

**Fachbereich
Automatisierung und Informatik**



„Datenbank – Grundlagen“

Normalisierungsprozess

**Dipl. Inf., Dipl.-Ing. (FH) Michael Wilhelm
Friedrichstraße 57 - 59
38855 Wernigerode**

**Raum: 2.202
Tel.: 03943/659-338
Fax: 03943/659-399
Email: mwilhelm@hs-harz.de**

Inhaltsverzeichnis

1	Normalisierungsprozess	3
1.1	Redundanzfreie Speicherung	3
1.2	Abhängigkeiten	3
1.2.1	Funktionale Abhängigkeit	3
1.2.2	Volle Abhängigkeit	4
1.2.3	Transitive Abhängigkeit	5
2	Erste Normalform	6
3	Zweite Normalform	8
4	Dritte Normalform	10
5	Indexverzeichnis	11

1 Normalisierungsprozess

Definition 1:

Der Normalisierungsprozess bezweckt die redundanzfreie Speicherung von Informationen innerhalb einer Datenbank. Dies geschieht durch die Aufteilung und Umgruppierung der Attribute innerhalb der Tabellen.

1.1 Redundanzfreie Speicherung

Definition 2:

Die redundanzfreie Speicherung bedeutet, dass kein Teil eines Datenbestandes weggelassen werden kann, ohne dass es zu Informationsverlusten führt.

Die redundanzfreie Speicherung verringert den Speicherplatz einer Datenbank und verhindert Mutationsanomalien.

Beispiel:

PNr	Name	Vorname	Automarke	Typ
1234	Müller	Herbert	Opel	Astra
2345	Meier	Hans	Toyota	Yaris
3456	Schmid	Petra	Porsche	911
1234	Müller	Herbert	Seat	Leon

In der Tabelle sind Autobesitzer mit Personalnummer, Name, Vorname und deren Autodaten gespeichert. Da Herr Müller zwei Autos besitzt, muss er zweimal in der Tabelle gespeichert werden. Damit sind diese Daten redundant gespeichert. Bei einer Änderung müssen beide Datensätze verändert werden. Falls diese Adressenänderung nur einmal vorgenommen wird, entsteht eine Mutationsanomalie, weil es zwei Müller gibt, deren Personalnummer identisch, deren Adresse aber unterschiedlich ist. Damit ist eine Datenkonsistenz nicht gewährleistet.

1.2 Abhängigkeiten

Innerhalb einer Tabelle unterscheidet man drei Abhängigkeiten.

- Funktionale Abhängigkeit
- Volle Abhängigkeit
- Transitive Abhängigkeit

Diese Abhängigkeiten beziehen sich immer auf die Attribute innerhalb einer Tabelle.

1.2.1 Funktionale Abhängigkeit

Definition 3:

*Ein Attribut bzw. eine Attributkombination B ist dann von einem Attribut oder einer Attributkombination A **funktional abhängig**, wenn zu einem bestimmten Attributwert von A **genau ein** Attributwert von B gehört. Aus dem Attributwert von A ergibt sich also eindeutig der Attributwert von B.*

Definition 4:- funktionale Abhängigkeit:

Ein Attribut B ist funktional abhängig von Attribut A, wenn zu einem bestimmten Wert von A höchstens ein Wert von B möglich ist.

$F: A \rightarrow B$

Einfache Schreibweise $A \rightarrow B$ bzw. $R.A \rightarrow R.B$.

Beispiel:

In der Tabelle „Personen“ (PNr, Name, PLZ, Ort) ist das Attribut „Name“ funktional voll abhängig vom Attribut „PNr“, da jeder Name mit einer bestimmten Personalnummer verknüpft ist. Es ist auch nicht möglich, die Telefonnummer aus dem Namen zu bestimmen.

1.2.2 Volle Abhängigkeit

Definition 5:

Ein Attribut bzw. eine Attributkombination B ist dann von einem Attribut oder einer Attributkombination A **voll abhängig**, wenn B nur von A, nicht jedoch schon von einem Teil der Attributkombination funktional abhängig ist.

Definition 6:- volle funktionelle Abhängigkeit:

Bei einem zusammengesetzten Identifikationsschlüssel ist ein Attribut B von dem Identifikationsschlüssel voll abhängig, wenn B von der gesamten Attributkombination, nicht bereits von Teilen der Attributkombination, funktional abhängig ist.

