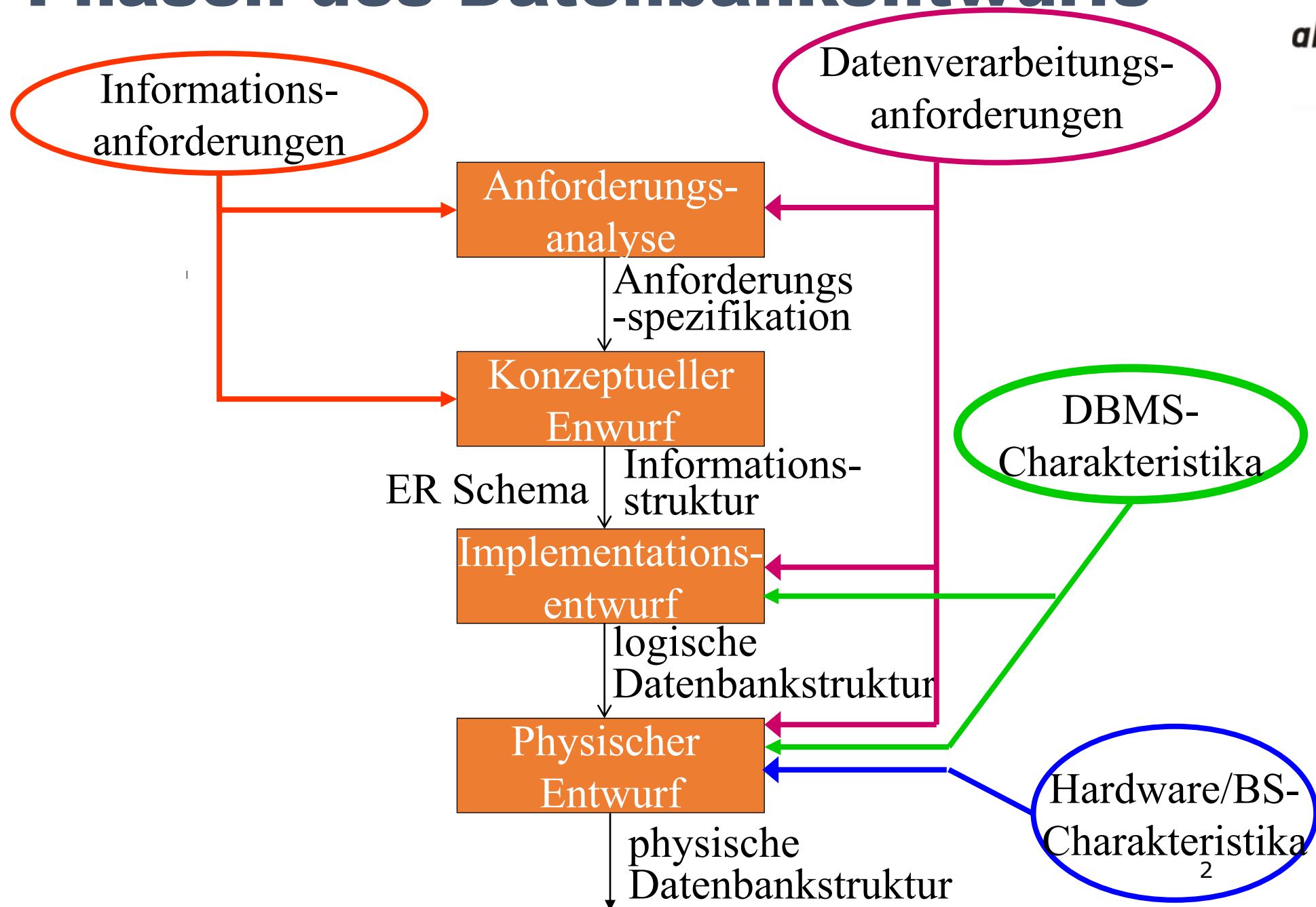


# Data Engineer

**Datenmodellierung:**

Normalformen – „Überprüfung der Redundanzen“

# Phasen des Datenbankentwurfs



# Relationale Entwurfstheorie – Normalformen



Warum Normalformen?

