



13. Übungsblatt

Für die Übung am Donnerstag, 08. Februar 2007,
 von 15:30 bis 17:00 Uhr in 13/222.

Aufgabe 1: Aktive DBS / ECA-Regeln

Gegeben seien zwei Relationen „Teil“ und „Zusammensetzung“ zur Modellierung einer Stückliste:

Teil			Zusammensetzung		
Tnr	Name	Preis	OTnr	UTnr	Anzahl
T1	Auto	25000	T1	T2	1
T2	Motor	5500	T1	T3	1
T3	Getriebe	3000	T1	T7	100
T4	Zylinder	500	T2	T4	6
T5	Vergaser	90	T2	T5	1
T6	Kolben	300	T2	T7	30
T7	Schraube1	3	T3	T7	10
			T4	T6	1
			T5	T7	10

a) Formulieren Sie für folgende Integritätsbedingungen ECA-Regeln.

- Teile werden nur dann eingefügt, wenn sie in eine Zusammensetzung eingehen.
- Der Preis eines Teils muss höher oder gleich der Summe der Preise seiner Bestandteile sein.
- Ändert sich der Preis eines Teils, so werden die Preise der Oberteile entsprechend angepasst.
- Der Preis eines Teils darf sich nicht um mehr als 10% erhöhen.
- Wenn ein Teil gelöscht wird, werden auch die Teile gelöscht, die dieses Teil benutzen.
- Wenn sich die Zusammensetzung ändert, wird der Preis entsprechend angepasst.

b) Folgende Transaktionen sollen auf dieser DB mit den ECA-Regeln ausgeführt werden. Für jede Transaktion gelte die ursprüngliche Datenbank.

- Das Teil T7 werde durch das Teil T8 (Schraube1.1, 4) ersetzt.
- Die Preise für T4 und T5 werden um 10% erhöht.
- T7 gehe nicht 10, sondern 15 mal in T5 ein.
- Der Preis von T3 wird um 10% erhöht, anschließend wird der Preis von T1 ausgegeben.
- Der Preis von T7 werde auf 2 gesenkt, T7 gehe 14 mal in T5 ein.
- T7 gehe 14 mal in T5 ein, der Preis von T7 werde auf 2 gesenkt.

Lösung:

a) Formulierung von ECA-Regeln

- Teile werden nur dann eingefügt, wenn sie in eine Zusammensetzung eingehen.
 Auswertung und Ausführung müssen deferred geschehen, da zunächst das neue Teil und anschließend ein entsprechendes Tupel in Tabelle *Zusammensetzung* in der gleichen Transaktion eingefügt werden müssen.

```
CREATE ECA-RULE in_zusammensetzung AS
AFTER INSERT ON Teil REFERENCING NEW AS Nt
CHECK DEFERRED
Nt.Tnr NOT IN
(SELECT OTnr FROM Zusammensetzung)
UNION
(SELECT UTnr FROM Zusammensetzung)
DO ROLLBACK
END ECA-RULE in_zusammensetzung
```

- Der Preis eines Teils muss höher oder gleich der Summe der Preise seiner Bestandteile sein.
 Es wird davon ausgegangen, dass die Bedingung zunächst erfüllt ist. Diese Regel muss nur dann überprüft werden, wenn sich der Preis eines Teils verringert. Es ist zu überprüfen, ob der Preis noch höher ist als die Summe der Preise der eingehenden Teile. Wird der Preis erhöht, ist das kein Problem, für die Oberteile wird der Preis nach Regel 3 angepasst. Ändert sich die Zusammensetzung eines Teils, gilt Regel 6.

```
CREATE ECA-RULE preis_bed AS
AFTER UPDATE OF Preis ON Teil REFERENCING OLD AS Ot, NEW AS Nt
CHECK IMMEDIATE
(Ot.Preis > Nt.Preis) AND
(Nt.Preis < (SELECT SUM(T.Preis*Z.Menge)
FROM Teil T, Zusammensetzung Z
WHERE Nt.Tnr = Z.OTnr AND T.Tnr = Z.UTnr))
DO ROLLBACK
END ECA-RULE preis_bed
```