$A = \{A_1, A_2, A_3, \dots, A_n\}$

$\forall A_i \subset A \exists R.A_i \rightarrow R.B$

$R.A_i \mapsto R.B$

1. Beispiel: Relation TELEFON(#Name, #Vorname, Telefon):

Name	Vorname	Telefon
Meier	Franz	77
Meier	Erhard	94
Müller	Josef	94
Schmidt	Erhard	44

Im obigen Beispiel ist die Telefonnummer funktional abhängig von Primärschlüssel „Name,Vorname“. Aus den Schlüssel kann eindeutig auf die Telefonnummer geschlossen werden. Dies gilt aber nicht im umgekehrten Fall, da die Telefonnummer 94 doppelt vergeben wurde.

Telefon = f(Name, Vorname)

2. Beispiel:

In der Tabelle „Verkauf“ (KNr, ANr, Kaufdatum, Name) ist das Kaufdatum voll abhängig vom ID-Schlüssel (KNr, ANr), weil das Kaufdatum weder vom Attribut KNr noch vom Attribut ANr funktional abhängig ist. Das Attribut Name ist nicht voll vom ID-Schlüssel abhängig, denn der Name gehört zur Kundennummer.

1.2.3 Transitive Abhängigkeit

Definition 7:

Ein Attribut bzw. eine Attributkombination C ist von einem Attribut oder einer Attributkombination A **transitiv abhängig**, wenn das Attribut B von A und das Attribut C von B funktional abhängig ist, aber nicht von C funktional abhängig ist.

Es gilt:

$$A \Rightarrow B$$

$$B \Rightarrow C$$

Dann gilt: $A \Rightarrow C$

Definition 8:- transitive Abhängigkeit:

Ein Attribut C ist transitiv abhängig von einem Attribut A , wenn es funktional abhängig ist von Attribut B , und das Attribut B funktional, voll funktional oder transitiv abhängig ist von Attribut A .

1. Beispiel:

In der Tabelle „Personen“ (PNr, AbtNr, Abteilung) ist das Attribut „Abteilung“ vom Attribut „PNr“ transitiv abhängig, weil Abteilung von „AbtNr“ und „AbtNr“ von „PNr“ funktional abhängig ist. „PNr“ ist aber nicht von der Abteilungsbezeichnung abhängig.

es gilt: $\text{Abteilung} \Rightarrow \text{AbtNr}$
 $\text{AbtNr} \Rightarrow \text{PNr}$

dann folgt daraus

$$\text{Abteilung} \Rightarrow \text{PNr}$$

2. Beispiel:

Relation MITARBEITER(#Personalnummer, Name, Vorname, Abteilung):

PersonalNr	Name	Vorname	Abteilung
653	Meier	Franz	47
697	Meier	Erhard	53
703	Müller	Josef	53

Tabelle 16.: Relation MITARBEITER

2 Erste Normalform

Definition 9: - 1. Normalform:

Eine Relation ist in erster Normalform, wenn die den Tupeltyp bildenden Attribute Datenprimitiva einfachen Datentyps sind.

Bei dieser Begriffsbestimmung wird die Einteilung der Datentypen verwendet, wie sie beispielsweise in der Programmiersprache Java, C++ und PASCAL verwendet wird.

Definition 10: - einfacher Datentyp:

Ein einfacher Datentyp ist durch einen Wertebereich charakterisiert, dessen Werte elementar sind, in dem Sinne, dass sie nicht weiter zerlegt werden können.

1. Beispiel:

Relation TEMPERATUR(#Ort, #Tag, Temperatur):

Ort	Tag	Temperatur
		07 14 21 Uhr
Halle	04-JAN-2000	4 12 5
Wernigerode	04-JAN-2000	3 13 4

Tabelle Temperatur

Diese Beispielrelation verstößt gegen die erste Normalform, da die Werte des Attributs Temperatur nicht elementar, bzw. atomar sind.

Ein mögliche Variante wäre:

Ort	Tag	Uhrzeit	Temperatur
Halle	04-JAN-2000	07	4
Halle	04-JAN-2000	14	12
Halle	04-JAN-2000	21	5
Wernigerode	04-JAN-2000	7	3
Wernigerode	04-JAN-2000	14	13
Wernigerode	04-JAN-2000	21	4

Tabelle Temperatur

2. Beispiel:

Der Normalisierungsprozess läuft über verschiedene Stufen und soll am Beispiel einer Autoverkaufsfirma erläutert werden.

Alle Autos sollen mit Marke, Typ und Seriennummer eingetragen werden. Alle Verkäufer und alle Kunden sollen erfasst werden. Ein Kunden muss aber mindestens ein Auto gekauft haben, bevor er in der Datenbank erfasst werden kann.