Vermeidung bzw. Beseitigung von

- Redundanz
- Inkonsistenzen

Vermeidung von Anomalien (2 Arten)

- Im Einbenutzerbetrieb
- Im Mehrbenutzerbetrieb

# Relationale Entwurfstheorie – Begriffe



- *Redundanz* (Informationstheorie): bezeichnet Informationen, die in einer Informationsquelle mehrfach vorhanden sind. Eine Informationseinheit ist dann redundant, wenn sie ohne Informationsverlust weggelassen werden kann.
- *Redundanz* (Sprach-/Kommunikationstheorie): bezeichnet mehrfache Nennung von Informationen, die für das Verständnis des Gesamtkontexts nicht notwendig sind.
- *Inkonsistenz*: bezeichnet einen Zustand, in dem zwei Dinge, die beide als gültig angesehen werden sollen, nicht miteinander vereinbar sind. Inkonsistenz bedeutet insbesondere Widersprüchlichkeit oder Unbeständigkeit von Daten (Zusammenhanglosigkeit).
- Für DO wichtiger Sachverhalt: Redundanz (also das mehrmalige Vorhandensein von Information) führt zu Inkonsistenz(en).

# Anomalien in DBs im Einbenutzerbetrieb



Im Einbenutzerbetrieb durch nicht normalisierte bzw. denormalisierte DB

- *Einfüge-Anomalie*: Wenn Einfügen eines Datensatzes nicht funktioniert, da nicht alle Informationen zum Primärschlüssel vorliegen.
- *Änderungs-Anomalie*: Wenn Attribute an mehreren Stellen gespeichert sind und bei Änderungen nicht überall angepasst werden – führt zu Inkonsistenzen
- *Lösch-Anomalie*: Wenn durch Löschen mehr Informationen entfernt werden als gewünscht. Entsteht wenn Datensatz mehrere unabhängige Informationen enthält. Es wird Information gelöscht, die noch benötigt wird.

# Anomalien in DBs im Mehrbenutzerbetrieb



Im Mehrbenutzerbetrieb durch unzulässigen parallelen DB-Zugriff

- *Verlorenes Update (Lost Update)*: Eine zweite Änderung der selben Information überschreibt eine erste Änderung.
- *Schreib-Lese-Konflikt (Dirty Read)*: Bei gleichzeitigem Lese-/Schreibzugriff wird bereits gelesen bevor Schreibzugriff bestätigt (committed) ist.
- *Nichtwiederholbares Lesen (Non-Repeatable Read)*: Wenn innerhalb einer Transaktion dieselbe Leseoperation nacheinander unterschiedliche Daten liefert.
- *Phantomproblem (Inconsistent Read)*: Wenn in einer Transaktion, die sich auf mehrere Datensätze mit einer angegebenen Eigenschaft bezieht, in einer gleichzeitig ablaufenden Transaktion neue Datensätze mit dieser Eigenschaft eingefügt, kann dies inkonsistente Daten der ersten Transaktion zur Folge haben.

# Relationale Entwurfstheorie – Normalformen



Im Moment 6 verschiedene (aufeinander aufbauende) Arten von Normalformen:

- 1. Normalform (1NF)
  - 2NF
  - 3NF
  - Boyce-Codd NF (BCNF)
  - 4NF
  - 5NF
- Stellen die Vermeidung von Anomalien sicher.
  - Aus Performanzgründen oft aufgehoben – Denormalisierung. Manchmal auch nur teilweise (für einzelne Tabellen bzw. Teile davon).
  - Immer sehr stark von der konkreten Anwendung abhängig, wie weit (und wo) normalisiert wird.
  - Aus praktischer Sicht meist bei 3NF beendet. Weitere NF benötigen zusätzliche Tabellen, welche den Abfrageaufwand erhöhen.

# Relationale Entwurfstheorie – Funktionale Abhängigkeiten



Eine Relation wird durch Attribute definiert. Bestimmen einige dieser Attribute eindeutig die Werte anderer Attribute, so spricht man von *funktionaler Abhängigkeit*.

