Normalformen

Übersicht

Kriterien eines guten DB-Entwurfs:

- so wenig Redundanzen wie möglich
- keine Einfüge-, Lösch-, Änderungsanomalien

1. Erste Normalform

Beispiel: (nicht 1. Normalform)

vorrat

(TEIL	LAGER)
101	{1,3}
102	{1,2,4}
112	{4}

Def.: Ein Relationsschema R ist in *1. Normalform (1.NF)*, wenn die Wertebereiche aller Attribute von R atomar sind.

1.NF <---------------> Wertebereiche atomar

vorrat (TEIL	LAGER)
101	1
101	3
102	1
102	2
102	4
112	4

F = { TEIL LAGER $\rightarrow$ MENGE LAGERADRESSE

LAGER $\rightarrow$ LAGERADRESSE }

Beispiel:

1. Lagerhaltungsrelation:

vorrat	(<u>TEIL</u>	<u>LAGER</u>	MENGE	LAGERADRESSE)
	101	1	25	Waaggasse 10
	102	3	410	Krugerstr. 42
	102	1	300	Waaggasse 10
	112	4	10	Brunnerstr. 105

2. Lager 1 wird verlegt:

vorrat	(<u>TEIL</u>	<u>LAGER</u>	MENGE	LAGERADRESSE)
	101	1	25	*Kroißberggasse 12*
	102	3	410	Krugerstr. 42
	102	1	300	*Waaggasse 10*
	112	4	10	Brunnerstr. 105

Redundanz bewirkt Anomalie nach Adreßänderung

Beispiel (Fortsetzung):

3. Teil 102 wird aus Lager 3 gelöscht

vorrat	(<u>TEIL</u>	<u>LAGER</u>	MENGE	LAGERADRESSE)
	101	1	25	Waaggasse 10
	~~102~~	~~3~~	~~410~~	~~Krugerstr. 42~~
	102	1	300	Waaggasse 10
	112	4	10	Brunnerstr. 105

Adresse von Lager 3 wird unauffindbar.

Beispiel(Fortsetzung)

4. Zerlegung in 2 Relationen

vorrat

(TEIL	LAGER	MENGE)
101	1	25
102	3	410
102	1	300
112	4	10

lager

(LAGER	LAGERADRESSE)
1	Waaggasse 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	*Kroißberggasse 12*
3	Krugerstr. 42
4	Brunnerstr. 105

Lageradresse bleibt konsistent

5 Wienerbergerstr. 10

Beispiel (Fortsetzung)

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	Kroißberggasse 12
3	Krugerstr. 42
4	Brunnerstr. 105
5	Wienerbergerstr. 10

Lageradresse bleibt erhalten.

2. Zweite Normalform

Eine funktionale Abhängigkeit $X \to Y$ heißt *volle funktionale Abhängigkeit*, wenn für keine Teilmenge $X' \subseteq X$ $X' \to Y$ gilt.

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

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

3. Dritte Normalform (Beispiel)

1. Personalrelation

Annahme: Jede Abteilung befindet sich in nur einem Gebäude.

personal	(<u>PERSONAL#</u>	ABTEILUNG	GEBÄUDE)
	1215	Buchhaltung	A12
	2410	Vertrieb	A14
	2412	PR	B8
	809	Buchhaltung	A12

F = {PERSONAL# $\rightarrow$ ABTEILUNG GEBÄUDE,

ABTEILUNG $\rightarrow$ GEBÄUDE}

Beispiel (Fortsetzung)

2. Verlegung der Buchhaltungs-Abteilung

personal	(<u>PERSONAL#</u>	ABTEILUNG	GEBÄUDE)
	1215	*Buchhaltung*	*A7*
	2410	Vertrieb	A14
	2412	PR	B8
	809	*Buchhaltung*	*A12*

Redundanz bewirkt Anomalie nach Adreßänderung.

Beispiel (Fortsetzung)

3. Entfernen des Angestellten 2412:

personal	(<u>PERSONAL#</u>	ABTEILUNG	GEBÄUDE)
	1215	Buchhaltung	A12
	2410	Vertrieb	A14
	~~2412~~	~~PR~~	~~B8~~
	809	Buchhaltung	A12

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

Beispiel (Fortsetzung)

4. Zerlegung in 2 Relationen

personal

(<u>PERSONAL#</u>	ABTEILUNG)
1215	Buchhaltung
2410	Vertrieb
2412	PR
809	Buchhaltung

abteilungen

(<u>ABT</u>	GEBÄUDE)
Buchhtg.	A12
PR	B8
Vertrieb	A14

Beispiel (Fortsetzung)

5. Verlegung der Buchhaltungs-Abteilung

personal		abteilungen	
(<u>PERSONAL#</u>	ABTEILUNG)	(<u>ABT</u>	GEBÄUDE)
1215	Buchhaltung	*Buchhtg.*	*A7*
2410	Vertrieb	PR	B8
2412	PR	Vertrieb	A14
809	Buchhaltung		

Gebäudeinformation bleibt konsistent.

