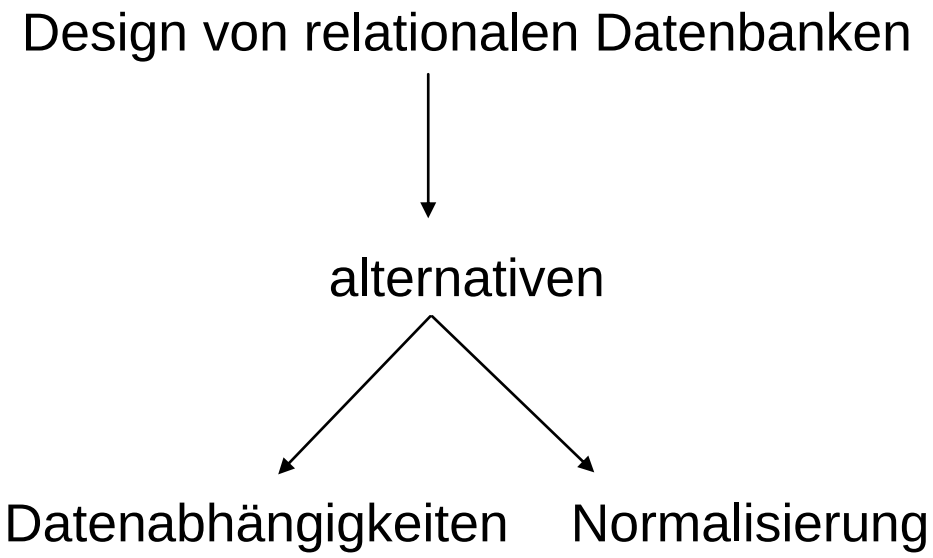
Design Theorie für relationale Datenbanken



Ziel: automatisches Datenbankdesign

Schlechtes Datenbank Design

Frage: Warum sind manche Datenbank-Entwürfe schlecht?

Beispiel: folgendes Relationenschema sei gegeben.

Lieferant (LName, LAdresse, Ware, Preis)

Bei diesem Schema-Entwurf kann es zu verschiedenen Problemen kommen:

1. **Redundanz.** Die Adresse des Lieferanten wird für jede Ware, die er liefert, wiederholt.
2. **Potentielle Inkonsistenz.** Als Konsequenz der Redundanz kann man die Adresse eines Lieferanten in einem Tupel verändern, sie aber in anderen Tupeln unberührt lassen. Auf diese Weise bekommt man zwei verschiedene Adressen für denselben Lieferanten.

3. **Einfüge-Anomalien.** Man kann keine Adresse für einen Lieferanten haben, wenn er nicht mindestens eine Ware liefert. Es ist möglich, in die Ware- und Preiskomponente NULL-Werte für diese Lieferanten einzugeben, dann müssten diese Tupel aber gelöscht werden, wenn eine Ware für diese Lieferanten eingegeben wird.
4. **Lösch-Anomalien.** Invers zu Problem 3 kann es passieren, dass man alle Waren eines Lieferanten löscht, und dadurch auch seine Adresse verliert.

Ein besseres Schema für dieses Beispiel erhält man durch Zerlegung (decomposition) der Relation Lieferant in zwei Relationen mit folgenden Schemata:

LA (LName, LAdresse)

LWP (LName, Ware, Preis)

Beispiel:

(1) Lagerhaltungsrelation

vorrat	(Teil	Lager	Menge	Lageradresse)
	101	1	25	Waagg. 10
	102	3	410	Krugerstr. 42
	102	1	300	Waag. 10
	112	4	10	Brunnerstr. 105

(2) Lager 1 wird verlegt

vorrat	(Teil	Lager	Menge	Lageradresse)
	101	1	25	Kroißbergg. 12
	102	3	410	Krugerstr. 42
	102	1	300	Waag. 10
	112	4	10	Brunnerstr. 105

Redundanz bewirkt Anomalie nach Adressänderung

(3) Teil 102 wird aus Lager 3 gelöscht

vorrat	(Teil	Lager	Menge	Lageradresse)
	101	1	25	Waagg. 10
	102	3	410	Krugerstr. 42
	102	1	300	Waag. 10
	112	4	10	Brunnerstr. 105

Adresse von Lager 3 wird unauffindbar.

