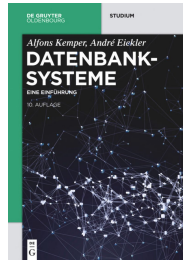




IB320: Datenbanken

Thomas Franzke M.Sc. (Fakultät Informatik)

1. Oktober 2024



(Quelle: <https://www.degruyter.com/document/isbn/978-3-11-044375-2/html>)

ISBN: 9783110443752

Ablauf und Praktikum

- Findet immer zweiwöchentlich statt, jeweils 4 Unterrichtseinheiten
- Praktikum: Leistungsnachweis und prüfungsrelevant!
- Verwendete SW: SQLite, Postgres, Java, ...
- Software ist auf den Hochschulrechnern verfügbar!
- Das Praktikum ist nicht dazu da, persönliche Hardware einzurichten
- Prüfung: Schriftlich, ... Minuten
- Hilfsmittel zur Prüfung: Geodreieck, Lineal

Inhaltsverzeichnis

- 1 Designprinzipien und Motivation
- 2 Konzeptueller Entwurf
- 3 ER-Diagramme, Chen-Notation, (min,max)-Notation, UML
- 4 Relationale Algebra
- 5 ... mehr im nächsten Foliensatz

Probleme beim konventionellen Ansatz mit Papier oder Dateisystem:

- Redundanz und Inkonsistenz
- Beschränkte Zugriffsmöglichkeiten
- Probleme des Mehrbenutzerbetriebs
- Verlust von Daten
- Integritätsverletzung
- Sicherheitsprobleme
- Entwicklungskosten

Redundanz und Inkonsistenz

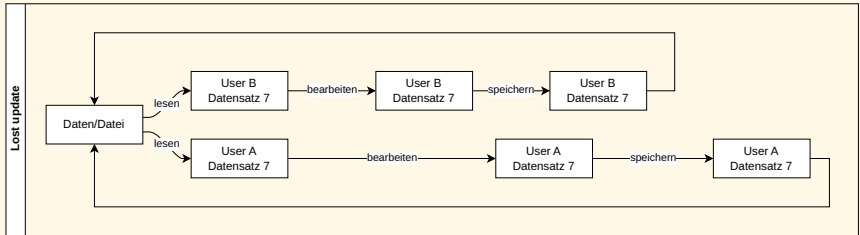
- Daten werden in einzelnen Dateien gespeichert
- Verschiedene Nutzer könnten Kopien anfertigen um Daten weiter zu nutzen
- Beispiel: Adressdaten von Studierenden werden sowohl in der Studierendenverwaltung genutzt und gleichzeitig in der Fakultät
- **Inkonsistenzen**

Beschränkte Zugriffsmöglichkeiten

- Es ist schwer, Daten in einzelnen Dateien logisch miteinander zu verknüpfen
- Im Dateisystem nicht möglich

Motivation DBMS Szenarien

Probleme des Mehrbenutzerbetriebs



- Problem: User A überschreibt Datensatz von User B

Integritätsverletzung

- Eventuell bestehen **Integritätsbedingungen**
- Beispiel: Studierende müssen zuerst Pflichtseminare abgelegt haben bevor sie zur Prüfung zugelassen werden
- Bedingungen sind zwingend einzuhalten
- **daher Ablehnung der Bearbeitung**
- Datenbasis wird immer in einen konsistenten Zustand überführt

Sicherheitsprobleme

- Nicht jeder Nutzer soll Zugriff auf Alle Daten bekommen
- Beispiel: Studierende sollen Kontaktinfos von Dozenten erhalten, aber nicht die Gehaltsdaten
- **Privilegien**

Drei Abstraktionsebenen:

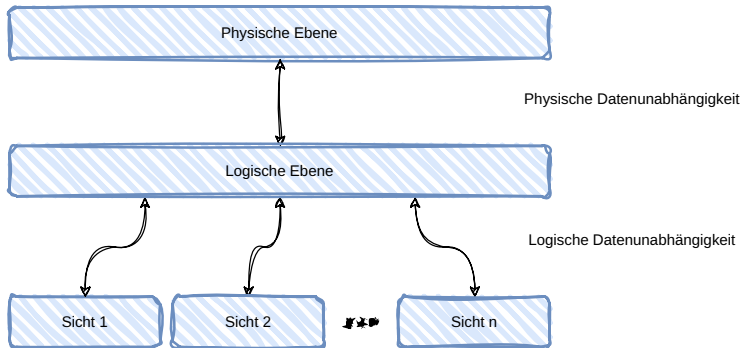
- ➊ **Physische Ebene:** Wie Daten im Hintergrundspeicher gespeichert werden (z.B. Platte)
- ➋ **Logische Ebene:** Datenbankschema, daher welche Daten gespeichert werden
- ➌ **Die Sichten:** Bereitstellung von Teilmengen der Informationen für Bedürfnisse jeweiliger Benutzer/Gruppen

Ähnlich dem OSI-Schichtenmodell ergibt sich auch hier ein "Dreischichtmodell"

- **Physische Datenunabhängigkeit:** Zwischen physischer- und logischer Schicht
- **Logische Datenunabhängigkeit:** Zwischen logischer Ebene und den Sichten

Zwischen logischer Ebene und den Sichten ist ein "Verbergen" von Änderungen natürlich nur begrenzt möglich. Beispielsweise kann das Ergebnis einer Sicht invariant bleiben, die Abfrage kann sich aber ändern.