Beispiel (Fortsetzung)

6. Löschen des Angestellten 2412

personal		abteilungen	
(<u>PERSONAL#</u>	ABTEILUNG)	(<u>ABT</u>	GEBÄUDE)
1215	Buchhaltung	Buchhtg.	A7
2410	Vertrieb	PR	B8
~~2412~~	~~PR~~	Vertrieb	A14

Gebäudeinformation bleibt erhalten.

Transitive Abhängigkeit

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, sodaß $A \notin Y$, $Y \not\equiv X$, $X \rightarrow Y$ und $Y \rightarrow A$ gilt.

Beispiel:

P# -> ABT, ABT -> ABTADR

ABTADR ist transitiv abhängig von P#

Dritte Normalform

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

Beispiel:

R= TEIL LAGER MENGE LAGERADR

F={TEIL LAGER -> MENGE LAGERADR,
LAGER -> LAGERADR}

nicht 3NF!!

4. Zerlegung von Relationenschemata

Im folgenden wollen wir annehmen, daß ein Relationenschema *R* aus 2 Teilen besteht:

S ... einer Menge von Attributen

F ... einer Menge funktionaler Abhängigkeiten über S

Ein *Relationales Datenbankschema* ist eine Menge von Relationenschemata: $R = \{R_1, R_2, ..., R_n\}$.

Ein Datenbankschema *R* ist in 1.NF, 2.NF, 3.NF, wenn jedes Relationenschema in *R* in 1.NF, respektive 2.NF, 3.NF ist.

Zerlegung von Relationenschemata

Eine Zerlegung eines Relationenschemas $R(S,F)$ ist eine Menge von Relationenschemata $\{R_1(S_1,F_1), R_2(S_2,F_2), ..., R_n(S_n,F_n)\}$ wobei

1. $S = S_1 \cup S_2 \cup ... \cup S_n$
2. $F_i \equiv (\{X \rightarrow Y \mid X \rightarrow Y \in F^+, XY \subseteq S_i\})$
3. $(F_1 \cup F_2 \cup ... \cup F_n)^+ \subseteq F^+$

Abhängigkeitstreue und verbundtreue Zerlegung

Die Zerlegung $\{R_1(S_1,F_1), ..., R_n(S_n,F_n)\}$ eines Relationenschemas $R(S,F)$ ist *abhängigkeitstreu (dependency preserving)*, wenn: $\{(F_1 \cup F_2 \cup ... \cup F_n)^+ = F^+$ (keine Abhängigkeit geht verloren).

Die Zerlegung ist *verbundtreu (verlustfrei, lossless)*, wenn für jede Relation $r(S)$ gilt:

$$r(S_1) \bowtie r(S_2) \bowtie ... \bowtie r(S_n) = r(S).$$

Beispiel: nichtabhängigkeitstreue Zerlegung

Sei $R(A\ B\ C\ D\ E)$ ein Relationsschema mit einer Menge von funktionalen Abhängigkeiten
$F = \{A \rightarrow BCD, CD \rightarrow E, EC \rightarrow B\}$.

Eine mögliche Zerlegung von *R*:

$\{R_1,R_2\}$ mit $R_1(A\ B\ C\ D)$ und $R_2(C\ D\ E)$.

Die Abhängigkeit $EC \rightarrow B$ wird weder durch die auf R_1 noch durch die auf R_2 anwendbaren Abhängigkeiten aus F^+ impliziert.

==> nicht abhängigkeitstreue Zerlegung!