- Ändert sich der Preis eines Teils, so werden die Preise der Oberteile entsprechend angepasst.
 Änderung eines Preises bewirkt in jedem Fall Folgeänderungen in Oberteilen, die Änderungen müssen sofort geschehen, sonst können die neuen Preise nicht in derselben Transaktion ausgegeben werden.

```
CREATE ECA-RULE update_preis AS
AFTER UPDATE OF Preis ON Teil REFERENCING OLD AS Ot, NEW AS Nt
DO IMMEDIATE
-- (ändere Preis der Oberteile entsprechend der vorkommenden Anzahl ab)
UPDATE Teil SET Preis = Preis + (Nt.Preis - Ot.Preis) *
(
SELECT Anzahl from Zusammensetzung
WHERE UTnr = Nt.Tnr AND OTnr = Tnr
)
WHERE Tnr IN (SELECT OTnr FROM Zusammensetzung WHERE UTnr = Nt.Tnr)
END ECA-RULE update_preis
```

- (4) Der Preis eines Teils darf sich nicht um mehr als 10% erhöhen.
Diese Änderung soll unmittelbar nach der Änderung überprüft werden.

```
CREATE ECA-RULE max_preis_erhoehung AS
AFTER UPDATE OF Preis ON Teil REFERENCING OLD AS Ot, NEW AS Nt
CHECK IMMEDIATE
(Ot.Preis * 1.1 < Nt.Preis )
DO IMMEDIATE ROLLBACK
END ECA-RULE max_preis_erhoehung
```

- (5) Wenn ein Teil gelöscht wird, werden auch die Teile gelöscht, die dieses Teil benutzen.

```
CREATE ECA-Rule folge_loeschungen AS
AFTER DELETE ON Teil REFERENCING OLD AS Ot
DO IMMEDIATE
(lösche die Elemente aus Zusammensetzung, die angeben,
wie sich das gelöschte Teil zusammensetzt)
DELETE FROM Zusammensetzung
WHERE Ot.TNr = OTnr

(lösche die Teile, die das gelöschten Teil enthalten)
DELETE FROM Teil
WHERE TNr IN ( SELECT OTnr
              FROM Zusammensetzung
              WHERE UTnr = Ot.Tnr)

(lösche die Elemente aus Zusammensetzung, die angeben, welche
Teile das gelöschte Teil enthielten)
DELETE FROM Zusammensetzung
WHERE Ot.TNr = UTnr

END ECA-RULE folge_loeschungen
```

Frage: Ist die dritte Delete-Anweisung erforderlich? Wenn ja, warum nicht? ;-)