Für eine Relation  $R(A_1, A_2, \dots, A_n)$  sind  $X$  und  $Y$  je eine echte Teilmenge der Attributmenge (z.B.  $X=(A_4, A_5)$  und  $Y=(A_8)$ ). Eine funktionale Abhängigkeit ( $X \rightarrow Y$ ) liegt vor, wenn es keine zwei Tupel geben kann, in denen für gleiche  $X$ -Werte verschiedene  $Y$ -Werte auftreten können. Umgekehrt kann es aber für gleiche  $Y$ -Werte verschiedene  $X$ -Werte geben.



# Relationale Entwurfstheorie – Normalformen

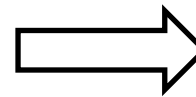


1. Normalform: Jedes Attribut darf nur atomar vorkommen, keine relationenwertige Attribute
2. Normalform: Jedes Nicht-Schlüssel Attribut vom Primärschlüssel voll funktional abhängig
3. Normalform: Jedes Nicht-Schlüssel Attribut nicht transitiv vom Primärschlüssel abhängig

# 1. Normalform (1NF)

Jedes Attribut darf nur atomar vorkommen, keine relationenwertige Attribute. Wenn bei Eintrag in einer Spalte mehrere Werte möglich, **KEINE 1NF!**

Eltern		
Vater	Mutter	Kinder
Johann	Martha	{Elsa, Luca, Josef}
Johann	Maria	{Theo, Kathi}
Heinz	Frida	{Franz}



Vater	Mutter	Kinder
Johann	Martha	Elsa
Johann	Martha	Luca
Johann	Martha	Josef
Johann	Maria	Theo
Johann	Maria	Kathi
Heinz	Frida	Franz

# 1. Normalform (1NF)

Eine Relation befindet sich in der 1. Normalform, wenn

- sie zweidimensional ist (Reihen und Spalten)
- wenn jeder Datensatz einen Sachverhalt der realen Welt repräsentiert und nur einmal vorkommt
- die Daten in einer Spalte einem Attribut entsprechen und das Attribut nur einmal in der Relation (Tabelle) vorkommt
- für jedes Attribut nur ein Wert eingetragen ist

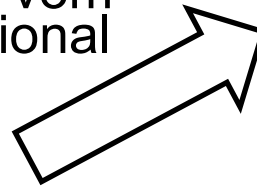
Bilden der 1NF:

Entfernen aller Mehrfacheinträge in einem Attribut. Jedem Attribut eines Datensatzes darf höchstens ein Wert (pro Zeile) zugewiesen sein.

## 2. Normalform (2NF)

Eine Relation ist in 2NF wenn

- sie in 1NF ist und
- jedes Nicht-Schlüssel Attribut vom Primärschlüssel (der auch aus mehreren Feldern bestehen kann) voll funktional abhängig ist. Wichtig ist, dass Datenfelder nicht nur von einem Teilschlüsselfeld, sondern vom gesamten Schlüssel funktional abhängig sind.



1NF

R.-Nr.	Datum	Name	Vorname	Straße	Hnr.	PLZ	Ort	Artikel	Anzahl	Preis	Währung
187	01.01.2012	Mustermann	Max	Musterstr.	1	12345	Musterort	Bleistift	5	1,00	Euro

2NF

Rechnung		
R.-Nr.	Datum	Knr.
187	01.01.2012	007

Kunde						
Knr.	Name	Vorname	Straße	Hnr.	PLZ	Ort
007	Mustermann	Max	Musterstr.	1	12345	Musterort

Rechnungsposition			
R.-P.-Nr.	R.-Nr.	Art.-Nr.	Anzahl
1	187	69	5

Artikel		
Art.-Nr.	Artikel	Preis
69	Bleistift	1,00

## 2. Normalform (2NF)

Vorgehensweise bei der Transformation von 1NF in 2NF:

- Zerlegen der Relation in kleinere Relationen, so dass in jeder Relation alle Nicht-Schlüsselfelder nur noch vom Primärschlüssel abhängen.
- Überprüfen des Primärschlüssels, wenn dieser aus mehreren Attributen besteht: Wenn es Attribute gibt, die nur von einem Teil des Primärschlüssels abhängen, so sind dieser Teil des Primärschlüssels und die zugehörigen Attribute in eine neue Relation zu bringen.