Datenunabhängigkeit



Je nachdem, in welcher Schicht gearbeitet wird, werden verschiedene Datenmodelle verwendet:

- Beschreibung der Datenobjekte in **Datendefinitionssprache** (DDL = Data Definition Language)
- Festlegung der anwendbaren Operatoren und deren Wirkung in **Datenmanipulationssprache** (DML = Data Manipulation Language)
- Konzeptioneller Entwurf z.B. als **Entity-Relationship-Modell**
- DML-Kommandos können z.B. **direkt** am Terminal oder über grafische Tools eingegeben werden oder
- **eingebettet** in eigene Programme über geeignete APIs (z.B. JDBC)

Der Entwurf kann in drei Abstraktionsebenen unterschieden werden:

- 1 Konzeptionelle Ebene: Unabhängig vom DBMS; Anwendersicht
- 2 Implementationsebene: Umsetzung der vorherigen Ebene in die Konzepte des darunterliegenden Datenbanksystems
- 3 Physische Ebene: Niedrigste Abstraktionsebene (in dieser Veranstaltung weniger relevant)

Das Vorgehen des Datenbankentwurfs ähnelt dem der allgemeinen Softwareentwicklung (Anforderungsanalyse, Spezifikation, Implementierung, ...)

Informationsstrukturanforderung

- Abstraktion in **Objekte** mit passenden Objekttypen
- **Attribute**, die diese Objekte beschreiben
- **Beziehungen** und Beziehungstypen zwischen den Objekten

Objektbeschreibung: Uni-Angestellte

- Anzahl: 1000
- Attribute
 - Personalnummer
 - Typ: char
 - Länge: 9
 - Wertebereich: 0...999.999.99
 - Anzahl Wiederholungen: 0
 - Definiertheit: 100%
 - Identifizierend: Ja
 - Gehalt
 - Typ: dezimal
 - Länge: (8,2)
 - Anzahl Wiederholungen: 0
 - Definiertheit: 90%
 - Identifizierend: Nein
 - Rang ...

Beziehungsbeschreibung: prüfen

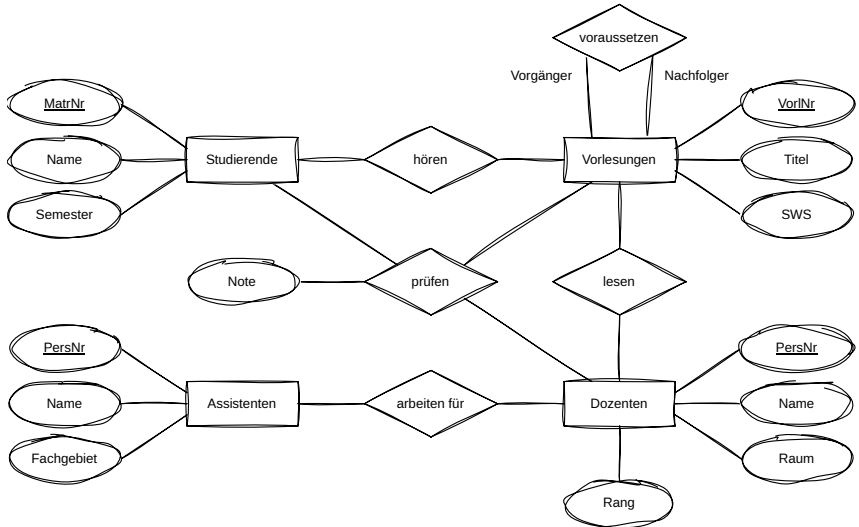
- Beteiligte Objekte
 - Professor/Dozent als Prüfer
 - Studierender als Prüfling
 - Vorlesung/Veranstaltung als Prüfungsstoff
- Attribute der Beziehung
 - Datum
 - Uhrzeit
 - Note
- Anzahl: 100000 pro Jahr

Elemente des ER-Diagramms

- **Entities** (Gegenstandstypen) - Rechtecke
- **Relationships** (Beziehungen)
 - Beziehungstypen - Rauten
 - Verbindungen - ungerichtete Kanten
- **Attribute**
 - Schlüsselattribute - unterstrichen
- **Rollen**
 - Text neben Verbindungen

Datenbankentwurf

Beispiel: ER-Modell konzeptuell



Ein Beziehungstyp R zwischen den Entitytypen E_1, E_2, \dots, E_n kann als Relation angesehen werden. Die Ausprägung der Beziehung R ist eine Teilmenge des kartesischen Produkts der beteiligten Entitytypen.

$$R \subseteq E_1 \times E_2 \times \dots \times E_n \quad (1)$$

- n entspricht dem Grad der Beziehung R .
- Die häufigsten Beziehungstypen sind **binär**.