- (6) Wenn sich die Zusammensetzung ändert, wird der Preis entsprechend angepasst.
Die Zusammensetzung kann sich bei folgenden Ereignissen ändern:
Update(Z.OTnr), Update(Z.UTnr), Update(Z.Anzahl), Delete(Z), Insert(Z)
Der Bedingungsteil bleibt leer. Die Preisänderungen der Oberteile müssen immediate geschehen. (ähnlich wie bei Regel 3).

```
CREATE ECA-RULE update_zus_otnr AS
AFTER UPDATE OF OTnr ON Zusammensetzung REFERENCING OLD AS Ot, NEW AS Nt
DO IMMEDIATE
-- (ändere Preis des alten Oberteils)
UPDATE Teil SET Preis = Preis -
( SELECT Preis*Ot.Anzahl FROM Teil
  WHERE Tnr = Ot.UTnr)
WHERE TNr = Ot.OTnr;
-- (ändere Preis des neuen Oberteils)
UPDATE Teil SET Preis = Preis +
( SELECT Preis*Nt.Anzahl FROM Teil
  WHERE Tnr = Nt.UTnr)
WHERE TNr = Nt.OTnr;
END ECA-RULE update_zus_otnr
```

```
CREATE ECA-RULE update_zus_utnr AS
AFTER UPDATE OF UTnr ON Zusammensetzung REFERENCING OLD AS Ot, NEW AS Nt
DO IMMEDIATE
(ziehe den Preis für das alte Unterteil ab, addiere den Preis
für das neue Unterteil)
UPDATE Teil SET Preis = Preis -
( SELECT Preis*Ot.Anzahl FROM Teil
  WHERE Tnr = Ot.UTnr)
+
( SELECT Preis*Nt.Anzahl FROM Teil
  WHERE Tnr = Nt.UTnr)
WHERE TNr = Ot.OTnr;
END ECA-RULE update_zus_utnr
```

```
CREATE ECA-RULE update_zus_anzahl AS
AFTER UPDATE OF Anzahl ON Zusammensetzung REFERENCING OLD AS Ot, NEW AS Nt
DO IMMEDIATE
UPDATE Teil SET Preis = Preis + (Nt.Anzahl - Ot.Anzahl) *
( SELECT Preis FROM Teil
  WHERE Tnr = Ot.UTnr)
WHERE TNr = Ot.OTnr;
END ECA-RULE update_zus_anzahl
```

```
CREATE ECA-RULE delete_zus AS
AFTER DELETE ON Zusammensetzung REFERENCING OLD AS Oz
DO IMMEDIATE
UPDATE Teil SET Preis = Preis - Oz.Anzahl *
( SELECT Preis FROM Teil
  WHERE Tnr = Oz.UTnr)
WHERE TNr = Oz.OTnr;
END ECA-RULE delete_zus
```

```
CREATE ECA-RULE insert_zus AS
AFTER INSERT ON Zusammensetzung REFERENCING NEW AS Nz
DO IMMEDIATE
UPDATE Teil SET Preis = Preis + Nz.Anzahl *
( SELECT Preis FROM Teil
  WHERE Tnr = Nz.UTnr)
WHERE TNr = Nz.OTnr;
END ECA-RULE insert_zus
```

b) Folgende Transaktionen sollen auf dieser DB mit den ECA-Regeln ausgeführt werden.

Für jede Transaktion gelte die ursprüngliche Datenbank.

(1) Das Teil T7 werde durch das Teil T8 (Schraube 1.1, 4) ersetzt.

Diese TA kann in verschiedenen Schritten durchgeführt werden, z.B.:

- Einfügen von T8
- Ändern der Zusammensetzung
- Löschen von T7

Es ergibt sich folgender Ablauf:

INSERT(T8) -> R1d

UPDATE(Z) -> R6i

R6i: UPDATE(T5.Preis = 100) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE(T2.Preis=5510) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE(T1.Preis=25010) -> R2i, R3i, R4i

R2i: -

R3i: -

R4i: -

R4i: -

R4i: (90 * 1.1 < 100) => ROLLBACK

Es werden mehrere Regeln gleichzeitig aktiviert. In unserer Lösung wurden diese in FIFO-Ordnung bearbeitet.

(2) Die Preise für T4 und T5 werden um 10% erhöht.

Folgende Operationen/Regeln werden ausgeführt:

UPDATE(T4.Preis=550) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE(T2=5800) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE(T1=25300) -> R2i, R3i, R4i

R2i: -

R3i: -

R4i: -

R4i: -

R4i: -

UPDATE(T5.Preis=99) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE(T2=5809) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE(T1=25309) -> R2i, R3i, R4i

R2i: -

R3i: -

R4i: -

R4i: -

R4i: -

(3) T7 gehe nicht 10, sondern 15 mal in T5 ein.

Folgende Operationen/Regeln werden ausgeführt:

UPDATE(Z) -> R6i

R6i: UPDATE(T5=105) -> R2i, R3i, R4i

...

R4: ROLLBACK

(4) Der Preis von T3 wird um 10% erhöht, anschließend wird der Preis von T1 ausgegeben.

UPDATE(T3 =3300) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE (T1=25300) -> R2i, R3i, R4i

R2i: -

R3i: -

R4i: -

R4i: -

Ausgabe T1=25300

(5) Der Preis von T7 werde auf 2 gesenkt, T7 gehe 14 mal in T5 ein.

