

# Kapitel 3: Das relationale DB-Modell & SQL

|                               | Relationales Datenmodell (RDM) | Objekt-orientierte Datenmodelle (OODM) | Objekt-relationale Datenmodelle | Semi-strukturierte Datenmodelle |
|-------------------------------|--------------------------------|--|---------------------------------|---------------------------------|
| Überblick über die Konzepte   | <b>3.1</b>                     | <b>4.1</b>                             | <b>5.1</b>                      | <b>6.1</b><br><b>6.2</b>        |
| Darstellung von Assoziationen |                                |  |                                 |                                 |
| Datendefinition               |                                |  |                                 |                                 |
| Anfragen                      |                                |  |                                 |                                 |
| Aktualisierungsoperationen    |                                |  |                                 |                                 |
| Spezifika                     | <b>3.2 SQL</b>                 | <b>4.2 ODMG</b>                        | <b>5.2, 5.3</b>                 |                                 |

# 3.1 RDM: Überblick über die Konzepte (1)

---

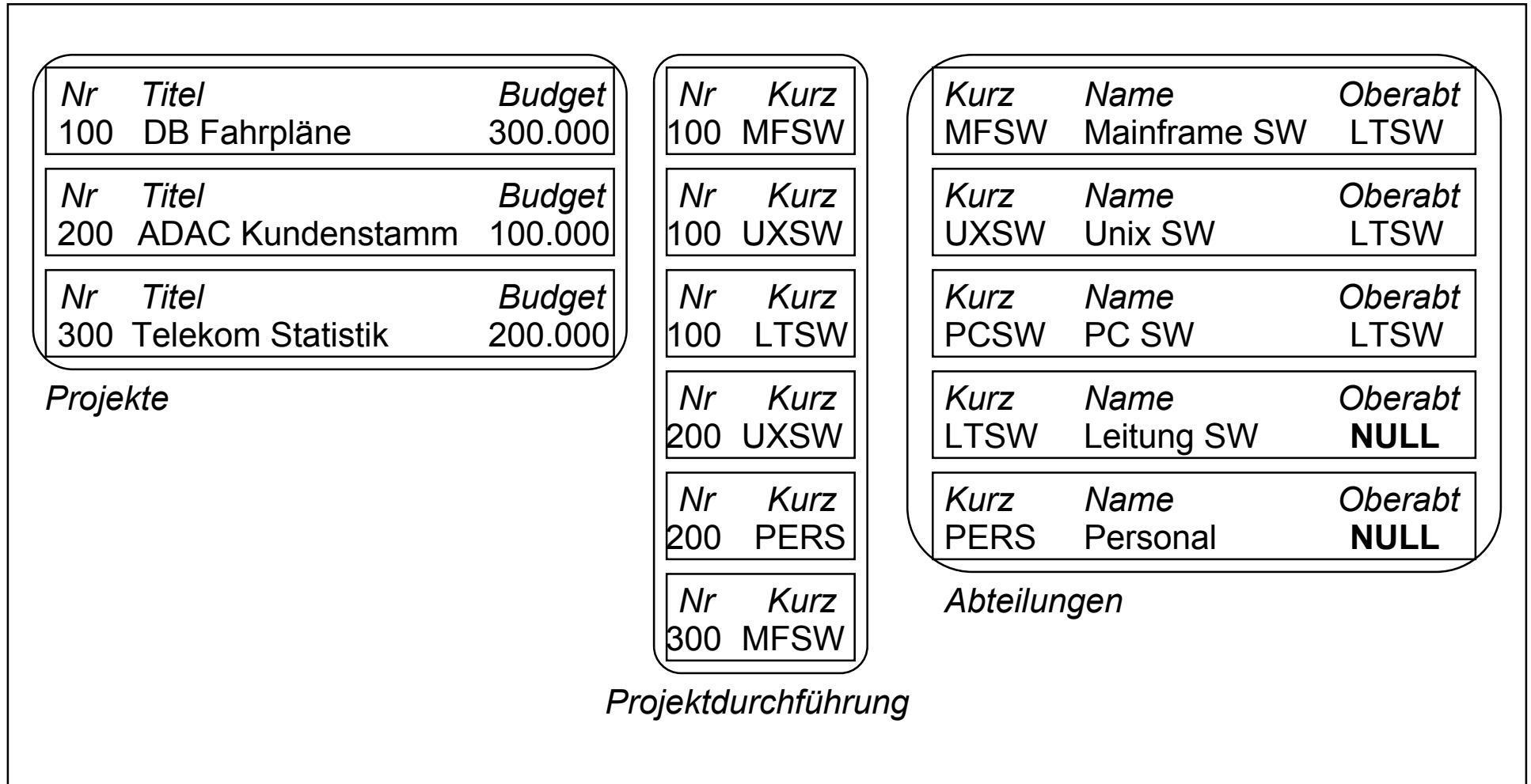
- ❑ Eine Datenbank ist ein Aggregat benannter *Relationen*.
- ❑ Eine Relation ist eine Menge von Elementen
  - deren Struktur durch Attribute definiert,
  - deren Identität durch Schlüssel realisiert und
  - deren Werte durch Domänen kontrolliert werden.
- ❑ Ein Relationenelement ist das Aggregat seiner 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.

# 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



Projektdatenbank

# RDM: Tabellen und Schlüssel (1)

---

Im relationalen Datenmodell unterscheidet man zwei Arten von Schlüsseln:

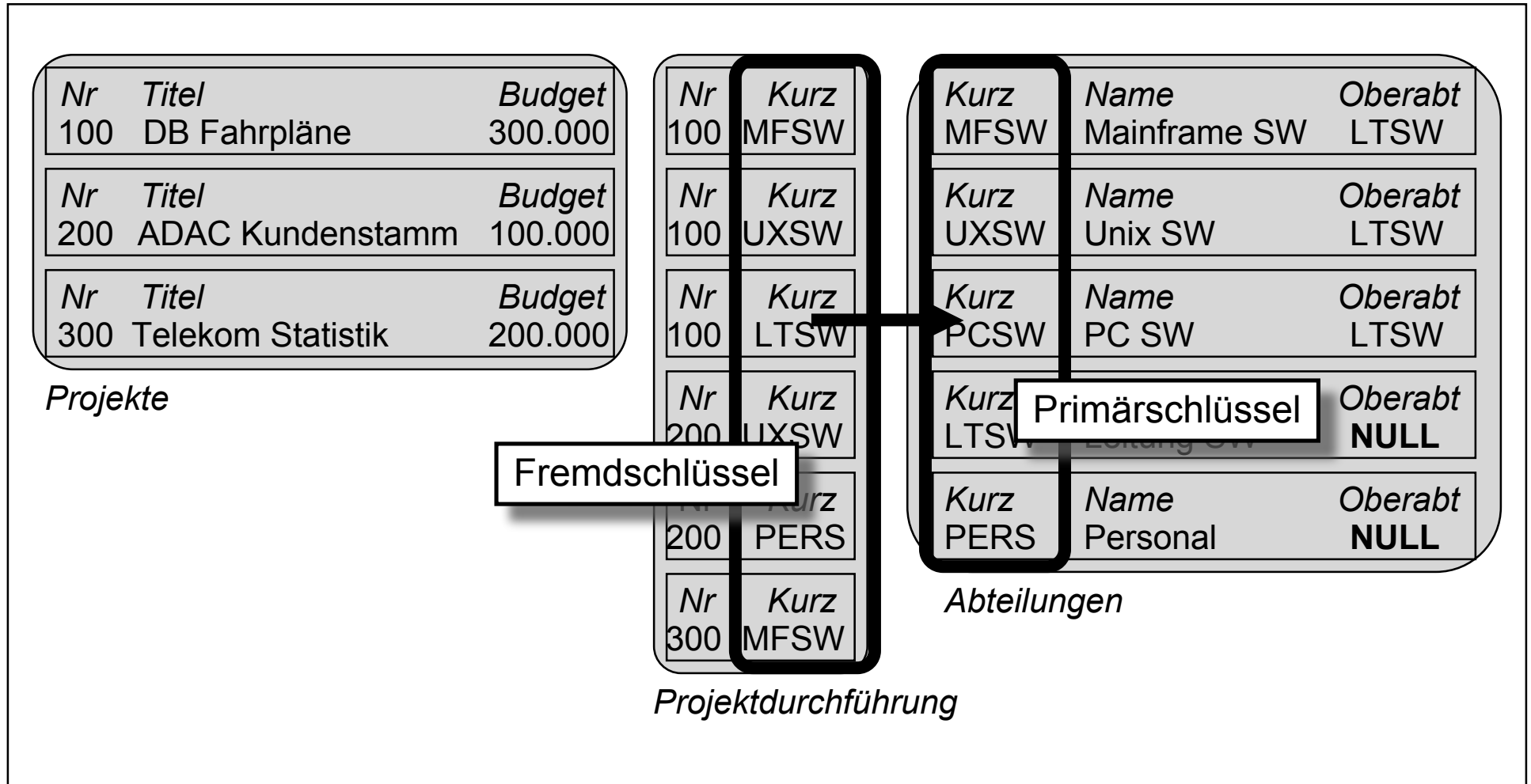
- ❑ **Primärschlüssel:** Ist ein Attribut oder eine minimale Attributkombination, das/die für jede Zeile der Tabelle eindeutige Werte bzw. Wertkombinationen enthält. Es kann mehrere Attribute mit dieser Eigenschaft geben (*Schlüsselkandidaten*).
- ❑ **Fremdschlüssel:** Ist ein Attribut oder Attributkombination,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

Duplikate sind in Tabellen nicht erlaubt  $\rightsquigarrow$  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  $\rightsquigarrow$  (Name, Oberabt), Kurz  $\rightsquigarrow$  Name, Kurz  $\rightsquigarrow$  Oberabt

# RDM: Tabellen und Schlüssel (2)



Projektdatenbank

# 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 Löschoperation, 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,
- spezielle Wertmengen festlegen.

Int

"Yes", "No", "Don't know"

❑ Operationen auf Attributen, wie z.B. der Vergleich zwischen Budget und Nummer, können auf ihre Zulässigkeit überprüft werden.

