

## Kapitel 3: Das relationale DB-Modell & SQL

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

Datenbanken und Informationssysteme

Das relationale Datenmodell 3.1.1

### 3.1 RDM: Überblick über die Konzepte (1)

- Eine Datenbank ist ein Aggregat benannter *Relationen*.
- Eine Relationen 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.

Datenbanken und Informationssysteme

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).
- Ein Primärschlüssel, der in Relation A zur Identifikation eines Tupels in Relation B benutzt wird, bezeichnet man als *Fremdschlüssel*.

## RDM: Projektdatenbank

<table border="1"> <thead> <tr> <th>Nr</th> <th>Titel</th> <th>Budget</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>DB Fahrpläne</td> <td>300.000</td> </tr> </tbody> </table>	Nr	Titel	Budget	100	DB Fahrpläne	300.000	<table border="1"> <thead> <tr> <th>Nr</th> <th>Kurz</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>MFSW</td> </tr> </tbody> </table>	Nr	Kurz	100	MFSW	<table border="1"> <thead> <tr> <th>Kurz</th> <th>Name</th> <th>Oberabt</th> </tr> </thead> <tbody> <tr> <td>MFSW</td> <td>Mainframe SW</td> <td>LTSW</td> </tr> </tbody> </table>	Kurz	Name	Oberabt	MFSW	Mainframe SW	LTSW												
Nr	Titel	Budget																												
100	DB Fahrpläne	300.000																												
Nr	Kurz																													
100	MFSW																													
Kurz	Name	Oberabt																												
MFSW	Mainframe SW	LTSW																												
<table border="1"> <thead> <tr> <th>Nr</th> <th>Titel</th> <th>Budget</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>ADAC Kundenstamm</td> <td>100.000</td> </tr> </tbody> </table>	Nr	Titel	Budget	200	ADAC Kundenstamm	100.000	<table border="1"> <thead> <tr> <th>Nr</th> <th>Kurz</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>UXSW</td> </tr> </tbody> </table>	Nr	Kurz	100	UXSW	<table border="1"> <thead> <tr> <th>Kurz</th> <th>Name</th> <th>Oberabt</th> </tr> </thead> <tbody> <tr> <td>UXSW</td> <td>Unix SW</td> <td>LTSW</td> </tr> </tbody> </table>	Kurz	Name	Oberabt	UXSW	Unix SW	LTSW												
Nr	Titel	Budget																												
200	ADAC Kundenstamm	100.000																												
Nr	Kurz																													
100	UXSW																													
Kurz	Name	Oberabt																												
UXSW	Unix SW	LTSW																												
<table border="1"> <thead> <tr> <th>Nr</th> <th>Titel</th> <th>Budget</th> </tr> </thead> <tbody> <tr> <td>300</td> <td>Telekom Statistik</td> <td>200.000</td> </tr> </tbody> </table> <p><i>Projekte</i></p>	Nr	Titel	Budget	300	Telekom Statistik	200.000	<table border="1"> <thead> <tr> <th>Nr</th> <th>Kurz</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>LTSW</td> </tr> <tr> <td>200</td> <td>UXSW</td> </tr> <tr> <td>200</td> <td>PERS</td> </tr> <tr> <td>300</td> <td>MFSW</td> </tr> </tbody> </table> <p><i>Projektdurchführung</i></p>	Nr	Kurz	100	LTSW	200	UXSW	200	PERS	300	MFSW	<table border="1"> <thead> <tr> <th>Kurz</th> <th>Name</th> <th>Oberabt</th> </tr> </thead> <tbody> <tr> <td>PCSW</td> <td>PC SW</td> <td>LTSW</td> </tr> <tr> <td>LTSW</td> <td>Leitung SW</td> <td>NULL</td> </tr> <tr> <td>PERS</td> <td>Personal</td> <td>NULL</td> </tr> </tbody> </table> <p><i>Abteilungen</i></p>	Kurz	Name	Oberabt	PCSW	PC SW	LTSW	LTSW	Leitung SW	NULL	PERS	Personal	NULL
Nr	Titel	Budget																												
300	Telekom Statistik	200.000																												
Nr	Kurz																													
100	LTSW																													
200	UXSW																													
200	PERS																													
300	MFSW																													
Kurz	Name	Oberabt																												
PCSW	PC SW	LTSW																												
LTSW	Leitung SW	NULL																												
PERS	Personal	NULL																												

