

# Datenbankmanagement

Theorie 4: Relationale Algebra

**Prof. Dr. Gregor Hülken**

1. Einführung und Überblick

2. Modellierung

3. Normalisierung

**4. Relationale Algebra**

5. Lookup etc. in der Praxis

6. SQL – Data Definition Language

7. SQL – Data Manipulation Language

8. SQL – Trigger

9. SQL – Funktionen / Prozeduren

10. SQL – Datenschutz

11. Transaktionen

## Inhalte

✓ Relationale Algebra

Relationale Integritätsregeln

Klassische Mengenoperationen

Spezielle Relationenoperationen

## Relationale Algebra

- Theoretische Grundlage relationaler Datenbanken
- Mit Hilfe der relationalen Algebra können Anfragen an die Datenbank formuliert werden

### ➤ Besteht aus:

- Relationalen Objekten
- Integritätsregeln
- Operationen
  - Klassische Mengenoperationen
  - Spezielle Relationenoperationen

## Relationale Objekte

Englische Bezeichnung	Deutsche Bezeichnung
Relation	Tabelle
Attribut	Spalte
Tupel	Datensatz, Zeile in der Tabelle
Domain	Wertebereich
Degree	Anzahl der Spalten
Candidate-Key	eindeutiger Schlüssel
Primary-Key	Hauptschlüssel / Primärschlüssel
Alternate-key	Zweitschlüssel / Sekundärschlüssel
Foreign-key	Fremdschlüssel

1. Einführung und Überblick
2. Modellierung
3. Normalisierung
<b>4. Relationale Algebra</b>
5. Lookup etc. in der Praxis
6. SQL – Data Definition Language
7. SQL – Data Manipulation Language
8. SQL – Trigger
9. SQL – Funktionen / Prozeduren
10. SQL – Datenschutz
11. Transaktionen

Inhalte

✓ Relationale Algebra

✓ Relationale Integritätsregeln

Klassische Mengenoperationen

Spezielle Relationenoperationen

## Relationale Integritätsregeln

Regel
Entity-Integrität
Referenzielle Integrität
Semantische Integrität
Ablaufintegrität

Die Inhalte einer Datenbank sollten idealerweise fehlerfrei und in sich schlüssig sein. Beim Anlegen von Tabellen werden auch gleich deren Integritätsregeln festgelegt. Somit werden fehlerhafte Datensätze gar nicht erst angenommen.

Quelle: Throll, M.; Bartosch, O. (2007): Einstieg in SQL

## 1. Entity Integrität

- Eine Menge von Relationen besitzt die Entity Integrität, wenn jede Relation einen Primärschlüssel besitzt

## 2. Referenzielle Integrität

- Eine Menge von Relationen besitzt die referentielle Integrität, wenn jeder Wert eines Fremdschlüssels einer Relation Wert eines Primärschlüssel in einer anderen Relation ist.
- Diese Art der Integrität erfordert, dass beim Einfügen von Werten in die Fremdschlüsselattribute (Detail) geprüft werden muss, ob der Wert im Primärschlüsselattribut (Master) vorkommt.
- Andererseits muss beim Löschen von Primärschlüsselwerten nachgeschaut werden, ob noch abhängige Fremdschlüsselwerte existieren und entsprechend reagiert werden.

## 1. Semantische Integrität

- Eine Menge von Relationen besitzt die semantische Integrität, wenn die Richtigkeit der Eingaben der Benutzer gewährleistet ist.
- Ist die semantische Integrität gegeben, so spricht man von einem konsistenten Datenzustand, wenn sie verletzt ist, von einem inkonsistenten Zustand der persistent gespeicherten Daten

## 2. Ablaufintegrität

- Eine Menge von Relationen besitzt die Ablaufintegrität, wenn mehrere Benutzer konkurrierend auf die Datenbank zugreifen können und sichergestellt ist, dass sich die Datenbank danach immer in einem korrekten Zustand befindet.