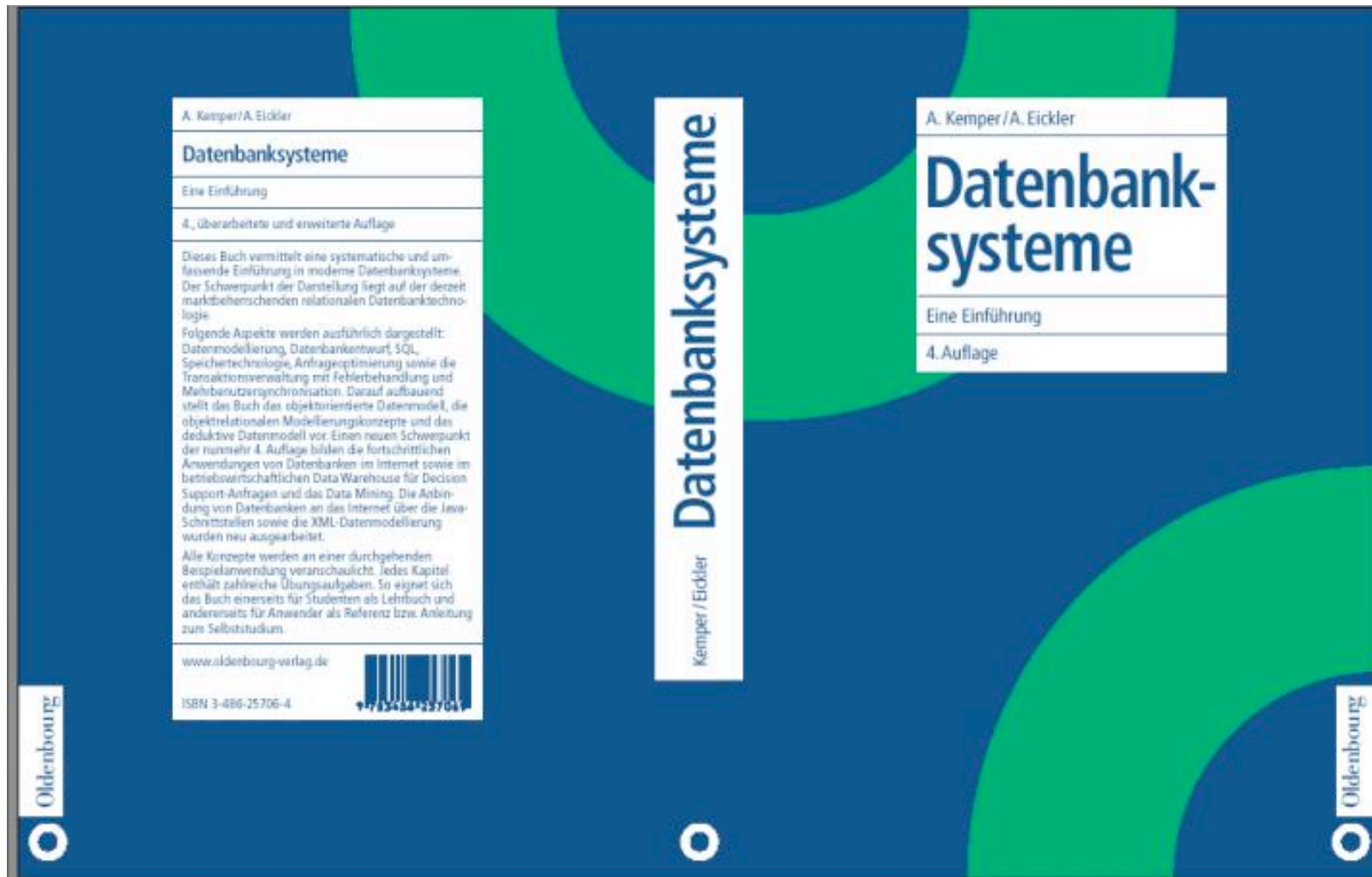
# RDM: Entwurf relationaler Schemata

---

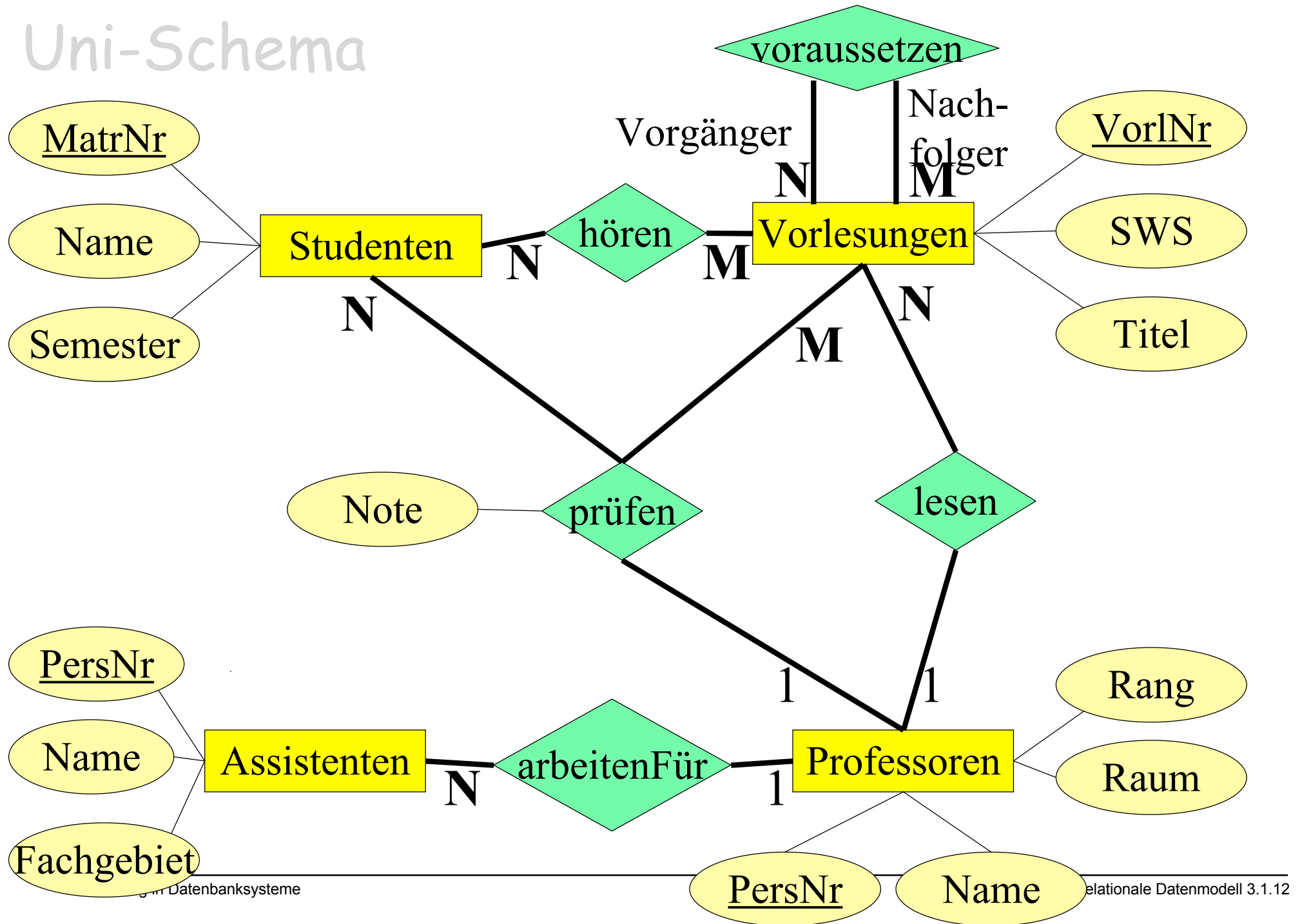
## Zwei alternative Methoden:

- ❑ Entwickle zunächst ein ER-Diagramm, leite daraus ein relationales Schema mit Entitäten- und Beziehungstabellen ab (vgl. C. Batini, S. Ceri, S.B. Navathe. Conceptual Database Design - An Entity Relationship Approach, Benjamin/Cummings, Redwood City, Kalifornien, 1992). bevorzugt
- ❑ 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