Datenbankentwurf

Funktionalitäten bei binären Beziehungen

Binäre Beziehungstypen R zwischen Entitytypen E_1 und E_2

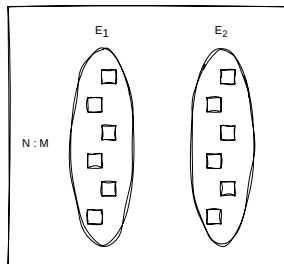
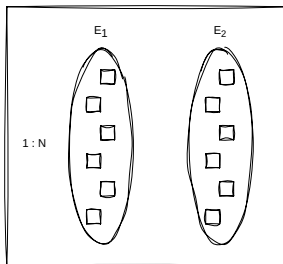
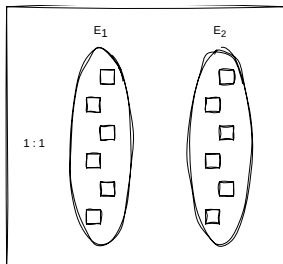
- **1:1-Beziehung**, falls jedem Entity e_1 aus E_1 höchstens ein e_2 aus E_2 zugeordnet (und umgekehrt) ist. Keine Zuordnung ist erlaubt.
- **1:N-Beziehung**, falls jedem Entity e_1 aus E_1 beliebig viele Entities aus E_2 zugeordnet sein können.
- **N:M-Beziehung**, wenn jedes Entity E_1 mit beliebig vielen Entities aus E_2 assoziiert werden kann und umgekehrt.

Bei der 1:N-Beziehung ist die **Richtung** entscheidend! Beispielsweise ist die Aussage "Eine Firma stellt eine Person ein" eine komplett andere als "Eine Person stellt eine Firma ein".

Datenbankentwurf

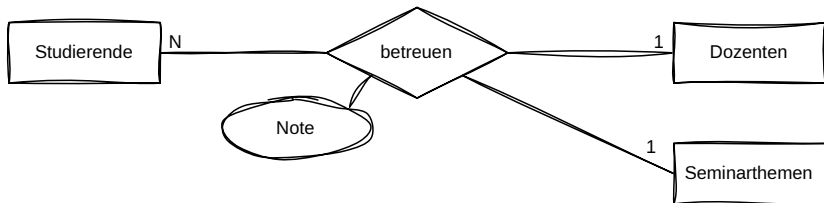
Übung: Beziehungen

Zeichnen Sie passende Verbindungen ein.
Hinweis: Es gibt nicht nur eine Lösung ;)



Datenbankentwurf

Funktionalitäten bei n-stelligen Beziehungen



Darstellung als partielle Funktion:

$$R : E_1 \times \dots \times E_{k-1} \times E_{k+1} \times \dots \times E_n \rightarrow E_k \quad (2)$$

Beispiel:

betreuen : Seminarthemen \times Studierende \rightarrow Dozenten

betreuen : Dozenten \times Studierende \rightarrow Seminarthemen

Datenbankentwurf

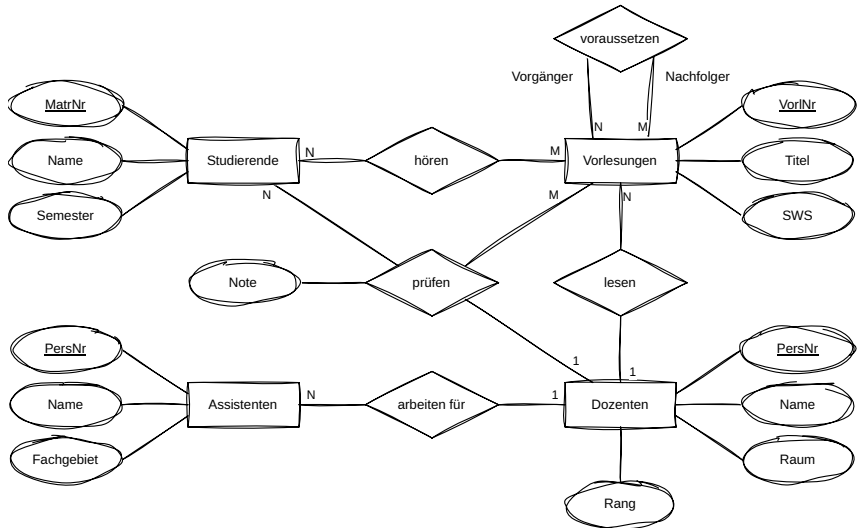
Funktionalitäten bei n -stelligen Beziehungen

Daraus ergeben sich folgende Konsistenzbedingungen:

- Studierende dürfen bei demselben Dozenten nur ein Seminarthema belegen
- Studierende dürfen dasselbe Seminarthema nur einmal bearbeiten (zumindest beim gleichen Professor)
- Dozenten können dasselbe Thema an verschiedene Studierende ausgeben
- Ein Thema kann von verschiedenen Dozenten an unterschiedliche Studierende vergeben werden