F = { Teil Lager $\rightarrow$ Menge Lageradresse

Lager $\rightarrow$ Lageradresse }

Weitere Anomalie:

Einfügen eines neuen Lagers, anfangs ohne Teile.

(4) Zerlegung (Decomposition) in zwei Relationen

vorrat (Teil	Lager	Menge)
101	1	25
102	3	410
102	1	300
112	4	10

lager (Lager	Lageradresse)
1	Waagg. 10
3	Krugerstr. 42
4	Brunnerstr. 105

(5) Verlegung von Lager 1

vorrat (Teil	Lager	Menge)
101	1	25
102	3	410
102	1	300
112	4	10

lager (Lager	Lageradresse)
1	Kroißberg. 12
3	Krugerstr. 42
4	Brunnerstr. 105

Lageradresse bleibt konsistent

(6) Löschen von Teil 112

vorrat (Teil	Lager	Menge)
101	1	25
102	3	410
102	1	300
~~112~~	~~4~~	~~10~~

lager (Lager	Lageradresse)
1	Kroißberg. 12
3	Krugerstr. 42
4	Brunnerstr. 105

Lageradresse bleibt erhalten

(7) Einfügen eines neuen Lagers (ohne Teile) geht problemlos

vorrat (Teil	Lager	Menge)
101	1	25
102	3	410
102	1	300

lager (Lager	Lageradresse)
1	Kroißberg. 12
3	Krugerstr. 42
4	Brunnerstr. 105
5	Kettenhofweg 13

Integritätsbedingungen

Integritätsbedingungen (Integrity Constraints) oder Einschränkungen beschränken die möglichen Inhalte einer Datenbank auf solche, die Abbild eines realen Objektes sein können.

Einschränkungen im ER-Modell:

- Schlüssel
- Art der Beziehung (one-to-one, one-to-many, ...)
- Weitere Einschränkungen mittels Business Rules

Einschränkungen in Datenbanken werden wie folgt unterschieden:

- **Statische Einschränkungen**:
  betreffen die Korrektheit vorkommender Datenbankinhalte, z.B. „das Gehalt eines Angestellten darf nicht negativ sein“.
- **Dynamische Einschränkungen**:
  Regeln, die Übergänge zwischen Datenbankzuständen beschränken, z.B. „das Gehalt eines Angestellten darf nur zunehmen“.

Statische Einschränkungen werden über sogenannte **Assertions** realisiert. Spezielle Formen von Assertions sind:

- **Wertebereich Bedingungen** (Domain Constraints):
  Wie im Abschnitt SQL gesehen, können den einzelnen Attributen einer Relation bei der Erstellung Wertebereiche (integer, varchar(x), real, not null, ...) zugewiesen werden.

- **Reverenzierende Bedingungen** (Referential Constraints):
  Oft wird gewünscht, dass Werte, die in einer Relation für eine gegebene Attributmenge auftreten ebenso in einer anderen Relation für gewisse Attribute existieren, z.B. bei der Überführung einer Generalisierung aus dem ER-Modell in das Relationen-Modell.
  Diese Bedingung wird i. allg. mittels sogenannter **Foreign Keys** realisiert.

- **Funkionale Abhängigkeiten** (Functional Dependencies)
  – im Folgenden beschrieben –

Dynamische Einschränkungen werden über sogenannte **Trigger** realisiert. Dies sind Ausdrücke, die automatisch vom System als Seiteneffekt einer DB-Modifikation ausgeführt werden.

Für einen Trigger muss definiert werden

- Welche Bedingungen führen zur Ausführung.
- Was soll getan werden.

Beispiel: (Bank)
Anstatt ein negatives Giro-Konto zuzulassen wird, falls der Saldo eines Giro-Kontos negativ wird der entsprechende Betrag einem Kredit-Konto zugewiesen und das Saldo des Giro-Kontos auf 0 gesetzt.