# Uni-Schema



# Relationale Darstellung von Entitytypen

---

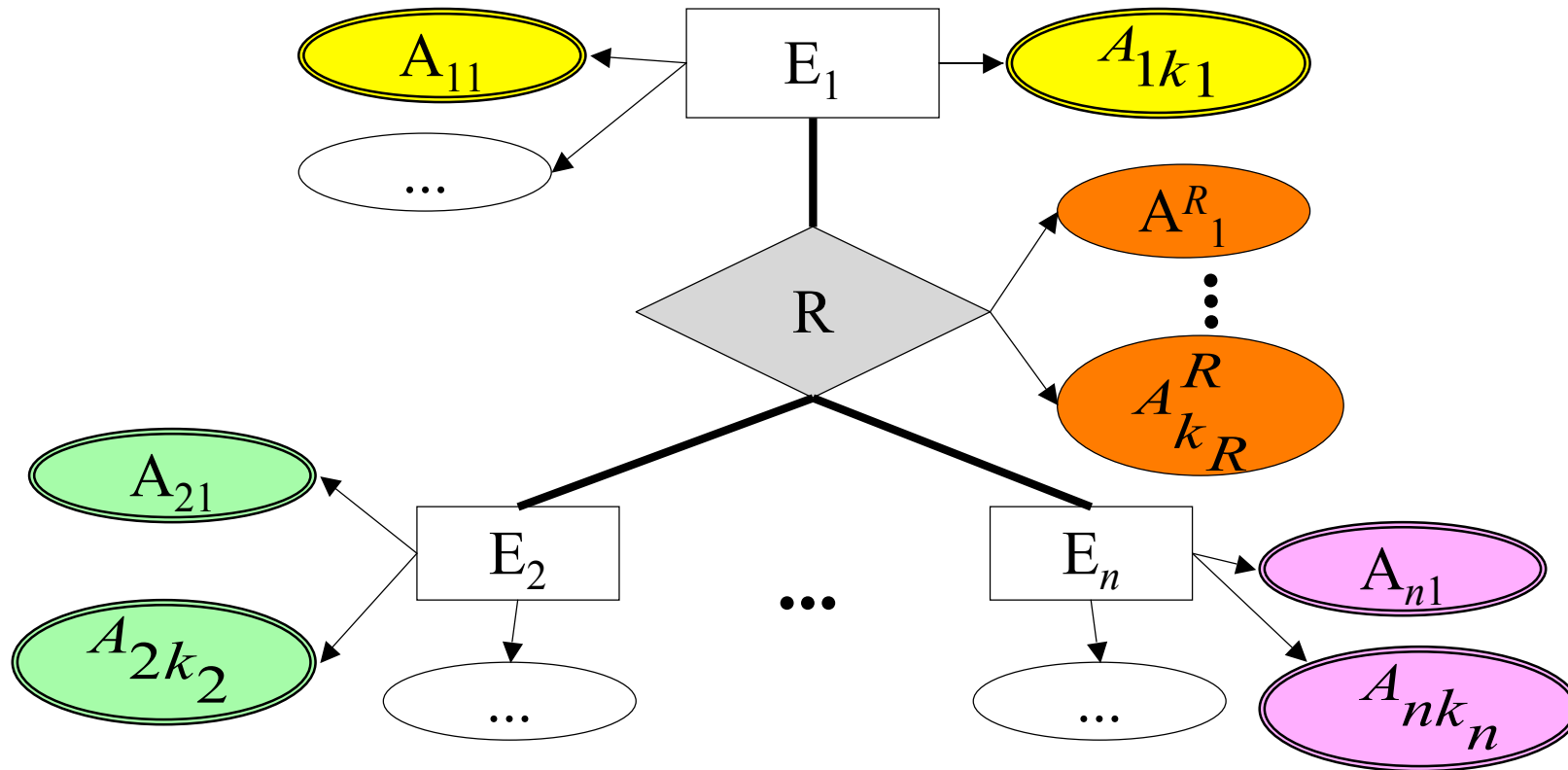
**Studenten:** {[MatrNr:integer, *Name: string*, *Semester: integer*]}

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

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

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

# Relationale Darstellung von Beziehungen



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

# Beziehungen unseres Beispiel-Schemas

---

**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]}

# Schlüssel der Relationen

---

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

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

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

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

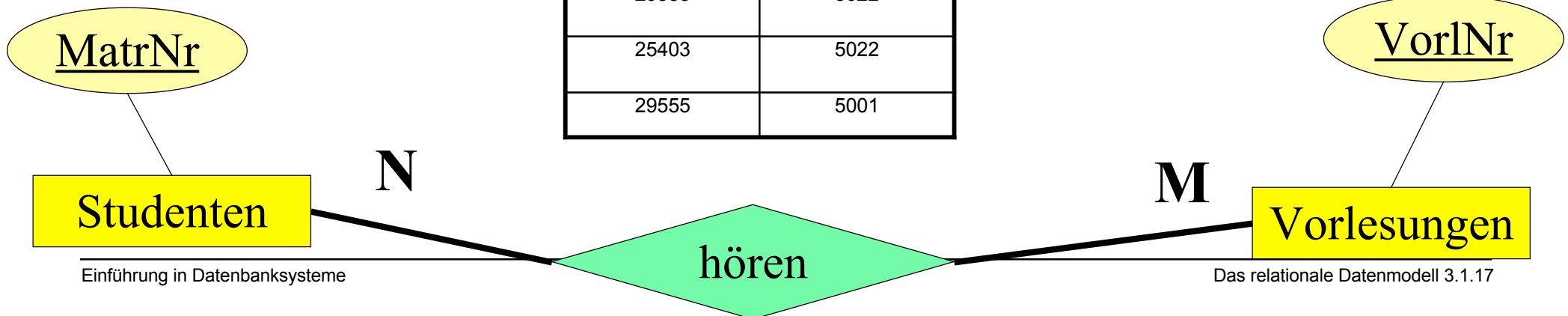
**prüfen** : {[MatrNr: integer, VorlNr: integer, PersNr: integer,  
Note: decimal]}

# Ausprägung der Beziehung *hören*

| Studenten     |     |
|---------------|-----|
| <i>MatrNr</i> | ... |
| 26120         | ... |
| 27550         | ... |
| ...           | ... |

| hören  |        |
|--------|--------|
| MatrNr | VorlNr |
| 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   |     |
|---------------|-----|
| <i>VorlNr</i> | ... |
| 5001          | ... |
| 4052          | ... |
| ...           | ... |



# Notation für Relationenschemata

---

Schema: Tabellename = {[Attr1: Typ1, Attr2: Typ2, ...]}

In eckigen Klammern [...] wird angegeben, wie die Tupel aufgebaut sind.

Die Mengenklammern sollen ausdrücken, daß es sich bei einer Relationenausprägung um eine Menge von Tupeln handelt.