Beispiel: verbundtreue Zerlegung

R = ({ABC}, {A → B, B → C}), ABC wird zerlegt in *r und s*

q

A	B	C
a1	b1	c1
a2	b2	c1

r

A	B
a1	b1
a2	b2

$\bowtie$ s

B	C
b1	c1
b2	c1

q’

A	B	C
a1	b1	c1
a2	b2	c1

Nicht verbundtreue Zerlegung

R = ({ABC}, {A → B, C → B}), ABC wird zerlegt in *r und s*

q

A	B	C
a1	b1	c1
a2	b1	c2

r

A	B
a1	b1
a2	b1

$\bowtie$ s

B	C
b1	c1
b1	c2

q’

A	B	C
a1	b1	c1
a1	b1	c2
a2	b1	c1
a2	b1	c2

Theorem

Eine Zerlegung von $R(S,F)$ in $\{R_1(S_1,F_1), R_2(S_2,F_2), ..., Rn(S_n,F_n)\}$ ist verbundtreu, wenn die Hülle eines der generierten Relationenschemata gleich S ist.

$\exists\, S_i : S_i^+ = S$ (S_i^+ bezüglich $F_1 \cup F2 ... \cup F_n$)

===> Zerlegung von S in $S_1, S_2, ..., S_n$ ist verbundtreu

5. Boyce-Codd Normalform

Eine Relation *R* in 1.NF ist in *Boyce-Codd Normalform (BC-NF)* bezüglich ihrer Menge funktionaler Abhängigkeiten *F*,

wenn kein Attribut transitiv von einem Schlüssel abhängt
oder (äquivalent)
falls für jede nicht-triviale Abhängigkeit $X \rightarrow Y$ gilt:
X ist ein Schlüssel von *R*.

BCNF ==> 3.NF ==> 2.NF ==> 1.NF

Beispiel: BCNF

R = {ÜBUNG, STUDENT, TUTOR}

F = {STUDENT ÜBUNG $\rightarrow$ TUTOR, f1
TUTOR $\rightarrow$ ÜBUNG} f2

Es ist nicht möglich, eine verbundtreue und abhängigkeitstreue Zerlegung von *R* zu finden.

Beispiel 1

R=GHIJKL, F= $\{GH \rightarrow I, GH \rightarrow J, I \rightarrow K, K \rightarrow L, I \rightarrow J\}$
Minimale Überdeckung F'=$\{GH \rightarrow I, I \rightarrow K, K \rightarrow L, I \rightarrow J\}$
Schlüssel: GH
2NF erfüllt
3NF nicht erfüllt wegen $I \rightarrow K, K \rightarrow L, I \rightarrow J$
daher auch BCNF nicht erfüllt

Beispiel 2

R=ABCDE, F=$\{AB \rightarrow CDE, CD \rightarrow AB, E \rightarrow C\}$
Schlüsselkandidaten: AB, CD, ED
2NF erfüllt
3NF erfüllt (C hängt von AB transitiv ab, aber keine Verletzung der 3NF weil C prim)
BCNF nicht erfüllt, weil $E \rightarrow C$ gilt und E nicht Oberschlüssel ist.

Beispiel 3

R=ABCDE, F=$AB \rightarrow CD, CD \rightarrow ABE$
Schlüssel: AB und CD
2NF, 3NF erfüllt
Auch BCNF erfüllt, alle linken Seiten sind Oberschlüssel

Beispiel 4

R=ABCDE, F=AB $\rightarrow$ C, C $\rightarrow$ D

Schlüssel: ABE (E nicht vergessen!!)

2 NF nicht erfüllt, weil C, D nur von AB, nicht aber von E abhängen
daher auch 3NF und BCNF nicht erfüllt

6. Verbundtreue Zerlegung in BCNF/3.NF

- Verfahren
- Bemerkungen zum Verfahren
- Beispiel
- Problem: Eingebettete Abhängigkeiten

Verfahren (1)

Gegeben sei ein Relationenschema *R* und eine Menge *F* von funktionalen Abhängigkeiten.

Mache aus *F* eine minimale Überdeckung ohne überflüssige Attribute. (Diese heißt weiterhin *F*).

Bestimme **alle** Schlüssel von *R* unter *F*.

Verfahren (2)

Wir prüfen zunächst, ob *F* eine “bösartige” Abhängigkeit enthält, d.h., eine Abhängigkeit der Form $X \rightarrow Y$, wobei

bei 3.NF-Zerlegung: *Y* nicht prim und transitiv abhängig vom Schlüssel *Z* über *X* ist.

bei BCNF-Zerlegung: *X* kein Schlüssel von *R* ist.

Eine solche Abhängigkeit würde die BCNF-Bedingungen für *(R,F)* verletzen.

Enthält F keine bösartige Abhängigkeit, dann ist *(R,F)* bereits in 3.NF/BCNF. Fertig!

