

# **Datenbanken: Normalisieren**

# Definition "Normalisieren"

= Daten einer DB werden so geordnet, dass redundanzfreie, klare Struktur entsteht.

# Definition "Relation"

= Struktur einer Datenbank (in unserer Vorstellung: = "Tabelle")

Vorüberlegung:  
Wann ist eine Relation eine gute Relation?

**Ist diese Relation eine gute Relation?**

**Notieren Sie: Nachteile**

Bestellung	Kunde
13.05.2010, 28.59 Euro	Josef Schmitt, Kundennr. 1774
13.09.2010, 99.01 Euro	Muriel Sauter, Kundennr. 131

**Bestellungen(Bestellung, Kunde)**

# Vorüberlegung: Wann ist eine Relation eine gute Relation?

Bestellung	Kunde
13.05.2010, 28.59 Euro	Josef Schmitt, Kundennr. 1774
13.09.2010, 99.01 Euro	Muriel Sauter, Kundennr. 131

## Bestellungen(Bestellung, Kunde)

Wie verhält sich diese Relation bei ...

- Sortieren nach Kundennummer
- Dateneingabe durch verschiedene Personen
- Ändern einer Kundenanschrift
- Berechnung der Gesamtsumme der Bestellungen im September 2010

# Ein zentrales Problem: Redundanz

<u>Best.- Datum</u>	<u>Nach- name</u>	<u>Vor- name</u>	<u>Wohn- ort</u>	<u>Kunden- Nr.</u>	<u>Re.- betrag</u>
01.09.10	Schmitt	Josef	München	1377	31.99
01.09.10	Maier	Marina	Hamburg	19991	188.45
03.09.10	Huber	Hannes	Hamburg	573	20.55
07.09.10	Schmitt	Josef	München	1377	61.99

**bestellungen(bestelldatum, nachname, vorname,  
wohntort, kunden-nr., rechnungsbetrag)**

**Kundendaten für Josef Schmitt sind mehrfach vorhanden**

- müssen mehrmals eingegeben werden
- müssen bei Änderung der Anschrift mehrfach geändert werden
- verstopfen beim Sortieren die Ausgabetabelle

# Ausgangszustand: **Chaos**

Bestellung	Kunde	Rechnung
01.09.10	Josef Schmitt, München, Kundennummer 1377	31.99, Kreditkarte
01.09.10	Marina Maier, Hamburg, Kundennummer: 19991	188.45, Vorkasse
03.09.10	Hannes Huber, Hamburg, Kundennummer: 573	20.55, Kreditkarte
07.09.10	Josef Schmitt, München, Kundennummer 1377	61.99, Kreditkarte

**ungeordnete, schlecht arrangierte Datenstrukturen**  
=  
**fehleranfällig**

# Ausgangszustand: Chaos

Bestellung	Kunde	Rechnung
01.09.10	Josef Schmitt, München, Kundennummer 1377	31.99, Kreditkarte
01.09.10	Marina Maier, Hamburg, Kundennummer: 19991	188.45, Vorkasse
03.09.10	Hannes Huber, Hamburg, Kundennummer: 573	20.55, Kreditkarte
07.09.10	Josef Schmitt, München, Kundennummer 1377	61.99, Kreditkarte

## Fehlermöglichkeiten:

- Einfügeanomalie** (Anlegen eines neuen Kunden führt zu unvollständigem Datensatz)
- Änderungsanomalie** (Wird Anschriftänderung nicht mehrfach vorgenommen, stimmen Daten nicht überein = Dateninkonsistenz)
- Löschanomalie** (Wird die Bestellung vom 1.9. storniert, gehen auch die Daten von Marina Maier verloren)

# 1. Normalform

## Attribute atomisieren

Bestellung	Kunde	Rechnung
01.09.10	Josef Schmitt, München, Kundennummer 1377	31.99, Kreditkarte
01.09.10	Marina Maier, Hamburg, Kundennummer: 19991	188.45, Vorkasse
03.09.10	Hannes Huber, Hamburg, Kundennummer: 573	20.55, Kreditkarte
07.09.10	Josef Schmitt, München, Kundennummer 1377	61.99, Kreditkarte



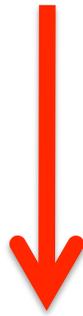
<u>Best.-Datum</u>	<u>Nachname</u>	<u>Vorname</u>	<u>Wohnort</u>	<u>Kunden-Nr.</u>	<u>Re.-betrag</u>	<u>Zahlungsweise</u>
01.09.10	Schmitt	Josef	München	1377	31.99	Kreditkarte
01.09.10	Maier	Marina	Hamburg	19991	188.45	Vorkasse
03.09.10	Huber	Hannes	Hamburg	573	20.55	Kreditkarte
07.09.10	Schmitt	Josef	München	1377	61.99	Kreditkarte

Keine weitere sinnvolle Zerlegung der Attribute möglich.

