

# Datenbanksysteme

Normalisierung

Jan Haase

2024

Abschnitt 5

---

# Themenübersicht

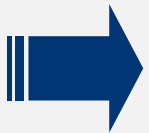
- Grundbegriffe und Datenbankentwurf
- Entity-Relationship-Modelle
- Relationales Datenbankmodell

## **Normalisierung**

- Arbeiten mit relationalen Datenbanken

# Gründe für die Normalisierung

- Verständlicheres Datenmodell für Anwender und Entwickler
- Vermeidung von Anomalien beim Einfügen, Löschen oder Ändern von Daten
- Eliminierung von Redundanzen
- Robusteres Datenmodell gegenüber Änderungen oder Erweiterungen
- Korrekte Abbildung der Realität



Erhöhung der Qualität des Datenmodells

Aber: Die Normalisierung hat auch Nachteile!

- Beim Zerlegen des Modells in viele Einzelrelationen leidet irgendwann die Übersichtlichkeit.
- In einer Anwendung müssen die Relationen erst wieder zusammengesetzt werden. Die Performanz ist schlechter.
- In auswertungsintensiven Datenbanken (Data Warehouse, Analysesysteme) wird daher oft auf eine vollständige Normalisierung verzichtet.

## Beispiel für Anomalien

Diese Modellierung verursacht mögliche Probleme:

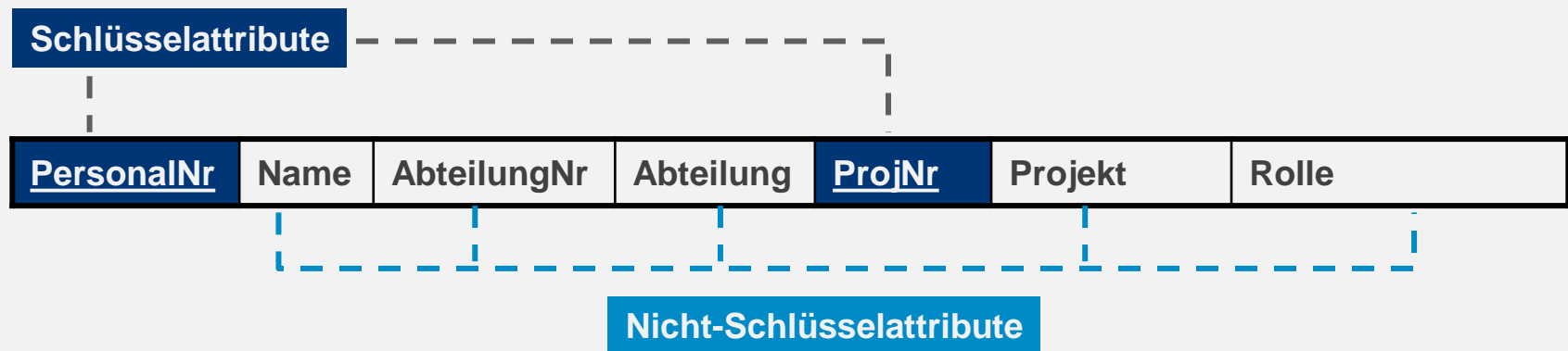
- **Einfügen: Insert-Anomalie** - Ein Projekt, dem noch keine Mitarbeiter zugeordnet sind, kann nicht angelegt werden.
- **Löschen: Deletion-Anomalie** - Wenn ein Projekt gelöscht werden soll, werden auch Mitarbeiter-Daten gelöscht.
- **Ändern: Update-Anomalie** - Wenn eine Abteilungszuordnung eines Mitarbeiters geändert wird, muss diese Änderung ggf. in mehreren Tabellenzeilen durchgeführt werden.

**Tabelle Projekte**

PersonalNr	Name	AbteilungNr	Abteilung	ProjNr	Projekt	Rolle
01	Müller	4711	IT	P10	SAP-Einführung	Projektleiter
01	Müller	4711	IT	P11	Datenbank	Projektmitarbeiter
02	Meier	4712	Einkauf	P10	SAP-Einführung	Projektmitarbeiter
03	...					

# Schlüssel: Erläuterung

- Ein **Schlüssel** ist eine Menge von Attributen (also eines oder mehrere), die eine Zeile (Tupel) einer Tabelle eindeutig identifiziert.
- Ein **Schlüsselkandidat** ist ein Schlüssel mit minimaler Anzahl an Attributen.
- Eine **Relation** kann mehrere Schlüsselkandidaten haben.
- Ein **Primärschlüssel** ist ein beliebig ausgewählter Schlüsselkandidat, der zur eindeutigen Identifizierung jeder Zeile benutzt wird. Besteht der Primärschlüssel aus mehreren Attributen (dies ist dann der Fall, wenn ein Attribut zur eindeutigen Identifizierung nicht ausreicht), wird er als *zusammengesetzter Primärschlüssel* bezeichnet.
- Ein **Schlüsselattribut** ist ein Attribut, das Teil mindestens eines Schlüsselkandidaten ist. Alle anderen Attribute sind *Nicht-Schlüsselattribute*.