Funktionale Abhängigkeiten

Funktionale Abhängigkeiten sind Aussagen über **alle möglichen** Relationen, die Instanzen (Werte) eines Realtionenschemas r(R) sein können.

- Aus r(R) kann nicht abgeleitet werden welche funktionalen Abhängigkeiten für R gelten.
- Von r(R) kann abgelesen werden, dass bestimmte funktionale Abhängigkeiten **nicht** gelten.

Definition von Funktionalen Abhängigkeiten

Im Folgenden bezeichnen:

V, W, X, Y, Z	Attributmengen,
A, B, C, … L	Attribute,
Q, R, S	Relationeneschemata
q(Q), r(R), s(S)	Relationen und
t, u, v, w	Tupel

Ist eine funktionale Abhängigkeit als statische Einschränkung auf einem Schema R definiert, so muss jede Relation r über R diese funktionale Abhängigkeit zu jedem Zeitpunkt erfüllen.

Definition Funktionale Abhängigkeit (FD):

Seien X und Y Teilmengen von R.

Eine Relation r(R) erfüllt (satisfies) die **funktionale Abhängigkeit** (functional dependency) FD $X \rightarrow Y$, wenn für je zwei (beliebige) Tupel $u, v \in r(R)$ gilt:

$$u(X) = v(X) \Rightarrow u(Y) = v(Y).$$

X heißt die linke Seite (left hand side) LHS, Y heißt die rechte Seite (right hand side) RHS von FD.

$$X \rightarrow Y \Leftrightarrow u(X) = v(X)$$
$$\Rightarrow u(Y) = v(Y)$$

Beispiel :

R = (PERSONALNR, ABTEILUNG, ADRESSE)

F = {PERSONALNR → ABTEILUNG,

ABTEILUNG → ADRESSE}

1)

PERSONALNR	ABTEILUNG	ADRESSE
12	Buchhaltung	1010 Wien
13	Verkauf	1030 Wien
11	Filiale	8010 Graz
14	Buchhaltung	1010 Wien

Abhängigkeiten erfüllt

2)

PERSONALNR	ABTEILUNG	ADRESSE
12	Buchhaltung	1010 Wien
13	Verkauf	1030 Wien
11	Filiale	8010 Graz
14	Buchhaltung	5010 Salzburg

Abhängigkeit ABTEILUNG → ADRESSE verletzt

Beispiel :

Gegeben ist die Relation r(R):

r	(A	B	C	D	E)
	a1	b1	c1	d1	e1
	a1	b2	c2	d2	e1
	a2	b1	c3	d3	e1
	a2	b1	c4	d3	e1
	a3	b2	c5	d1	e1

Geben Sie an, welche der folgenden Abhängigkeiten r erfüllt.

(a) $A \rightarrow D$

(b) $AB \rightarrow D$

(c) $C \rightarrow BDE$

(d) $E \rightarrow A$

(e) $A \rightarrow E$

(f) $A \rightarrow BC$

Definition Schlüssel:

Gegeben seien ein Relationenschema R und eine Menge F von FDs.

$X \subseteq R$ ist ein **Oberschlüssel** für R

$\Leftrightarrow$

$X \rightarrow R$

X ist ein **Schlüssel** für R

$\Leftrightarrow$

$X \rightarrow R$ und X minimal ($\neg(\forall A \in R: X \backslash A \rightarrow R)$)

- Die meisten Datenbanksysteme ermöglichen funktionale Abhängigkeiten durch die Definition eines Schlüssels (Key).

- Dies erlaubt eine effiziente Implementierung.

Beispiel:

Personal	Schlüssel **PersNr.**	Tel	Abteilung	Name
	2	55	Verkauf	Otto
	3	56	Buchhaltung	Müller

Triviale FD: PersNr $\rightarrow$ Tel., Abteilung, Name

Normalformen

Was sind Kriterien eines guten Entwurfs?