UPDATE(T7 =2) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE (T5=80) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE (T2=5490) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE (T1=24990) -> R2i, R3i, R4i

R2i: -

R3i: -

R4i: -

R4i: -

R4i: -

UPDATE (T3=2990) ->R2i, R3i, R4i

R2i: -

R3i: UPDATE (T1=24980) -> R2i, R3i, R4i

R2i: -

R3i: -

R4i: -

UPDATE (T2=5460) -> R2i, R3i, R4i

R2i: -

R3i: UPDATE T1(24950) ->R2i, R3i, R4i

R2: -

R3: -

R4: -

UPDATE (T1=24850) R2i, R3i, R4i

R2i: -

R3i: -

R4i: -

R4i: -

UPDATE (Z) -> R6i
 R6i: UPDATE (T5=88)
 R2i: -
 R3i: UPDATE (T2=5468)
 R2i: -
 R3i: UPDATE (T1=24858),
 R2: -
 R3: -
 R4: -
 R4i: -

(6) T7 gehe 14 mal in T5 ein, der Preis von T7 werde auf 2 gesenkt.

UPDATE Z -> R6i
 R6i: UPDATE (T5=102)
 R2i: -
 R3i: UPDATE (T2=5512)
 R2i: -
 R3i: UPDATE (T1=25012)
 R2i: -
 R3i: -
 R4i: -
 R4: -
 R4i: ROLLBACK

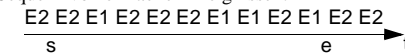
Aufgabe 2: Aktive DBS / Ereignisalgebra

Zur Definition von Ereignissen stellt ein aktives Datenbanksystem (aDBS) dem Benutzer eine Ereignissprache zur Verfügung, mit deren Hilfe er festlegen kann, welche Ereignisse das Eintreten einer bestimmten Situation im Datenbanksystem indizieren können. Die Ereignissprache stellt Konstrukte zur Formulierung der Ereignisse zur Verfügung. Die Formulierung von einfachen Ereignissen sind dabei absolute bzw. relative Zeitangaben oder der Beginn bzw. das Ende von Datenbankoperationen.

Darüber hinaus gibt es auch zusammengesetzte Ereignisse, die in einer einfachen Algebra beschrieben werden können. In dieser Algebra können Ereignisse mittels Ereigniskonstruktoren verknüpft werden. Das Eintreten eines zusammengesetzten Ereignisses entspricht dem Zeitpunkt, zu dem das letzte Ereignis der Kombination eingetreten ist. Ereignisse können auf bestimmte Überwachungsintervalle [Intervallanfang – Intervallende] begrenzt werden. Die folgenden Ereigniskonstruktoren sollen betrachtet werden:

- Disjunktion: $E=(E1 | E2)$
- Sequenz: $E=(E1; E2)$
- Konjunktion: $E=(E1, E2)$
- Negation: NOT E
- Stern-Operator: *E [s - e]
- Times-Operator: TIMES (n, E) [s - e]

Gegeben sei folgende Sequenz von einfachen Ereignissen:



Geben Sie jeweils an, wie oft folgende Ereignisse signalisiert werden:

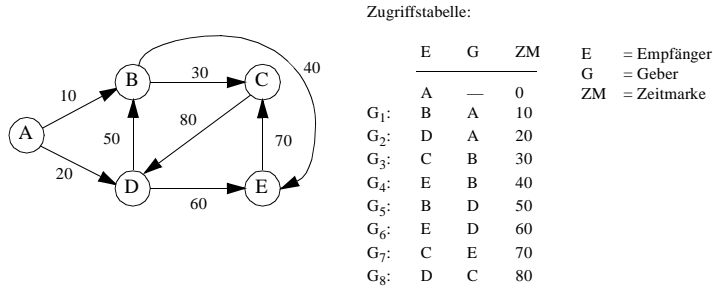
- | | |
|-----------------------------|--------------------------------|
| a) (E1; E2) | g) NOT (E1; E2) [s - e] |
| b) (E1 E2) | h) (E2; E1 [s - e]) |
| c) (E1, E2) | i) TIMES (4, E2) [s - e] |
| d) *(E1; E2) [s - e] | j) TIMES (2, E1) [s - e] |
| e) *(E1, E2) [s - e] | k) TIMES (2, (E1, E2)) [s - e] |
| f) (*E1 [s - e]; E2)[s - e] | l) TIMES (2, (E1; E2)) [s - e] |