# Schlüssel: Definition und Beispiel

## Definition: Schlüssel

Gegeben sei eine Tabelle und eine Menge  $M$ , die alle Attribute der Tabelle enthält. Gegeben sei weiterhin eine Attributsmenge  $A$  der Tabelle.

**Wenn  $A \rightarrow M$  gilt, heißt  $A$  Schlüssel der Tabelle.**

Eine Attributsmenge  $A$  ist also ein Schlüssel, wenn man aus den Attributen aus  $A$  eindeutig alle Belegungen der Attribute der Tabelle berechnen kann.

## Schlüssel

### ● Schlüssel:

- $\{\text{PersonalNr}, \text{ProjNr}\} \rightarrow \{\text{PersonalNr}, \text{ProjNr}, \text{Name}, \text{AbteilungsNr}, \text{Abteilung}, \text{ProjNr}, \text{Projekt}, \text{Rolle}\}$

### ● Weitere Schlüssel:

- $\{\text{PersonalNr}, \text{ProjNr}, \text{Name}\}$
- $\{\text{PersonalNr}, \text{ProjNr}, \text{AbteilungsNr}\}$
- $\{\text{PersonalNr}, \text{ProjNr}, \text{Name}, \text{AbteilungsNr}, \text{Abteilung}, \text{ProjNr}, \text{Projekt}, \text{Rolle}\}$

PersonalNr	Name	AbteilungsNr	Abteilung	ProjNr	Projekt	Rolle
------------	------	--------------	-----------	--------	---------	-------

# Funktionale Abhängigkeit / voll funktionale Abhängigkeit

## Definition: *funktionale* Abhängigkeit

Ein Attribut oder eine Attributkombination Y ist *funktional abhängig* von Attribut oder Attributkombination X derselben Relation, wenn zu einem bestimmten Wert von X höchstens ein Wert von Y möglich ist.

D. h. kennt man die Werte der Attribute / des Attributs in X, lassen sich daraus eindeutig die Werte der Attribute/ des Attributs aus Y bestimmen.

Schreibweise: FD  $\{X\} \rightarrow \{Y\}$  oder kurz  $X \rightarrow Y$ .

*Die funktionale Abhängigkeit kann stets nur aus dem Anwendungskontext abgeleitet werden!*

## Definition: *voll funktionale* Abhängigkeit

Ein Attribut oder Attributkombination Y ist *voll funktional abhängig* von Attribut oder Attributkombination X derselben Relation, wenn Y von der gesamten Attributkombination X funktional abhängig ist, aber nicht bereits von Teilen derselben (d. h. X ist minimal).

Im Falle eines einzelnen Attributes in X ist also eine funktionale Abhängigkeit immer auch voll funktional.

# Beispiel

**Tabelle *Projekte***

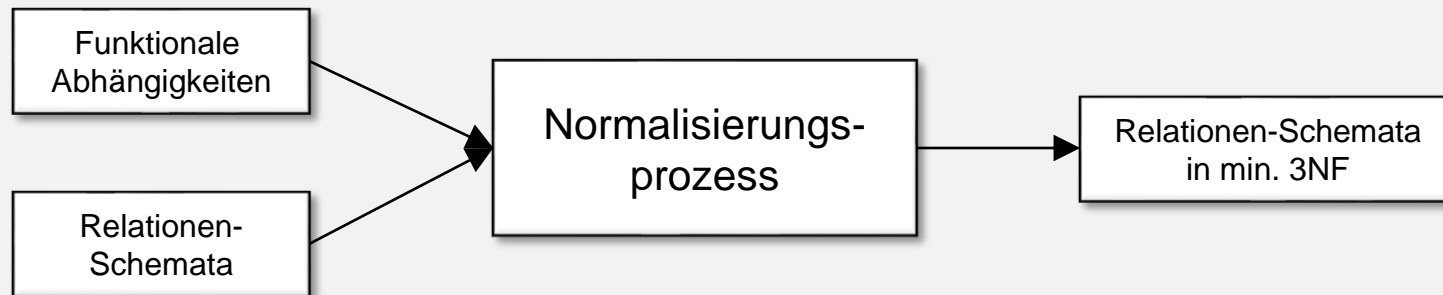
<u>PersonalNr</u>	<u>Name</u>	AbteilungNr	Abteilung	<u>ProjNr</u>	Projekt	<u>Rolle</u>
01	Müller	4711	IT	P10	SAP-Einführung	Projektleiter
01	Müller	4711	IT	P11	Datenbank	Projektmitarbeiter
02	Meier	4712	Einkauf	P10	SAP-Einführung	Projektmitarbeiter
03	...					