Datenbankentwurf

Beispiel: ER-Modell mit Funktionalitäten (Chen-Notation)



Datenbankentwurf

Die (min, max)-Notation

Die obige Darstellung ist nicht sehr präzise, da

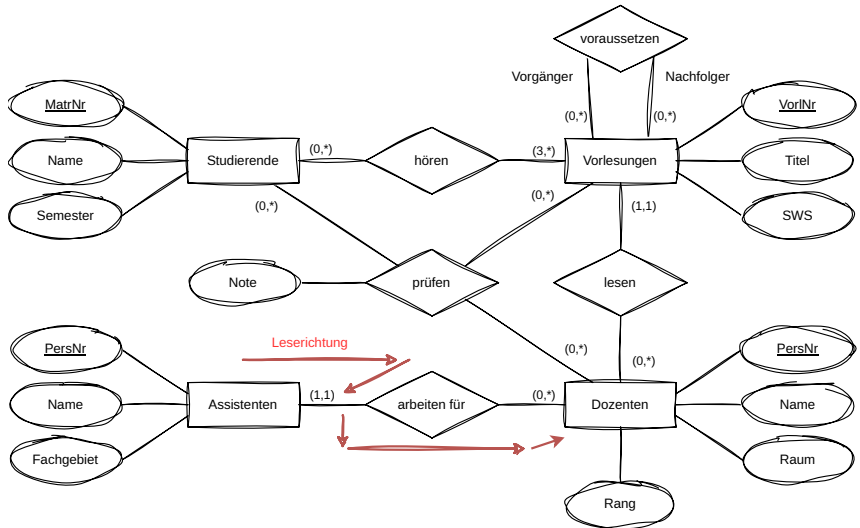
- Nur Extremwerte (eins oder viele) angegeben werden
- Wann immer möglich, sollten auch Unter- bzw. Obergrenzen möglich sein
- Existenzabhängigkeit (muss oder kann)?

Die (min, max)-Notation

- min gibt an, dass in einer Entity mindestens min-Tupel einer Relation R vorhanden sein müssen
- max gibt an, dass höchstens max-Tupel vorhanden sind
- Wenn es Entities geben darf, die gar nicht an der Beziehung teilnehmen, so ist $\text{min} = 0$
- Wenn ein Entity beliebig oft an der Beziehung teilnehmen darf, so wird die max-Angabe durch * ersetzt

Datenbankentwurf

Beispiel: ER-Modell (min, max)-Darstellung



Datenbankentwurf (min, max)-Notation

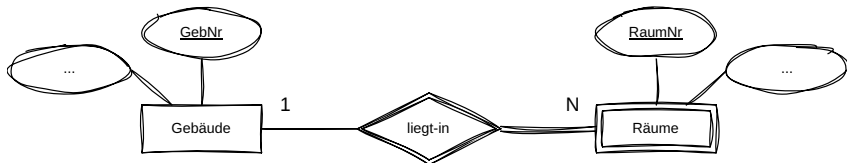
Die Beziehungen werden, wie im Bild dargestellt, gelesen:

- Assistenten arbeiten für genau einen Dozenten
- Dozenten haben keinen oder mehrere Assistenten
- Vorlesungen werden genau von einem Dozenten gelesen
- Dozenten können mehrere Vorlesungen lesen (oder auch nicht)
- Vorlesungen werden von mindestens 3 Studierenden gehört
- Besonders schlaue Studierende besuchen keine Vorlesung < /ironie >

Leserichtung "invers" zur einfachen Kardinalitäten- oder Krähenfußdarstellung!

Datenbankentwurf

Existenzabhängige Entitytypen

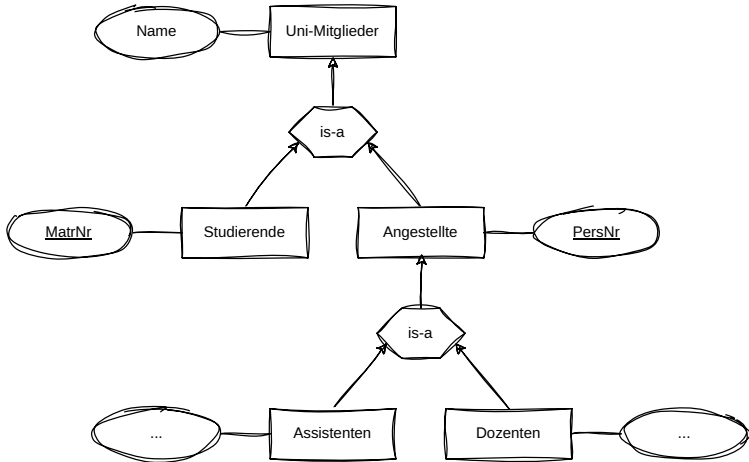


- Weak entities/relationships, die
- in ihrer Existenz von einer anderen, übergeordneten Entity abhängig
- und oft nur in Kombination mit dem Schlüssel dieser Entity eindeutig identifizierbar sind
- Warum kann diese Beziehung keine M:N-Funktionalität haben?
- Warum gibt es diese Darstellung nicht bei der (min, max)-Notation?

