelle nach dem obigen Modell verbotene leere Felder. Die Lösung ergibt sich durch Einführen der Beziehungstabelle *Ehe* mit den Fremdschlüsseln *Maenner.Name* und *Frauen.Name*

Richtige Lösung:

Name
Adam
Udo
Detlef
Ike

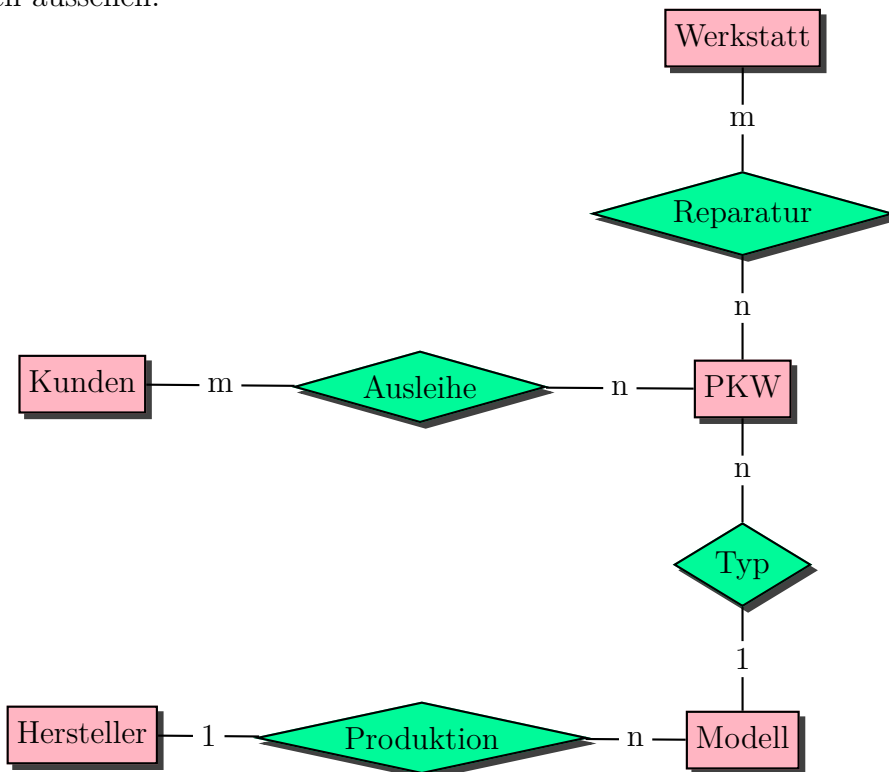
Ehemann	Ehefrau
Adam	Eva
Ike	Tina

Name
Berta
Eva
Tina

2.2 Transformationsbeispiel

Die Autovermietung *FlottFlitz* vermietet ihre PKW an ihre Kunden. Die PKW müssen von Zeit zu Zeit auch gewartet oder repariert werden. Dies geschieht in verschiedenen Werkstätten. Von allen Wagen wird natürlich das Modell und der Hersteller mit entsprechenden Daten gespeichert.