Lösung:

- | | |
|-----------------------------------|-------------------------------------|
| a) (E1; E2) => 4 | k) TIMES (2, (E1, E2)) [s - e] => 2 |
| b) (E1 E2) => 12 | l) TIMES (2, (E1; E2)) [s - e] => 1 |
| c) (E1, E2) => 4 | |
| d) *(E1; E2) [s - e] => 1 | |
| e) *(E1, E2) [s - e] => 1 | |
| f) (*E1 [s - e]; E2) [s - e] => 1 | |
| g) NOT (E1; E2) [s - e] => 0 | |
| h) (E2; E1 [s - e]) => 4 | |
| i) TIMES (4, E2) [s - e] => 1 | |
| j) TIMES (2, E1) [s - e] => 2 | |

Aufgabe 3: Entzug von Zugriffsrechten

Gegeben sei folgender Autorisierungsgraph mit zeitabhängiger Interpretation und die zugehörige Zugriffstabelle, die Vergaben bzgl. eines Zugriffsrechtes P an einem Objekt O (G_i) jeweils mit der GRANT-Option enthält.



a) Wie verändern sich der Autorisierungsgraph und die Zugriffstabelle jeweils durch folgende Zugriffsrechtrücknahmen:

- (1) Entzug von G₁ zum Zeitpunkt 90 (REVOKE₁: B, A, 90)
- (2) Entzug von G₂ zum Zeitpunkt 90 (REVOKE₂: D, A, 90)

b) Zeichnen Sie nun den entsprechenden Autorisierungsgraphen mit zeitunabhängiger Interpretation und geben Sie die zugehörige Zugriffstabelle an.

Welche Zugriffsrechtvergaben hätten in der obigen Zugriffstabelle nicht gewährt werden dürfen? Welche Auswirkungen haben jeweils dieselben Zugriffsrechtrücknahmen aus der Teilaufgabe (a) auf den Autorisierungsgraphen und die Zugriffstabelle?

Lösung:

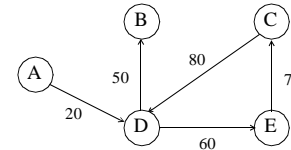
Siehe Härder: Implementierung von Datenbanksystemen, 1978, pp. 180

a) Autorisierungsgraph mit zeitabhängiger Interpretation

- (1) REVOKE₁: B, A, 90
 - (i) G₁ aus der Zugriffstabelle entfernen (A -> B)
 - (ii) Weiter mit Entzug P von B:
Finde die eingehende Kante in B mit minimaler ZM => t_{min}(B) = 50 (D -> B)
Lösche alle ausgehenden Kanten aus B mit ZM < t_{min}(B) = 50
=> B -> C und B -> E löschen
 - (iii) Weiter mit Entzug P von C:
t_{min}(C) = 70 => Es gibt keine ausgehenden Kanten aus C zu löschen.

- (iv) Weiter mit Entzug P von E:
t_{min}(E) = 60 => Es gibt keine ausgehenden Kanten aus E zu löschen.

Zugriffstabelle:



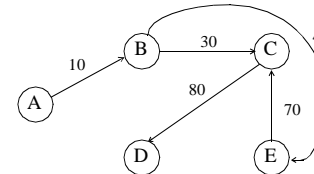
	E	G	ZM
	A	—	0
G ₂ :	D	A	20
G ₅ :	B	D	50
G ₆ :	E	D	60
G ₇ :	C	E	70
G ₈ :	D	C	80

E = Empfänger
G = Geber
ZM = Zeitmarke

(2) REVOKE₂: D, A, 90

- (i) G₂ aus der Zugriffstabelle entfernen (A -> D)
- (ii) Weiter mit Entzug P von D:
t_{min}(D) = 80 (C -> D)
Lösche alle ausgehenden Kanten aus B mit ZM < t_{min}(D) = 80
=> D -> B und D -> E löschen
- (iii) Weiter mit Entzug P von B:
t_{min}(B) = 10 => Es gibt keine ausgehenden Kanten aus B zu löschen.
- (iv) Weiter mit Entzug P von E:
t_{min}(E) = 40 => Es gibt keine ausgehenden Kanten aus E zu löschen.