Verfahren (3)

Gibt es eine bösartige Abhängigkeit $X \rightarrow Y$ in F, dann splitten wir R in zwei Teile R_1 und R_2 mit:

$$R_1 = R - Y \text{ und } R_2 = XY$$

Definition von F[X] (Projektion einer Abhängigkeitsmenge F auf eine Attributmenge X):

$F[X] = \{ f \in F \mid f = V \rightarrow W$ und V, W beide Teilmengen von $X\}$

Verfahren (4)

Wir berechnen nun Überdeckungen F_1 und F_2 für R_1 und R_2, aus denen sich alle in R_1 bzw. R_2 gültigen funktionalen Abhängigkeiten ableiten lassen. D.h. wir berechnen Abhängigkeitsmengen F_1 und F_2 über R_1 bzw. R_2, für die gilt:

$$F_1^+ = F^+ [R_1] \text{ und } F_2^+ = F^+ [R_2]$$

Das Verfahren wird nun rekursiv auf *(R1,F1)* und *(R2,F2)* angewendet.

Die Menge aller Subschemata in 3.NF/BCNF, die im Laufe des Verfahrens erzeugt werden, ist eine Zerlegung in 3.NF/BCNF des ursprünglichen Schemas R.

Bemerkungen (1)

1. Das Verfahren terminiert immer, da die Kardinalität der erzeugten Schemata in jedem Schritt abnimmt. Schlimmstenfalls erhält man Schemata der Kardinalität 2 (also mit 2 Attributen): diese sind immer in BCNF.

2. Eigentlich müßte man testen, ob F^+ (und nicht nur F) eine bösartige Abhängigkeit enthält. Es kann allerdings gezeigt werden, daß F^+ eine bösartige Abhängigkeit genau dann enthält, wenn F eine bösartige Abhängigkeit enthält.

Bemerkungen (2)

3. Es ist leicht zu sehen, daß die Zerlegung verbundtreu ist. Jede Einzelzerlegung ist verbundtreu: aus $R_1 = R - Y$ und $R_2 = \underline{X}Y$ kann durch den Verbund $R_1[X = X]R_2$ wieder R gewonnen werden, da X ein Schlüssel für R_2 ist.

4. Das schwierigste Teilproblem des Verfahrens besteht in der Berechnung von F_1 und F_2 aus F, R_1, R_2. Dieses Problem wird später gesondert behandelt (siehe Algorithmus RBR).

Bemerkungen (3)

5. Die während des Verfahrens entstehenden (bzw. verwendeten) Relationenschemata können als Baum mit Wurzel *(R,F)* dargestellt werden. Die Blätter des Baumes sind die Bestandteile der Zerlegung (siehe auch das folgende Beispiel).

6. Gibt es in einer Stufe verschiedene bösartige Abhängigkeiten, dann kann das Verfahren - je nach Auswahl der bösartigen Abhängigkeit - zu verschiedenen Zerlegungen führen.

Beispiel:

(Ullman, S. 238-240)

man betrachte ein Schema U = (C,T,H,R,S,G):

Die Attribute haben folgende Bedeutung:

- C: Course (Vorlesung)
- T: Teacher (Vortragender)
- H: Hour (Stunde)
- R: Room (Raum)
- S: Student
- G: Grade (Note)

Beispiel (Fortsetzung)

R = $CT\underline{H}R\underline{S}G$
F = {C → T, CS → G, HR → C, HS → R, TH → R}

$R1$ = $\underline{CS}G$
$F1$ = {CS → G}

$R2$ = $CT\underline{H}R\underline{S}$
$F2$ = { C → T, TH → R, HR → C, HS → R}

Beispiel (Fortsetzung)

$R2$ = $CT\underline{H}R\underline{S}$
$F2$ = { C → T, TH → R, HR → C, HS → R}

$R2.1$ = $\underline{C}T$
$F2.1$ = {C → T}

$R2.2$ = $C\underline{H}R\underline{S}$
$F2.2$ = {CH → R, HR → C, HS → R}

Beispiel (Fortsetzung)

$R2.2$ = $C\underline{H}R\underline{S}$
$F2.2$ = {CH → R, HR → C, HS → R}

$R2.2.1$ = $\underline{CH}R$
$F2.2.1$ = {CH → R, HR → C}

$R2.2.2$ = $C\underline{HS}$
$F2.2.2$ = {HS → C}

Beispiel (Fortsetzung)

Dazu geg.: Menge F von funktionalen Abhängigkeiten:

$C \rightarrow T$ jede Vorlesung hat nur einen Vortragenden

$HR \rightarrow C$ in einem bestimmten Hörsaal kann zu einem bestimmten Zeitpunkt nur höchstens eine Vorlesung stattfinden

$HT \rightarrow R$ ein Vortragender kann zu einem bestimmten Zeitpunkt nur in höchstens einem Hörsaal unterrichten

$CS \rightarrow G$ jeder Student hat zu jeder Vorlesung höchstens eine Note

$HS \rightarrow R$ jeder Student kann sich zu einem bestimmten Zeitpunkt in höchstens einem Hörsaal aufhalten.

