

Kapitel 3: Das relationale DB-Modell & SQL

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

Einführung in Datenbanksysteme

Das relationale Datenmodell 3.1.1

3.1 RDM: Überblick über die Konzepte (1)

- Eine Datenbank ist ein Aggregat benannter *Relationen*.
- Eine Relation ist eine Menge von Elementen
 - deren Struktur durch Attribute definiert,
 - deren Identität durch Schlüssel realisiert und
 - deren Werte durch Domänen kontrolliert werden.
- Ein Relationenelement ist das Aggregat seiner Attribute.
- Relationen werden meist durch *Tabellen* dargestellt, wobei jede Tabelle aus Zeilen und Spalten besteht.
- Jede Zeile repräsentiert ein Element der Relation und wird auch als *Tupel* bezeichnet.
- Die Zahl der Zeilen ist variabel und wird *Kardinalität* der Relation genannt.
- Die Spalten der Tabellen enthalten die *Attribute* der Relation.

Einführung in Datenbanksysteme

Das relationale Datenmodell 3.1.2

RDM: Überblick über die Konzepte (2)

- ❑ Die Zahl der Spalten einer Tabelle wird im Schema festgelegt und als *Grad* der Relation bezeichnet.
- ❑ Jeder Spalte ist eine *Domäne* zugeordnet, welche die zulässigen Werte für das Attribut in allen Zeilen festlegt.
- ❑ Jede Tabelle besitzt einen *Primärschlüssel*, der ein einzelnes Attribut oder eine Kombination von Attributen ist, die eine eindeutige Identifikation jedes Tupels innerhalb der Tabelle gestattet.
- ❑ Beziehungen zwischen Datenobjekten werden durch Identifikation des referenzierten Objektes über seinen Primärschlüssel repräsentiert (➡ assoziative Identifikation).
- ❑ Einen Schlüssel, der in Relation A zur Identifikation eines Tupels in Relation B benutzt wird, bezeichnet man als *Fremdschlüssel*.

RDM: Projektdatenbank

Nr	Titel	Budget
100	DB Fahrpläne	300.000
200	ADAC Kundenstamm	100.000
300	Telekom Statistik	200.000

Projekte

Nr	Kurz
100	MFSW

Nr	Kurz
100	UXSW

Nr	Kurz
100	LTSW

Nr	Kurz
200	UXSW

Nr	Kurz
200	PERS

Nr	Kurz
300	MFSW

Projektdurchführung

Kurz	Name	Oberabt
MFSW	Mainframe SW	LTSW

Kurz	Name	Oberabt
UXSW	Unix SW	LTSW

Kurz	Name	Oberabt
PCSW	PC SW	LTSW

Kurz	Name	Oberabt
LTSW	Leitung SW	NULL

Kurz	Name	Oberabt
PERS	Personal	NULL

Abteilungen

Projektdatenbank

RDM: Tabellen und Schlüssel (1)

Im relationalen Datenmodell unterscheidet man zwei Arten von Schlüsseln:

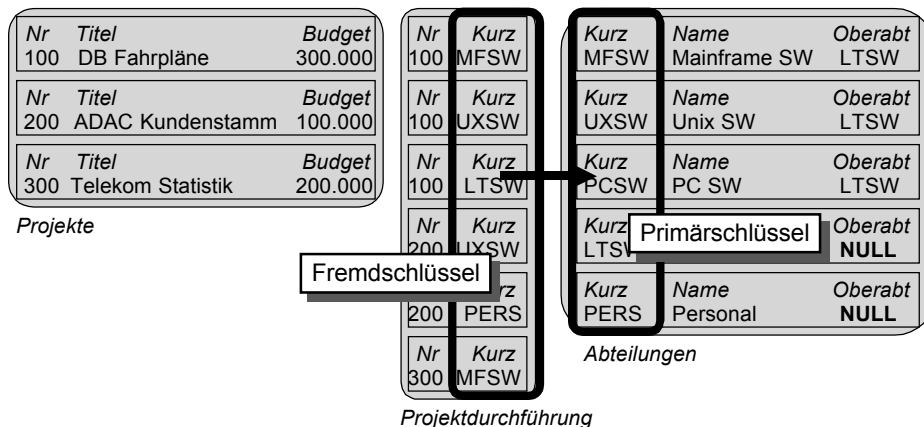
- ❑ **Primärschlüssel:** Ist ein Attribut oder eine minimale Attributkombination, das/die für jede Zeile der Tabelle eindeutige Werte bzw. Wertkombinationen enthält. Es kann mehrere Attribute mit dieser Eigenschaft geben (*Schlüsselkandidaten*).
- ❑ **Fremdschlüssel:** Ist ein Attribut oder Attributkombination, das/die als Werte Primärschlüsselwerte anderer Tabellen annimmt. Er ist eine Referenz auf ein anderes Objekt.

Duplikate sind in Tabellen nicht erlaubt \Rightarrow die Gesamtheit aller Attribute bildet automatisch einen Schlüsselkandidaten. Oft ist jedoch die Einführung eines künstlichen Schlüssels z.B. einer eindeutigen Nummer (ID) sinnvoll.

Eine Relation mit Primärschlüssel repräsentiert eine Funktion von den Primärschlüsselattributen zu den Nicht-Schlüsselattributen.

Beispiel: Kurz \Rightarrow (Name, Oberabt), Kurz \Rightarrow Name, Kurz \Rightarrow Oberabt

RDM: Tabellen und Schlüssel (2)



Projektdatenbank

RDM: Assoziationen

Bei der Benutzung des Relationenmodells unterscheidet man meist zwei Arten der Verwendung von Tabellen:

- ❑ **Entitätentabellen** stellen Objekte (Entitäten) der Anwendung dar (z.B. Projekte, Abteilungen). Jede Zeile einer solchen Tabelle beschreibt ein Objekt.
- ❑ **Beziehungstabellen** werden unter Verwendung von Fremdschlüsseln zur Darstellung von Assoziationen zwischen Objekten verwendet (z.B. Projektdurchführung).