Kundenname	Adresse	Automarke	Typ	Seriennr	Verkäufer	Datum
Meier	Planetenweg 7	VW Opel	Golf Astra	123456 345678	Schmid Peter	23.02.2002 07.08. 2002
Müller	Altstadt 12	VW	Golf	388721	Frey	17.06. 2002
Steffen	Gartenstr. 7	VW	Polo	222245	Schmid	15.07. 2002
Steffen	Augasse 12	Audi	A8	122154	Frey	13.11. 2002
		Opel	Vectra	445321		
					Schenk	

Bemerkungen:

Der Opel Vectra wurde noch nicht verkauft, muss Zwecks aber Inventur in der Liste stehen. Der Verkäufer Schenk hat zwar noch kein Auto verkauft, sollte aber auch aus ähnlichen Gründen in der Liste erscheinen. Die weitere Frage ist, ob die Kunden „Steffen“ und „Meier“ den selben Vertreter „Schmid“ als Ansprechpartner haben. Oder ob es sich um zwei Personen handelt.

Schlussfolgerung:

Einführen weitere Attribute (KNr, ANr, VNr):

KNr	Kunden name	Adresse	ANr	Auto-marke	Typ	Seriennr	VNr	Verkäufer	Datum
1	Meier	Planetenweg 7	1	VW	Golf	123456	1	Schmid	23.02.2002
1	Meier	Planetenweg 7	2	Opel	Astra	345678	2	Peter	07.08. 2002
2	Müller	Altstadt 12	3	VW	Golf	388721	3	Frey	17.06. 2002
3	Steffen	Gartenstr. 7	4	VW	Polo	222245	1	Schmid	15.07. 2002
4	Steffen	Augasse 12	5	Audi	A8	122154	3	Frey	13.11. 2002
			6	Opel	Vectra	445321			
							4	Schenk	

Tabelle „Geschäftsdaten“ in der 1. Normalform

Neu eingefügt wurden Nummern für die Kunden, Autos und Verkäufer.

Definition 11:

Eine Tabelle befindet sich in der 1. Normalform, wenn alle Attribute nur einfache Attributwerte aufweisen, wobei auch Nullwerte zugelassig sind.

Weitere Bemerkungen:

Es sind immer noch Redundanzen in der Tabelle. Die Namen bzw. Adressen von Meier und Steffen sind doppelt. Ebenso erscheint der Verkäufer Schmid mehrfach.

Des Weiteren existieren in der Tabelle Gruppen von Attributen, die unabhängig von einander existieren können. Zum Beispiel sind die Autoattribute unabhängig vom Kunden.

3 Zweite Normalform

Die zweite Normalform betrifft nur Tabellen mit einer Kombination von Attributen für den ID-Schlüssel. Als Kriterium für zweite Normalform gilt, dass alle nicht zum ID-Schlüssel gehörigen Attribute einer Tabelle vom ganzen ID-Schlüssel und nicht nur von einzelnen Attributen davon funktional abhängig sein müssen.

Definition 12: - 2. Normalform:

Eine Relation liegt in zweiten Normalform vor, wenn sie sich in ersten Normalform befindet und wenn für die Menge A der Attribute der Identifikationsschlüsselattribute gilt:

$R.A \rightarrow R.B$

wobei die Menge B alle Nichtschlüsselattribute umfasst.

oder

Definition 13:

Eine Tabelle befindet sich in der zweiten Normalform, wenn sie schon in der ersten Normalform ist und jedes **nicht** zum ID-Schlüssel gehörende Attribut **voll** vom ID-Schlüssel abhängig ist. Es können sich also nur Tabellen mit zusammengesetzten ID-Schlüsseln in der zweiten Normalform befinden.

Es folgen die nach Gebieten aufgeteilten Tabellen der „Geschäftsdaten“:

KNr	Kunden name	Adresse
1	Meier	Planetenweg 7
2	Müller	Altstadt 12
3	Steffen	Gartenstr. 7
4	Steffen	Augasse 12

Tabelle Kunden

ANr	Automarke	Typ	Seriennr
1	VW	Golf	123456
2	Opel	Astra	345678
3	VW	Golf	388721
4	VW	Polo	222245
5	Audi	A8	122154
6	Opel	Vectra	445321

Tabelle Autos

KNr	ANr	VNr	Verkäufer	Datum
1	1	1	Schmid	23.02. 2002
1	2	2	Peter	07.08. 2002
2	3	3	Frey	17.06. 2002
3	4	1	Schmid	15.07. 2002
4	5	3	Frey	13.11. 2002

Tabelle Verkäufe

Datenbankschreibeweise:

Kunden(#KNr, Kundenname, Adresse)

Autos(#ANr, Automarke, Seriennummer)

Verkäufe(#KNr, #ANr, Datum, VNr, Verkäufer)

Für die zweite Normalform sind nur Tabellen mit zusammengesetzten Schlüssel zu betrachten. Dieses wäre die Tabelle „Verkäufe“. Die Attribute „Datum“, „VNr“, „Verkäufer“ sind voll vom zusammengesetzten ID-Schlüssel abhängen. Somit befindet sich diese Tabelle mindestens in der zweiten Normalform.