Projektdatenbank

## RDM: Tabellen und Schlüssel (1)

Im relationalen Datenmodell unterscheidet man zwei Arten von Schlüsseln:

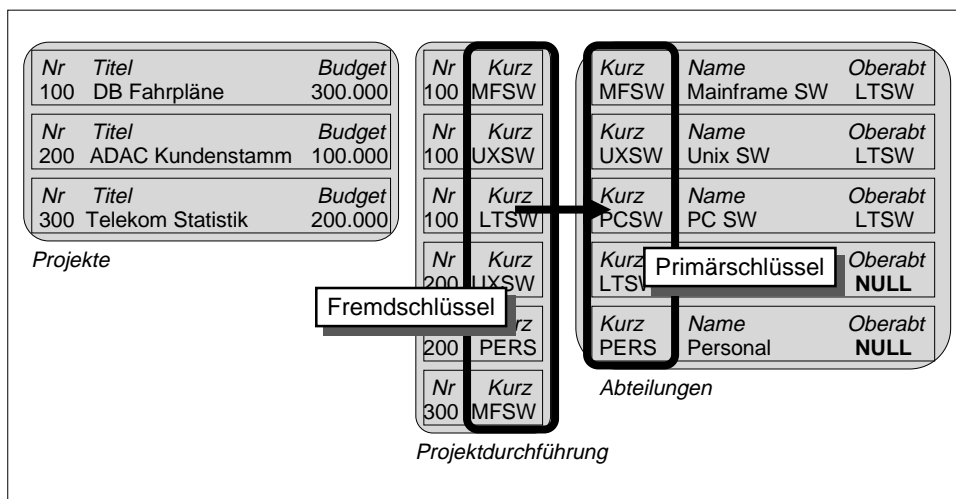
- ❑ **Primärschlüssel:** Ist ein Attribut oder eine minimale Attributkombination, das/die für jede Zeile der Tabelle eindeutige Werte bzw. Wertkombinationen enthält. Es kann mehrere Attribute mit dieser Eigenschaft geben (*Schlüsselkandidaten*).
- ❑ **Fremdschlüssel:** Ist ein Attribut oder Attributkombination,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

Duplikate sind in Tabellen nicht erlaubt → 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 → (Name, Oberabt), Kurz → Name, Kurz → 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 referenzierte 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.

Das Domänenkonzept ist in den meisten RDBMS nicht vollständig realisiert!

## 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).

## RDM: Normalisierung (1)

Offensichtlich ist folgende Tabellenstruktur für die Projektdatenbank unvorteilhaft (Entitäten werden redundant gespeichert).

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

*Projektdurchführung*

**Problem:** Nicht-Schlüsselattribute funktional abhängig von Teilen des Primärschlüssels

## RDM: Normalisierung (2)

Bei dieser Modellierung treten folgende Probleme auf:

- Änderungsanomalie**
- Löschanomalie**
- Einfügeanomalie**

da z.B. die Änderung des Budget des Projektes 100 die Änderung von 3 Tupeln erfordert.

**Beobachtung:** Im Ausschnitt der Realität (FIS) bestehen folgende funktionale Abhängigkeiten zwischen Attributen:  
 Nr → Titel  
 Nr → Budget

Redundanzvermeidung durch Normalisierungsregeln: Definiere Aufspaltung von Tabellen basierend auf Kenntnis der funktionalen Abhängigkeiten.

Hier: Auslagerung der Spalten *Titel* und *Budget* mit Primärschlüssel *Nr* in eine separate Tabelle *Projekte*.

## RDM: Normalformen

### Definitionen:

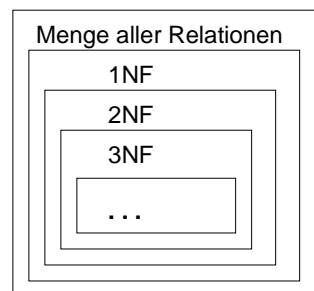
**1. NF:** Eine Relation ist in 1NF genau dann, wenn die Domänen aller Attribute elementar sind (keine geschachtelten Tabellen, Relationen, ...).