## 2. Normalform (2NF)

<u>PersonalNr</u>	Name	Vorname	<u>AbtNr</u>	AbtBez	<u>ProjNr</u>	ProjBez	Tätigkeit	Stunden	Stundenlohn
0001	Eichenu	Maria	1	Personal				25	50,00
0002	Glahn	Stefanie	2	Einkauf	1	Umfrage	Leiterin	55	30,00
0002	Glahn	Stefanie	2	Einkauf	3	Analyse	Bearbeiterin	70	30,00
0003	Kirsch	Karin	2	Einkauf	1	Umfrage	Bearbeiterin	25	50,00
0004	Wiedner	Karl	3	Verkauf	2	Messe	Leiter	160	35,00
0004	Wiedner	Karl	3	Verkauf	3	Analyse	Leiter	25	50,00
0005	Leitner	Josef	3	Verkauf	2	Messe	Vortrag	80	30,00

Hier 1NF, jedoch 2NF verletzt, da zusammengesetzter Primärschlüssel (PersonalNr, AbtNr, ProjNr) Abhängigkeiten erzeugt, die 2NF verletzen!

## 2. Normalform (2NF)

- Der Name von Mitarbeitern hängt von nicht relevanten Attributen ab, z.B.: AbtNr od. ProjNr.
- AbtBez hängt nicht vom gesamten Primärschlüssel ab, sondern nur von AbtNr, genauso wie ProjBez nur von ProjNr abhängt.
- Der Sachverhalt, welcher Mitarbeiter in welcher Funktion an welchem Projekt arbeitet ist ebenfalls in eine eigene Relation zu verlagern.

Dadurch ergeben sich 4 neue Relationen.

Personal/Mitarbeiter

Attribut	Datenty p
<u>Personal Nr</u>	Zahl
Name	Text
Vorname	Text
AbtNr	Zahl
Attribut	Datenty p
<u>AbtNr</u>	Zahl
Name	Text

Abteilung

Attribut	Datenty p
<u>ProjNr</u>	Zahl
ProjBez	Text

Arbeitet\_An

Attribut	Datenty p
<u>ProjNr</u>	Zahl
<u>PersonalNr</u>	Zahl
Tätigkeit	Text
Stunden	Zahl
Stundenloh n	Währun g

# 3. Normalform (3NF)

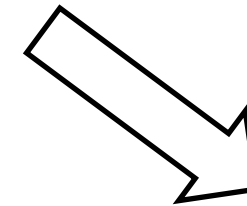
Eine Relation ist in 3NF, wenn sie in der 2NF ist und jedes Nicht-Schlüssel Attribut *nicht* transitiv vom Primärschlüssel abhängig ist.

Kunde						
Knr.	Name	Vorname	Straße	Hnr.	PLZ	Ort
007	Mustermann	Max	Musterstr.	1	12345	Musterort

Alle Datenfelder sind nur vom gesamten Schlüssel abhängig und es treten untereinander keine Abhängigkeiten auf.

Ist ein Nicht-Schlüsselfeld über ein anderes Nicht-Schlüsselfeld identifizierbar, handelt es sich um eine **transitive Abhängigkeit** (man könnte auch von *indirekter Abhängigkeit* sprechen).

Es existiert ein Algorithmus (*Synthesealgorithmus*), der eine Relation in die 3. Normalform transformiert.



Kunde					
Knr.	Name	Vorname	Straße	Hnr.	PLZ
007	Mustermann	Max	Musterstr.	1	12345

Postleitzahl	
PLZ	Ort
12345	Musterort



# 3. Normalform (3NF)

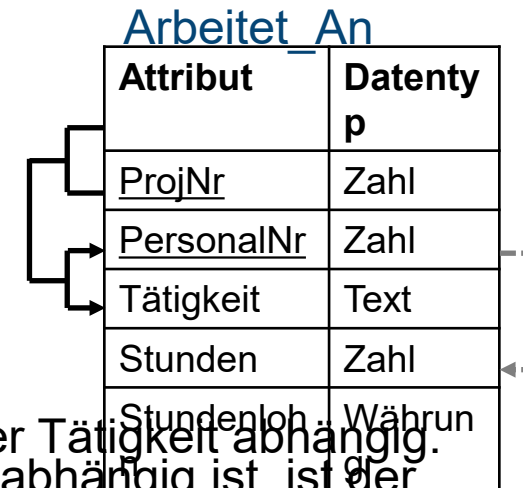
<u>PersonalNr</u>	<u>ProjNr</u>	Tätigkeit	Stunden	Stundenlohn
0002	1	Leiterin	55	50,00
0002	3	Bearbeiterin	70	30,00
0003	1	Bearbeiterin	25	30,00
0004	2	Leiter	160	50,00

In diesem Beispiel ist der Stundenlohn von der Art der Tätigkeit abhängig. Indem dass die Tätigkeit von ProjNr und PersonalNr abhängig ist, ist der Stundenlohn transitiv (indirekt) ebenfalls von den beiden Schlüsselattributen abhängig.