Frage:

Ist der Datenbestand komplett erhalten geblieben?

2. Beispiel:

Relation LAGER(#Lager, Adresse, Teil, Menge):

Lager	Adresse	Teil	Menge
1	Hubergasse 12	101	25
3	Krugstraße 34	102	410
1	Waagegasse 10	102	30
4	Brunnen-straße 2	103	10

Tabelle 13.: Relation LAGER (alt)

In dieser Relation sind für das Lager 1 zwei Adressen angegeben (Datenanomalie). Weiterhin gehen durch das Löschen, z.B. des Teils 102, aus dem Lager 3 auch die anderen Daten über das Lager 3, z.B. die Adresse verloren.

Die Nichtschlüsselattribute, z.B. Menge, sind nicht voll funktional abhängig vom Identifikationsschlüssel.

Ausweg bietet die Bildung von zwei Relationen:

Relation LAGER(#Lager, Adresse):

Lager	Adresse
1	Hubergasse 12
3	Krugstraße 34
4	Brunnen-straße 2

Tabelle 14.: Relation LAGER (neu)

Relation MENGE(#Lager, #Teil, Menge):

Lager	Teil	Menge
1	101	25
3	102	410
1	102	30
4	103	10

Tabelle 15.: Relation MENGE

4 Dritte Normalform

Definition 14:

Bei der dritten Normalform werden nun auch die Abhängigkeiten der nicht zum ID-Schlüssel einer Tabelle gehörenden Attribute untereinander untersucht. Dabei gilt, dass kein Nichtschlüssel-Attribut von einem anderen Nichtschlüssel-Attribut funktionell abhängig sein darf.

oder

Definition 15: - 3. Normalform:

Eine Relation ist in dritten Normalform, wenn sie sich in ersten Normalform befindet und kein Nichtschlüsselattribut transitiv abhängig ist vom Identifikationsschlüssel.

Eine Relation in dritten Normalform befindet sich automatisch auch in zweiten Normalform.

oder

Definition 16:

Ein Tabelle befindet sich in der dritten Normalform, wenn sie schon in der zweiten Normalform (bzw. mit einfachen ID-Schlüssel in der ersten Normalform) ist und **kein** Nichtschlüssel-Attribut vom ID-Schlüssel **transitiv** abhängig ist.

Für das Auto-Beispiel gilt, dass die Tabelle „Auto“ und „Kunden“ sich in der dritten Normalform befinden. Für die Tabelle „Verkäufe“ gilt das nicht. Hier kann man aus der Verkäufersnummer auf den Verkäufersnamen schließen. Die Verkäufersnummer ihrerseits ist aber vom ID-Schlüssel „KNR, ANr“ funktional abhängig. Somit besteht eine transitive Abhängigkeit zwischen dem ID-Schlüssel KNR, ANr und dem Attribut „Verkäufer“. Die Tabelle muss weiter aufgespalten werden.

KNr	ANr	Datum	VNr
1	1	23.02. 2002	1
1	2	07.08. 2002	2
2	3	17.06. 2002	3
3	4	15.07. 2002	1
4	5	13.11. 2002	3

Tabelle Verkäufe

VNr	Verkäufer
1	Schmid
2	Peter
3	Frey
4	Schenk

Tabelle Verkäufer

Datenbankschreibeweise:

Verkäufe (#KNr, #ANr, Datum, VNr)

Verkäufer (#VNr, Verkäufer)

Tabellen, welche sich in der dritten Normalform befinden, werden als normalisiert bezeichnet. Die dargestellten Informationen sind redundanzfrei.

5 Indexverzeichnis

	A	
Abhängigkeit		3
	D	
Dritte Normalformt		10
	E	
Erste Normalform		6
	F	
Funktionale Abhängigkeit		3
	N	
Normalierungsprozess		3
	R	
Redundanz		3
	T	
Transitive Abhängigkeit		5
	V	
Volle Abhängigkeit		4
	Z	
Zweite Normalform		8