**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 von Schlüsselkandidaten abhängt.

...

(vgl. S.M. Lang, P.C. Lockemann. Datenbankeinsatz. Springer, Berlin u.a., 1995)



Abhängigkeitsbewahrende Zerlegung möglich

## RDM: NF2-Erweiterung

Eine Erweiterung des RDM um mengenwertige Attribute führt zu NF<sup>2</sup>-Tabellen (Non First Normal Form-Tabellen, NFNF-Tabellen).

<i>Kurz</i>	<i>Name</i>	<i>Oberabt</i>	<i>Projekte</i>												
MFSW	Mainframe SW	LTSW	<table border="1"> <tr> <td><i>Nr</i></td> <td><i>Titel</i></td> <td><i>Budget</i></td> </tr> <tr> <td>100</td> <td>DB Fahrpläne</td> <td>300.000</td> </tr> <tr> <td><i>Nr</i></td> <td><i>Titel</i></td> <td><i>Budget</i></td> </tr> <tr> <td>300</td> <td>Telekom Statistik</td> <td>200.000</td> </tr> </table>	<i>Nr</i>	<i>Titel</i>	<i>Budget</i>	100	DB Fahrpläne	300.000	<i>Nr</i>	<i>Titel</i>	<i>Budget</i>	300	Telekom Statistik	200.000
<i>Nr</i>	<i>Titel</i>	<i>Budget</i>													
100	DB Fahrpläne	300.000													
<i>Nr</i>	<i>Titel</i>	<i>Budget</i>													
300	Telekom Statistik	200.000													
...	...	...	...												
PCSW	PC SW	LTSW	<table border="1"> <tr> <td><i>Nr</i></td> <td><i>Titel</i></td> <td><i>Budget</i></td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </table>	<i>Nr</i>	<i>Titel</i>	<i>Budget</i>									
<i>Nr</i>	<i>Titel</i>	<i>Budget</i>													
UXSW	Unix SW	LTSW	<table border="1"> <tr> <td><i>Nr</i></td> <td><i>Titel</i></td> <td><i>Budget</i></td> </tr> <tr> <td>100</td> <td>DB Fahrpläne</td> <td>300.000</td> </tr> <tr> <td><i>Nr</i></td> <td><i>Titel</i></td> <td><i>Budget</i></td> </tr> <tr> <td>200</td> <td>ADAC Kundenstamm</td> <td>100.000</td> </tr> </table>	<i>Nr</i>	<i>Titel</i>	<i>Budget</i>	100	DB Fahrpläne	300.000	<i>Nr</i>	<i>Titel</i>	<i>Budget</i>	200	ADAC Kundenstamm	100.000
<i>Nr</i>	<i>Titel</i>	<i>Budget</i>													
100	DB Fahrpläne	300.000													
<i>Nr</i>	<i>Titel</i>	<i>Budget</i>													
200	ADAC Kundenstamm	100.000													
...	...	...	...												

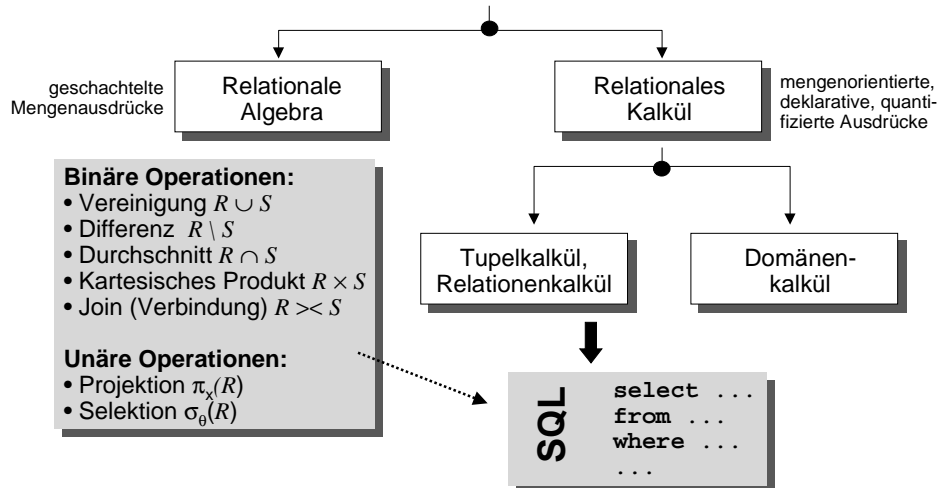
Abteilungen2

Datenbanken und Informationssysteme

Das relationale Datenmodell 3.1.15

## RDM: Anfragen

Relationale Anfragesprachen im Überblick:



Datenbanken und Informationssysteme

Das relationale Datenmodell 3.1.16

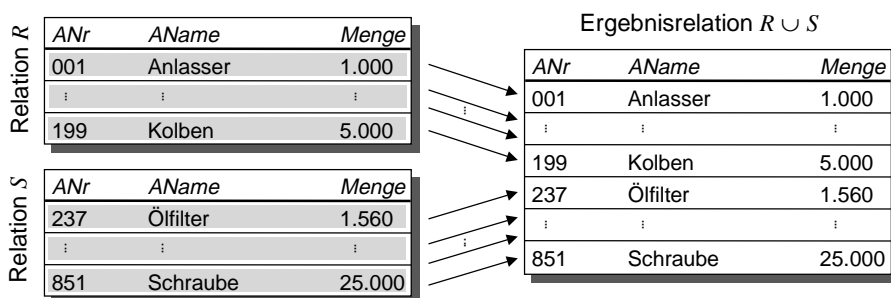


## RDM: Relationale Algebra - Anfragen (1)

### Vereinigung $R \cup S$ :

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

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

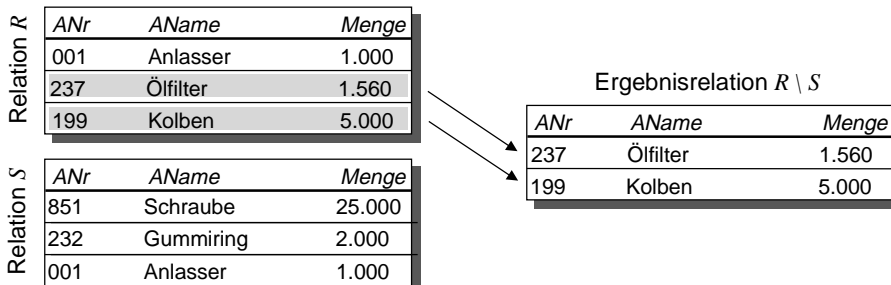


## RDM: Relationale Algebra - Anfragen (2)

### 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.

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

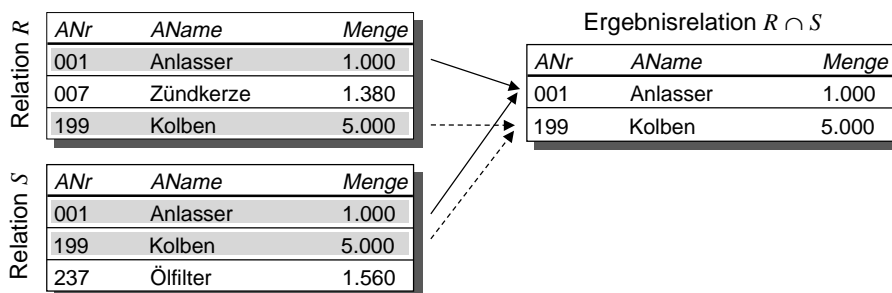


## RDM: Relationale Algebra - Anfragen (3)

### 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.

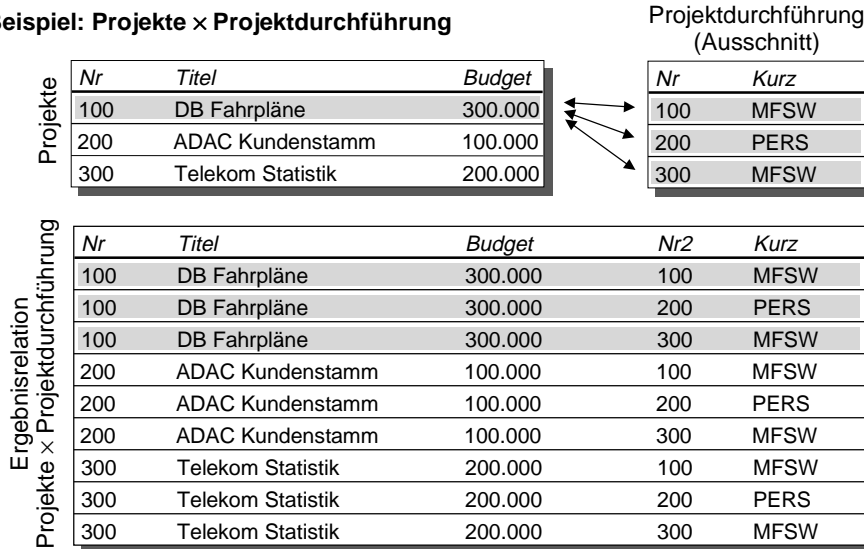
$$R \times S := \{ (r_1, \dots, r_n, s_1, \dots, s_m) \mid (r_1, \dots, r_n) \in R, (s_1, \dots, s_m) \in S \}$$

#### Beispiel:

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

## RDM: Relationale Algebra - Anfragen (5)

Beispiel: Projekte × Projektdurchführung



## 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 gemäß des Prädikats  $\theta$  hergestellt.
- Im allgemeinen Fall (*Theta-Join*) vergleicht das Prädikat  $\theta$  Attribute aus den Relationen R und S.

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

□ Beispiele:

- $Projekte \bowtie_{(Nr \neq Nr_2)} Projektdurchführung$  (s. nächste Folie)
- $Projekte \bowtie_{(Budget > Nr) \wedge (Titel \neq Kurz)} Projektdurchführung$

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

## RDM: Relationale Algebra - Anfragen (7)

Beispiel:  $\text{Projekte} \times_{(Nr \neq Nr)} \text{Projektdurchf\u00fchrung}$

Projekte	Nr	Titel	Budget
	100	DB Fahrpl\u00e4ne	300.000
	200	ADAC Kundenstamm	100.000
	300	Telekom Statistik	200.000