1:1 und 1:n Beziehungen lassen sich durch Aufnahme eines Fremdschlüsselattributs in eine Entitätentabelle repräsentieren (z.B. Attribut Oberabteilung in Abteilungen). Falls keine assoziierte Entität existiert, muß der ausgezeichnete Nullwert **NULL** als Fremdschlüsselwert verwendet werden. **NULL** ist kein zulässiger Primärschlüsselwert.

Generell: Bei 1:1 und 1:n Beziehungen sind keine Beziehungstabellen notwendig.

Beachte: Im relationalen Modell existieren zur Verknüpfung von Objekten keine Zeiger oder vergleichbare Strukturen. Das relationale Modell basiert auf assoziativer Identifikation mit Hilfe von Werten.

RDM: Datendefinition

Schemadefinition der Projektdatenbank:

```
create table Projekte
( Nr integer not null,
  Titel char(30) not null,
  Budget decimal(10,2) not null,
  primary key(Nr) );
```

```
create table Projektdurchfuehrung
( Nr integer not null,
  Kurz char(4) not null,
  primary key(Nr, Kurz) );
```

```
create table Abteilungen
( Kurz char(4) not null,
  Name char(30) not null,
  Oberabt char(4),
  primary key(Kurz) );
```

Referentielle Integrität in SQL: Kapitel 3.2

RDM: Referentielle Integrität

Referentielle Integrität: Zu jedem benutzten Fremdschlüssel existiert ein Tupel mit einem entsprechenden Primärschlüsselwert in der referenzierten Tabelle.

Überprüfung der referentiellen Integrität ist notwendig beim

- Einfügen eines neuen Fremdschlüsselwertes in eine Beziehungstabelle. Das referenzierte Objekt mit diesem Wert als Primärschlüssel muß existieren.
- Löschen eines Tupels aus einer Entitätentabelle. Auf dieses Tupel dürfen keine Referenzen bestehen. Gibt es noch Referenzen, bieten sich mehrere Möglichkeiten an:
 - Eine Fehlermeldung wird erzeugt.
 - Propagierung der Löschoption, das referenzierende Tupel wird ebenfalls gelöscht (→ *kaskadiertes Löschen*).
 - Die Referenzen können durch Setzen des Fremdschlüssels auf einen Nullwert ungültig gemacht werden, sofern dieser nicht Bestandteil des Schlüssels ist.

RDM: Domänen

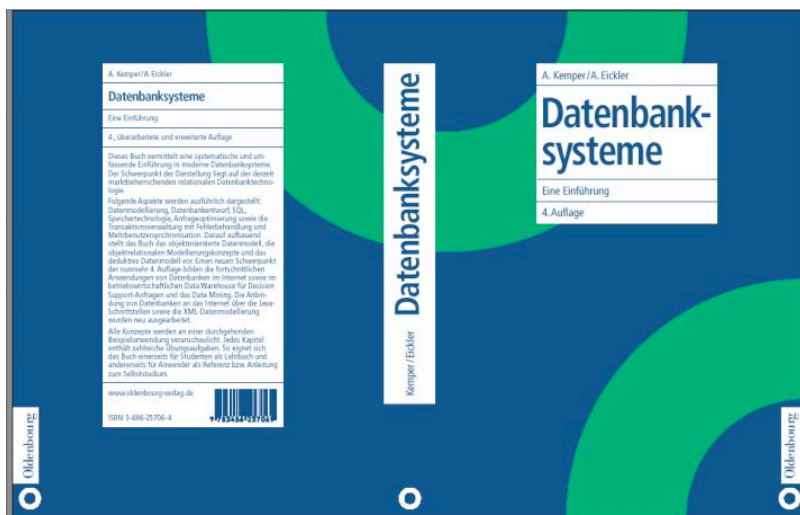
- Domänen legen zulässige Wertebereiche für Attribute fest. Sie sind mit Typen vergleichbar und können
 - mit vordefinierten Typen übereinstimmen, `Int`
 - spezielle Wertmengen festlegen. `"Yes", "No", "Don't know"`
- Operationen auf Attributen, wie z.B. der Vergleich zwischen Budget und Nummer, können auf ihre Zulässigkeit überprüft werden.