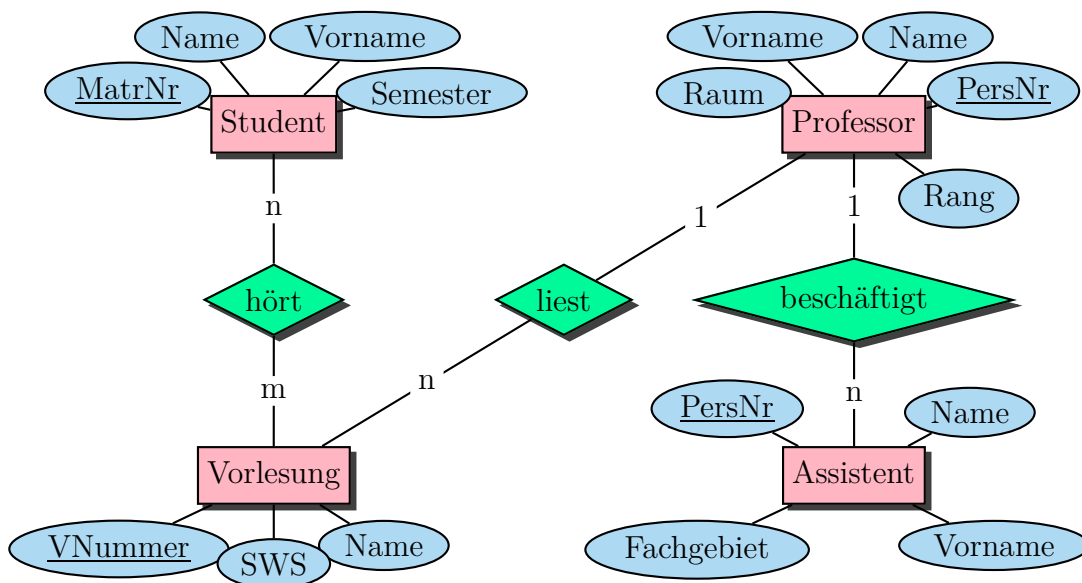
Ein vereinfachtes ER-Diagramm (Attribute wurden weggelassen) könnte dann folgendermaßen aussehen:



- Nach **Regel 1** werden alle Entitätstypen zu einer Tabelle:
 Kunden(Nummer, Name, Vorname, Strasse, PLZ, Ort, Geburtsdatum, Telephon, DatumFührerscheinerwerb)
 PKW(Kennzeichen, Modell, Erstzulassung, kmStand)
 Werkstatt(Name, Strasse, PLZ, Ort, Telephon, Kontaktperson)
 Modell(Name, Hersteller, Leistung, Laenge, Breite, Hubraum)
 Hersteller(Name, Strasse, PLZ, Ort, Telephon, Kontaktperson)
- Nach **Regel 2** wird auch jede n:m-Beziehung zu einer eigenen Tabelle:
 Ausleihe(fortlaufendeNummer, Kundennummer, Kennzeichen, Ausleihdatum, Rueckgabedatum)
 Reparatur(Vorgangsnummer, Werkstatt, Kennzeichen, Datum, Dauer, Art)
- Nach **Regel 4a** wird der Primärschlüssel des Herstellers als Fremdschlüsselattribut beim Modell aufgenommen:
 Modell(Name, Leistung, Laenge, Breite, Hubraum, **Herstellername**)
 Beim PKW verfährt man ebenso:
 PKW(Kennzeichen, Erstzulassung, kmStand, **Modellname**)

2.3 Transformationsübung

Das ER-Diagramm von Übungsbeispiel 1 könnte etwa so aussehen:



Transformiere das ER-Diagramm mit den entsprechenden Regeln in geeignete Tabellen.

3 Relationenalgebra

3.1 Das relationale Datenbankmodell

Die uns aus der Praxis bekannten SQL-Anfragen, die wir eher intuitiv entwickelten, können aus theoretischer Sicht etwas systematischer dargestellt werden. Die Erfindung der relationalen Algebra durch EDGAR F. CODD in den 70er Jahren revolutionierte die Datenbankwelt. Mit den Operatoren dieser Algebra können alle Fragen an eine Datenbank in einer symbolischen Notation festgehalten werden. In der Praxis sind dann allerdings wieder die SQL-Anweisungen wichtiger.

Das *relationale Datenbankmodell* basiert auf dem mathematischen Begriff der *Relation*.

Eine Relation R ist eine Teilmenge des kartesischen Produktes einer Liste von Wertebereichen $W_1, W_2, W_3, \dots, W_n$:

$$R \subseteq W_1 \times W_2 \times W_3 \times \dots \times W_n$$

Elemente einer Relation sind die n -Tupel der Art $(v_1, v_2, v_3, \dots, v_n)$ mit $v_i \in W_i$. Die Größe n bezeichnet man als Grad der Relation R .

Beispiel: Eine einfache Schülerliste einer AG der Jahrgangsstufe 5 hat die Attribute *Nummer*, *Nachname*, *Vorname* und *Klasse*. Also gibt es die Wertemengen

- $W_1 = \mathbb{N}$
- $W_2 = \{Abelmann, Achenbach, \dots, Zwackelmann\}$
- $W_3 = \{Anna, Berta, Cäsar, \dots, Zenzi\}$
- $W_4 = \{5a, 5b, 5c, 5d\}$

Dann sind z. B.