Beispiel (Fortsetzung)

Leicht zu sehen: *HS* ist einziger (minimaler) Schlüssel dieses Schemas

Beispiel (Fortsetzung)

Diese Abhängigkeiten sind nicht ganz leicht zu finden:

z.B. muß beim Übergang von *CTHRS* auf *CHRS* eine neue Abhängigkeit $CH \rightarrow R$ angegeben werden.

Problem:

$F \models CH \rightarrow R$,

aber

$CH \rightarrow R$ ist nicht Element von F.

$CH \rightarrow R$ entsteht durch Verschmelzung der Abhängigkeiten $C \rightarrow T$ und $TH \rightarrow R$ durch Anwendung der Pseudotransivitätsregel.

Eingebettete Abhängigkeiten

- Problemstellung
- 1. Lösungsversuch
- A-Resolventen
- Umformung funktionaler Abhängigkeiten mit *fold, unfold*
- Algorithmus Reduction by Resolution

Problemstellung

Gegeben seien S und F und ein Subschema S_1 von S. (Wir nehmen an, daß F minimal ist und keine überflüssigen Attribute hat.)

Man finde eine Überdeckung G für S_1, die genau alle auf S_1 gültigen Abhängigkeiten repräsentiert.

Mathematisch ausgedrückt:

Finde G mit $G^+ = F^+[S_1]$.

(Die Menge $F^+[S_1]$ wird auch die Menge der in S_1 *eingebetteten* Abhängigkeiten genannt.)

1. Lösungsversuch

Eine ziemlich triviale Lösung dieses Problems besteht in $G = F^+[S_1]$. Diese Lösung ist in doppelter Hinsicht schlecht:

(a) es muß die gesamte Menge F^+ aller aus F folgenden Abhängigkeiten berechnet werden (z.B. mittels Armstrong Axiomen) und dann auf S_1 projiziert werden --> großer Aufwand (immer exponentiell)

(b) die Überdeckung $F^+[R]$ ist von maximaler Größe und braucht daher viel Speicherplatz (unhandlich)

Eine bessere Methode ist die im folgenden beschriebene.

A-Resolventen

Sei f eine Abhängigkeit der Form $XAY \rightarrow Z$, wobei A ein Einzelattribut ist, und X und Y möglicherweise leere Attributemengen darstellen und sei g eine Abhängigkeit der Form $V \rightarrow A$. Die aus f und g ableitbare Abhängigkeit $h: XVY \rightarrow Z$ nennen wir die A-*Resolvente* von f und g.

Beispiel:

Die beiden Abhängigkeiten $AC \rightarrow D$ und $DEF \rightarrow B$ haben als D-Resolvente $AEFC \rightarrow B$.

Bemerkung:

Jede A-Resolvente von f und g folgt aus $\{f,g\}$ durch Pseudotransitivität.

unfold - fold

Sei *G* eine Menge von funktionalen Abhängigkeiten. Wir bezeichnen mit *unfold(G)* die Menge jener Abhängigkeiten, die aus Elementen von *G* durch Aufsplitten der rechten Seite in Einzelattribute gewonnen wird.

Umgekehrt bezeichnen wir mit *fold(G)* jene Überdeckung von *G*, die durch Zusammenfassen aller Abhängigkeiten mit gleicher linker Seite gewonnen wird.

Beispiel:

G = { AB $\rightarrow$ CD, AB $\rightarrow$ E, AF $\rightarrow$ EGH }

unfold(G) = { AB $\rightarrow$ C, AB $\rightarrow$ D, AB $\rightarrow$ E, AF $\rightarrow$ E, AF $\rightarrow$ G, AF $\rightarrow$ H }

fold(G) = { AB $\rightarrow$ CDE, AF $\rightarrow$ EGH}

Bemerkung:

G, unfold(G) und fold(G) sind immer äquivalente Überdeckungen; im allgemeinen gilt:

fold (unfold(G)) <> G.