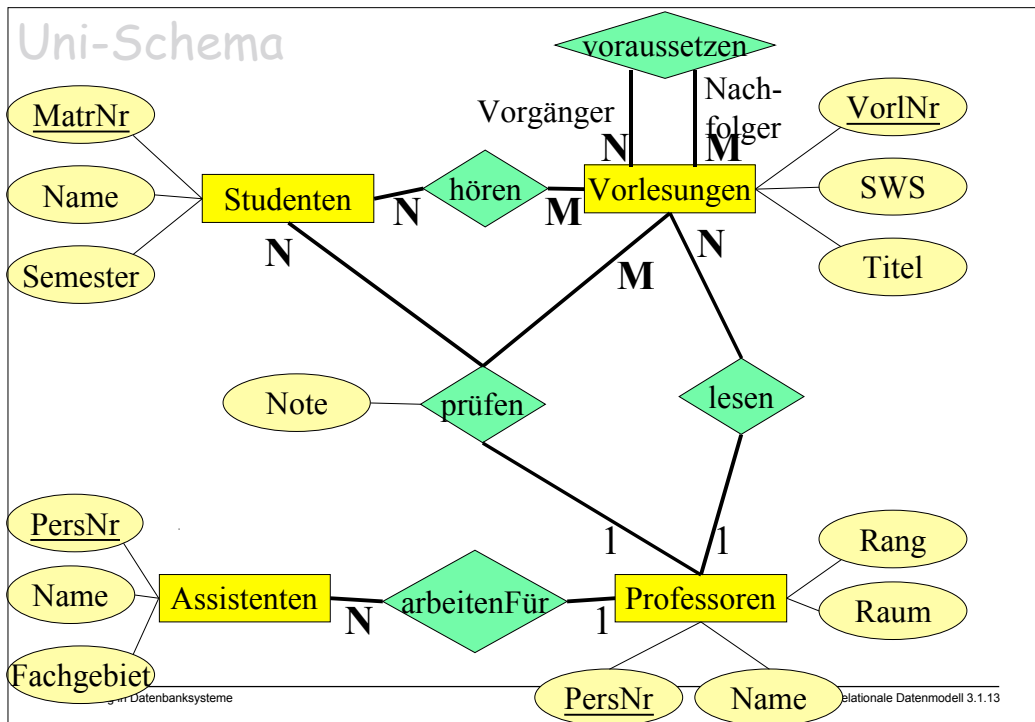
RDM: Entwurf relationaler Schemata

Zwei alternative Methoden:

- ❑ Entwickle zunächst ein ER-Diagramm, leite daraus ein relationales Schema mit Entitäten- und Beziehungstabellen ab bevorzugt
(vgl. C. Batini, S. Ceri, S.B. Navathe. Conceptual Database Design - An Entity Relationship Approach, Benjamin/Cummings, Redwood City, Kalifornien, 1992).
- ❑ Sammle funktionale Abhängigkeiten aus der Anforderungsdefinition und erzeuge daraus ein relationales Schema in Normalform (Im Trend 1970...80). Ausführlich in der Literatur beschrieben (vgl. S.M. Lang, P.C. Lockemann. Datenbankeinsatz. Springer, Berlin u.a., 1995).

Acknowledgments





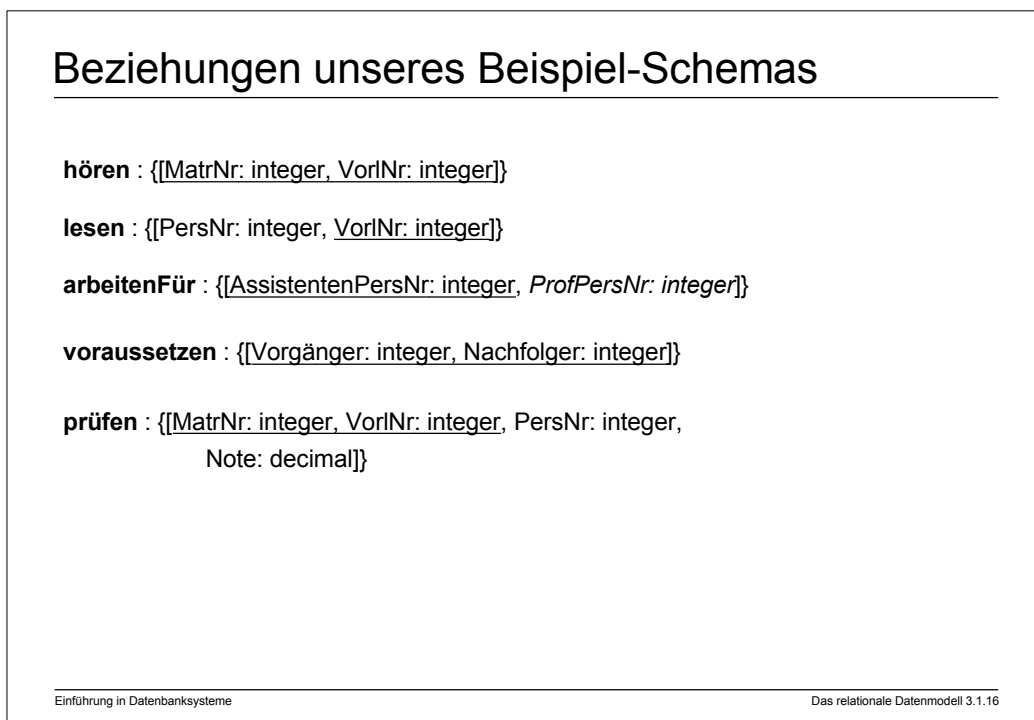
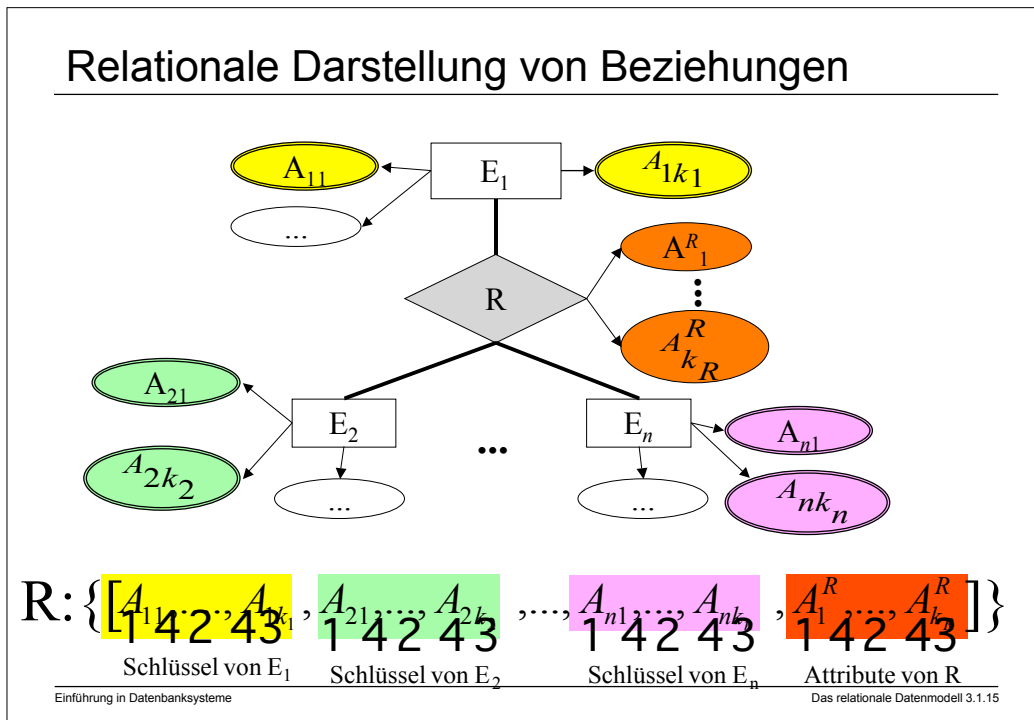
Relationale Darstellung von Entitytypen