- $S_1 = (5, Achenbach, Michael, 5a)$
- $S_2 = (3, Abelmann, Jens, 5c)$
- $S_3 = (2, Zwackelmann, Zensi, 5d)$

drei 4-Tupel, die eine drei-elementige Relation $R = \{S_1, S_2, S_3\}$ bilden. Üblicherweise schreibt man diese Relation mit dem Relationennamen *AG-Liste* als Tabelle auf:

AG-Liste

Nummer	Nachname	Vorname	Klasse
5	Achenbach	Michael	5a
3	Abelmann	Jens	5c
2	Zwackelmann	Zensi	5d

Den Zusammenhang von relationalem Modell und dem ERM zeigt die folgende Tabelle:

Relationales Modell	Beschreibung	ERM	im ER-Diagramm
Relationenname	Name der Tabelle	Name d. Entitätstyps	Name im Rechteck
Attribut	Spalte einer Tabelle	Attribut	Oval
Relationenschema	Menge von Attributen	Entitätstyp	Rechteck mit Ovalen
Tupel	Zeile einer Tabelle	Entität	-
Relation	Menge von Zeilen	Entitätsmenge	-

Die **Operationen auf Datenbanken** können dann wie folgt gegliedert werden:

- Mengentheoretische Operationen

Operator	Schreibweise	Bedeutung
Durchschnitt	$R \cap S$	Schnittmenge
Vereinigung	$R \cup S$	Vereinigungsmenge
Differenz	$R \setminus S$	Differenz von Mengen
Produkt	$R \times S$	Kartesisches Produkt zweier Mengen

- Relationsoperationen

Operator	Schreibweise	Bedeutung
Selektion	$\sigma_{\text{Formel}}(R)$	Auswahl von Tupeln gemäß der Formel, Streichung von Zeilen
Projektion	$\pi_{\text{Attribute}}(R)$	Auswahl von Attributen gemäß der Formel, Streichung von Spalten
Join	$R \bowtie_{\text{Ausdruck}} S$	Verknüpfung zweier Relationen zu einer neuen mit den Attributen beider Tabellen über gemeinsames Attribut

- Umbenennung

Zur Erläuterung der Operationen soll von folgendem Datenbankschema ausgegangen werden (↑ soll einen Fremdschlüssel kennzeichnen):

Lehrer(ID, Vorname, Nachname)

Lehrerin(ID, Vorname, Nachname)

Klassenleitungsteam(↑LehrerID, ↑LehrerinID)

Schulleitung(ID, Vorname, Nachname)

lehrer

<u>ID</u>	Vorname	Nachname
Me	Peter	Meier
Sz	Peter	Schulz
Bm	Hans	Baum

lehrerin

<u>ID</u>	Vorname	Nachname
Be	Petra	Blume
Sr	Clara	Sommer
Kr	Helga	Kremer

klassenleitungsteam

↑LehrerID	↑LehrerinID
Bm	Be
Me	Kr
Me	Sr

schulleitung

<u>ID</u>	Vorname	Nachname
Bm	Hans	Baum
Kr	Helga	Kremer

3.2 Durchschnitt

Aus zwei Tabellen mit gleichen Attributen werden alle gemeinsamen Tupel zu einer neuen Tabelle vereinigt.