- Beispiel funktionale Abhängigkeit
  - Das Attribut **Name** ist funktional abhängig vom Attribut *PersonalNr*, da von der *PersonalNr* eindeutig auf den Namen geschlossen werden kann (umgekehrt nicht, da es z. B. mehrere Mitarbeiter mit dem Namen Müller geben kann).  $PersonalNr \rightarrow Name$
- Beispiel voll funktionale Abhängigkeit
  - Das Attribut **Rolle** ist voll funktional abhängig von der Attributkombination *PersonalNr* und *ProjNr*, da die Rolle eines Mitarbeiters projektabhängig variiert. Nur unter Angabe beider Nummern kann man eindeutig auf die Rolle schließen.  $PersonalNr, ProjNr \rightarrow Rolle$



# Normalisierung

- Prozess:



# 1. Normalform (1NF)

Eine Relation ist in der ersten Normalform (1NF), wenn alle Attribute nur **atomare Werte** enthalten, d. h. pro Attribut/Zeile gibt es nur einen Wert.

## Nicht normalisierte Relation:

<u>PersonalNr</u>	Name	AbteilungNr	Abteilung	<u>ProjNr</u>	Projekt	Rolle
01	Müller	4711	IT	P10	SAP-Einführung	Projektleiter
				P11	Datenbank	Projektmitarbeiter
02	Meier	4712	Einkauf	P10	SAP-Einführung	Projektmitarbeiter
03	...					



## 1 NF:

<u>PersonalNr</u>	Name	AbteilungNr	Abteilung	<u>ProjNr</u>	Projekt	Rolle
01	Müller	4711	IT	P10	SAP-Einführung	Projektleiter
01	Müller	4711	IT	P11	Datenbank	Projektmitarbeiter
02	Meier	4712	Einkauf	P10	SAP-Einführung	Projektmitarbeiter
03	...					

## 2. Normalform (2NF)

Eine Relation ist in der zweiten Normalform (2NF), wenn sie in 1NF ist und jedes Nicht-Schlüssel-Attribut von jedem Schlüsselkandidaten **voll funktional abhängig** ist.

Name, AbteilungNr und Abteilung hängen nur von der PersonalNr ab.

Projekt hängt nur von der ProjNr ab.

<u>PersonalNr</u>	Name	AbteilungNr	Abteilung	<u>ProjNr</u>	Projekt	Rolle
01	Müller	4711	IT	P10	SAP-Einführung	Projektleiter
01	Müller	4711	IT	P11	Datenbank	Projektmitarbeiter
02	Meier	4712	Einkauf	P10	SAP-Einführung	Projektmitarbeiter
03	...					



Tabelle muss in 2NF umgewandelt werden.

## 2. Normalform (2NF)

- Zur Bildung der 2. Normalform ist eine Zerlegung in mehrere Relationen erforderlich.
- Relationen mit einelementigen Schlüsselkandidaten sind immer in 2NF.

### 1 NF:

<u>PersonalNr</u>	Name	AbteilungNr	Abteilung	<u>ProjNr</u>	Projekt	Rolle
01	Müller	4711	IT	P10	SAP-Einführung	Projektleiter
...						



### 2 NF:

**Tabelle Personal**

<u>PersonalNr</u>	Name	AbteilungNr	Abteilung
01	Müller	4711	IT
02	Meier	4712	Einkauf
03	...		

**Tabelle Projekte**

<u>ProjNr</u>	Projekt
P10	SAP-Einführung
P11	Datenbank

**Tabelle Rollen**

<u>PersonalNr</u>	<u>ProjNr</u>	Rolle
01	P10	Projektleiter
01	P11	Projektmitarbeiter
...		

### 3. Normalform (3NF)

Eine Relation ist in dritter Normalform (3NF), wenn sie in 2NF ist und keine funktionalen Abhängigkeiten zwischen Nicht-Schlüsselattributen (sog. *transitive Abhängigkeiten*) existieren.

**Tabelle *Personal***

<u>PersonalNr</u>	Name	AbteilungNr	Abteilung
01	Müller	4711	IT
02	Meier	4712	Einkauf
03	Schulze	4712	Einkauf
...			

Abteilung hängt zwar vom Schlüssel PersonalNr ab, könnte aber auch aus dem Nichtschlüsselattribut AbteilungNr abgeleitet werden.