- So wenig Redundanz wie möglich
- Keine Einfüge-, Lösch-, Änderungsanomalien

Erste und Zweite Normalform

Beispiel: (nicht 1. Normalform)

vorrat (Teil Lager)

Teil	Lager
101	{1, 3}
102	{1, 2, 4}
112	{4}

Definition:

Ein Relationenschema R ist in 1. Normalform (1NF), wenn die Wertebereiche aller Attribute von R atomar sind.

1.NF

$\Leftrightarrow$

Wertebereiche atomar

Definition:

Eine funktionale Abhängigkeit $X \rightarrow Y$ heißt **volle funktionale Abhängigkeit**, wenn für keine Teilmenge $X' \subset X$ $X' \rightarrow Y$ gilt. Y heißt dann **voll funktional** abhängig von X.

$$X \rightarrow Y \text{ voll funktional}$$
$$\Leftrightarrow$$
$$\nexists\, X' \subset X : X' \rightarrow Y$$

$$X \text{ Schlüssel von } R$$
$$\Leftrightarrow$$
$$X \rightarrow R \text{ voll funktional}$$

$$\begin{cases} X \rightarrow Y \\ X \text{ Oberschlüssel} \end{cases}$$

Definition:

Ein Attribut A heißt **prim** in R, wenn es in einem Schlüssel von R enthalten ist, sonst heiße es nicht **prim**.

A ist prim

$\Leftrightarrow$

X: X ist Schlüssel, $A \in X$

Definition:

Ein Relationenschema R ist in **2. Normalform (2NF)**, wenn es in 1NF ist und jedes nicht prime Attribut voll funktional von jedem Schlüssel von R abhängig ist.

2. NF

$\Leftrightarrow$

A nicht prim $\Rightarrow$ Schlüssel $\rightarrow$ A voll funktional

Die 2. NF ist verletzt, wenn ein Teil eines Schlüssels ein Nicht-Schlüsselattribut funktional bestimmt.

Beispiel:

R = ABCD F = {AB→CD, B→D}

B→D verletzt 2. NF

Zerlegung in:

R1 = (ABC) mit F1 = {AB→C}

R2 = (BD) mit F2 = {B→D}

Dritte Normalform

Beispiel:

(1) Personalrelation

Annahme: Jede Abteilung befindet sich in nur einem Gebäude.

personal (Personal# Abteilung Gebäude)

Personal#	Abteilung	Gebäude
1215	Buchh.	A12
2410	Vertrieb	A14
2412	PR	B8
809	Buchh.	A12

F = {Personal# $\rightarrow$ Abteilung Gebäude,

Abteilung $\rightarrow$ Gebäude}

(in 2NF)

(2) Verlegung der Buchhaltungsabteilung:

personal (Personal# Abteilung Gebäude)

1215	*Buchh.*	*A7*
2410	Vertrieb	A14
2412	PR	B8
809	*Buchh.*	*A12*

Redundanz bewirkt Anomalie nach Adressänderung.

(3) Entfernen des Angestellten 2412:

personal (Personal# Abteilung Gebäude)

1215	Buchh.	A12
2410	Vertrieb	A14
~~*2412*~~	~~*PR*~~	~~*B8*~~
809	Buchh.	A12

Ist der Abteilung PR vorübergehend kein Personal zugeteilt, dann geht die Gebäudeinformation verloren.

(4) Zerlegung in zwei Relationen

personal		abteilungen	
(Personal#	Abt)	(Abt	Gebäude)
1215	Buchh.	Buchh.	A12
2410	Vertrieb	Vertrieb	A14
2412	PR	PR	B8
809	Buchh.		

(5) Verlegung der Buchhaltungsabteilung

personal		abteilungen	
(Personal#	Abt)	(Abt	Gebäude)
1215	Buchh.	*Buchh.*	*A7*
2410	Vertrieb	Vertrieb	A14
2412	PR	PR	B8
809	Buchh.		

Gebäudeinformation bleibt konsistent.