Projektdurchf\u00fchrung  
(Ausschnitt)

Nr	Kurz
100	MFSW
200	PERS
300	MFSW

Ergebnisrelation	Nr	Titel	Budget	Nr2	Kurz
	100	DB Fahrpl\u00e4ne	300.000	200	PERS
	100	DB Fahrpl\u00e4ne	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 Verkn\u00fcpfung von Tabellen \u00fcber ihre Fremdschl\u00fcsselwerte erlaubt.

- Beispiel:

- $\text{Projekte} \times \times \text{Projektdurchf\u00fchrung} := \text{Projekt} \times \times_{Nr = Nr} \text{Projektdurchf\u00fchrung}$

- In diesem Fall betrachtet  $\theta$  nur die Gleichheit zwischen Fremdschl\u00fcssel und Prim\u00e4rschl\u00fcssel, 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)

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

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

→

Ergebnisrelation $\pi_{(Nr, Budget)}$ (Projekte)	
Nr	Budget
100	300.000
200	100.000
300	200.000

## RDM: Relationale Algebra - Anfragen (10)

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

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

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

Relation R	ANr	AName
	001	Anlasser
	007	Zündkerze
	199	Kolben

→

Ergebnisrelation $\sigma_{ANr < 199}(R)$	
ANr	AName
001	Anlasser
007	Zündkerze

## RDM: Relationale Algebra: Zusammenfassung

---

### Vorteil:

- Einfache, mathematische Behandlung, z.B.  $(R \times S) \times T = R \times (S \times T)$

### Nachteile:

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

## RDM: Anfragen im relationalen Kalkül (1)

---

### Grundlage: Quantifizierte Formeln (Boolesches Ergebnis)

- Basis:** Jede Bedingung  $\theta$ , mit  $\theta \in \{=, \neq, <, \leq, >, \geq\}$ , die zwei Tupelkomponenten verknüpft, ist eine Formel.