- *Hinweise:*
  - Aufbauend auf der 3NF lassen sich noch weitere NF bilden
  - Die 3NF ist jedoch für die meisten praktischen Anwendungen ausreichend.
  - Ein durch ERM-Abbildung erstelltes Relationenmodell ist bei guter Modellierung fast immer in 3NF → trotzdem Prüfung nötig.

## 3. Normalform (3NF)

- Zur Bildung der 3. Normalform ist eine Zerlegung in mehrere Relationen erforderlich.

### 2 NF:

**Tabelle *Personal***

<u>PersonalNr</u>	Name	AbteilungNr	Abteilung
01	Müller	4711	IT
02	Meier	4712	Einkauf
03	Schulze	4712	Einkauf



### 3 NF:

**Tabelle *Personal***

<u>PersonalNr</u>	Name	<u>AbteilungNr</u>
01	Müller	4711
02	Meier	4712
03	Schulze	4712

**Tabelle *Abteilung***

<u>AbteilungNr</u>	Abteilung
4711	IT
4712	Einkauf

## Zusammenfassung (1/2)

- **Gründe und Nachteile der Normalisierung**
  - Qualität des Datenmodells
    - Verständlicheres Datenmodell
    - Vermeidung Anomalien: Insert-, Deletion-, Update-Anomalie
    - Eliminierung von Redundanzen
    - Robusteres Datenmodell
    - Korrekte Realitätsabbildung
  - Aber:
    - Komplexität steigt
    - Zusammenführung in Anwendung: Performance-Einbußen
    - ggfs. Verzicht auf vollständige Normalisierung
- **Schlüssel**
  - Schlüssel, Schlüsselkandidat, Schlüsselattribut, Primärschlüssel
- **Funktionale und voll funktionale Abhängigkeit**

## Zusammenfassung (2/2)

- **Normalformen**

- 1. Normalform
  - Nur atomare Werte
- 2. Normalform
  - 1.NF
  - jedes Nicht-Schlüssel-Attribut voll funktional abhängig
    - Zerlegung in mehrere Relationen
- 3. Normalform
  - 2.NF
  - Auflösung transitiver Abhängigkeiten
    - Zerlegung in mehrere Relationen



# Übungsaufgabe 5

In einer Firma wird folgende Tabelle „Aufgabeliste“ zur Verwaltung von Aufträgen genutzt.

AuftragsNr	Eingangsdatum	ProduktNr	ProduktName	Preis	Lieferant
1	11.02.2003	10	Reis	0.51	Meier
1	11.02.2003	12	Mais	0.53	Müller
2	11.02.2003	10	Reis	0.51	Meier
2	11.02.2003	11	Hirse	0.51	Schmidt

Dabei gelten folgende funktionale Abhängigkeiten:

- (1) {AuftragsNr} → {Eingangsdatum}
- (2) {ProduktNr} → {ProduktName, Preis, Lieferant}
- (3) {Produktname} → {Preis, Lieferant}

Weitere funktionale Abhängigkeiten lassen sich aus den angegebenen Abhängigkeiten berechnen, weiterhin können die Mengen auf den linken Seiten nicht verkleinert werden.

- a) Geben Sie die möglichen Schlüsselkandidaten für die Tabelle an
- b) Formen Sie die Tabelle in Tabellen in zweiter Normalform und dann in dritter Normalform um, die die gleiche Aussagekraft haben.
- c) Was würde sich ändern, wenn es mehrere Lieferanten für ein Produkt geben würde, also (3) durch {Produktname, Lieferant} → {Preis} ersetzt würde?

# Übungsaufgabe 6

Nehmen Sie an, dass zur Verwaltung von Studierenden folgende Tabelle genutzt wird. Jeder Student arbeitet parallel in genau einer Firma und hat dort einen betrieblichen Betreuer.

MatNr	Name	TelefonNr	Studium	Firma	Betreuer	Fach	Note

Dabei gelten folgende funktionale Abhängigkeiten (weitere lassen sich daraus berechnen)

- I. {MatNr} → {Name, TelefonNr, Studium, Firma, Betreuer}
- II. {TelefonNr} → {MatNr}
- III. {MatNr, Fach} → {Note}
- IV. {TelefonNr, Fach} → {Note}
- V. {Firma} → {Betreuer}

- a) Begründen Sie formal, warum {TelefonNr} → Firma gilt
- b) Nennen Sie die Schlüsselkandidaten
- c) Bringen Sie die Tabelle mit den Standardverfahren in die zweite Normalform
- d) Bringen Sie die Tabellen aus c) mit dem Standardverfahren in die dritte Normalform