1. Einführung und Überblick

2. Modellierung

3. Normalisierung

**4. Relationale Algebra**

5. Lookup etc. in der Praxis

6. SQL – Data Definition Language

7. SQL – Data Manipulation Language

8. SQL – Trigger

9. SQL – Funktionen / Prozeduren

10. SQL – Datenschutz

11. Transaktionen

## Inhalte

✓ Relationale Algebra

✓ Relationale Integritätsregeln

✓ Klassische Mengenoperationen

Spezielle Relationenoperationen

## Klassische Mengenoperationen

Englische Bezeichnung	Deutsche Bezeichnung
Product	Kartesisches Produkt
Union	Vereinigung
Intersection	Durchschnitt
Difference	Differenz

## Kartesisches Produkt (Product)

- Auch Kreuzprodukt genannt, ist eine Grundoperation der relationalen Algebra und damit lässt sie sich nicht durch andere Operationen simulieren
- Doppelte Tupel treten nicht in der Ergebnismenge auf

R1		
ID	Name	Vorname
1	Müller	Frank
2	Meier	Karl
3	Schmidt	Dieter

R2	
Strasse	Ort
Domagkstrasse	Münster
Kölner Strasse	Wuppertal

R1 x R2				
ID	Name	Vorname	Strasse	Ort
1	Müller	Frank	Domagkstrasse	Münster
1	Müller	Frank	Kölner Strasse	Wuppertal
2	Meier	Karl	Domagkstrasse	Münster
2	Meier	Karl	Kölner Strasse	Wuppertal
3	Schmidt	Dieter	Domagkstrasse	Münster
3	Schmidt	Dieter	Kölner Strasse	Wuppertal

## Vereinigung (Union)

- Bei der Vereinigung  $R \cup S$  werden alle Tupel der Relation R mit allen Tupeln der Relation S zu einer einzigen Relation vereint.
- Voraussetzung dafür ist, dass R und S das gleiche Relationenschema haben. Das heißt, sie haben gleiche Attribute und Attributtypen.
- Duplikate werden bei der Vereinigung gelöscht.

$$R \cup S := \{t | t \in R \vee t \in S\}$$

R1

A	B	C
1	2	3
4	5	6

R2

A	B	C
4	5	6
7	8	9

R1  $\cup$  R2

A	B	C
4	5	6
7	8	9
1	2	3

## Differenz

- Bei der Operation  $R \setminus S$  oder  $R - S$  werden aus der ersten Relation  $R$  alle Tupel entfernt, die auch in der zweiten Relation  $S$  vorhanden sind.

R1

A	B	C
1	2	3
4	5	6

R2

A	B	C
4	5	6
7	8	9

R1 - R2

A	B	C
1	2	3

$R - S := R \setminus S := \{t | t \in R \wedge t \notin S\}$

## Symmetrische Differenz

- Bei der symmetrischen Differenz  $R \Delta S$  handelt es sich um die Menge aller Tupel, die entweder in  $R$  oder in  $S$  aber nicht in beiden gleichzeitig enthalten sind.

R1

A	B	C
1	2	3
4	5	6

R2

A	B	C
4	5	6
7	8	9

R1 Δ R2

A	B	C
1	2	3
7	8	9

$R \Delta S := \{t | (t \in R \vee t \in S) \wedge t \notin R \cap S\}$

1. Einführung und Überblick

2. Modellierung

3. Normalisierung

**4. Relationale Algebra**

5. Lookup etc. in der Praxis

6. SQL – Data Definition Language

7. SQL – Data Manipulation Language

8. SQL – Trigger

9. SQL – Funktionen / Prozeduren

10. SQL – Datenschutz

11. Transaktionen

## Inhalte

✓ Relationale Algebra

✓ Relationale Integritätsregeln

✓ Klassische Mengenoperationen

✓ Spezielle Relationenoperationen

## Spezielle Relationenoperationen

Englische Bezeichnung	Deutsche Bezeichnung
Selektion	Zeilenselektion
Projection	Projektion
Join	Verknüpfung
Division	Division

## Selektion

- Mit der Selektion werden bestimmte Tupel, sprich Zeilen, aus einer bestehenden Relation ausgewählt
- Auf einer Relation  $R(A_1, \dots, A_n)$  mit den Attributwerten  $a_1, \dots, a_n$  wird das Selektionsprädikat  $B$  als Abbildung  $B: R \rightarrow \{ \text{wahr, falsch, unknown} \}$ , also  $B(a_1, \dots, a_n) \in \{ \text{wahr, falsch, unknown} \}$  erklärt

R1

ID	Name	Vorname
1	Müller	Frank
2	Müller	Thomas
3	Meier	Karl
4	Schmidt	Dieter

R2: Selektion ( $ID < 3$ )