Datenbankentwurf

Generalisierung im konzeptuellen Entwurf

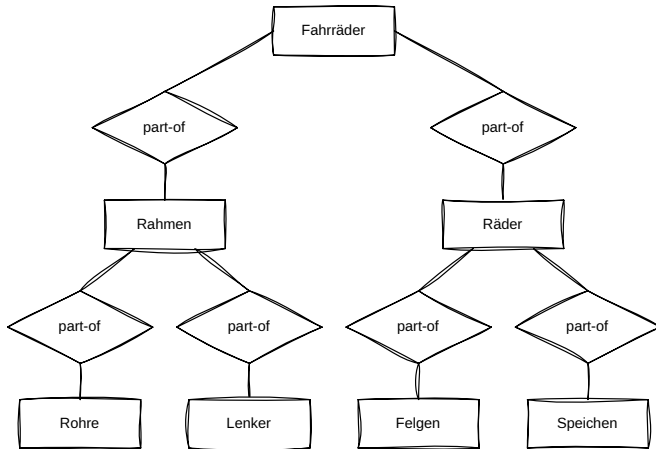
Kennen Sie das auch aus dem SW-Engineering?



Datenbankentwurf

Aggregation im konzeptuellen Entwurf

Kennen Sie das auch aus dem SW-Engineering?

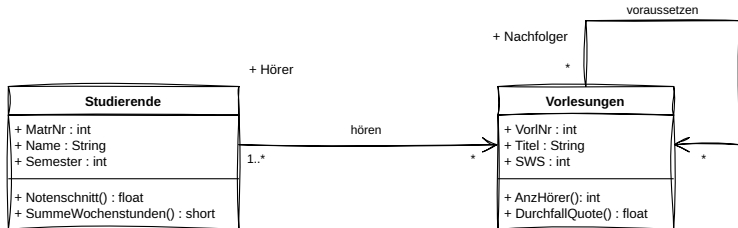


Datenbankentwurf

Konzeptuelle Modellierung mit UML

Das Klassendiagramm ist gut verwendbar, allerdings sind manche Eigenschaften unpassend:

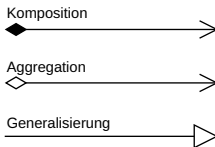
- Attribute sind immer public, da diese in der DB sichtbar sein müssen
- Methoden passen nicht so recht zur Datenbank, wohl aber zur dahinter liegenden Anwendung
- Schlüsselattribute sind nicht darstellbar



Datenbankentwurf

Konzeptuelle Modellierung mit UML

- Generalisierung: Pfeil in Richtung der Oberklasse
- Aggregation: Ungefüllte Raute auf Seite des übergeordneten Objekts
- Komposition: Gefüllte Raute auf Seite des übergeordneten Objekts; existenzabhängig, entspricht der Multiplizität 1



Übung

Machen Sie sich zum Thema Krähenfussnotation schlau.

Das relationale Modell

Das relationale Modell

Mathematischer Formalismus

- Mengenorientierte Verarbeitung der Daten
- Sehr einfach strukturiert über flache Tabellen
- Wertebereiche (Domänen = D_n) dürfen nur unstrukturierte, atomare Werte enthalten (keine Structs)
- Eine Relation R ist def. als Teilmenge des kartesischen Produkts der n Domänen:

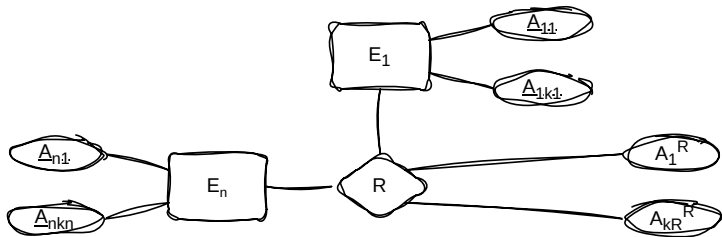
$$R \subseteq D_1 \times \dots \times D_n$$

$$\textit{Telefonbuch} \subseteq \textit{string} \times \textit{string} \times \textit{integer}$$

Telefonbuch		
Name	Straße	Telefon
Jean Luc Picard	La Barre	555 102 673
Barney Gumble	Moes Tavern	555 712 58
Sheldon Cooper	2311 Northern Los Robles Avenue	555 101 011
...