Mengenschreibweise:	$\text{lehrer} \cap \text{schulleitung}$						
SQL-Schreibweise:	<code>SELECT * FROM lehrer WHERE lehrer.ID IN (SELECT ID FROM schulleitung);</code>						
Ergebnis:	Schnittmenge der Tabellen <i>lehrer</i> und <i>schulleitung</i> <table border="1"> <thead> <tr> <th><u>ID</u></th> <th>Vorname</th> <th>Nachname</th> </tr> </thead> <tbody> <tr> <td>Bm</td> <td>Hans</td> <td>Baum</td> </tr> </tbody> </table>	<u>ID</u>	Vorname	Nachname	Bm	Hans	Baum
<u>ID</u>	Vorname	Nachname					
Bm	Hans	Baum					

3.3 Vereinigung

Zwei Tabellen mit gleichen Attributen werden zu einer neuen Tabelle vereinigt. Doppelte Tupel werden entfernt:

Mengenschreibweise:	$\text{lehrer} \cup \text{lehrerin}$																					
SQL-Schreibweise:	<code>(SELECT * FROM lehrer) UNION (SELECT * FROM lehrerin);</code>																					
Ergebnis:	Vereinigungsmenge der Tabellen <i>lehrer</i> und <i>lehrerin</i> <table border="1"> <thead> <tr> <th><u>ID</u></th> <th>Vorname</th> <th>Nachname</th> </tr> </thead> <tbody> <tr> <td>Me</td> <td>Peter</td> <td>Meier</td> </tr> <tr> <td>Sz</td> <td>Peter</td> <td>Schulz</td> </tr> <tr> <td>Bm</td> <td>Hans</td> <td>Baum</td> </tr> <tr> <td>Be</td> <td>Petra</td> <td>Blume</td> </tr> <tr> <td>Sr</td> <td>Clara</td> <td>Sommer</td> </tr> <tr> <td>Kr</td> <td>Helga</td> <td>Kremer</td> </tr> </tbody> </table>	<u>ID</u>	Vorname	Nachname	Me	Peter	Meier	Sz	Peter	Schulz	Bm	Hans	Baum	Be	Petra	Blume	Sr	Clara	Sommer	Kr	Helga	Kremer
<u>ID</u>	Vorname	Nachname																				
Me	Peter	Meier																				
Sz	Peter	Schulz																				
Bm	Hans	Baum																				
Be	Petra	Blume																				
Sr	Clara	Sommer																				
Kr	Helga	Kremer																				

3.4 Differenz

Aus zwei Tabellen mit gleichen Attributen werden aus der ersten Tabelle die Tupel ausgewählt, die in der zweiten Tabelle nicht enthalten sind.

Mengenschreibweise:	$\text{lehrer} \setminus \text{schulleitung}$									
SQL-Schreibweise:	<code>SELECT * FROM lehrer WHERE lehrer.ID NOT IN (SELECT ID FROM schulleitung);</code>									
Ergebnis:	Differenzmenge von <i>lehrer</i> und <i>schulleitung</i> <table border="1"> <thead> <tr> <th><u>ID</u></th> <th>Vorname</th> <th>Nachname</th> </tr> </thead> <tbody> <tr> <td>Me</td> <td>Peter</td> <td>Meier</td> </tr> <tr> <td>Sz</td> <td>Peter</td> <td>Schulz</td> </tr> </tbody> </table>	<u>ID</u>	Vorname	Nachname	Me	Peter	Meier	Sz	Peter	Schulz
<u>ID</u>	Vorname	Nachname								
Me	Peter	Meier								
Sz	Peter	Schulz								

3.5 Produkt bzw. kartesisches Produkt

Es werden alle Zeilen einer Tabelle mit allen Zeilen einer zweiten Tabelle verknüpft, was im Normalfall eine ziemlich sinnlose Tabelle ergibt.