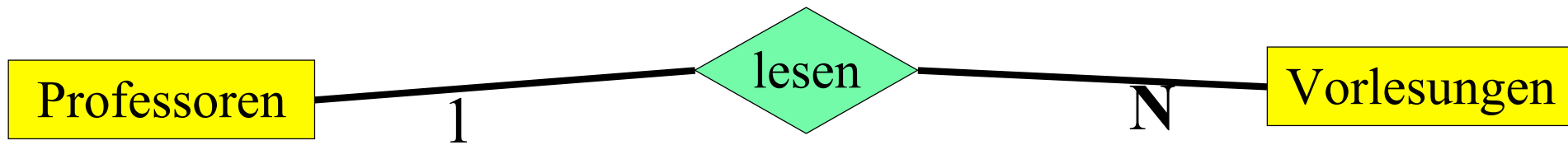
Manchmal werden die Attribute auch als Menge benötigt:

Wir schreiben für das Schema der Tabelle  $\mathcal{R}$ :  $\mathcal{R} = \{\text{Attr1}, \text{Attr2}, \dots\}$ .

Eine konkrete Relation R ist eine Teilmenge des Kreuzproduktes von  $\text{dom}(\text{Attr1}) \times \text{dom}(\text{Attr2}) \times \dots$

# Verfeinerung des relationalen Schemas

---



## 1:N-Beziehung

Initial-Entwurf

*Vorlesungen* : {[VorNr, Titel, SWS]}

*Professoren* : {[PersNr, Name, Rang, Raum]}

*lesen*: {[VorNr, PersNr]}

# Verfeinerung des relationalen Schemas

---

## 1:N-Beziehung

Initial-Entwurf

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

**Professoren** : {[PersNr, Name, Rang, Raum]}

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

Verfeinerung durch Zusammenfassung

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

**Professoren** : {[PersNr, Name, Rang, Raum]}

## Regel

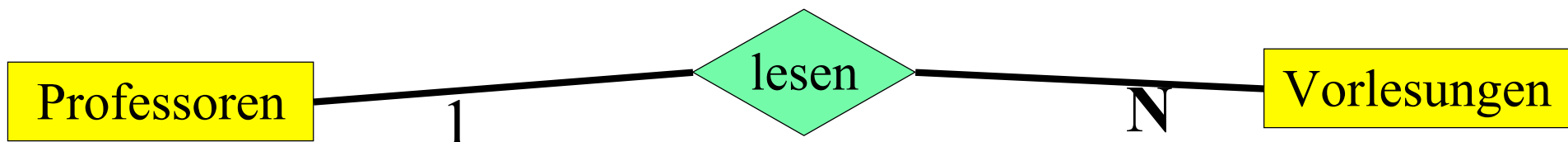
Relationen mit gleichem Schlüssel kann man zusammenfassen

**aber nur diese und keine anderen!**

# 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    |

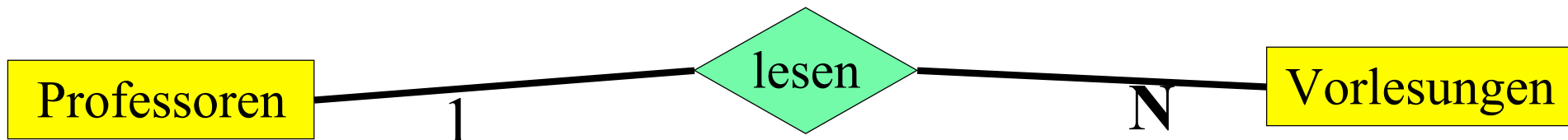
| Vorlesungen |                      |     |             |
|-------------|----------------------|-----|-------------|
| VorlNr      | Titel                | SWS | Gelesen Von |
| 5001        | Grundzüge            | 4   | 2137        |
| 5041        | Ethik                | 4   | 2125        |
| 5043        | 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 |                      |     |
|-------------|----------------------|-----|
| VorINr      | 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   | ??    |
|             |            |      |      |       |

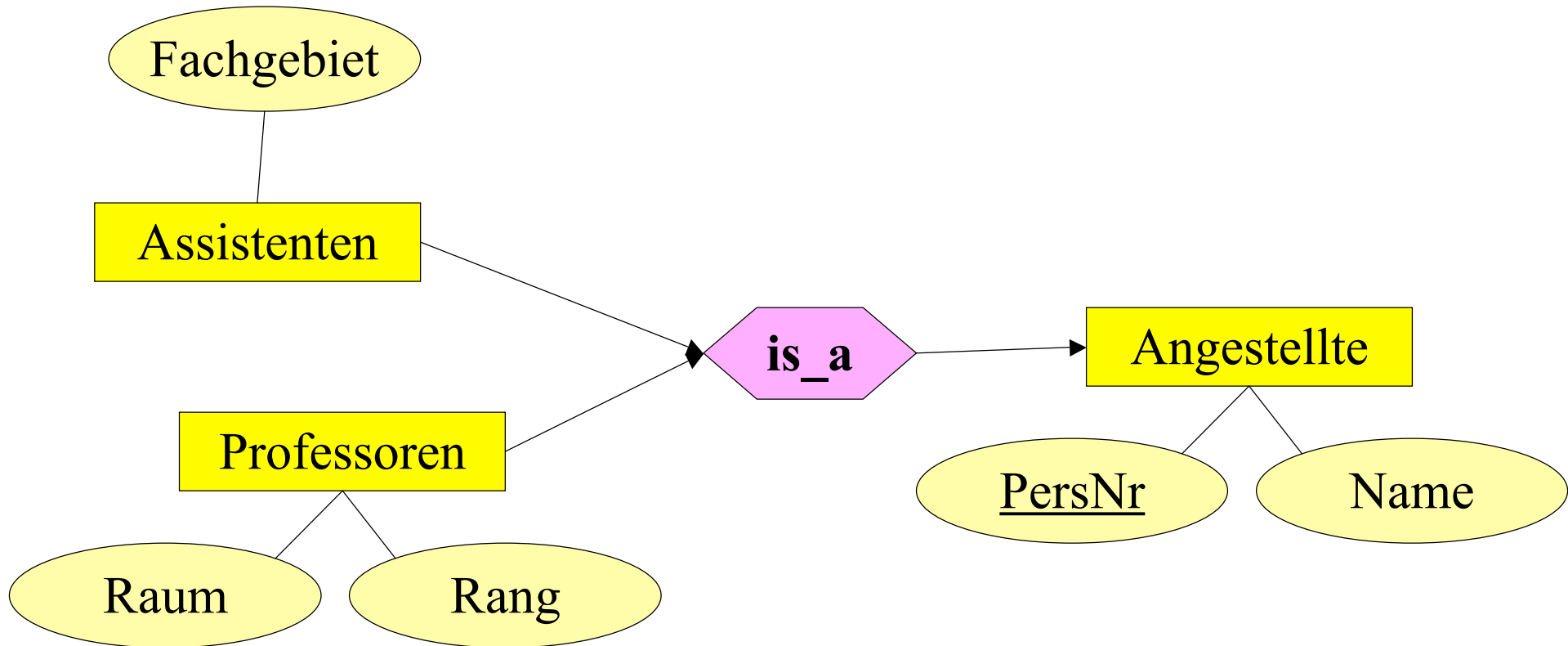
| Vorlesungen |                      |     |
|-------------|----------------------|-----|
| VorlNr      | Titel                | SWS |
| 5001        | Grundzüge            | 4   |
| 5041        | Ethik                | 4   |
| 5043        | Erkenntnistheorie    | 3   |
| 5049        | Mäeutik              | 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]}

