

# Vortrag 2 Abfragen mit Python und Datentypen

## Datenbanken

Lukas Wais

Codersbay

Version: 19. April 2023

# Inhaltsverzeichnis

Wiederholung Normalformen

Datentypen

Abfragen mit Python

Voraussetzungen und Bibliotheken

Abfragen in Python

# Definitionen

- ▶ **Normalisieren:** Die Daten werden so geordnet, dass eine klare und redundanzfreie Struktur entsteht
- ▶ **Relation:** Struktur einer DB (bei uns aktuell = Tabelle)

## Diese Tabelle soll in die 3NF übertragen werden

Bestellung	Kunde	Rechnung
01.09.10	Josef Schmitt, München, Kundennummer: 1337	31.99, Kreditkarte
01.09.10	Maria Maier, Hamburg, Kundennummer: 19991	188.45, Vorkasse
03.09.10	Hanner Huber, Hamburg, Kundennummer: 573	20.55, Kreditkarte
07.09.10	Josef Schmitt, München, Kundennummer: 1337	61.99, Kreditkarte

Ausgangspunkt: Chaos; ungeordnete, schlechte Datenstruktur → Anomalien möglich

# 1NF Attribute atomisieren

<u>Best.-Datum</u>	Nachname	Vorname	Wohnort	<u>Kunden-Nr.</u>	Re.-Betrag	Zahlungs- Weise
01.09.10	Schmitt	Josef	München	1337	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	1337	61.99	Kreditkarte

bestellungen(bestellung, kunde, rechnung)



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

## 2NF Attribute einem Schlüssel zuordnen

<u>Best.-Datum</u>	<u>Kunden-Nr.</u>	Re.-Betrag	Zahlungs- Weise
01.09.10	1337	31.99	Kreditkarte
01.09.10	19991	188.45	Vorkasse
03.09.10	573	20.55	Kreditkarte
07.09.10	1337	61.99	Kreditkarte

Relation Bestellungen

Nachname	Vorname	Wohnort	<u>Kunden-Nr.</u>
Schmitt	Josef	München	1337
Maier	Marina	Hamburg	19991
Huber	Hannes	Hamburg	573

Relation Kunden

## 2NF Erklärung

Warum die beiden Tabellen?

- ▶ Die Kundendaten hängen nur von einem Teil des Primärschlüssels ab, von der Kundennummer
- ▶ Das Bestelldatum hat keinen Einfluss auf den Nachnamen, deshalb ist eine Zerlegung notwendig

Jedes Attribut hängt vom gesamten Primärschlüssel ab

## 2NF textuell

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



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

## 3NF transitive Abhängigkeiten beseitigen

<u>BestellID</u>	Best.-Datum	Re.-Betrag	Zahlungs- weise	Kunden-Nr.
1	01.09.10	31.99	Kreditkarte	1337
2	01.09.10	188.45	Vorkasse	19991
3	03.09.10	20.55	Kreditkarte	573
4	07.09.10	61.99	Kreditkarte	1337

Relation Bestellungen

Nachname	Vorname	Wohnort	<u>Kunden-Nr.</u>
Schmitt	Josef	München	1337
Maier	Marina	Hamburg	19991
Huber	Hannes	Hamburg	573

Relation Kunden

## 3NF Erklärung

### Mathematische Transitivität

Eine Relation  $\mathfrak{R}$  angewendet auf die Menge  $X$  ist *transitiv*, wenn gilt

$$\text{für alle } x, y, z \in X : (x\mathfrak{R}y \wedge y\mathfrak{R}z) \Rightarrow x\mathfrak{R}z$$

wobei  $x\mathfrak{R}y$  die infix Notation für  $(x, y) \in \mathfrak{R}$  ist.

Beispiel Ordnungsrelation: Sei  $x = 8, y = 4, z = 2$ , dann gilt  $x > y$  und  $y > z$  daraus folgt, dass  $x$  größer sein muss als  $z$ .

# 3NF Erklärung

## Mathematische Transitivität

Eine Relation  $\mathfrak{R}$  angewendet auf die Menge  $X$  ist *transitiv*, wenn gilt

$$\text{für alle } x, y, z \in X : (x\mathfrak{R}y \wedge y\mathfrak{R}z) \Rightarrow x\mathfrak{R}z$$

wobei  $x\mathfrak{R}y$  die infix Notation für  $(x, y) \in \mathfrak{R}$  ist.

Beispiel Ordnungsrelation: Sei  $x = 8, y = 4, z = 2$ , dann gilt  $x > y$  und  $y > z$  daraus folgt, dass  $x$  größer sein muss als  $z$ .

- ▶ Problem: Der Nachname ist direkt abhängig von der Kundennummer.
- ▶ Die Kundennummer ist direkt abhängig vom Bestelldatum.
- ▶ Daraus folgt, dass der Nachname indirekt (= transitiv) abhängig von dem Bestelldatum ist.

## 3NF textuell

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



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



# Auszug der Datentypen von mariaDB

- ▶ BOOLEAN
  - ▶ Synonym für TINYINT(1)
- ▶ INT
  - ▶ Kann vorzeichenbehaftet sein
- ▶ BLOB ... binary large object
  - ▶ große binäre Datenobjekte (Bilder, Musik, ...)
- ▶ Date
  - ▶ Es gibt einige verschiedene Datums- und Zeitdatentypen
  
- ▶ Alle Datentypen: <https://mariadb.com/kb/en/data-types/>

## mariaDB Treiber

Um eine Datenbankabfrage zu ermöglichen ist es notwendig den richtigen Treiber für das jeweilige DBMS zu installieren. Denke an die Installation des Plugins. Für Python reicht es die Bibliothek mit pip zu installieren. Dafür einfach folgende Zeile in den Terminal eingeben:

▶ `pip install mariadb`

Unter Umständen muss man noch zusätzliche Pakete installieren, deshalb Fehlermeldungen **genau** lesen.

Der Connector ist selbstverständlich Open Source

<https://github.com/mariadb-corporation/mariadb-connector-python>.

## Einfaches SELECT mit Konsolenausgabe

```
import mariadb                                # do not forget the import

mydb = mariadb.connect(                       # DB setup
    host="localhost",
    user="cb_user",
    password="codersbayuser",
    database="nation"
)

mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM regions") # execute the query
data = mycursor.fetchall()                # get the result

data = list(data)                           # convert data to a list
print(data)                                 # work with the list
```