Mengenschreibweise:	$\text{lehrer} \times \text{lehrerin}$																																																												
SQL-Schreibweise:	<code>SELECT * FROM lehrer, lehrerin;</code>																																																												
Ergebnis:	<p>Kartesisches Produkt aus lehrer und lehrerin</p> <table border="1"> <thead> <tr> <th><u>ID</u></th> <th>Vorname</th> <th>Nachname</th> <th><u>ID</u></th> <th>Vorname</th> <th>Nachname</th> </tr> </thead> <tbody> <tr><td>Me</td><td>Peter</td><td>Meier</td><td>Be</td><td>Petra</td><td>Blume</td></tr> <tr><td>Sz</td><td>Peter</td><td>Schulz</td><td>Be</td><td>Petra</td><td>Blume</td></tr> <tr><td>Bm</td><td>Hans</td><td>Baum</td><td>Be</td><td>Petra</td><td>Blume</td></tr> <tr><td>Me</td><td>Peter</td><td>Meier</td><td>Sr</td><td>Clara</td><td>Sommer</td></tr> <tr><td>Sz</td><td>Peter</td><td>Schulz</td><td>Sr</td><td>Clara</td><td>Sommer</td></tr> <tr><td>Bm</td><td>Hans</td><td>Baum</td><td>Sr</td><td>Clara</td><td>Sommer</td></tr> <tr><td>Me</td><td>Peter</td><td>Meier</td><td>Kr</td><td>Helga</td><td>Kremer</td></tr> <tr><td>Sz</td><td>Peter</td><td>Schulz</td><td>Kr</td><td>Helga</td><td>Kremer</td></tr> <tr><td>Bm</td><td>Hans</td><td>Baum</td><td>Kr</td><td>Helga</td><td>Kremer</td></tr> </tbody> </table>	<u>ID</u>	Vorname	Nachname	<u>ID</u>	Vorname	Nachname	Me	Peter	Meier	Be	Petra	Blume	Sz	Peter	Schulz	Be	Petra	Blume	Bm	Hans	Baum	Be	Petra	Blume	Me	Peter	Meier	Sr	Clara	Sommer	Sz	Peter	Schulz	Sr	Clara	Sommer	Bm	Hans	Baum	Sr	Clara	Sommer	Me	Peter	Meier	Kr	Helga	Kremer	Sz	Peter	Schulz	Kr	Helga	Kremer	Bm	Hans	Baum	Kr	Helga	Kremer
<u>ID</u>	Vorname	Nachname	<u>ID</u>	Vorname	Nachname																																																								
Me	Peter	Meier	Be	Petra	Blume																																																								
Sz	Peter	Schulz	Be	Petra	Blume																																																								
Bm	Hans	Baum	Be	Petra	Blume																																																								
Me	Peter	Meier	Sr	Clara	Sommer																																																								
Sz	Peter	Schulz	Sr	Clara	Sommer																																																								
Bm	Hans	Baum	Sr	Clara	Sommer																																																								
Me	Peter	Meier	Kr	Helga	Kremer																																																								
Sz	Peter	Schulz	Kr	Helga	Kremer																																																								
Bm	Hans	Baum	Kr	Helga	Kremer																																																								

3.6 Selektion

Es werden die Zeilen ausgewählt, die eine bestimmte Bedingung erfüllen.

Mengenschreibweise:	$\sigma_{\text{Vorname} = \text{'Peter'}}(\text{lehrer})$									
SQL-Schreibweise:	<code>SELECT * FROM lehrer WHERE Vorname = 'Peter';</code>									
Ergebnis:	<p>Nur Lehrer mit dem Vornamen Peter werden gewählt.</p> <table border="1"> <thead> <tr> <th><u>ID</u></th> <th>Vorname</th> <th>Nachname</th> </tr> </thead> <tbody> <tr><td>Me</td><td>Peter</td><td>Meier</td></tr> <tr><td>Sz</td><td>Peter</td><td>Schulz</td></tr> </tbody> </table>	<u>ID</u>	Vorname	Nachname	Me	Peter	Meier	Sz	Peter	Schulz
<u>ID</u>	Vorname	Nachname								
Me	Peter	Meier								
Sz	Peter	Schulz								

3.7 Projektion

Es werden nur bestimmte Spalten ausgewählt. Doppelte Zeilen werden entfernt.

Mengenschreibweise:	$\pi_{\text{Vorname}}(\text{lehrer})$			
SQL-Schreibweise:	<code>SELECT DISTINCT Vorname FROM lehrer;</code>			
Ergebnis:	<p>Nur die Vornamen (ohne Doppelungen) werden geliefert.</p> <table border="1"> <tbody> <tr><td>Vorname</td></tr> <tr><td>Peter</td></tr> <tr><td>Hans</td></tr> </tbody> </table>	Vorname	Peter	Hans
Vorname				
Peter				
Hans				

3.8 Join

Ein Join ist die Bildung eines kartesischen Produktes gefolgt von einer Selektion.