Eine Änderung des Stundenlohns für eine bestimmte Tätigkeit erfordert die Änderung in mehreren Tupeln.

Arbeitet\_An

Attribut	Datentyp
<u>ProjNr</u>	Zahl
<u>PersonalNr</u>	Zahl
Tätigkeit	Text
Stunden	Zahl
Stundenlohn	Währung



# 3. Normalform (3NF)

<u>Personal Nr</u>	<u>ProjN r</u>	Tätigkeits Nr	Stunde n
0002	1	1	55
0002	3	2	70
0003	1	2	25
0004	2	1	160
0004	3	1	25
0005	2	3	80

<u>Tätigkeits Nr</u>	Tätigkeit	Stundenlo hn
1	LeiterIn	50,00
2	Bearbeiteri n	30,00
3	Vortrag	35,00

Um von 2NF in 3NF zu transformieren sind alle transitiven Abhängigkeiten durch Teilen der Relation in mehrere Relationen zu entfernen.

Dabei müssen Nicht-Schlüsselfelder immer direkt vom gesamten Schlüsselfeld abhängig sein.

# 3. Normalform (3NF)

Zusätzliche Schlüsselfelder:  
Lange Textfelder sind für Schlüsselfelder ungeeignet, da sie mehr Speicherplatz für den Index benötigen. Außerdem zieht eine Änderung des Textes auch eine Änderung in den Relationen nach sich, die in Beziehung zu dieser Relation stehen.

Durch Hinzufügen eines weiteren Schlüsselfeldes kann dieses Merkmal umgangen werden.

<u>Tätigkeits Nr</u>	Tätigkeit	Stundenlohn
1	LeiterIn	50,00
2	BearbeiterIn	30,00
3	Vertrag	33,00

Dabei ist es ratsam, *numerische, ganzzahlige Schlüssel* zu verwenden, da textbasierte Schlüssel anfällig für Benutzerfehler sind und SQL Funktionen zur automatischen Schlüsselgenerierung nur teilweise verfügbar sind.

# Boyce-Codd Normalform (BCNF)

- Die einzelnen Abhängigkeiten von einzelnen Schlüsseln oder Schlüsselattributen untereinander werden bis zu 3NF nicht berücksichtigt. In der BCNF werden sie beseitigt.
- Eine Relation befindet sich in BCNF, wenn kein Attribut funktional abhängig von einer Attributgruppe ohne Schlüsseleigenschaft ist.
- Das bedeutet, dass in einer Relation keine funktionalen Abhängigkeiten zwischen einem minimalen Schlüssel (nicht zusammengesetzten Schlüssel) und den Attributen bestehen dürfen. Die meisten Relationen der 3NF befinden sich bereits in BCNF. Nur die Relationen, die mehrere überlappende, minimale Schlüssel besitzen, die eine funktionale Abhängigkeit erzeugen und somit wiederum Redundanzen hervorrufen, sind dzu überführen.

# Boyce-Codd Normalform (BCNF)

Ein Relationenschema ist in BCNF, wenn es in der 3. NF ist und jede *Determinante* (Attributmenge, von der andere Attribute funktional abhängen) ein Superschlüssel ist (oder die Abhängigkeit ist trivial).

Alternative Formulierung: wenn für alle funktionalen Abhängigkeiten  $\alpha \rightarrow \beta$ , die auf R gelten, gilt: Die funktionale Abhängigkeit ist trivial oder  $\alpha$  ist ein Superschlüssel für R.

Die BCNF verhindert, dass Teile zweier aus mehreren Feldern zusammengesetzten Schlüsselkandidaten voneinander abhängig sind.