Studenten: {[MatrNr:integer, Name: string, Semester: integer]}

Vorlesungen: {[VorlNr:integer, Titel: string, SWS: integer]}

Professoren: {[PersNr:integer, Name: string, Rang: string, Raum: integer]}

Assistenten: {[PersNr:integer, Name: string, Fachgebiet: string]}



Schlüssel der Relationen

hören : {[MatrNr: integer, VorlNr: integer]}

lesen : {[PersNr: integer, VorlNr: integer]}

arbeitenFür : {[AssistentenPersNr: integer, ProfPersNr: integer]}

voraussetzen : {[Vorgänger: integer, Nachfolger: integer]}

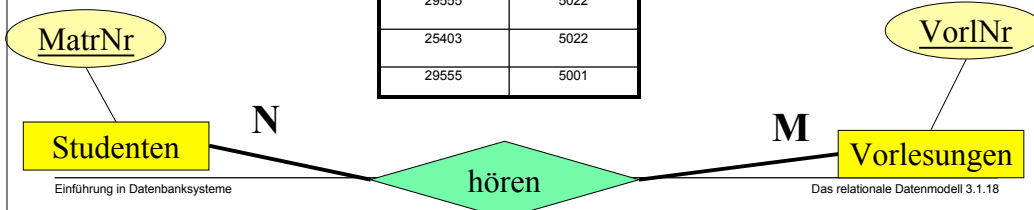
prüfen : {[MatrNr: integer, VorlNr: integer, PersNr: integer,
Note: decimal]}

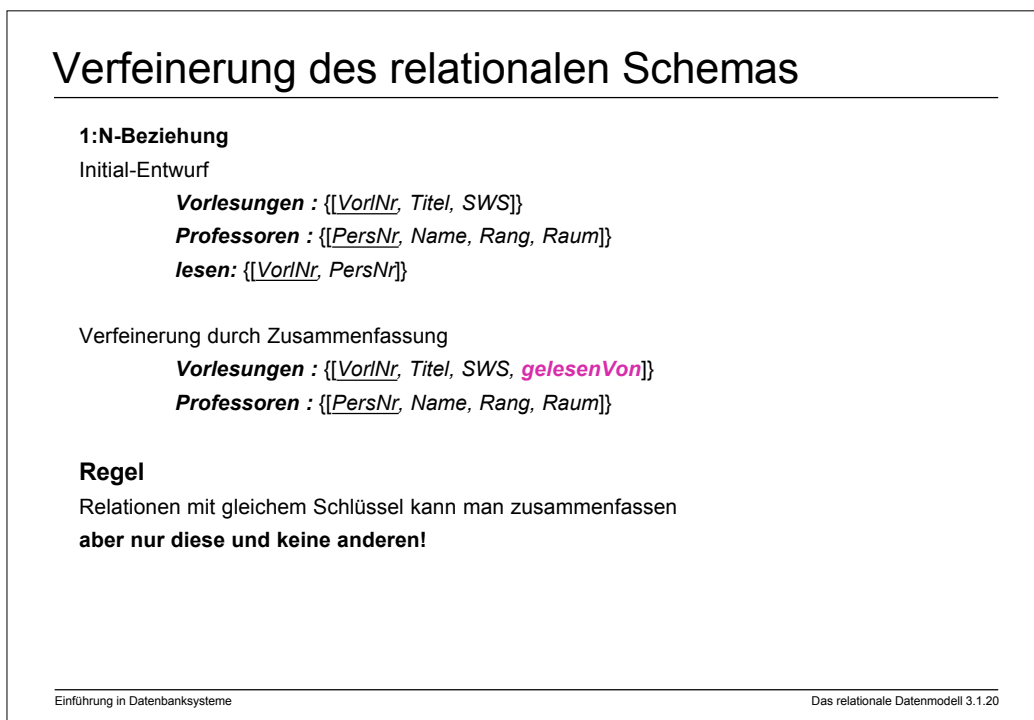
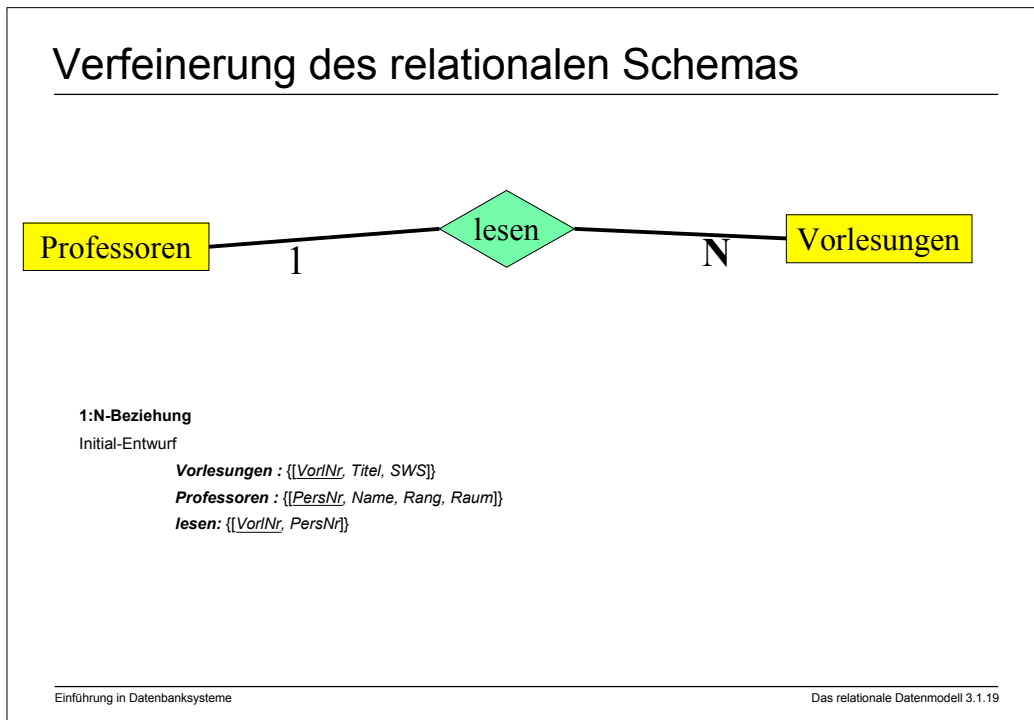
Ausprägung der Beziehung *hören*