Mengenschreibweise:	$\text{lehrer} \bowtie_{\text{ID}} \text{klassenleitungsteam}$																				
SQL-Schreibweise:	<pre>SELECT * FROM lehrer JOIN klassenleitungsteam ON lehrer.ID = klassenleitungsteam.LehrerID; Alternativ: SELECT * FROM lehrer, klassenleitungsteam WHERE lehrer.ID = klassenleitungsteam.LehrerID;</pre>																				
Ergebnis:	<p>Lehrer, die auch in einem Klassenleitungsteam sind, werden geliefert:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Vorname</th> <th>Nachname</th> <th>LehrerID</th> <th>LehrerinID</th> </tr> </thead> <tbody> <tr> <td>Bm</td> <td>Hans</td> <td>Baum</td> <td>Bm</td> <td>Be</td> </tr> <tr> <td>Me</td> <td>Peter</td> <td>Meier</td> <td>Me</td> <td>Kr</td> </tr> <tr> <td>Me</td> <td>Peter</td> <td>Meier</td> <td>Me</td> <td>Sr</td> </tr> </tbody> </table>	ID	Vorname	Nachname	LehrerID	LehrerinID	Bm	Hans	Baum	Bm	Be	Me	Peter	Meier	Me	Kr	Me	Peter	Meier	Me	Sr
ID	Vorname	Nachname	LehrerID	LehrerinID																	
Bm	Hans	Baum	Bm	Be																	
Me	Peter	Meier	Me	Kr																	
Me	Peter	Meier	Me	Sr																	

3.8.1 Left-Join

Beim **Left-Join** werden auch Zeilen aus der ersten Tabelle aufgeführt, die keinen Partner in der zweiten Tabelle haben. Mit NULL wird aufgefüllt.

Mengenschreibweise:	$\text{lehrer} \bowtie_{\text{ID}} \text{klassenleitungsteam}$																									
SQL-Schreibweise:	<pre>SELECT * FROM lehrer LEFT JOIN klassenleitungsteam ON lehrer.ID = klassenleitungsteam.LehrerID;</pre>																									
Ergebnis:	<p>Alle Lehrer erscheinen, auch wenn sie keinen Partner haben:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Vorname</th> <th>Nachname</th> <th>LehrerID</th> <th>LehrerinID</th> </tr> </thead> <tbody> <tr> <td>Me</td> <td>Peter</td> <td>Meier</td> <td>Me</td> <td>Kr</td> </tr> <tr> <td>Me</td> <td>Peter</td> <td>Meier</td> <td>Me</td> <td>Sr</td> </tr> <tr> <td>Sz</td> <td>Peter</td> <td>Schulz</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>Bm</td> <td>Hans</td> <td>Baum</td> <td>Bm</td> <td>Be</td> </tr> </tbody> </table>	ID	Vorname	Nachname	LehrerID	LehrerinID	Me	Peter	Meier	Me	Kr	Me	Peter	Meier	Me	Sr	Sz	Peter	Schulz	NULL	NULL	Bm	Hans	Baum	Bm	Be
ID	Vorname	Nachname	LehrerID	LehrerinID																						
Me	Peter	Meier	Me	Kr																						
Me	Peter	Meier	Me	Sr																						
Sz	Peter	Schulz	NULL	NULL																						
Bm	Hans	Baum	Bm	Be																						

3.8.2 Right-Join

Bei einem **Right-Join** werden auch die Zeilen aus der zweiten Tabelle aufgeführt, die keinen Partner in der ersten Tabelle haben. Es wird mit NULL aufgefüllt.