# 1. Normalform

## Attribute atomisieren

bestellungen(bestellung, kunde, rechnung)



bestellungen(best\_datum, vorname, nachname, wohnort, kundennr, re-betrag)

Keine weitere sinnvolle Zerlegung der Attribute möglich.

## 2. Normalform

### Attribute einem Schlüssel zuordnen

Best.-Datum	Nachname	Vorname	Wohnort	Kunden-Nr.	Re.-betrag	Zahlungsweise
01.09.10	Schmitt	Josef	München	1377	31.99	Kreditkarte
01.09.10	Maier	Marina	Hamburg	19991	188.45	Vorkasse
03.09.10	Huber	Hannes	Hamburg	573	20.55	Kreditkarte
07.09.10	Schmitt	Josef	München	1377	61.99	Kreditkarte

Kundendaten (Nachname usw.) hängen nur von einem Teil des Primärschlüssels ab (nämlich von Kundennummer); Bestelldatum hat keinen Einfluss auf Nachnamen → Zerlegung der Relation notwendig

Bestellungsdatum	Knummer	Rechnungsbetrag	Zahlungsweise
01.09.10	1377	31.99	Kreditkarte
01.09.10	19991	188.45	Vorkasse
03.09.10	573	20.55	Kreditkarte
07.09.10	1377	61.99	Kreditkarte

*Relation 1: Bestellungen*

Nachname	Vorname	Wohnort	Knummer
Schmitt	Josef	München	1377
Maier	Marina	Hamburg	19991
Huber	Hannes	Hamburg	573

*Relation 2: Kunden*

**Jedes Attribut hängt vom gesamten Primärschlüssel ab.**

## 2. Normalform

# Attribute einem Schlüssel zuordnen

bestellungen(best\_datum, nachname, vorname,  
wohnort, kundennr, re-betrag, zahlungsweise)



bestellungen(best\_datum,  $\hat{\uparrow}$ kundennr, re-betrag, zahlungsweise)

kunden(kundennr, nachname, vorname, wohnort)

**Jedes Attribut hängt vom gesamten Primärschlüssel ab.**

# 3. Normalform

## Transitive Abhängigkeiten beseitigen

**direkt abhängig**

BestellID	Best.-tum	Nach-name	Vor-name	Wohn-ort	Kunden-Nr.	Re.-betrag	Zahlungs-weise
1	01.09.10	Schmitt	Josef	München	1377	31.99	Kreditkarte
2	01.09.10	Maier	Marina	Hamburg	19991	188.45	Vorkasse
3	03.09.10	Huber	Hannes	Hamburg	573	20.55	Kreditkarte
4	07.09.10	Schmitt	Josef	München	1377	61.99	Kreditkarte

**Relation 1: Bestellungen**

Bestellungsdatum	Knummer	Rechnungsbetrag	Zahlungsweise
01.09.10	1377	31.99	Kreditkarte
01.09.10	19991	188.45	Vorkasse
03.09.10	573	20.55	Kreditkarte
07.09.10	1377	61.99	Kreditkarte

**Relation 2: Kunden**

Nachname	Vorname	Wohnort	Knummer
Schmitt	Josef	München	1377
Maier	Marina	Hamburg	19991
Huber	Hannes	Hamburg	573

**Keine transitiven Abhängigkeiten mehr vorhanden.**

# 3. Normalform

## Transitive Abhängigkeiten beseitigen

bestellungen(bestellID, bestDatum, nachname, vorname,  
wohnort, kundenr, re-betrag, zahlungsweise)



bestellungen(bestellID, bestDatum, re-betrag, zahlungsweise, ↑kundenr)  
kunden(kundenr, nachname, vorname, wohnort)

Nachname "direkt" abhängig von Kundennummer.

Kundennummer "direkt" abhängig von Bestell-ID.

Damit: Nachname "indirekt" (= transitiv) abhängig von Bestell-ID.

(→ Daten, die direkt von Kunden-Nr. abhängig sind, sind evtl. redundant!)

# Zusammenfassung: Normalisierung bis 3NF

## Normalisieren bis zur 3. Normalform - Vorgehen

Solange nicht 1NF erreicht	
Kann ich einzelne Attribute sinnvoll in unterschiedliche Attribute aufteilen?	
J	N
Attribute atomisieren	1NF liegt vor!
Solange nicht 2NF erreicht	
Hat die Relation einen zusammengesetzten Primärschlüssel?	
J	N
Gibt es Attribute, die nur von einem Teil des Primärschlüssels (von einem Teilschlüssel) abhängen?	
J	N
Aus diesem Teilschlüssel muss eine neue Relation gebildet werden, damit die 2NF erreicht wird!	2NF liegt vor!
Solange nicht 3NF erreicht	
Gibt es Nichtschlüsselattribute, die funktional von einem Nichtschlüsselattribut abhängen (und damit: transitiv abhängig vom Primärschlüssel sind)?	
J	N
Abhängige Daten in eigene Relation auslagern; Primärschlüssel der neuen Relation bleibt als Fremdschlüssel in der alten Relation.	3NF liegt vor!