Es existiert ein Zerlegungsalgorithmus der relationale Schemata in die Boyce-Codd Normalform überführt.

# 4. Normalform (4NF)

- In der 4. Normalform befinden sich Relationen, die sich in der 3NF befinden und die keine paarweise auftretenden mehrwertigen Abhängigkeiten der Attribute enthalten.
- Eine mehrwertige Abhängigkeit besteht, wenn mehrere Nicht-Schlüsselattribute direkt vom Schlüssel abhängig sind, aber nicht voneinander abhängig sind.

<u>PersonalNr</u>	ProjektNr	Kinder
0002	1	Hans
0002	1	Gerd
0003	2	Johanna
0003	2	Jens
0003	3	Johanna
0003	3	Jens

In der Tabelle ist sowohl das Attribut ProjektNr als auch das Attribut Kinder direkt vom Attribut PersonalNr abhängig.

## 4. Normalform (4NF)

- Ist es möglich, eine Relation in zwei Relationen zu zerlegen und die ursprünglich Relation durch einen Natural Join dieser beiden Relationen wiederherzustellen, befindet sich die Ausgangsrelation noch nicht in der 4. Normalform.
- Eine Zerlegung in die 4. Normalform ist nicht immer möglich bzw. nötig. Wenn zwischen den Attributen inhaltliche Zusammenhänge bestehen, kann sich die Relation bereits in der 4NF befinden.

## 4. Normalform (4NF)

**Ein Relationenschema ist dann in der 4. Normalform, wenn es in der BCNF ist und nur noch triviale mehrwertige Abhängigkeiten (MWA) enthält.**

**Einfach ausgedrückt: Es darf innerhalb einer Relation nicht mehrere  $1:n$  oder  $m:n$ -Beziehungen zu einem Schlüsselwert geben, die thematisch/inhaltlich nichts miteinander zu tun haben. Gehört etwa zu einem Schlüsselwert  $i$ -mal Attribut  $a$ , aber davon UNABHÄNGIG auch  $j$ -mal Attribut  $b$ , ist die 4NF verletzt.**

**Anschaulich ausgedrückt: Die 4NF untersucht n-äre Beziehungen (mehr als zwei Tabellen stehen gleichzeitig in Beziehung) und ob diese korrekt modelliert wurden.**

**Eine mehrwertige Abhängigkeit (engl. multivalued dependency (MVD))**

**$\alpha \twoheadrightarrow \beta$  beschreibt die Abhängigkeit einer Menge von Attributen  $\beta$  von einer Menge aus Attributen  $\alpha$ .**



## 5. Normalform (5NF)

- Ist eine verlustlose Zerlegung in Einzelabhängigkeiten in der 4NF nicht möglich, werden in der 5NF weitere Primärschlüssel hinzugefügt. Dies geschieht so lange, bis nur noch Einzelabhängigkeiten der Attribute von einem oder mehreren Primärschlüsseln bestehen (keine Join Dependencies). Kann eine Relation nicht durch einen Join aus Einzelrelationen erstellt werden, befindet sie sich in 5NF.
- Die 4NF, 5NF sowie BCNF führen zu einer größeren Menge an Relationen.

## 5. Normalform (5NF)

Eine Relation ist in 5NF, wenn sie in der 4NF ist und keine mehrwertigen Abhängigkeiten enthält, die voneinander abhängig sind.

Einfach ausgedrückt: Es darf innerhalb einer Relation nicht mehrere  $1:n$ - oder  $m:n$ -Beziehungen zu einem Schlüsselwert geben, die thematisch/inhaltlich miteinander verknüpft sind. Gehört etwa zu einem Schlüsselwert  $i$ -mal Attribut  $a$ , aber davon ABHÄNGIG auch  $j$ -mal Attribut  $b$ , ist die 5NF verletzt.

# 5. Normalform (5NF)

Die räumliche Zuordnung von Arbeitsgruppen und Arbeitsplätzen ist in einer Relation gespeichert.

Diese Relation befindet sich in 3NF, da der einzige Primärschlüssel die Kombination aller drei Attribute ist.

Es liegen aber Redundanzen vor, da jede Arbeitsgruppe mehrere Arbeitsplätze benötigt und es in jedem Raum mehrere Arbeitsplätze gibt.