Mengenschreibweise:	$\text{klassenleitungsteam} \bowtie_{\text{ID}} \text{lehrer}$																									
SQL-Schreibweise:	<pre>SELECT * FROM klassenleitungsteam RIGHT JOIN lehrer ON lehrer.ID = klassenleitungsteam.LehrerID;</pre>																									
Ergebnis:	<p>Alle Lehrer erscheinen, auch wenn sie keinen Partner haben:</p> <table border="1"> <thead> <tr> <th>LehrerID</th> <th>LehrerinID</th> <th>ID</th> <th>Vorname</th> <th>Nachname</th> </tr> </thead> <tbody> <tr> <td>Me</td> <td>Kr</td> <td>Me</td> <td>Peter</td> <td>Meier</td> </tr> <tr> <td>Me</td> <td>Sr</td> <td>Me</td> <td>Peter</td> <td>Meier</td> </tr> <tr> <td>NULL</td> <td>NULL</td> <td>Sz</td> <td>Peter</td> <td>Schulz</td> </tr> <tr> <td>Bm</td> <td>Be</td> <td>Bm</td> <td>Hans</td> <td>Baum</td> </tr> </tbody> </table>	LehrerID	LehrerinID	ID	Vorname	Nachname	Me	Kr	Me	Peter	Meier	Me	Sr	Me	Peter	Meier	NULL	NULL	Sz	Peter	Schulz	Bm	Be	Bm	Hans	Baum
LehrerID	LehrerinID	ID	Vorname	Nachname																						
Me	Kr	Me	Peter	Meier																						
Me	Sr	Me	Peter	Meier																						
NULL	NULL	Sz	Peter	Schulz																						
Bm	Be	Bm	Hans	Baum																						

3.9 Umbenennung

Ein Attribut wird umbenannt bzw. mit einem Alias versehen:

Mengenschreibweise:	$\rho_{Kuerzel \leftarrow ID}(\text{lehrerin})$												
SQL-Schreibweise:	SELECT ID AS Kuerzel, Vorname, Nachname FROM lehrerin;												
Ergebnis:	Ein Attribut erhält ein Alias. <table border="1"><thead><tr><th><i>Kuerzel</i></th><th>Vorname</th><th>Nachname</th></tr></thead><tbody><tr><td>Be</td><td>Petra</td><td>Blume</td></tr><tr><td>Sr</td><td>Clara</td><td>Sommer</td></tr><tr><td>Kr</td><td>Helga</td><td>Kremer</td></tr></tbody></table>	<i>Kuerzel</i>	Vorname	Nachname	Be	Petra	Blume	Sr	Clara	Sommer	Kr	Helga	Kremer
<i>Kuerzel</i>	Vorname	Nachname											
Be	Petra	Blume											
Sr	Clara	Sommer											
Kr	Helga	Kremer											

3.10 Aufgaben:

- Die Tabellen *Kollegium1* und *Kollegium2* zeigen für die Kollegien zweier Schulen, wer welches Bistro gerne besucht. Die Tabelle *Angebot* gibt an, welche Getränke die Bistros ausschenken. Die Tabelle *Vorliebe* gibt an, wer welches Getränk mag.

Kollegium1

Name	Bistro
Anna	Alternative
Anna	Chic
Bruno	Alternative
Bruno	Basic
Chantal	Alternative
Chantal	Basic
Chantal	Chic

Kollegium2

Name	Bistro
Arne	Basic
Bruno	Alternative
Bruno	Basic
Claudia	Basic
Claudia	Chic

Angebot

Bistro	Getränk
Alternative	Tee
Alternative	Wasser
Basic	Wasser
Basic	Kaffee
Chic	Kaffee
Dukat	Tee

Vorliebe

Name	Getränk
Anna	Tee
Anna	Kaffee
Bruno	Wasser
Chantal	Wasser
Chantal	Kaffee

Erstelle in den folgenden Teilaufgaben mit Hilfe der Relationenalgebra jeweils eine neue Tabelle. Formuliere die zugehörigen Datenbankabfragen mit Operatoren der Relationenalgebra.

- Erstelle eine Tabelle aller Kolleginnen und Kollegen beider Schulen mit den bevorzugten Bistros.
- Erstelle einen Tabellenauszug aus *Kollegium1* mit den Bistros, die Anna gerne besucht.
- Es ist Mittagspause in Schule 1. Wer auch in Schule 2 unterrichtet, kann nicht mitkommen, weil dort pädagogischer Tag ist. Modelliere diese Situation mit der Relationenalgebra.
- Erstelle eine Namensliste von *Kollegium1*.
- Erstelle eine Namensliste der Kollegen, die an beiden Schulen unterrichten.
- Erstelle eine Tabelle mit allen möglichen Bistrosbesuchen der Lehrkräfte der ersten Schule unabhängig von den Vorlieben.
- Erstelle die Tabelle, die beschreibt, welche Getränke die Lehrkräfte der ersten Schule in ihren bevorzugten Bistros trinken können.
- Erstelle die Tabelle, die beschreibt, welche Getränke die Lehrkräfte der ersten Schule in ihren bevorzugten Bistros genießen können (also auch mögen).