Das relationale Modell

Relationale Darstellung von Beziehungen



$$R : \{ \underbrace{[A_{11}, \dots, A_{1k_1}]}_{\text{Schlüssel von } E_1}, \underbrace{[A_{21}, \dots, A_{2k_2}]}_{\text{Schlüssel von } E_2}, \dots, \underbrace{[A_{n1}, \dots, A_{nk_n}]}_{\text{Schlüssel von } E_n}, \underbrace{[A_1^R, \dots, A_{kR}^R]}_{\text{Attribute von R}} \}^1$$

¹Geschweifte Klammern = Menge, eckige Klammern = Tupel

Das relationale Modell

Relationale Darstellung von Beziehungen

- Erster Schritt: Für jeden Beziehungstypen eine eigene Relation definieren
- In der Schemaverfeinerung mit anderen Relationen zusammenfassen und evtl. wieder eliminieren (nicht bei M:N Beziehungen)
- Die Relation R enthält alle Schlüsselattribute der beteiligten Entities als **Fremdschlüssel**
- und alle Attribute von R
- Klar: Evtl. Umbenennen nötig, da Namen nicht mehrfach vorkommen dürfen
- Info: Der identifizierende/"wichtigste"/Haupt- Schlüssel einer Entity heisst **Primärschlüssel**

Das relationale Modell

Relationale Darstellung von Beziehungen

Beispiel: Studierende - hören - Vorlesungen

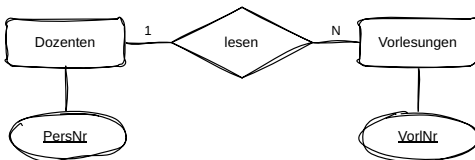
Studierende	
MatrNr	...
261200	...
275500	...
...	...

Vorlesungen	
VorlNr	...
5001	...
4052	...
...	...

hören	
MatrNr	VorlNr
261200	5001
261200	4052
275500	4052
275500	5001
...	...

Das relationale Modell

Verfeinerung des rel. Schemas - 1:N-Beziehungen



- Bei der 1:N-Beziehung existieren viele Datensätze einer Entity (hier die Vorlesung), aber nur genau ein Bezug auf einen Datensatz der anderen Entity (Dozent)
- Das bedeutet, dass eine Richtungsabhängigkeit besteht:
- Ein Dozent liest möglicherweise viele Vorlesungen, eine Vorlesung wird aber von exakt einem Dozenten gelesen

Idee:

Intuitiv könnte man also den Primärschlüssel von der 1-Seite in die Entity auf der N-Seite integrieren.

Das relationale Modell

Verfeinerung des relationalen Schemas

Regel:

Nur Relationen mit gleichem Schlüssel zusammenfassen.

Vorlesungen : {[VorlNr, Titel, SWS]}

Dozenten : {[PersNr, Name, Rang, Raum]}

lesen : {[VorlNr, PersNr]}

Nach Anwendung der Regel:

Vorlesungen : {[VorlNr, Titel, SWS, **gelesenVon²**]}

Dozenten : {[PersNr, Name, Rang, Raum]}

²eigentlich PersNr

Das relationale Modell

Verfeinerung des relationalen Schemas

Vorlesungen			
VorlNr	Titel	SWS	gelesenVon
1010	Grundlagen d. Informatik	4	1001
1015	Grundlagen d. theoretischen Informatik	4	1002
1020	Digitaltechnik	2	1003
1031	Mathematik I	7	1004
1040	Mathematik II	6	1004
1061	Software Engineering I	4	1005
1150	Programmieren I	6	1001
...

Dozenten			
PersNr	Name	Rang	Raum
1001	Nazareth	Prof. Dr.	K013
1002	Siebert	Prof. PhD	J119
1003	Franzke	LfbA. M.Sc.	K014
1004	Sagraloff	Prof. Dr.	K010
1005	Schröter	Prof. Dr.	K020
...

Das relationale Modell

Verfeinerung des relationalen Schemas

Achtung, hier folgt exorbitanter Bockmist!

Nach anderer Verfeinerung:

Vorlesungen : {[VorlNr, Titel, SWS]}

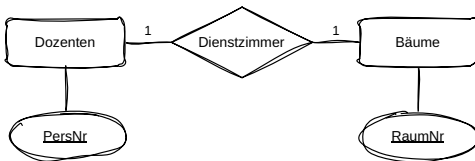
Dozenten : {[PersNr, liestVorl, Name, Rang, Raum]}

Dozenten				
PersNr	liestVorl	Name	Rang	Raum
1001	1010	Nazareth	Prof. Dr.	K013
1002	1015	Siebert	Prof. PhD	J119
1003	1020	Franzke	LfbA. M.Sc.	K014
1004	1031	Sagraloff	Prof. Dr.	K010
1004	1040	Sagraloff	Prof. Dr.	K010
1005	1061	Schröter	Prof. Dr.	K020
1001	1150	Nazareth	Prof. Dr.	K013
...

< /exorbitanterBockmist! >

Das relationale Modell

Verfeinerung des rel. Schemas - 1:1 Beziehungen



Viel einfacher, da die Fremdschlüssel frei positioniert werden können.