Zugriffstabelle:

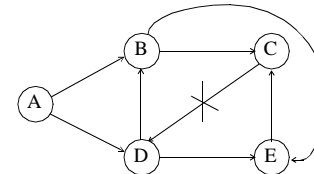


	E	G	ZM
	A	—	0
G ₁ :	B	A	10
G ₃ :	C	B	30
G ₄ :	E	B	40
G ₇ :	C	E	70
G ₈ :	D	C	80

E = Empfänger
G = Geber
ZM = Zeitmarke

b) Autorisierungsgraph mit zeitunabhängiger Interpretation

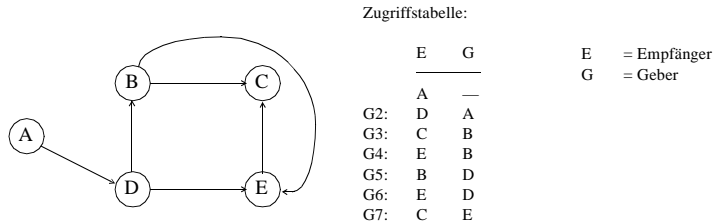
Zugriffstabelle:



	E	G
	A	—
G ₁ :	B	A
G ₂ :	D	A
G ₃ :	C	B
G ₄ :	E	B
G ₅ :	B	D
G ₆ :	E	D
G ₇ :	C	E
G₈:	D	C

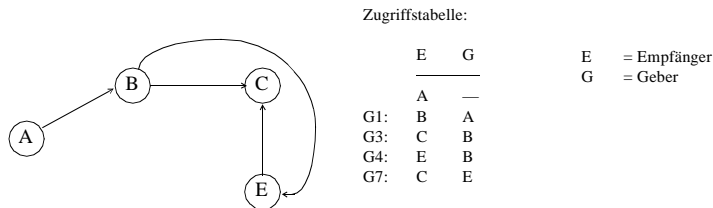
E = Empfänger
G = Geber

- (1) G8 darf nicht gewährt werden, weil sie Zyklen verursachen.
Sei $N(E)$ die Anzahl der Geber, die dem Empfänger E jeweils das gleiche Zugriffsrecht auf das Objekt P gegeben haben.
- (2) REVOKE1: B, A
- (i) G1 aus der Zugriffstabelle entfernen (A -> B)
 - (ii) Weiter mit Entzug P von B:
 $N(B) = 1 > 0 \Rightarrow$ Rekursiver Entzug kann an dieser Stelle nicht fortgesetzt werden weil, B das gleiche Zugriffsrecht von D besitzt.



REVOKE2: D, A

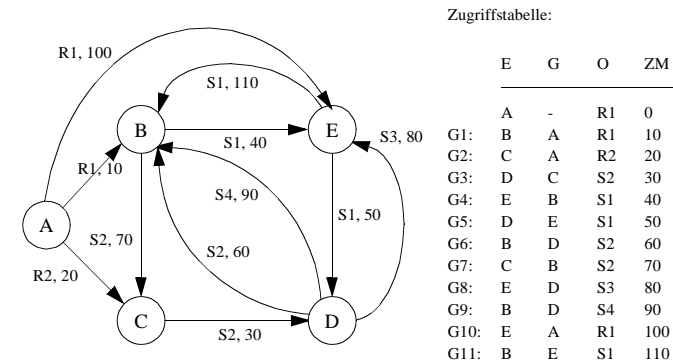
- (i) G2 aus der Zugriffstabelle entfernen (A -> D)
- (ii) Weiter mit Entzug P von D:
 $N(D) = 0 \Rightarrow$ Lösche alle ausgehenden Kanten aus D \Rightarrow D -> B und D -> E löschen
- (iii) Weiter mit Entzug P von B:
 $N(B) = 1 > 0 \Rightarrow$ Rekursiver Entzug kann an dieser Stelle nicht fortgesetzt werden weil, B das gleiche Zugriffsrecht von A besitzt.
- (iv) Weiter mit Entzug P von E:
 $N(E) = 1 > 0 \Rightarrow$ Rekursiver Entzug kann an dieser Stelle nicht fortgesetzt werden weil, E das gleiche Zugriffsrecht von B besitzt.