(6) Löschen des Angestellten 2412

personal		abteilungen	
(Personal#	Abt)	(Abt	Gebäude)
1215	Buchh.	Buchh.	A12
2410	Vertrieb	Vertrieb	A14
~~*2412*~~	~~*PR*~~	PR	B8
809	Buchh.		

Gebäudeinformation bleibt erhalten

Definition:

Gegeben sei eine Menge funktionaler Abhängigkeiten F über einem Relationenschema R, $X \subseteq R$, $A \in R$. A ist **transitiv abhängig von X**, wenn es eine Attributmenge $Y \subset R$ gibt so, dass $X \rightarrow Y \in F^+$ und $Y \rightarrow A \in F^+$ sind und $Y \rightarrow X \notin F^+$, $A \notin XY$ gilt.

$$X \rightarrow A \text{ transitiv}$$

$$\Leftrightarrow$$

$$\exists Y,\ Y \neq X:\ X \rightarrow Y \rightarrow A,\ A \text{ nicht prim}$$

Definition:

Ein Relationenschema R in 1. NF ist in 3. Normalform (3NF), wenn kein nichtprimes Attribut von einem Schlüssel in R transitiv abhängig ist.

3.NF

$\Leftrightarrow$

A nicht prim $\Rightarrow$ Schlüssel $\rightarrow$ A nicht transitiv

3.NF $\Rightarrow$ 2.NF $\Rightarrow$ 1.NF

Die 3. NF ist **verletzt**, wenn ein Nicht-Schlüsselattribut **transitiv** von einem Schlüssel abhängt.

Beispiel:

R = $\underline{A}BC = \{A \rightarrow B, B \rightarrow C\}$

$A \rightarrow B \rightarrow C$ verletzt 3. NF

Zerlegung in:

R1 = ($\underline{A}B$) mit F1 = $\{A \rightarrow B\}$ und

R2 = ($\underline{B}C$) mit F2 = $\{B \rightarrow C\}$

Beispiel:

Personen			
Name	PLZ	Ort	Straße
Hans Muster	60308	Frankfurt	Kettenhofweg 23
Klaus Fischer	63069	Offenbach	Bahnhofstr. 12
Julia Knopf	63069	Offenbach	Waldweg 34

Diese Relation ist in 2NF, da sie nur ein primes Attribut besitzt. Das Problem bei dieser Relation ist, dass der Ort von dem nicht-primen Attribut PLZ abhängt:

{ PLZ } → { Ort }

Damit ist diese Relation nicht in dritter Normalform. Hierdurch können ebenfalls wieder Anomalien auftreten. Betrachten wir das reale Problem, dass bei der Eingabe von Daten Fehler gemacht werden können und ein Ortsname daher in verschiedenen Schreibweisen in der Relation vorkommen kann → Inkonsistenz.

Die Lösung wird auch in diesem Fall durch die Zerlegung der Relation in mehrere, der 3NF genügenden Relationen erreicht.

In diesem Fall wird die Relation Personen durch die zwei folgenden Relationen ersetzt:

- Orte { PLZ, Ort }
- Personen { Name, PLZ, Straße }

Orte	
PLZ	Ort
60308	Frankfurt
63069	Offenbach

Personen		
Name	PLZ	Straße
Hans Muster	60308	Kettenhofweg 23
Klaus Fischer	63069	Bahnhofstr. 12
Julia Knopf	63069	Waldweg 34

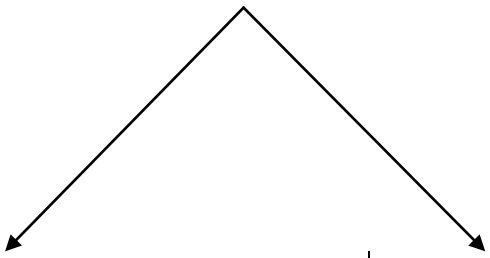
LIEFERANT(<u>LName</u>, LAdr, <u>Ware</u>, Preis)