Der Primärschlüssel besitzt somit mehrwertige Abhängigkeiten.

<u>Arbeitsgruppe</u>	Raum	Platz
1	1	1
1	1	2
1	1	3
2	1	4
2	1	5
3	3	1
3	3	2
3	3	3

# 5. Normalform (5NF)

- Die mehrwertigen Abhängigkeiten lassen sich durch eine Aufteilung in zwei Tabellen beseitigen.
- Die beiden Ergebnisrelationen befinden sich dann in 4NF und ebenfalls in 5NF, da eine verlustfreie Zerlegung möglich ist.

<u>Arbeitsgruppe_Raum</u>	<u>Arbeitsgruppe</u>	<u>Raum</u>
11	1	1
21	2	1
33	3	3

<u>Arbeitsgruppe_Raum</u>	<u>Platz</u>
11	1
11	2
11	3
21	4
21	5
33	1
33	2
33	3

# 3NF Merkgeregeln

1. Ist die Relation in 1. Normalform und besteht der Primärschlüssel aus nur einem Attribut, so liegt automatisch die 2. Normalform vor.
2. Ist eine Relation in 2. Normalform und besitzt sie außer dem Primärschlüssel höchstens ein weiteres Attribut, so liegt die Tabelle in 3. Normalform vor.
3. Die 3. Normalform ist die letzte und muss die 1. und 2. Bedingung erfüllen (die Werte hängen nur vom Schlüssel ab, und nicht von Nichtschlüsselattributen, und keine Transitivitäten enthalten).

aus Wikipedia,

[https://de.wikipedia.org/wiki/Normalisierung\\_\(Datenbank\)](https://de.wikipedia.org/wiki/Normalisierung_(Datenbank))

# Praktische Umsetzung der NF

# 1. Normalform - Regel

Regel:

Jedes Attribut kann nur 1x auftreten. (Gilt auch für Gruppen von Attributen mit vergleichbarem Inhalt)

D.h. 1 : N-Beziehungen sind in einer Entität nicht erlaubt.

Sie bilden eigene Entitäten unter Mitnahme des Schlüsselbegriffs

Aus der ursprünglichen Entität Person werden in der 1. Normform die drei Entitäten Person, Kind und Zulagen gebildet.

# 1. Normalform PERSONAL

[1.NF Personal.jpg](#)

Personaldatenbank.xls – Register 2



# 2. Normalform – Regel

Bei Entitäten mit zusammengesetztem Schlüssel muß jedes Attribut von allen Schlüsselteilen direkt abhängig sein.

Durch die 2. Normalform entstehen zwei neue Entitäten: Fähigkeit und Zulage-Daten.

Die Felder

FBEZ (Fähigkeitsbezeichnung) bzw.

GZUBEZ (Bezeichnung der Gehaltszulage) und

GZULAG (Gehaltszulagenbetrag)

hängen jeweils nur vom zweiten Schlüsselteil ab: {Nummer der Fähigkeit} bzw. GZUART (Gehaltszulagenart)

des zusammengesetzten Schlüssels ab.

# 2. Normalform PERSONAL

[2.NF Personal.jpg](#)

Personaldatenbank.xls – Register 3

# 3. Normalform Regel

Ein Attribut darf nicht direkt von einem anderen Nicht-Schlüssel-Attribut abhängig sein.

In unserem Beispiel sind die Felder BAKNAM (Bankname) und BAKORT (Bankort) eindeutig vom Nichtschlüsselattribut BANKL (Bankleitzahl) abhängig. Das Gleiche trifft für das Feld GEBANS: (Gebäudeanschrift zu, das vom Feld GEBSL (Gebäudeschlüssel abhängig ist. In beiden Fällen werden neue Entitäten gebildet.

Das Feld KSTLKAME (Name des Kostenstellenleiters) kann, da von *P#* eindeutig abhängig, entfallen. Die Abhängigkeit existiert bereits in der Entität Person.

# 3. Normalform PERSONAL

[3.NF Personal.jpg](#)

Personaldatenbank.xls – Register 4

Für dieses Beispiel haben Sie bereits die Datenanalyse gegeben:

Baufirma.xls

erstellen Sie die 3.NF