# Vereinbarung zur Notation

---

Sei  $\mathcal{R} = \{A, B, C, D\}$  ein Relationenschema.

Seien  $r$  und  $t$  Tupel aus einer konkreten Relation  $R$  gemäß dem Schema  $\mathcal{R}$ .

Sei weiterhin  $\alpha \subseteq \mathcal{R}$ .

Wir vereinbaren:

$r.\alpha = t.\alpha$  soll heißen, daß für alle  $A$  aus  $\alpha$  gilt:  $r.A = t.A$ .

# Funktionale Abhängigkeiten

---

Schema

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

Ausprägung  $R$

Seien  $\alpha \subseteq \mathcal{R}$ ,  $\beta \subseteq \mathcal{R}$

$\alpha \rightarrow \beta$  genau dann wenn  $\forall r, s \in R$  mit  $r.\alpha = s.\alpha \Rightarrow r.\beta = s.\beta$

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

$$\{A\} \rightarrow \{B\}$$

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

*Nicht:*  $\{B\} \rightarrow \{C\}$

*Notationskonvention:*

$$CD \rightarrow B$$

# 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 → Vater,Mutter

Kind,Opa → Oma

Kind,Oma → Opa

# Schlüssel

---

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

$$\square \alpha \rightarrow \mathcal{R}$$

Wir nennen  $\alpha$  Super-Schlüssel, weil noch nichts darüber ausgesagt wird, daß der Schlüssel  $\alpha$  minimal ist.

$\beta$  ist voll funktional abhängig von  $\alpha$  genau dann wenn gilt

$$\square \alpha \rightarrow \beta \text{ und}$$

$$\square \alpha \text{ kann nicht mehr verkleinert werden, d.h.}$$

- $\forall A \in \alpha$  folgt, dass  $(\alpha - \{A\}) \rightarrow \beta$  nicht gilt, oder kürzer
- $\forall A \in \alpha: \neg((\alpha - \{A\}) \rightarrow \beta)$

Notation für volle funktionale Abhängigkeit:  $\alpha \rightarrow^{\bullet} \beta$   
 $\alpha \subseteq \mathcal{R}$  ist ein Kandidaten-Schlüssel, falls folgendes gilt:

$$\square \alpha \rightarrow^{\bullet} \mathcal{R}$$

# 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:

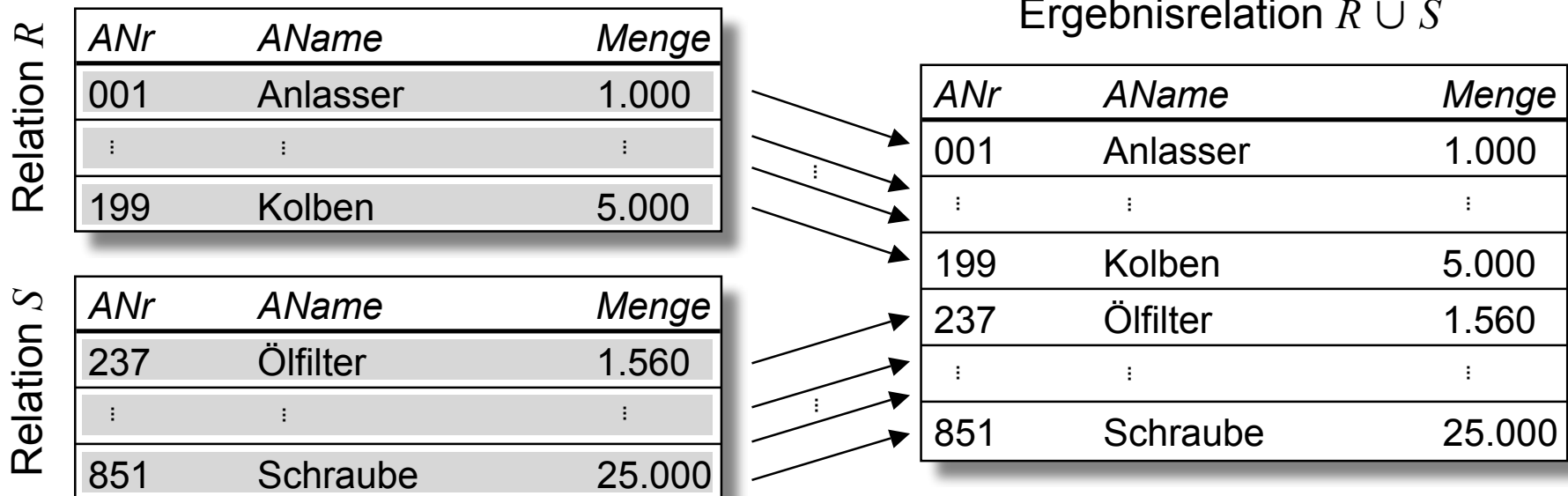
- ❑ {Raum} → {PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung}
- ❑ {PLZ} → {Landesregierung}