Studenten	
<u>MatrNr</u>	...
26120	...
27550	...
...	...

hören	
<u>MatrNr</u>	<u>VorlNr</u>
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

Vorlesungen	
<u>VorlNr</u>	...
5001	...
4052	...
...	...

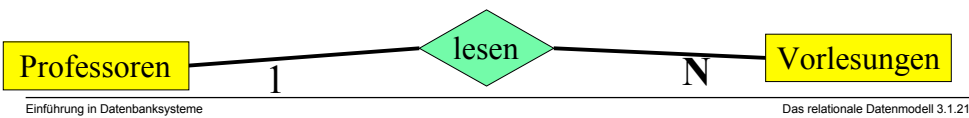




Ausprägung von *Professoren* und *Vorlesung*

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

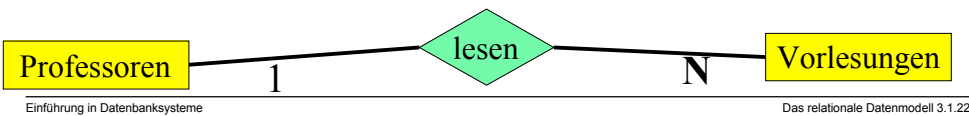
Vorlesungen			
VorNr	Titel	SWS	Gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137



Vorsicht: So geht es NICHT

Professoren				
PersNr	Name	Rang	Raum	liest
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
...
2134	Augustinus	C3	309	5022
2136	Curie	C4	36	??

Vorlesungen		
VorNr	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
5216	Bioethik	2
5259	Der Wiener Kreis	2
5022	Glaube und Wissen	2
4630	Die 3 Kritiken	4



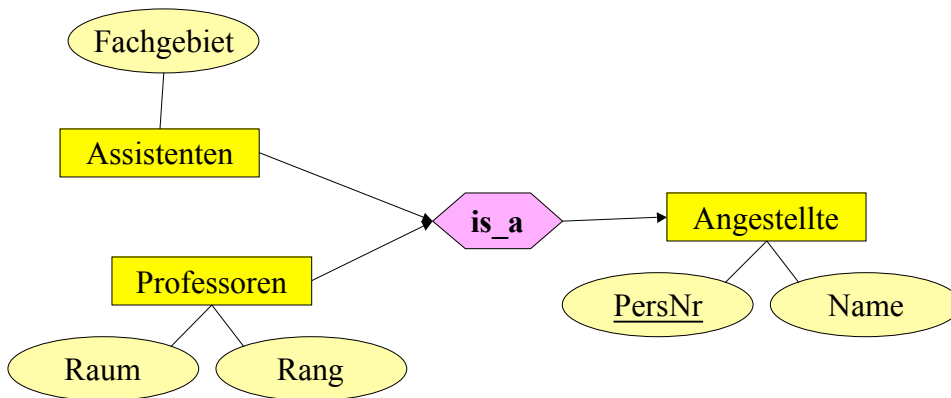
Anomalien

Professoren				
PersNr	Name	Rang	Raum	liest
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
...
2134	Augustinus	C3	309	5022
2136	Curie	C4	36	??

