




Eine Normalisierung sollte in jedem Fall mindestens theoretisch durchgeführt werden, um alle Redundanzen zu erkennen. Ob diese normalisierte Datenbank dann auch implementiert wird, muß nach Abwägung von Platzgewinn und zusätzlicher Sicherheit gegenüber den komplizierteren Abfragen entschieden werden. In vielen Fällen ist kontrollierte Redundanz ein guter Kompromiss.

	<b>Redundanz</b> Ein Eintrag in einer Datenbasis ist <b>redundant</b> , wenn er in der Datenbasis mehrfach gespeichert oder aus bereits vorhandenen Einträgen der Datenbasis abgeleitet werden kann. Redundanz ist aus mindestens zwei Gründen unerwünscht: <ul style="list-style-type: none"> <li>• sie kann zu sogenannten Anomalien (Mutationsanomalien) führen,</li> <li>• sie führt zu einer schlechteren Ausnutzung des Speicherplatzes.</li> </ul>
	<b>Anomalie</b> <b>Anomalien</b> sind inkonsistente Datenbasiszustände, die durch Einfügen, Löschen oder Ändern von Datenbasiseinträgen hervorgerufen werden. Eine typische Anomalie ist z.B. die Änderung nur an einem von mehreren redundanten Einträgen.
	<b>Normalisierung</b> Ziel der <b>Normalisierung</b> ist, Redundanz und damit Anomalien zu vermeiden. Die Normalisierung ist ein notwendiger Schritt bei der Datenmodellierung, denn sie liefert eine formale Beschreibung der vorhandenen Redundanz in der Datenbank. Normalisierung kann für jedes Datenmodell durchgeführt werden; meist wird sie am Beispiel des relationalen Modells erklärt.

Grundlage der Normalisierung sind Normalformen (häufig NF abgekürzt). Die erste Normalform bezieht sich auf die Struktur einzelner Attribute, die höheren Normalformen auf Abhängigkeiten zwischen Attributen sowie Schlüsseigenschaften von Attributen. Im Zuge der Normalisierung werden Relationen solange in neue Relationen zerlegt, bis die Datenbasis den vorgegebenen Normalformen genügt.

① . Normalform				
Eine Relation ist in 1. Normalform, wenn alle ihre Attribute einfache (skalare oder atomar) Werte sind, d.h. sie haben keine innere Struktur. <table border="1" data-bbox="199 1317 1369 1615"> <tr> <th data-bbox="199 1319 1369 1370">② . Normalform</th></tr> <tr> <td data-bbox="199 1370 1369 1462">                     Eine Relation ist in 2. Normalform, wenn sie in 1. NF ist und jedes nicht zum Identifikationsschlüssel gehörige Attribut voll funktional** von diesem abhängig ist.                     <table border="1" data-bbox="236 1462 1310 1585"> <tr> <th data-bbox="236 1464 1310 1516">③ . Normalform</th></tr> <tr> <td data-bbox="236 1516 1310 1588">                         Eine Relation ist in 3. Normalform, wenn sie in 2. NF ist und kein Attribut, das nicht zum Identifikationsschlüssel gehört, transitiv*** von diesem abhängt.                     </td></tr> </table> </td></tr> </table>	② . Normalform	Eine Relation ist in 2. Normalform, wenn sie in 1. NF ist und jedes nicht zum Identifikationsschlüssel gehörige Attribut voll funktional** von diesem abhängig ist. <table border="1" data-bbox="236 1462 1310 1585"> <tr> <th data-bbox="236 1464 1310 1516">③ . Normalform</th></tr> <tr> <td data-bbox="236 1516 1310 1588">                         Eine Relation ist in 3. Normalform, wenn sie in 2. NF ist und kein Attribut, das nicht zum Identifikationsschlüssel gehört, transitiv*** von diesem abhängt.                     </td></tr> </table>	③ . Normalform	Eine Relation ist in 3. Normalform, wenn sie in 2. NF ist und kein Attribut, das nicht zum Identifikationsschlüssel gehört, transitiv*** von diesem abhängt.
② . Normalform				
Eine Relation ist in 2. Normalform, wenn sie in 1. NF ist und jedes nicht zum Identifikationsschlüssel gehörige Attribut voll funktional** von diesem abhängig ist. <table border="1" data-bbox="236 1462 1310 1585"> <tr> <th data-bbox="236 1464 1310 1516">③ . Normalform</th></tr> <tr> <td data-bbox="236 1516 1310 1588">                         Eine Relation ist in 3. Normalform, wenn sie in 2. NF ist und kein Attribut, das nicht zum Identifikationsschlüssel gehört, transitiv*** von diesem abhängt.                     </td></tr> </table>	③ . Normalform	Eine Relation ist in 3. Normalform, wenn sie in 2. NF ist und kein Attribut, das nicht zum Identifikationsschlüssel gehört, transitiv*** von diesem abhängt.		
③ . Normalform				
Eine Relation ist in 3. Normalform, wenn sie in 2. NF ist und kein Attribut, das nicht zum Identifikationsschlüssel gehört, transitiv*** von diesem abhängt.				