# RDM: Relationale Algebra - Anfragen (1)

## Vereinigung $R \cup S$ :

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

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

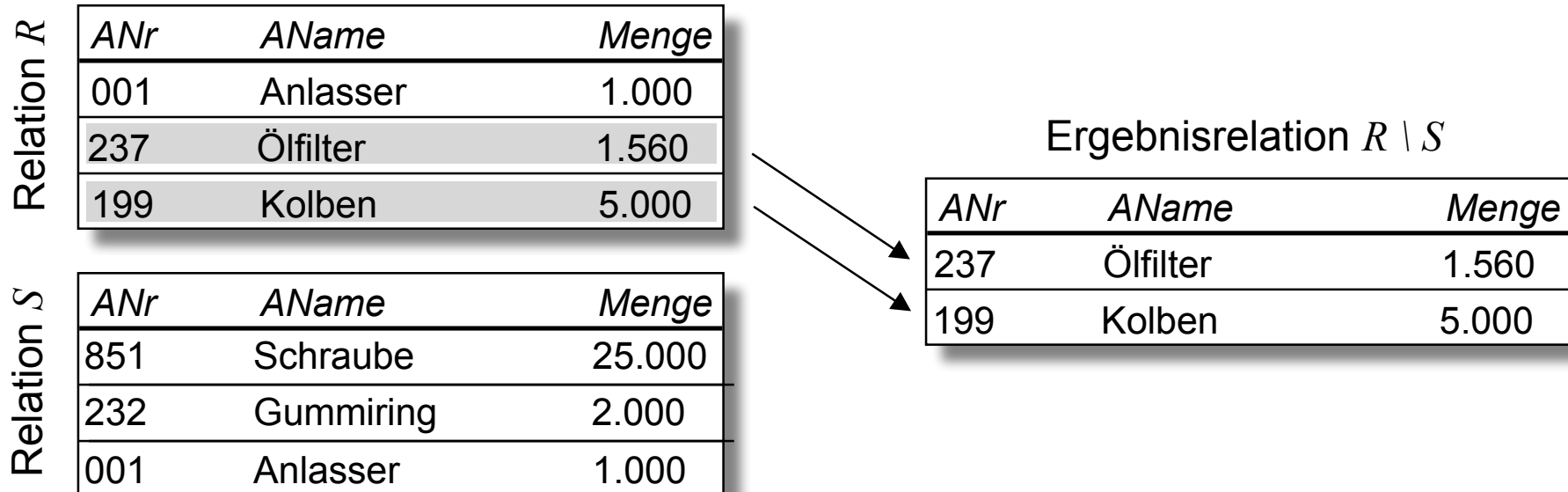


# RDM: Relationale Algebra - Anfragen (2)

## 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.

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

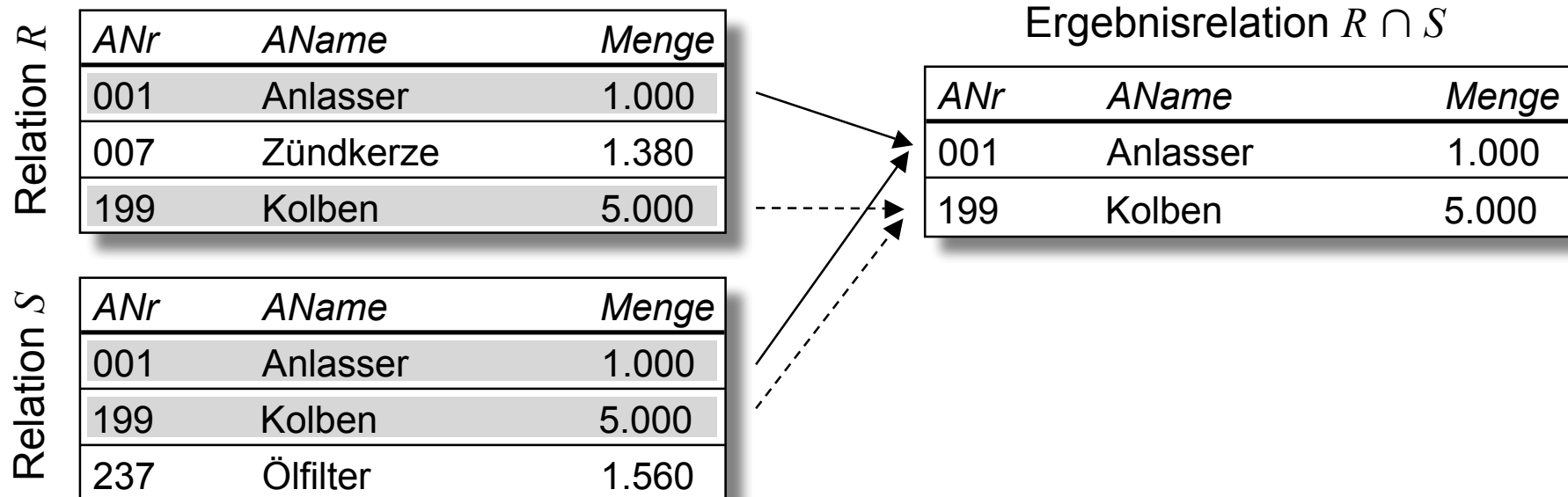


# RDM: Relationale Algebra - Anfragen (3)

## 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.

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



# 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  $R \times S$   $(k \cdot l)$  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.

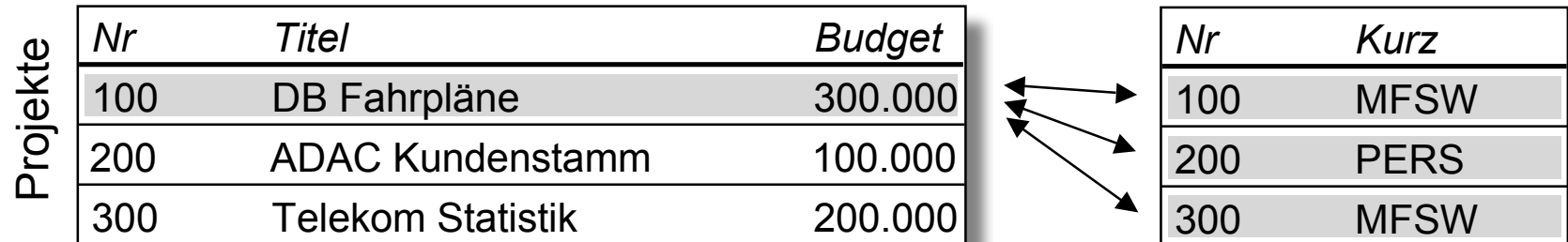
$$R \times S := \{ (r_1, \dots, r_n, s_1, \dots, s_m) \mid (r_1, \dots, r_n) \in R, (s_1, \dots, s_m) \in S \}$$