Vorlesungen		
VorNr	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
5216	Bioethik	2
5259	Der Wiener Kreis	2
5022	Glaube und Wissen	2
4630	Die 3 Kritiken	4

- Update-Anomalie: Was passiert wenn Sokrates umzieht
- Lösch-Anomalie: Was passiert wenn „Glaube und Wissen“ wegfällt
- Einfügeanomalie: Curie ist neu und liest noch keine Vorlesungen

Relationale Modellierung der Generalisierung



Angestellte: {[PersNr, Name]}

Professoren: {[PersNr, Rang, Raum]}

Assistenten: {[PersNr, Fachgebiet]}

Funktionale Abhängigkeiten

Schema

$$\mathcal{R} = \{A, B, C, D\}$$

Ausprägung R

Seien $r \in \mathcal{R}$, $s \in \mathcal{R}$

$r \twoheadrightarrow s$ genau dann wenn $r.A = s.A \wedge r.B = s.B$

R			
A	B	C	D
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

$$\{A\} \twoheadrightarrow \{B\}$$

$$\{C, D\} \twoheadrightarrow \{B\}$$

Nicht: $\{B\} \twoheadrightarrow \{C\}$

Notationskonvention:

$$CD \twoheadrightarrow B$$

Beispiel

Stammbaum				
Kind	Vater	Mutter	Opa	Oma
Sofie	Alfons	Sabine	Lothar	Linde
Sofie	Alfons	Sabine	Hubert	Lisa
Niklas	Alfons	Sabine	Lothar	Linde
Niklas	Alfons	Sabine	Hubert	Lisa
...	Lothar	Martha
...

Kind \twoheadrightarrow Vater, Mutter

Kind, Opa \twoheadrightarrow Oma

Kind, Oma \twoheadrightarrow Opa

Schlüssel

\mathcal{R} ist ein Super-Schlüssel, falls folgendes gilt:

\mathcal{R}

ist voll funktional abhängig von \mathcal{R} genau dann wenn gilt

\mathcal{R} und

\mathcal{R} kann nicht mehr verkleinert werden, d.h.

- \mathcal{R} folgt, dass $(\mathcal{R} \setminus \{A\})$ nicht gilt, oder kürzer
- $\mathcal{R} : ((\mathcal{R} \setminus \{A\}) \rightarrow \mathcal{R})$

Notation für volle funktionale Abhängigkeit: $\mathcal{R} \twoheadrightarrow \mathcal{R}$

\mathcal{R} ist ein Kandidaten-Schlüssel, falls folgendes gilt:

$\mathcal{R} \twoheadrightarrow \mathcal{R}$

Schlüsselbestimmung

Städte			
Name	BLand	Vorwahl	EW
Frankfurt	Hessen	069	650000
Frankfurt	Brandenburg	0335	84000
München	Bayern	089	1200000
Passau	Bayern	0851	50000
...

Kandidaten-schlüssel von *Städte*:

$\{Name, BLand\}$

$\{Name, Vorwahl\}$

Beachte, dass 2 kleinere Städte dieselbe Vorwahl haben können

Bestimmung funktionaler Abhängigkeiten

Professoren: {[PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung]}

- {PersNr} → {PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung}
- {Ort,Bland} → {EW, Vorwahl}
- {PLZ} → {Bland, Ort, EW}
- {Bland, Ort, Straße} → {PLZ}
- {Bland} → {Landesregierung}
- {Raum} → {PersNr}

Zusätzliche Abhängigkeiten, die aus obigen abgeleitet werden können:

- {Raum} → {PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung}
- {PLZ} → {Landesregierung}

RDM: Normalformen

Definitionen:

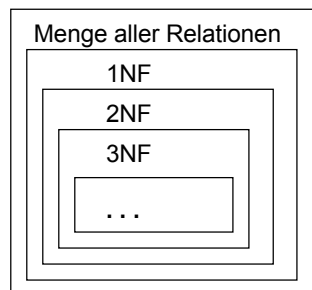
1. NF: Eine Relation ist in 1NF genau dann, wenn die Domänen aller Attribute elementar sind (keine geschachtelten Tabellen, Relationen, ...).

2. NF: Eine Relation ist in 2NF genau dann, wenn sie in 1NF ist und jedes Nichtschlüsselattribut von den Schlüsselkandidaten voll funktional abhängig ist.

3. NF: Eine Relation ist in 3NF genau dann, wenn sie in 2NF ist und kein Nichtschlüsselattribut transitiv funktional von Schlüsselkandidaten abhängt.

...

--> Wir beschäftigen uns mit dem Thema in einer anderen Vorlesung



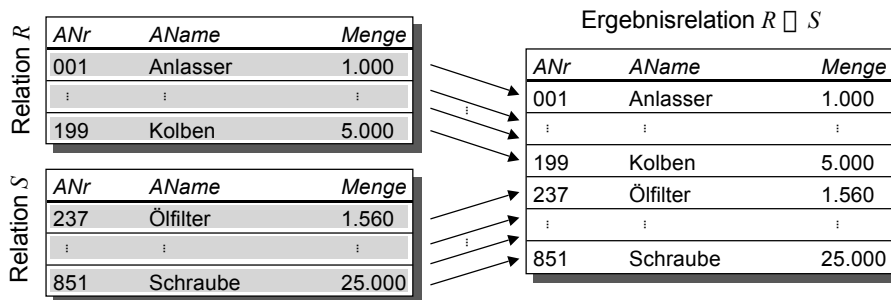
Abhängigkeitsbewahrende Zerlegung möglich

RDM: Relationale Algebra - Anfragen (1)

Vereinigung $R \cup S$:

- Alle Tupel zweier Relationen werden in einer Ergebnisrelation zusammengefaßt.
- Das Ergebnis enthält keine Duplikate.