2. Übersetze die oben angegebenen Anweisungen von der Mengenschreibweise in die SQL-Syntax und teste die Befehle an der Beispieldatenbank *Bistrobeispiel*.
3. Gegeben sind die folgenden Tabellen:

<i>Tabelle1</i>	<i>Tabelle2</i>	<i>Tabelle3</i>	<i>Tabelle4</i>																																						
<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="text-align: center;">A</th> <th style="text-align: center;">B</th> <th style="text-align: center;">C</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">4</td> <td style="text-align: center;">2</td> <td style="text-align: center;">8</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">6</td> <td style="text-align: center;">7</td> <td style="text-align: center;">3</td> </tr> </tbody> </table>	A	B	C	4	2	8	2	2	1	6	7	3	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="text-align: center;">C</th> <th style="text-align: center;">D</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">8</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">6</td> </tr> </tbody> </table>	C	D	8	2	3	6	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="text-align: center;">B</th> <th style="text-align: center;">E</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">5</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">4</td> <td style="text-align: center;">4</td> </tr> <tr> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> </tr> </tbody> </table>	B	E	5	3	4	4	5	4	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="text-align: center;">B</th> <th style="text-align: center;">C</th> <th style="text-align: center;">D</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">4</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">9</td> </tr> </tbody> </table>	B	C	D	1	2	1	4	2	1	1	2	9
A	B	C																																							
4	2	8																																							
2	2	1																																							
6	7	3																																							
C	D																																								
8	2																																								
3	6																																								
B	E																																								
5	3																																								
4	4																																								
5	4																																								
B	C	D																																							
1	2	1																																							
4	2	1																																							
1	2	9																																							

Führe die folgenden relationalen Operationen durch und stelle jeweils die Ergebnistabelle auf. Beschreibe die Aufgabenstellung mittels der behandelten Symbolik.

- (a) Selektion von Tabelle1 mit der Bedingung $B=2$.
- (b) Projektion von Tabelle3 auf E.
- (c) Join von Tabelle1 und Tabelle2 nach dem gemeinsamen Attribut C.
- (d) (Selektion von Tabelle1 mit $B>C$) vereinigt mit (Selektion von Tabelle1 mit $A<5$)

4 Quellen

Viele Inhalte und Beispiele habe ich ohne besonderen Hinweis bei diversen Autoren abgeschrieben. Ich beanspruche keineswegs, hier eine eigene geistige Leistung dokumentiert zu haben.

- SIEGFRIED SPOLWIG, *Johann Penon* u.a. auf <http://www.oszhdh.be.schule.de/gymnasium/faecher/informatik/datenbanken/index.htm>
- THOMAS MOHR u.a. auf <http://www.inf-schule.de>
- CHRISTOPH REEG auf <http://reeg.junetz.de>
- TINO HEMPEL auf <http://www.tinohempel.de/>
- JÜRGEN BURKERT, ROLAND LÄCHA, Datenbanken, Hessisches Landesinstitut für Pädagogik (HeLP), Wiesbaden 2002
- ANDREAS MEIER, Relationale Datenbanken, Springer-Verlag 2001
- BERNHARD SCHRIEK, Informatik mit Java - Band III, Nili-Verlag Werl 2007
- MICHAEL UNTERSTEIN, GÜNTER MATTHIESSEN, Relationale Datenbanken und SQL in Theorie und Praxis, Springer Vieweg 2012
- CARL AUGUST ZEHNDER, Informationssysteme und Datenbanken, vdf Hochschulverlag AG an der ETH Zürich 1998
- <http://de.wikipedia.org>