### ❑ Beispiel:

- Projekte  $\times$  Projektdurchführung (s. nächste Folie)

# RDM: Relationale Algebra - Anfragen (5)

## Beispiel: Projekte × Projektdurchführung



Ergebnisrelation  
Projekte × Projektdurchführung

| <i>Nr</i> | <i>Titel</i>      | <i>Budget</i> | <i>Nr2</i> | <i>Kurz</i> |
|-----------|-------------------|---------------|------------|-------------|
| 100       | DB Fahrpläne      | 300.000       | 100        | MFSW        |
| 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       | 200        | PERS        |
| 200       | ADAC Kundenstamm  | 100.000       | 300        | MFSW        |
| 300       | Telekom Statistik | 200.000       | 100        | MFSW        |
| 300       | Telekom Statistik | 200.000       | 200        | PERS        |
| 300       | Telekom Statistik | 200.000       | 300        | MFSW        |

# RDM: Relationale Algebra - Anfragen (6)

---

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

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

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

### ❑ Beispiele:

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

- ❑ Die Ergebnisrelation enthält die Zeilen des kartesischen Produkts der Relationen R und S, die  $\theta$  erfüllen.

# RDM: Relationale Algebra - Anfragen (7)

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

Projektdurchführung  
(Ausschnitt)

| Projekte | <i>Nr</i> | <i>Titel</i>      | <i>Budget</i> |
|----------|-----------|-------------------|---------------|
|          | 100       | DB Fahrpläne      | 300.000       |
|          | 200       | ADAC Kundenstamm  | 100.000       |
|          | 300       | Telekom Statistik | 200.000       |

| <i>Nr</i> | <i>Kurz</i> |
|-----------|-------------|
| 100       | MFSW        |
| 200       | PERS        |
| 300       | MFSW        |

| Ergebnisrelation | <i>Nr</i> | <i>Titel</i>      | <i>Budget</i> | <i>Nr2</i> | <i>Kurz</i> |
|------------------|-----------|-------------------|---------------|------------|-------------|
|                  |           |                   |               |            |             |
|                  | 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:

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

- In diesem Fall betrachtet  $\theta$  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: **Projekte**  $\bowtie_{(Nr = Nr)}$  **Projektdurchführung**

Projektdurchführung  
(Ausschnitt)

Projekte

| <i>Nr</i> | <i>Titel</i>      | <i>Budget</i> |
|-----------|-------------------|---------------|
| 100       | DB Fahrpläne      | 300.000       |
| 200       | ADAC Kundenstamm  | 100.000       |
| 300       | Telekom Statistik | 200.000       |

| <i>Nr</i> | <i>Kurz</i> |
|-----------|-------------|
| 100       | MFSW        |
| 200       | PERS        |
| 300       | MFSW        |

Ergebnisrelation

| <i>Nr</i> | <i>Titel</i>      | <i>Budget</i> | <i>Nr2</i> | <i>Kurz</i> |
|-----------|-------------------|---------------|------------|-------------|
|           |                   |               |            |             |
| 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)

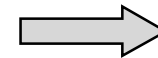
Projektion  $\pi_{(r_{f_1}, \dots, r_{f_n})} (R)$ :

- 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.

$$\pi_{(r_{f_1}, \dots, r_{f_n})}(R) := \{ (r_{f_1}, \dots, r_{f_n}) \mid (r_1, \dots, r_m) \in R \}$$

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

| Projekte | <i>Nr</i> | <i>Titel</i>      | <i>Budget</i> |
|----------|-----------|-------------------|---------------|
|          | 100       | DB Fahrpläne      | 300.000       |
|          | 200       | ADAC Kundenstamm  | 100.000       |
|          | 300       | Telekom Statistik | 200.000       |



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

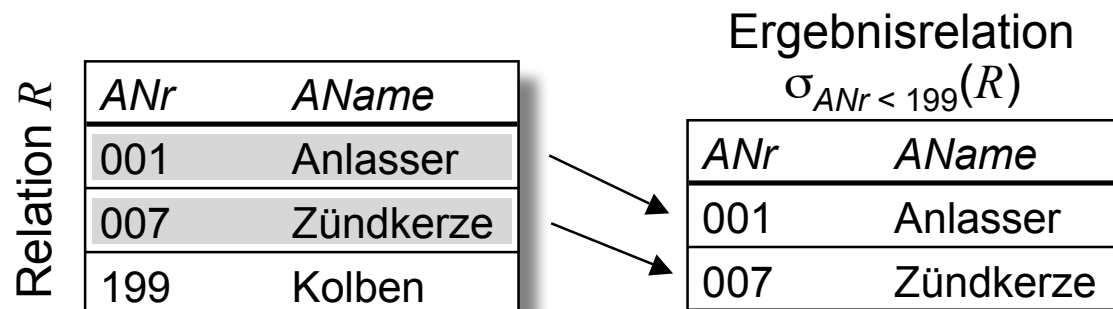
| <i>Nr</i> | <i>Budget</i> |
|-----------|---------------|
| 100       | 300.000       |
| 200       | 100.000       |
| 300       | 200.000       |

# RDM: Relationale Algebra - Anfragen (11)

## Selektion $\sigma_{\theta}(R)$ :

- ❑ Bestimmte Tupel einer Relation werden ausgewählt und in der Ergebnisrelation vereinigt.
- ❑ Zur Auswahl der zu übernehmenden Tupel dient das Prädikat  $\theta : R \rightsquigarrow \{ true, false \}$ , 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.

$$\sigma_{\theta}(R) := \{ r \in R \mid \theta(r) \}$$



# RDM: Relationale Algebra: Zusammenfassung

---

## Vorteil:

- ❑ Orthogonale Kombination der Konstrukte
- ❑ Einfache, mathematische Behandlung, z.B.  $(R \times S) \times T = R \times (S \times T)$
- ❑ Einfache (naive) Implementierung möglich
- ❑ Optimierung möglich

## Nachteile:

- ❑ Eingeschränkte Ausdrucksmächtigkeit auf Relationenebene (Summe, Mittelwert, Kardinalität)
- ❑ Reine Anfragesprache
- ❑ Optimierung nicht trivial