$$R \cup S := \{ r \mid r \in R \vee r \in S \}$$



Einführung in Datenbanksysteme

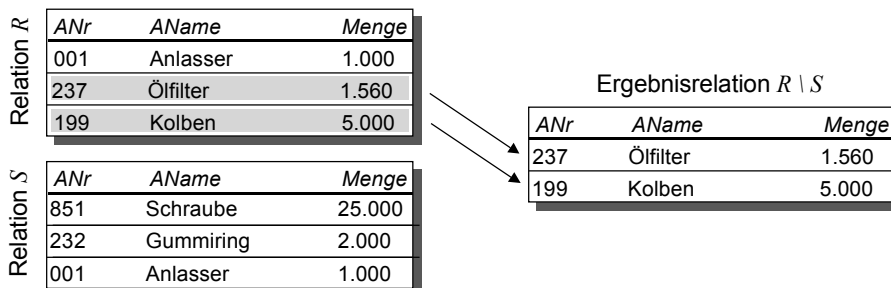
Das relationale Datenmodell 3.1.31

RDM: Relationale Algebra - Anfragen (2)

Differenz $R \setminus S$:

- Die Tupel zweier Relationen werden miteinander verglichen.
- Die in der ersten, nicht aber in der zweiten Relation befindlichen Tupel werden in die Ergebnisrelation aufgenommen.

$$R \setminus S := \{ r \mid r \in R \wedge r \notin S \}$$



Einführung in Datenbanksysteme

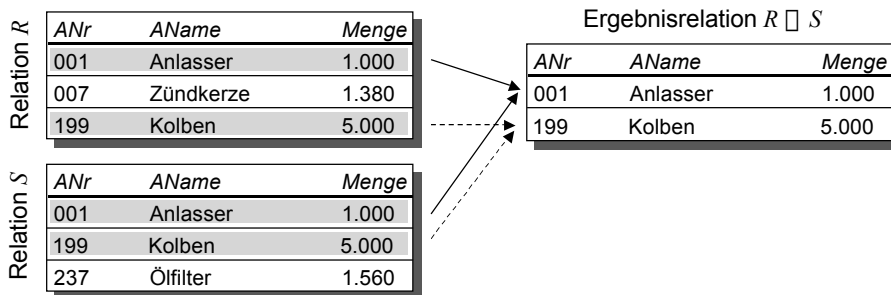
Das relationale Datenmodell 3.1.32

RDM: Relationale Algebra - Anfragen (3)

Durchschnitt $R \cap S$:

- Alle Tupel, die sowohl in der Relationen R als auch in der Relation S enthalten sind, werden in der Ergebnisrelation zusammengefaßt.

$$R \cap S := \{ r \mid r \in R \cap r \in S \}$$



RDM: Relationale Algebra - Anfragen (4)

Kartesisches Produkt $R \times S$:

- Alle Tupel zweier Relationen R und S werden kombinatorisch miteinander verbunden. Wenn die Relation R n Spalten und die Relation S m Spalten umfaßt, dann besitzt $R \times S$ (n+m) Spalten.
- Wenn die Relation R k Zeilen und die Relation S l Zeilen umfaßt, dann besitzt $R \times S$ (k*l) Zeilen.
- Um eindeutige Attributbezeichnungen in der Ergebnisrelation zu gewährleisten, müssen Attribute, die in den Relationen R und S gleich bezeichnet sind, vor der Bildung des kartesischen Produkts umbenannt werden.

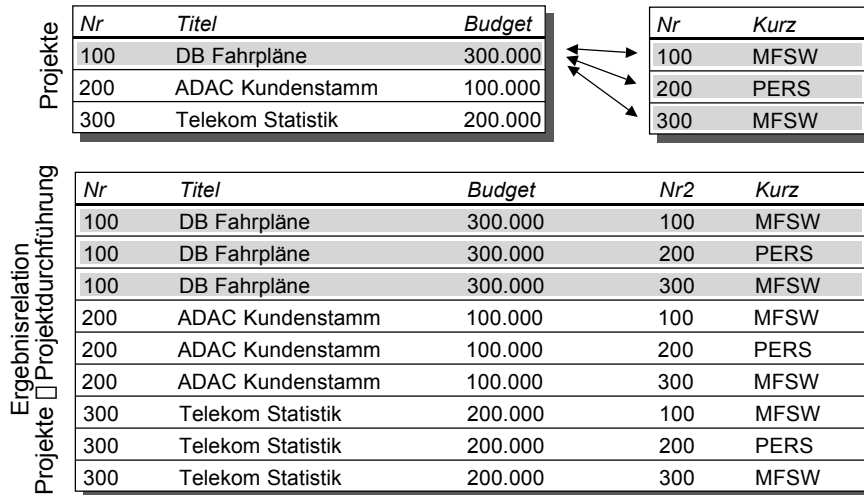
$$R \times S := \{ (r_1, \dots, r_n, s_1, \dots, s_m) \mid (r_1, \dots, r_n) \in R, (s_1, \dots, s_m) \in S \}$$

Beispiel:

- Projekte \times Projektdurchführung (s. nächste Folie)

RDM: Relationale Algebra - Anfragen (5)

Beispiel: Projekte \bowtie Projektdurchführung



Einführung in Datenbanksysteme

Das relationale Datenmodell 3.1.35

RDM: Relationale Algebra - Anfragen (6)

Join (Verbindung) $R \bowtie_{\sigma} S$:

- Eine Verbindung zwischen zwei Relationen wird in einer Kombination von kartesischem Produkt und nachfolgender Selektion (σ) gemäß des Prädikats σ hergestellt.
- Im allgemeinen Fall (*Theta-Join*) vergleicht ein (beliebiges) Prädikat σ mehrere Attribute aus den Relationen R und S (Spezialfall: Equi-Join).

$$R \bowtie_{\sigma} S := \sigma_{\sigma}(R \times S)$$

Beispiele:

- $Projekte \bowtie_{(Nr \neq Nr)} Projektdurchführung$ (s. nächste Folie)
- $Projekte \bowtie_{(Budget > 150000) \wedge (Nr = Nr)} Projektdurchführung$

- Die Ergebnisrelation enthält die Zeilen des kartesischen Produkts der Relationen R und S, die σ erfüllen.