Aufgabe 4: Entzug von wertabhängigen Zugriffsrechten

Gegeben seien folgender Autorisierungsgraph mit zeitabhängiger Interpretation und die zugehörige Zugriffstabelle. Dabei werden wertabhängige Autorisierungen mit Hilfe von Sichten formuliert. Es sei ein Zugriffsrecht P an den Objekten R1, R2, S2, S3 und S4 jeweils mit der GRANT-Option gewährt worden und es gelte dabei die dezentrale Vergabe der Zugriffsrechte durch Eigentümer der Objekte. Weiterhin seien:

$$S1 \subset R1; S2 \subset R2; S3 \subset S1 \cup S2 \text{ und } S3 \supset S1, S3 \supset S2; S4 \subset S3$$



- a) Wie verändern sich der Autorisierungsgraph und die Zugriffstabelle jeweils durch folgende Zugriffsrechtsrücknahme:
- (1) Entzug von G1 zum Zeitpunkt 120 (REVOKE1: B, A, R1, 120)
 - (2) Entzug von G2 zum Zeitpunkt 120 (REVOKE2: C, A, R2, 120)
- b) Zeichnen Sie nun den entsprechenden Autorisierungsgraphen mit zeitunabhängiger Interpretation und geben Sie die zugehörige Zugriffstabelle an. Welche Zugriffsrechtsvergaben hätten in der obigen Zugriffstabelle nicht gewährt werden dürfen? Welche Auswirkungen haben jeweils dieselben Zugriffsrechtsrücknahmen aus der Teilaufgabe (a) auf den Autorisierungsgraphen und die Zugriffstabelle?

Lösung:

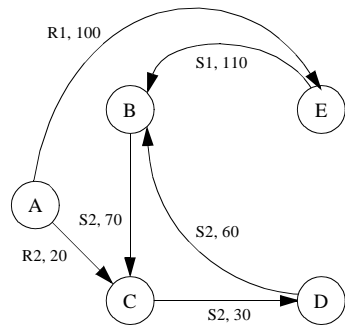
Siehe Härder: Implementierung von Datenbanksystemen, 1978, pp. 195

Beim Entzug einer Relation bzw. Sicht müssen alle auf ihr basierenden Sichten eliminiert werden. Ein Zugriffsrecht muß nicht nur rekursiv entzogen werden, sondern es muß auch von Sichten, die mit Hilfe des in Frage kommenden Zugriffsrechts definiert wurden, zurückgezogen werden.