Beispiel:  $x.a = y.b, x.a > y.b, \dots$

- Klammerung und Negation:** Falls  $f$  eine Formel ist, so sind dies auch  $(f)$  und  $\neg(f)$ .
- Boolesche Operationen:** Falls  $f$  und  $g$  Formeln sind, so sind auch  $f \wedge g$  und  $f \vee g$  Formeln.
- Quantoren:** Falls  $f$  Formel ist und  $x$  als freie (Tupel-)Variable enthält, so sind  $\exists x(f)$  und  $\forall x(f)$  Formeln, z.B.  $\exists x(x.a = 3)$ .
- Abschluß:** Genau die durch die vorigen Vorschriften erzeugbaren Ausdrücke sind Formeln.

Ein relationaler Kalkülausdruck (z.B. `select x.a, y.b ... where f`) mit  $x, y$  als freie Tupelvariablen in  $f$  liefert ein relationenwertiges Ergebnis, nämlich alle Tupel mit den Komponenten  $x.a, y.b, \dots$ , die für  $f$  den Wahrheitswert `true` liefern (Relationale Quantoren). Im relationalen Kalkülausdruck sind alle Tupelvariablen gebunden.

## RDM: Anfragen im relationalen Kalkül (2)

Im **Tupelkalkül** müssen die Tupelvariablen  $x, y, \dots$  der logischen und relationalen Quantoren explizit an Relationen gebunden werden.

Beispiel:

```
x in Projekte;

select x.Nr, x.Budget
where x.Budget > 100000
```

Projekte

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

Ergebnisrelation

Nr	Budget
100	300.000
300	200.000

## RDM: Anfragen im relationalen Kalkül (3)

Im Domänenkalkül beziehen sich die verwendeten Variablen  $x, y, \dots$  nicht auf die existierenden Tupel einer Relation, sondern auf die Wertebereiche ( $\rightarrow$  *Domänen*) von Attributen.

Beispiel:

```
x as int, y as float;

select x, y
where Projekte(Nr: x, Budget: y) ^ (y > 250000)
```

Projekte

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

Nr	Budget
100	300.000

## RDM: Anfragen in SQL (1)

### Die Anfragesprache SQL:

Iterationsabstraktion mit Hilfe des **select from** **where**-Konstrukts:

- ❑ **select**-Klausel: Spezifikation der Projektionsliste für die Ergebnistabelle
- ❑ **from**-Klausel: Festlegung der angefragten Tabellen
- ❑ **where**-Klausel: Selektionsprädikat, mit dessen Hilfe die Ergebnistupel aus dem kartesischen Produkt der beteiligten Tabellen selektiert werden

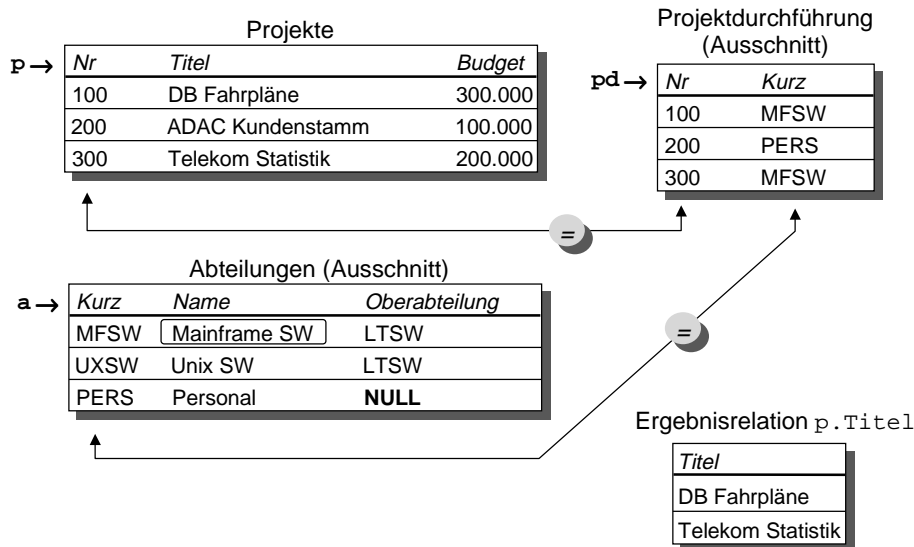
**Join:** Mehrere Tabellen werden wertbasiert, z.B. über gleiche Werte in zusammengehörigen Primärschlüssel/ Fremdschlüssel-Paaren, miteinander verknüpft.

