

# Relationenbegriff

## Mathematische Definition

Sind  $W_1, W_2, \dots, W_n$  nichtleere Mengen, dann ist jede nichtleere Teilmenge der Produktmenge

$$PM = W_1 * W_2 * \dots * W_n \text{ eine } n\text{-stellige}$$

**Relation**  $R \subseteq W_1 \times W_2 \times \dots \times W_n$

$W_1, W_2, \dots, W_n$  sind die Wertebereiche (Menge aller Werte = Domäne) der Attribute  $A_1, A_2, \dots, A_n$  von Entities.  
 $n$  ist der Grad (degree) der Relation.

## Darstellung als Tabelle

Mitarbeiter	MITNR	VORNAME	NAME	ANSCHRIFT	ALTER	← Attributnamen
Zeilen/ Tupel	101	Peter	Silie	Dresden	5	} Attributwerte
	102	Mario	Nette	Dresden	6	
	103	Klaus	Uhr	Radebeul	20	
	104	Otto	Graffie	Freital	17	
	105	Kurt	Isane	Friedersdorf	35	
	106	Paul	Aner	Merseburg	25	
	↑ Primär- schlüssel	Nichtschlüsselspalten Datenspalten			Domäne	

# Charakteristika des relationalen Modells

- K1: Es gibt eine Menge von Relationen unterschiedlichen Grades über den Attributwerten.
- K2: Die Relationen sind untereinander gleichberechtigt.
- K3: Die Relationen sind zeitlich veränderlich (Einfügen, Löschen, Ändern von Tupeln).
- K4: Jede Relation hat dabei charakteristische Eigenschaften.
  - K41: Jedes Tupel der Relation kommt nur einmal vor.
  - K42: Die Reihenfolge der Tupel ist beliebig.
  - K43: Die Reihenfolge der Spalten ist auch beliebig, da die Bezugnahme auf die Spalten über die eindeutigen Attributnamen und nicht über eine Spaltennummer erfolgt.  
Attributnamen müssen in einer Relation unterschiedlich sein.  
Die Reihenfolge wird einmal vorgegeben, bleibt dann bestehen.
  - K44: Es gibt genau einen Primärschlüssel, der die Tupel eindeutig identifiziert.

# Datendefinition im relationalen Datenmodell

Eine im relationalen Datenbanksystem agierende Datenbeschreibungssprache (DDL) muß die im relationalen Modell vorhandenen Komponenten definieren:

- Name der Relation,
- Attributnamen,
- Wertebereiche,
- Primärschlüssel,
- ggf. Integritätsbedingungen

# Datendefinition im relationalen Datenmodell - Beispiel

<b>Relation:</b>	<b>Mitarbeiter</b>	
	Attribute	<u>Mitarbnr</u> ; INT Name; CHAR(20) Geburtsdatum; DATE Gehalt; NUMERIC(8,2)
<b>Relation:</b>	<b>Abteilung</b>	
	Attribute	<u>Abteilnr</u> ; INT Bezeichnung; CHAR(15) Raum; CHAR(5) Leiter; INT
	Integritätsbedingung	Leiter → Mitarbeiter.Mitarbnr 100 ≤ Raum < 451
<b>Relation:</b>	<b>Mitabt</b>	
	Attribute	<u>Mitarbnr</u> ; INT <u>Abteilnr</u> ; INT Anteil; NUMERIC(3,1)
	Integritätsbedingung	(Mitarbnr, Abteilnr) ist Primärschlüssel; Mitarbnr → Mitarbeiter.Mitarbnr; Abteilnr → Abteilung.Abteilnr; 0,1 ≤ Anteil ≤ 1,0

# Normalformen (1 bis 3) nach Codd

## Erste Normalform

Eine Relation ist in der **ersten Normalform** (1. NF), wenn alle Attribute nur atomare Werte enthalten. Das bedeutet, dass in der Relation keine Wiederholgruppen vorhanden sein dürfen, die selbst Relationen sein können.

# Normalformen (1 bis 3) nach Codd

## Zweite Normalform

Eine Relation ist in der **zweite Normalform** (2. NF), wenn sie sich in der ersten Normalform befindet und zusätzlich jedes Nichtsschlüsselattribut voll funktional vom Gesamtschlüssel abhängig ist, nicht aber von einzelnen Schlüsselteilen.

### Funktionale Abhängigkeit

In einer Relation  $R(A, B)$  ist das Attribut (bzw. die Attributkombination)  $B$  von dem Attribut (bzw. der Attributkombination)  $A$  **funktional abhängig**, falls zu jedem Wert des Attributs  $A$  genau ein Wert des Attributs  $B$  gehört.

**Darstellung:**  $R.A \rightarrow R.B$

### Volle funktionale Abhängigkeit

In einer Relation  $R(S1, S2, B)$  ist das Attribut (bzw. die Attributkombination)  $B$  von den Attributen  $S1, S2$  **voll funktional abhängig**, wenn  $B$  von den zusammengesetzten Attributen  $(S1, S2)$  funktional abhängig ist, aber nicht von einem einzelnen Attribut  $S1$  oder  $S2$ .