Einführung in Datenbanksysteme

Das relationale Datenmodell 3.1.36

RDM: Relationale Algebra - Anfragen (7)

Beispiel: $\text{Projekte} \bowtie_{(Nr \neq Nr)} \text{Projektdurchführung}$

Projekte	Nr	Titel	Budget
	100	DB Fahrpläne	300.000
	200	ADAC Kundenstamm	100.000
	300	Telekom Statistik	200.000

Projektdurchführung
(Ausschnitt)

Nr	Kurz
100	MFSW
200	PERS
300	MFSW

Ergebnisrelation	Nr	Titel	Budget	Nr2	Kurz	
	100	DB Fahrpläne	300.000	200	PERS	
	100	DB Fahrpläne	300.000	300	MFSW	
	200	ADAC Kundenstamm	100.000	100	MFSW	
	200	ADAC Kundenstamm	100.000	300	MFSW	
	300	Telekom Statistik	200.000	100	MFSW	
	300	Telekom Statistik	200.000	200	PERS	

RDM: Relationale Algebra - Anfragen (8)

Join (Verbindung): Fortsetzung

□ Von besonderer Bedeutung im RDM ist der *Natural Join*, da er eine Verknüpfung von Tabellen über ihre Fremdschlüsselwerte erlaubt.

• Beispiel:

– $\text{Projekte} \bowtie \text{Projektdurchführung} := \text{Projekt} \bowtie_{Nr = Nr} \text{Projektdurchführung}$

– In diesem Fall betrachtet □ nur die Gleichheit zwischen Fremdschlüssel und Primärschlüssel, die den gleichen Attributnamen (*Nr*) besitzen.

□ Weitere abgeleitete *Joinoperationen* (*Semi-Join*, *Outer-Join*, ...) und die *Division* zweier Relationen sind beschrieben in:

- S.M. Lang, P.C. Lockemann. Datenbankeinsatz. Springer, Berlin u.a., 1995.

RDM: Relationale Algebra - Anfragen (9)

Natural Join: $\bowtie_{(Nr-Nr)}$ Projekte \bowtie Projektdurchführung

Projekte

Nr	Titel	Budget
100	DB Fahrpläne	300.000
200	ADAC Kundenstamm	100.000
300	Telekom Statistik	200.000

Projektdurchführung (Ausschnitt)

Nr	Kurz
100	MFSW
200	PERS
300	MFSW

Ergebnisrelation

Nr	Titel	Budget	Nr2	Kurz
100	DB Fahrpläne	300.000	100	MFSW
200	ADAC Kundenstamm	100.000	200	PERS
300	Telekom Statistik	200.000	300	MFSW

RDM: Relationale Algebra - Anfragen (10)

Projektion $\pi_{(r_1, \dots, r_n)}(R)$:

- n Spalten einer m-stelligen Relation R werden über ihren Namen ausgewählt.
- Dadurch entsteht eine n-stellige Relation ($n \leq m$).
- Die Reihenfolge der Spalten in der Ergebnisrelation kann definiert werden.
- Duplikatelimination in der Ergebnisrelation.

$$\pi_{(r_1, \dots, r_n)}(R) := \{ (r_1, \dots, r_n) \mid (r_1, \dots, r_m) \in R \}$$

□ Beispiel: $\pi_{(Nr, Budget)}(\text{Projekte})$

Projekte

Nr	Titel	Budget
100	DB Fahrpläne	300.000
200	ADAC Kundenstamm	100.000
300	Telekom Statistik	200.000



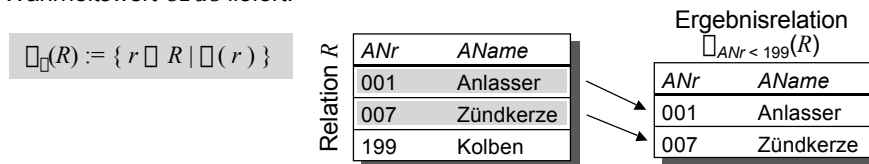
Ergebnisrelation $\pi_{(Nr, Budget)}(\text{Projekte})$

Nr	Budget
100	300.000
200	100.000
300	200.000

RDM: Relationale Algebra - Anfragen (11)

Selektion $\sigma(R)$:

- Bestimmte Tupel einer Relation werden ausgewählt und in der Ergebnisrelation vereinigt.
- Zur Auswahl der zu übernehmenden Tupel dient das Prädikat $\sigma: R \mapsto \{true, false\}$, in dem die Attributbezeichner als Eingabevariablen dienen.
- Anwendung dieses Prädikats auf jedes Tupel der Ausgangsrelation, indem die Werte des Tupels unter den jeweiligen Attributen für die Variablen eingesetzt werden.
- In die Ergebnisrelation werden alle Tupel übernommen, für die das Prädikat den Wahrheitswert *true* liefert.



Einführung in Datenbanksysteme

Das relationale Datenmodell 3.1.41

RDM: Relationale Algebra: Zusammenfassung

Vorteil:

- Einfache, mathematische Behandlung, z.B. $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
- Einfache (naive) Implementierung durch typisierte (!?) algebraische Signatur

Nachteile:

- Eingeschränkte Ausdrucksmächtigkeit auf Relationenebene (Summe, Mittelwert, Kardinalität)
- Reine Anfragesprache
- Prozedurale Formulierung des Lösungswegs (\mapsto Expressions) statt deklarativer Spezifikation des Ergebnisses (\mapsto Kalküle)

Einführung in Datenbanksysteme

Das relationale Datenmodell 3.1.42