\*) Ein Attribut bzw. eine Attributskombination R.B ist **funktional abhängig** von einem Attribut bzw. einer Attributskombination R.A der Relation R, wenn es zu jedem Wert von R.A höchstens einen Wert R.B gibt.

\*\*) R.B ist **voll funktional abhängig** von R.A, wenn es nicht bereits von Teilen von R.A funktional\* abhängig ist.

\*\*\*) R.C ist **transitiv abhängig** von R.A, wenn  $R.A \rightarrow R.B$  und  $R.B \rightarrow R.C$ , nicht aber  $R.B \rightarrow R.A$ .

Nachfolgend werden an einem durchgängigen Beispiel die Schritte der Normalisierung exemplarisch durchgeführt. Ausgangspunkt sind Daten (ohne Schlüsselattribute) die auszugsweise einer Abrechnungstabelle entnommen seien könnten.

Tabelle Arbeit (nicht normalisiert oder auch 0.Normalform)				
Mitarbeiter	Projekt	Tätigkeit	Zeit	Stundenlohn
Adler, Anton	Stadion	Montage	4	10.00€
	Firma	Reinigung, Aufbau	3,2	8.00€, 9.50€
Busch, Bernd	Stadion	Abbau, Reinigung	12,4	9.00€, 8.00€
	Zoo	Test	5	12.00€
	Kino	Fehlersuche, Test	3,4	15.00€, 12.00€

①

Beim Übergang zur 1.Normalform werden alle Zellen mit Mehrfacheinträgen aufgeteilt, so dass nur atomare Informationen enthalten bleiben (z.B.: aus Mitarbeiter wird Name und Vorname). Zusätzlich ist jeder Datensatz über einen Primärschlüssel (hier zusammengesetzter Primarykey aus M\_Nr, P\_Nr und T\_Nr) eindeutig identifizierbar.

Tabelle Arbeit (1.Normalform)								
M_Nr	Name	Vorname	P_Nr	Projekt	T_Nr	Tätigkeit	Zeit	Stundenlohn
1	Adler	Anton	1	Stadion	100	Montage	4	10.00€
1	Adler	Anton	2	Firma	200	Reinigung	3	8.00€
1	Adler	Anton	2	Firma	300	Aufbau	2	9.50€
2	Busch	Bernd	1	Stadion	400	Abbau	12	9.00€
2	Busch	Bernd	1	Stadion	200	Reinigung	4	8.00€
2	Busch	Bernd	3	Zoo	600	Test	5	12.00€
2	Busch	Bernd	4	Kino	500	Fehlersuche	3	15.00€
2	Busch	Bernd	4	Kino	600	Test	4	12.00€

②

Bei der Überführung in die 2.Normalform werden die voll funktionalen Abhängigkeiten (durch Auslagerung) aufgelöst. Es ist einsichtig, dass die Projektbezeichnung schon durch die Projektnummer eindeutig identifiziert werden kann, also nur von der P\_Nr funktional abhängig ist. P\_Nr ist Teil des Primärschlüssels. Somit ist die Projektbezeichnung nur von einem Teil des Primärschlüssels abhängig und nicht vom gesamten (also nicht voll funktional) Schlüssel.

Tabelle Arbeit (2.Normalform)					
T_Nr	Tätigkeit	Zeit	Stundenlohn	T_M_Nr	T_P_Nr
100	Montage	4	10.00€	1	1
200	Reinigung	3	8.00€	1	2
300	Aufbau	2	9.50€	1	2
400	Abbau	12	9.00€	2	1
200	Reinigung	4	8.00€	2	1
600	Test	5	12.00€	2	3
500	Fehlersuche	3	15.00€	2	4
600	Test	4	12.00€	2	4

Mitarbeiter (2NF)		
M_Nr	Name	Vorname
1	Adler	Anton
2	Busch	Bernd

Projekt (2NF)	
P_Nr	Projekt
1	Stadion
2	Firma
3	Zoo
4	Kino

③

Hier wird die transitive Abhängigkeit des Stundenlohns aufgelöst. Der Lohn ist abhängig von der Tätigkeit (und diese ist funktional abhängig von der Tätigkeitsnummer). Somit ist der Stundenlohn indirekt (transitiv) abhängig von T\_Nr und wird ausgelagert.

Tabelle Arbeit (3.Normalform)			
T_Nr	Zeit	T_M_Nr	T_P_Nr
100	4	1	1
200	3	1	2
300	2	1	2
400	12	2	1
200	4	2	1
600	5	2	3
500	3	2	4
600	4	2	4

Tätigkeit (3NF)		
T_Nr	Tätigkeit	Stundenlohn
100	Montage	10.00€
200	Reinigung	8.00€
300	Aufbau	9.50€
400	Abbau	9.00€
500	Fehlersuche	15.00€
600	Test	12.00€

Mitarbeiter (3NF)		
M_Nr	Name	Vorname
1	Adler	Anton
2	Busch	Bernd

Projekt (3NF)	
P_Nr	Projekt
1	Stadion
2	Firma
3	Zoo
4	Kino