Dozenten : {[PersNr, Name, Rang]}

Räume : {[RaumNr, Größe, Lage]}

Dienstzimmer : {[PersNr, RaumNr]}

Das relationale Modell

Verfeinerung des rel. Schemas - 1:1 Beziehungen

Nach Verfeinerung:

Dozenten : $\{[\underline{\text{PersNr}}, \text{Name}, \text{Rang}, \text{RaumNr}]\}$
Räume : $\{[\underline{\text{RaumNr}}, \text{Größe}, \text{Lage}]\}$

oder:

Dozenten : $\{[\underline{\text{PersNr}}, \text{Name}, \text{Rang}]\}$
Räume : $\{[\underline{\text{RaumNr}}, \text{Größe}, \text{Lage}, \text{PersNr}]\}$

Frage:

Welche Relation ist besser in Bezug auf Null-Werte?

Relationale Algebra

- Selektion derjenigen **Tupel** einer Relation die das Selektionsprädikat erfüllen

$$\sigma_F(R)$$

$$\sigma_{Semester > 10}(Studenten)$$

Hier werden die "Dauerstudenten" gefiltert:

- Selektionsprädikat: $F \mapsto Semester > 10$
- Relation: $R \mapsto Studenten$

$\sigma_{Semester > 10}(Studenten)$		
MatrNr	Name	Semester
1235687	Alter Stinker	18
3535222	Otto Faulpelz	12

- Auswählen einzelner/mehrerer **Spalten** einer Relation

$$\Pi_{N_1, N_2, \dots, N_n}(R)$$

$$\Pi_{Rang}(Dozenten)$$

$\Pi_{Rang}(Dozenten)$
Rang
Prof. Dr.
LB

Duplikate können natürlich auftreten. Das Eliminieren dieser trägt zur Laufzeitverbesserung bei.

- Zusammenfassen von zwei Relationen mit gleichen Attributnamen und Domänen (Typen)

$$\Pi_{N_1, N_2, \dots, N_n}(R_1) \cup \Pi_{N_1, N_2, \dots, N_n}(R_2)$$

$$\Pi_{PersNr, Name}(Assistenten) \cup \Pi_{PersNr, Name}(Dozenten)$$

So könnte eine Generalisierung modelliert werden

Eine Duplikateliminierung muss auf jeden Fall erfolgen. Warum?

Relationale Algebra

Mengendifferenz

- Für zwei Relationen R und S mit **gleichem Schema**
- Die Menge der Tupel, die in R aber nicht in S vorkommen
- Beispiel:

$$\Pi_{MatrNr}(Studierende) - \Pi_{MatrNr}(pruefen)$$

Liefert alle Studierende, die noch keine Prüfung abgelegt haben.

Relationale Algebra

Kartesisches Produkt (Kreuzprodukt)

$$R \times S \quad \text{!!! important !!!}$$

- Enthält alle $|R| \times |S|$ möglichen Paare von Tupeln aus R und S
- Das Schema der Ergebnisrelation ist die Vereinigung der Attribute aus R und S

Tabelle _A				Tabelle _B	
A ₁	A ₂	A ₃		B ₁	B ₂
1	2	3		101	102
4	5	6		103	104
7	8	9		105	106

Tabelle _A × Tabelle _B				
A ₁	A ₂	A ₃	B ₁	B ₂
1	2	3	101	102
1	2	3	103	104
1	2	3	105	106
4	5	6	101	102
4	5	6	103	104
4	5	6	105	106
7	8	9	101	102
7	8	9	103	104
7	8	9	105	106

Relationale Algebra Umbenennung

- Falls eine Relation mehrfach in einer Anfrage verwendet wird, könnte es notwendig sein, eine zweite Kopie der Relation zu benutzen
- Diese muss dann natürlich umbenannt werden

$$\rho^3_{R'}(R)$$

³Das ist ein 'Rho'

Relationale Algebra

Der relationale Verbund - Join

Da das Kreuzprodukt sehr große Datenmengen und "Aufblähungen" erzeugt, wurde das Mittel des Joins eingeführt.

Folgende Join-Arten sind möglich:

- Der natürliche Verbund (Natural join)
- Allgemeiner Join (Join, Theta-Join)
- Linker äußerer Join (Left outer join)
- Rechter äußerer Join (Right outer join)
- Semi-Join von L mit R (oder umgekehrt)
- Anti Semi-Join von L mit R (oder umgekehrt)

$$R \bowtie S$$

- Attribute von R: $A_1, \dots, A_m, B_1, \dots, B_k$
- Attribute von S: $B_1, \dots, B_k, C_1, \dots, C_n$
- R und S haben nur B_1, \dots, B_k gleichbenannte Attribute

$$R \bowtie S = \Pi_{A_1, \dots, A_m, R.B_1, \dots, R.B_k, C_1, \dots, C_n} (\sigma_{R.B_1=S.B_1 \wedge \dots \wedge R.B_k=S.B_k} (R \times S))$$

Selektieren derjenigen Tupel, deren Attributwerte für gleichbenannte Attribute der beiden Argumentrelationen gleich sind.