Bestimmung der Projekttitel, an denen die Abteilung für *Mainframe Software* arbeitet:

```
select p.Titel
from Projekte p,
     Projektdurchfuehrung pd,
     Abteilungen a
where p.Nr = pd.Nr
and a.Kurz = pd.Kurz
and a.Name = 'Mainframe SW';
```

hier: *Join* über die Tabellen *Projekte*, *Abteilungen* und *Projektdurchführung* mit Selektion und Projektion

## RDM: Anfragen in SQL (2)





## RDM: Anfragen in SQL (3)

---

### **Bewertung:**

- Deklarative Formulierung der Anfrage, Auswertungsreihenfolge un spezifiziert
- Tupelkalkül: Die Variablen (hier:  $p$ ,  $pd$  und  $a$ ) sind explizit an die Tupel der Relationen gebunden.
- Zusätzlich erlaubt SQL einige Algebra-Operationen (`union`).

## RDM: Aktualisierungsoperationen (1)

---

### **Klassen von SQL-Modifikationsoperationen auf Datenbanken:**

Operationen auf Daten, die sowohl Entitäten der realen Welt als auch Beziehungen zwischen solchen Objekten repräsentieren:

- Operationen zum Einfügen neuer Daten (`insert`-Befehl)
- Operationen zum Ändern von Daten (`update`-Befehl)
- Operationen zum Löschen von Daten (`delete`-Befehl)

## RDM: Aktualisierungsoperationen (2)

---

### Generische Operationen:

#### insert-Statement:

- Fügt ein einziges Tupel ein, dessen Attributwerte als Parameter übergeben werden.
- Fügt eine Ergebnistabelle ein.

```
insert into Projektdurchfuehrung
(400, 'XYZA')
```

```
insert into Projektdurchfuehrung
(Nr, Kurz)
select p.Nr, a.Kurz
from Projekte p, Abteilungen a
where p.Titel = 'Telekom Statistik'
and a.Name = 'Unix SW'
```

#### update-Statement:

- Selektion (des) der betreffenden Tupel(s)
- Neue Werte oder Formeln für zu ändernde Attribute

```
update Projekte
set Budget = Budget * 1.5
where Budget > 150000
```

## RDM: Aktualisierungsoperationen (3)

---

#### delete-Statement:

- Selektion (des) der betreffenden Tupel(s)

```
delete
from Projektdurchfuehrung
where Kurz = 'MFSW';
```

## RDM: Bewertung (1)

---

### Positive Beiträge des relationalen Datenmodells:

- Es bietet einen Satz von im Modell implizit enthaltenen generischen Operatoren (modellspezifisch, nicht anwendungsabhängig).
- Algebraische Sichtweise: Selektion, Projektion, natürlicher Verbund, Mengenoperationen, Umbenennung
- Positive Eigenschaften der Operatoren:
  - deskriptive Anfragesprache (→ Datenmodell, Datenunabhängigkeit)
  - Abgeschlossenheit (→ Orthogonalität der Operatoren)
  - Optimierbarkeit
  - Sicherheit: Jeder syntaktisch korrekte Ausdruck liefert in endlicher Zeit ein endliches Ergebnis.

## RDM: Bewertung (2)

---

### Beschränkungen des relationalen Datenmodells:

- Beschränkungen der Datenstrukturen
  - primitive vordefinierte Domänen (keine ADTs)
  - "flache" Tupel
  - atomare Attribute
  - homogene, ungeordnete Mengen (keine Listen oder Gruppen)
  - keine Unterstützung für Generalisierung und Spezialisierung
- Beschränkung der Identifikationsmechanismen
  - benutzerdefinierte, wertbasierte Schlüsselwerte
  - keine Garantie der referentiellen Integrität

## RDM: Bewertung (3)

---

- ❑ Beschränkung der Anfragesprache und Datenmanipulationssprache
  - vordefinierte Operatoren für atomare Attribute
  - vordefinierte (generische) Operatoren für Mengen
  - keine algorithmische Vollständigkeit (*Turing completeness*)