**Darstellung:**  $R.S1, R.S2 \rightarrow R.B$

# Normalformen (1 bis 3) nach Codd

## Dritte Normalform

Eine Relation ist in der **dritten Normalform** (3. NF), wenn sie sich in der zweiten Normalform befindet und zusätzlich jedes Nichtsschlüsselattribut nicht transitiv von einem Schlüsselattribut abhängig ist.

### Transitive Abhängigkeit

In einer Relation  $R(S, A, B)$  ist das Attribut  $B$  vom Attribut  $S$  (Schlüssel), der auch ein zusammengesetzter Schlüssel sein kann, **transitiv abhängig**, wenn  $A$  von  $S$  funktional abhängig ist,  $S$  jedoch nicht von  $A$  und  $B$  von  $A$  funktional abhängig ist.

**Darstellung:**  $R.S \rightarrow R.A \rightarrow R.B$  ( $R.A \not\rightarrow R.S$ )

Transitive Abhängigkeit ist immer eine mehrfache Abhängigkeit über mehrere Attribute.

# Bildung der Normalformen 1 bis 3 im Überblick

**Unnormalisierte Form**



**Abtrennung von Attributen, die selbst Relationen sind  
(Abtrennung von Wiederholgruppen)**

**Erste Normalform**



**Abtrennung von nicht voll funktionalen Abhängigkeiten  
zwischen den Schlüssel- und den  
Nichtschlüsselattributen (Abtrennung von  
unbegründeten Attributen, die nicht voll zur  
Beschreibung der Relationen benötigt werden)**

**Zweite Normalform**



**Abtrennung von Transitivitäten (Abtrennung von  
indirekten Attributzuweisungen)**

**Dritte Normalform**



# Codd'sche Regeln I

Auf dem relationalen Datenmodell aufbauende DBMS müssen nach Codd folgende Regeln genügen:

## 1. Informationsregel

Alle Informationen in einer relationalen Datenbasis sind auf genau eine Weise dargestellt, durch Werte in Tabellen.

## 2. Identifizierung

Jedes Objekt einer relationalen Datenbasis ist durch die Werte seiner Primärschlüsselattribute eindeutig identifiziert.

Der Primärschlüssel, der beim Kreieren der Tabelle deklariert wird, ist eine Spalte oder eine Kombination von Spalten.

## 3. Nullwerte

In einer relationalen Datenbasis wird jedes Datenelement mit unbekanntem Wert durch denselben Nullwert repräsentiert.

Dieser Wert ist unabhängig vom Domänen- oder Datentyp.

Es muss möglich sein, Nullwerte für bestimmte Attribute zu verbieten.

## 4. Data-Dictionary

Die Meta-Daten werden auf der logischen Ebene wie gewöhnliche Daten behandelt, so dass dieselbe DML für Abfragen verwendet werden kann.

# Codd'sche Regeln II

## 5. Umfassende Abfragensprache

Ein relationales System unterstützt mehrere Sprachen (z.B. SQL, QBE). Es muss jedoch eine Sprache geben, deren Anweisungen in einer exakt definierten Syntax verfügbar sind, und die alle folgenden Einrichtungen unterstützt:

- Tabellen-Definition
- View-Definition
- Datenmanipulation (Unterstützung der Operatoren der Relationenalgebra)
- Integritätsregeln
- Autorisierung
- Transaktionen-Verwaltung (Commit, Rollback)

## 6. View-Update

Alle Views, die theoretisch änderbar sind, müssen mit der DML änderbar sein.

## 7. Update-Level

Update-, Insert-, Delete-Operationen müssen auf einem Niveau verfügbar sein, das dem System die Möglichkeit der Optimierung lässt.

## 8. Physische Datenunabhängigkeit

## 9. logische Datenunabhängigkeit

## Codd'sche Regeln III

### 10. Integritätsbedingungen

- Entity-Integrität:

Keine Komponente des Primärschlüssels darf einen Nullwert enthalten.

- Referentielle Integrität:

Fremdschlüsselwerte müssen mit einem Primärschlüssel derselben Domäne korrespondieren, wobei die Domäne ein unterlegter „Pool“ von typbehafteten Werten ist, aus dem eine oder mehrere Spalten ihre Wertebereiche beziehen.

- Definition zusätzlicher Integritätsregeln mit Hilfe einer Dialogsprache
- Speicherung der Integritätsregeln im Data-Dictionary

### 11. Verteilungstransparenz

Die Terminalaktivitäten und Programme sind unabhängig von der Verteilung der Daten.

### 12. Nicht-Unterlaufbarkeit

Wenn ein relationales System über eine Eintupelschnittstelle verfügt, so dürfen die Integritätsregeln damit nicht unterlaufen werden. Dies gilt auch für alle anderen Nutzerschnittstellen des Systems.