Die Menge aller *nichttrivialen* A-Resolventen, die unmittelbar (in einem Schritt) aus *F* erzeugt werden können, bezeichnen wir mit *RES(F,A)*.

Mit "nichttrivial" bezeichnen wir hier jene Abhängigkeiten der Form $V \rightarrow A$ für die A in V nicht vorkommt.

Algorithmus RBR (Reduction by Resolution)

Gottlob, 1987 (modifizierte Version)

```
INPUT: Schema S,
          Abhängigkeitsmenge F für S
          Subschema S1 von S
OUTPUT: Überdeckung G mit G+ = F+[S1]
BEGIN
  G: = unfold(F);
  X: = S - S1;
  WHILE X <> EmptySet DO
     BEGIN
     wähle ein Attribut A aus X
     X: = X - {A};
     G: = (G ∪ RES(G,A) );
        {alle Abh., die A enthalten, wegschmeissen}
     END
  G: = fold(G);
  RETURN G
END.
```

Beispiel: Prüfungsverwaltung

Bsp: Prüfungsverwaltung (Fortsetzung)

R = {MATRNR NAME GEBDAT ALTER DA_BETREUER
INSTITUT SEMESTER LVA NOTE}
F = {MATRNR SEMESTER LVA → NOTE
MATRNR → NAME GEBDAT ~~ALTER~~ DA_BETREUER ~~INSTITUT~~
DA_BETREUER → INSTITUT}
GEBDAT → ALTER

R1 = {MATRNR NAME GEBDAT ALTER
DA_BETREUER INSTITUT}
F1 = {MATRNR → GEBDAT ALTER
DA_BETREUER
DA_BETREUER → INSTITUT
GEBDAT → ALTER}

R2 = {MATRNR SEMESTER LVA NOTE}
F2 = { MATRNR SEMESTER LVA
→ NOTE}

Bsp.: Prüfungsverwaltung (Fortsetzung)

R1 = {MATRNR NAME GEBDAT
ALTER DA_BETREUER INSTITUT}
F1 = { MATRNR → NAME GEBDAT DA_BETREUER
DA_BETREUER → INSTITUT
GEBDAT → ALTER}

R1.1 = {DA_BETREUER INSTITUT}
F1.1 = {DA_BETREUER → INSTITUT}

R1.2 = {MATRNR NAME GEBDAT
ALTER DA_BETREUER}
F1.2 = {MATRNR →
NAME GEBDAT
DA_BETREUER,
GEBDAT → ALTER }

R1.2 = {MATRNR NAME GEBDAT
ALTER DA_BETREUER}
F1.2 = {MATRNR → NAME GEBDAT DA_BETREUER
GEBDAT → ALTER}

R1.2.1 = {MATRNR NAME GEBDAT
DA_BETREUER}
F1.2.1 = {MATRNR → NAME GEBDAT
DA_BETREUER}

R1.2.2 = {GEBDAT ALTER}
F1.2.2 = {GEBDAT → ALTER}

Synthesealgorithmus (Prinzip)

Geg.: R = (S,F)
Ges.: 3. NF, verbund- und abhängigkeitstreu

1. Bilde eine minimale Überdeckung von F.
2. Bilde Gruppen von Abhängigkeiten. $X \rightarrow A$ und $Y \rightarrow B$ sind in derselben Gruppe, falls $X^+ = Y^+$.
3. Bilde je Gruppe ein Relationenschema.
4. Falls kein Relationenschema einen Schlüssel K für R enthält, füge ein Relationenschema K hinzu.

Synthesealgorithmus (Beispiel)

R = A D Z S K B

F = { $A \rightarrow D$	$A^+ = AD$	R1
$AK \rightarrow B$	$AK^+ = AKB$	R2
$ZK \rightarrow S$	$ZK^+ = ZKSADB$	R3
$ZS \rightarrow A$	$ZS^+ = ZSAD$	R4
$DZ \rightarrow S$ }	$DZ^+ = DZSA$	

RBR-Algorithmus; Beispiele

R = A D Z S K B

A ... Aufführung, D ... Dirigent, Z ... Uhrzeit,
S ... Konzertsaal, K ... Kritiker, B ... Beurteilung

$F = \{A \rightarrow D, SZ \rightarrow A, DZ \rightarrow S, AK \rightarrow B, KZ \rightarrow S\}$

$F[ASKZ] = \{SZ \rightarrow A, AZ \rightarrow S, KZ \rightarrow S\}$