Relationale Algebra

Der natürliche Verbund - Beispiel

$(\text{Studierende} \bowtie \text{ hoeren}) \bowtie \text{Vorlesungen}$

- Zuerst Auswertung des Verbunds von Studierende mit hören.
- m = Anzahl Attribute von 'Studierende', n = Anzahl Attribute von 'hören'
- $m+n-1$ -stellige Ergebnisrelation, da ein gleichbenanntes Attribut 'MatrNr'
- Analog dann mit der Ergebnisrelation und 'Vorlesungen'
- Die Klammer ist nicht nötig, da Join assoziativ (und kommutativ)

$(\text{Studierende} \bowtie \text{ hoeren}) \bowtie \text{Vorlesungen}$						
MatrNr	Name	Semester	VorlNr	Titel	SWS	gelesenVon
11212345	Meier	5	10730	SPS-Programmierung	4	1003
11215563	Müller	3	10730	Datenbanken	4	1003
...

Relationale Algebra

Der allgemeine Join

$$R \bowtie_{\theta} S$$

- θ ist ein beliebiges Prädikat über die Attribute A aus R und B aus S
- Das Ergebnis hat $m+n$ Attribute
- Gleiche Attribute werden mit dem '.' eindeutig benannt
- Beispiel: $R.A_1$ oder $S.B_1$

$$R \bowtie_{A_1 > B_1 \wedge A_2 = B_2 \wedge A_3 < B_5} S$$

Der Theta-Join⁴ ist lediglich eine vereinfachte Formulierung des Kreuzprodukts: $R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$

⁴Diesen Join werden Sie hauptsächlich im Praktikum verwenden

Relationale Algebra

Weitere Joins

Left outer join:

L						R							
A	B	C				C	D	E					
a_1	b_1	c_1				c_1	d_1	e_1					
a_2	b_2	c_2				c_3	d_2	e_2					

 $\text{]}\times$

Resultat				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	-	-

Right outer join:

L						R							
A	B	C				C	D	E					
a_1	b_1	c_1				c_1	d_1	e_1					
a_2	b_2	c_2				c_3	d_2	e_2					

 $\times[$

Resultat				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
-	-	c_3	d_2	e_2

Relationale Algebra

Weitere Joins

(Full) outer join:

L] × [R			=	Resultat				
A	B	C		C	D	E		A	B	C	D	E
a_1	b_1	c_1		c_1	d_1	e_1		a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2		c_3	d_2	e_2		a_2	b_2	c_2	-	-
								-	-	c_3	d_2	e_2

Semi-Join von L mit R:

L			⋈	R			=	Resultat		
A	B	C		C	D	E		A	B	C
a_1	b_1	c_1		c_1	d_1	e_1		a_1	b_1	c_1
a_2	b_2	c_2		c_3	d_2	e_2				

Relationale Algebra

Weitere Joins

Semi-Join von R mit L:

L				R				Resultat		
A	B	C		C	D	E		C	D	E
a_1	b_1	c_1	\times	c_1	d_1	e_1	$=$	c_1	d_1	e_1
a_2	b_2	c_2		c_3	d_2	e_2				

Anti Semi-Join von L mit R:

L				R				Resultat		
A	B	C		C	D	E		A	B	C
a_1	b_1	c_1	\triangleright	c_1	d_1	e_1	$=$	a_2	b_2	c_2
a_2	b_2	c_2		c_3	d_2	e_2				

Der Mengendurchschnitt:

$$R \cap S = R - (R - S)$$

- Vorher: $R - S$ ist die Menge aller Tupel, die in R , aber nicht in S vorkommen
- Versuchen Sie dies grafisch darzustellen

Wird eingesetzt für Anfragen mit **Allquantifizierung**

$$R \div S$$

Beispiel: Suchen der Matrikelnummer derjenigen Studierender, die **alle vierstündigen** Vorlesungen besucht haben:

$$hoeren \div \Pi_{VorINr}(\sigma_{SWS=4}(Vorlesungen))$$

- Für eine Division muss gelten, dass $R \subseteq S$,
- dann enthält das Schema der Ergebnisrelation die Attribute $A_R - A_S$ (= alle Attribute die nicht in S sind)

Relationale Algebra

Die relationale Division

Beispiel:

H	
M	V
m_1	v_1
m_1	v_2
m_1	v_3
m_2	v_2
m_2	v_3

÷

L
V
v_1
v_2

=

$H \div L$
M
m_1

- Es werden nur diejenigen Tupel aus H ausgewählt, die alle Tupel in L aufweisen
- In diesem Fall nur m_1 , da es für m_2 kein v_1 gibt

Relationale Algebra

Gruppierung und Aggregation⁶

- Gruppieren von Tupel mit gleichen Attributwerten für eine Liste von Attributnamen
- Auf jede Gruppe wird eine Aggregatfunktion angewendet
- Typ. Aggregatfunktionen: count, sum, avg, min, max

Beispiel:

$\Gamma_{Semester; count(*)}(Studierende)$

$\Gamma_{Semester; count(*)}(Studierende)$	
Semester	count(*)
18	2
12	17
10	23
8	40
6	80
4	120
2	145

⁵Das ist ein 'Gamma'

⁶Hier ist nicht die Aggregation aus dem ER-Schema gemeint!