FD = { LName $\rightarrow$ LAdr,

LName, Ware $\rightarrow$ Preis }

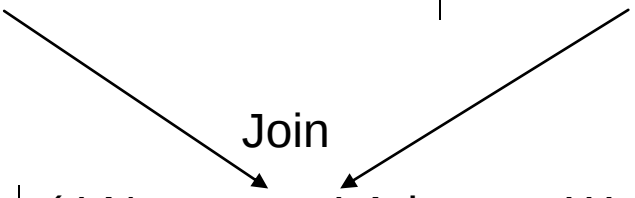
LIEF1(<u>LAdr, Ware</u>) LIEF2(<u>LName, Ware</u>, Preis)

LIEFERANT	(<u>LName</u>,	LAdr,	<u>Ware</u>,	Preis)
	l_1	a_1	w	p_1
	l_2	a_2	w	p_2



LIEF1	(<u>LAdr,</u>	<u>Ware</u>)
	a_1	w
	a_2	w

LIEF2	(<u>LName,</u>	<u>Ware</u>,	Preis)
	l_1	w	p_1
	l_2	w	p_2



Join

LIEFERANT	(<u>LName</u>,	LAdr,	<u>Ware</u>,	Preis)
	l_1	a_1	w	p_1
	l_2	a_1	w	p_2
	l_1	a_2	w	p_1
	l_2	a_2	w	p_2

Beispiel:

PLA (PLZ	Stadt	StraßeNr)
30419	Hannover	Schaumburgstr. 2
30419	Hannover	Quedlinburger Weg 17
30477	Hannover	Auf dem Brinke 7
60325	Frankfurt	Robert-Mayer-Straße 11

Es liegen folgende FDs vor:

PLZ $\rightarrow$ Stadt

Stadt, StraßeNr $\rightarrow$ PLZ

Als Schlüssel würde sich hier daher folgendes anbieten:

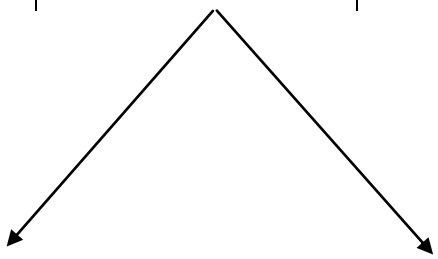
(PLZ, StraßeNr) oder (Stadt, StraßeNr)

Es gibt also keine nicht-prim Attribute, die Relation ist also in 3. NF.

Dennoch liegt hier eine Redundanz vor, da zu einer Postleitzahl die Stadt mehrmals gespeichert wird (z.B. PLZ = 30419). Die 3. NF reicht also nicht immer aus.

PLA

(PLZ	Stadt	StraßeNr)
30419	Hannover	Schaumburgstr. 2
30419	Hannover	Quedlinburger Weg 17
30477	Hannover	Auf dem Brinke 7
60325	Frankfurt	Robert-Mayer-Straße 11

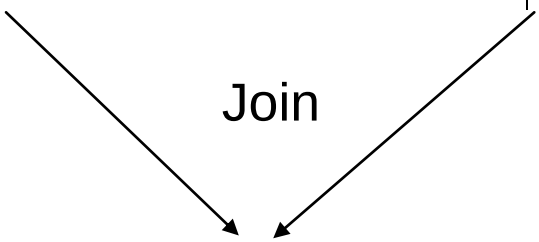


PLA1

(PLZ	Stadt)
30419	Hannover
30477	Hannover
60325	Frankfurt

PLZ2

(PLZ	StraßeNr)
30419	Schaumburgstr. 2
30419	Quedlinburger Weg 17
30477	Auf dem Brinke 7
60325	Robert-Mayer-Straße 11



Join

PLA

(PLZ	Stadt	StraßeNr)
30419	Hannover	Schaumburgstr. 2
30419	Hannover	Quedlinburger Weg 17
30477	Hannover	Auf dem Brinke 7
60325	Frankfurt	Robert-Mayer-Straße 11

Zerlegung ist verbundstreu!

... aber nicht abhängigkeitstreu!

Stadt, StraßeNr $\rightarrow$ PLZ geht verloren