ID	Name	Vorname
1	Müller	Frank
2	Müller	Thomas



## Projektion

- Mit der Projektion werden Attribute (Spalten) aus einer bestehenden Relation ausgewählt und evtl. deren Reihenfolge in der Ergebnismenge vertauscht.
- Projektion  $\pi_{\{B_1, \dots, B_j\}}(R) = \{ (a_{i1}, \dots, a_{ij}) \mid (a_1, \dots, a_n) \in R \}$
- Doppelte Tupel werden in der Ergebnismenge unterdrückt

R1

ID	Name	Vorname
1	Müller	Frank
2	Müller	Thomas
3	Meier	Karl
4	Schmidt	Dieter

R\* (ID, Name)

ID	Name
1	Müller
2	Müller
3	Meier
4	Schmidt

## Division

- Wird eingesetzt wenn die Frage: „für alle“ enthält
- Anschaulich gesprochen enthält  $R:S$  also diejenigen Attribute aus  $R$ , welche in jeder Kombination mit den Attributen aus  $S$  in  $R$  vorkommen.

R

Name	Vorname	Alter
Müller	Toni	50
Maier	Hans	48
Hohl	Helmut	70
Schmidt	Toni	50
Schmidt	Franz	48

S

Vorname	Alter
Toni	50
Franz	48

$R:S$

Name
Schmidt

## JOIN

- Join Operationen verbinden 2 Relationen ähnlich dem kartesischen Produkt.
- Es werden jedoch nur Tupel ausgewählt, die in Beziehung zueinander stehen.

### ➤ Mögliche Ausprägungen sind

- Theta Join
- Equi Join
- Natural Join
- Left Outer Join
- Right Out Join

## Theta Join

- Der Theta-Join ist nach Definition eine Operation, die sich aus Selektion und kartesischem Produkt (hier  $R1 \times R2$ ) ableiten lässt.
- Vergleichsoperatoren sind: =, <, >, ≤, ≥, <>, !=

R1

ID1	Name	Vorname
1	Müller	Frank
2	Meier	Karl
3	Schmidt	Dieter

R2

ID2	Strasse	Ort
1	Domagkstrasse	Münster
2	Kölner Strasse	Wuppertal

R (R1, R2, ID1=ID2)

ID1	Name	Vorname	ID2	Strasse	Ort
1	Müller	Frank	1	Domagkstrasse	Münster
2	Meier	Karl	2	Kölner Strasse	Wuppertal

Merke: =, <, >, ≤, ≥, <>, !=

## Equi Join

- Ein Equi Join ist ein Theta Join, der im Selektionsprädikat nur den Vergleichsoperatoren „=“, „<“, „>“, „<=“, „>=“ zulässt

Merke: Equi = GLEICH

## Natural Join

- Ist gleich dem Equi Join, nur dass zusätzlich die Attribute, die doppelt vorkommen, nur einmal aufgelistet werden.

Merke: Natural = 1 X GLEICH

## Beispiel Natural Join

R1

A1	A2
1	A
2	B
3	C

R2

A1	B2	B3
1	X	V
2	Y	WW

Natural Join aus R1,R2

A1	A2	B2	B3
1	A	X	V
2	B	Y	WW

## Left Outer Join

**Left Outer Join** zweier Relationen R1 und R2 ist ein Natural Join, bei dem alle Tupel der linken Relation, hier R1, die im Natural Join unterdrückt werden, als Tupel mit aufgeführt und in den Spalten, die zu R2 gehören, mit *NULL* - Werten aufgefüllt werden

R1

A1	A2
1	A
2	B
3	C

R2

A1	B2	B3
1	X	V
2	Y	WW

Left Outer Join aus (R1,R2, R1.A1=R2.A1)

A1	A2	B2	B3
1	A	X	V
2	B	Y	WW
3	C	NULL	NULL

Merke: Left outer = mit linken Daten

## Right Outer Join

**Right Outer Join** zweier Relationen R1 und R2 ist ein Natural Join, bei dem alle Tupel der rechten Relation, hier R2, die im Natural Join unterdrückt werden, als Tupel mit aufgeführt und in den Spalten, die zu R1 gehören, mit *NULL* - Werten aufgefüllt werden

R1

A1	A2
1	A
2	B
3	C

R2

A1	B2	B3
1	X	V
4	Y	WW

Right Outer Join aus (R1,R2, R2.A1=R1.A1)

A1	A2	B2	B3
1	A	X	V
4	NULL	Y	WW

*Merke: right outer = mit rechten Daten*