a) Autorisierungsgraph mit zeitabhängiger Interpretation

(1) REVOKE1: B, A, R1, 120

- (i) G1 aus der Zugriffstabelle entfernen (A -> B)
- (ii) Weiter mit Entzug von R1 auf B:
 $t_{\min}(B, R1) = \infty \Rightarrow$ Es gibt keine ausgehenden Kanten aus B zu löschen.
 $t_{\min}(B, R1 \cup S1) = 110 \Rightarrow$ Lösche G4: E, B, S1, 40
 $t_{\min}(B, R1 \cup S1 \cup S3) = 110 \Rightarrow$ Es gibt keine ausgehenden Kanten aus B zu löschen.
 $t_{\min}(B, R1 \cup S1 \cup S3 \cup S4) = 90 \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
- (iii) Weiter mit Entzug von S1 auf E:
 $t_{\min}(E, R1 \cup S1) = 100 \Rightarrow$ Lösche G5: D, E, S1, 50
 $t_{\min}(E, R1 \cup S1 \cup S3) = 80 \Rightarrow$ Es gibt keine ausgehenden Kanten aus E zu löschen.
 $t_{\min}(E, R1 \cup S1 \cup S3 \cup S4) = 80 \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
- (iv) Weiter mit Entzug von S1 auf D:
 $t_{\min}(D, R1 \cup S1) = \infty \Rightarrow$ Es gibt keine ausgehenden Kanten aus D zu löschen.
 $t_{\min}(D, R1 \cup S1 \cup S3) = \infty \Rightarrow$ Lösche G8: E, D, S3, 80
 $t_{\min}(D, R1 \cup S1 \cup S3 \cup S4) = \infty \Rightarrow$ Lösche G9: B, D, S4, 90
- (v) Weiter mit Entzug von S3 auf E:
 $t_{\min}(E, R1 \cup S1 \cup S3) = 100 \Rightarrow$ Es gibt keine ausgehenden Kanten aus E zu löschen.
 $t_{\min}(E, R1 \cup S1 \cup S3 \cup S4) = 100 \Rightarrow$ Keine ausgehenden Kanten aus E zu löschen.
- (vi) Weiter mit Entzug von S4 auf B:
 $t_{\min}(B, R1 \cup S1 \cup S3 \cup S4) = 100 \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.



Zugriffstabelle:

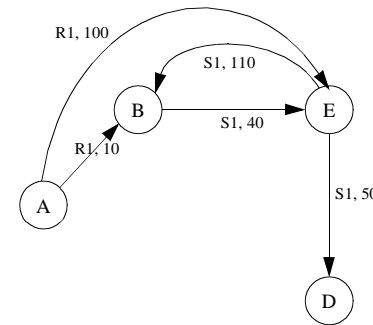
	E	G	O	ZM
A	-	R1	0	
G2:	C	A	R2	20
G3:	D	C	S2	30
G6:	B	D	S2	60
G7:	C	B	S2	70
G10:	E	A	R1	100
G11:	B	E	S1	110

(2) REVOKE2: C, A, R2, 120

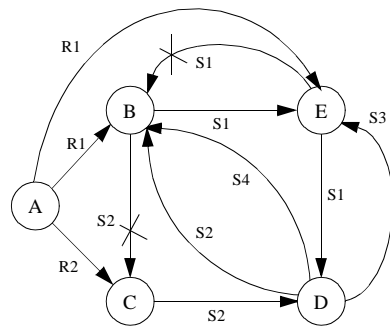
- (i) G2 aus der Zugriffstabelle entfernen (A -> C)
- (ii) Weiter mit Entzug von R2 auf C:
 $t_{\min}(C, R2) = \infty \Rightarrow$ Es gibt keine ausgehenden Kanten aus C zu löschen.
 $t_{\min}(C, R2 \cup S2) = 70 \Rightarrow$ Lösche G3: D, C, S2, 30
 $t_{\min}(C, R2 \cup S2 \cup S3) = 70 \Rightarrow$ Es gibt keine ausgehenden Kanten aus C zu löschen.
 $t_{\min}(C, R2 \cup S2 \cup S3 \cup S4) = 70 \Rightarrow$ Keine ausgehenden Kanten aus C zu löschen.
- (iii) Weiter mit Entzug von S2 auf D:
 $t_{\min}(D, R2 \cup S2) = \infty \Rightarrow$ Lösche G6: B, D, S2, 60
 $t_{\min}(D, R2 \cup S2 \cup S3) = \infty \Rightarrow$ Lösche G8: E, D, S3, 80
 $t_{\min}(D, R2 \cup S2 \cup S3 \cup S4) = \infty \Rightarrow$ Lösche G9: B, D, S4, 90
- (iv) Weiter mit Entzug von S2 auf B:
 $t_{\min}(B, R2 \cup S2) = \infty \Rightarrow$ Lösche G7: C, B, S2, 70
 $t_{\min}(B, R2 \cup S2 \cup S3) = \infty \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
 $t_{\min}(B, R2 \cup S2 \cup S3 \cup S4) = \infty \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
- (v) Weiter mit Entzug von S3 auf E:
 $t_{\min}(E, R2 \cup S2 \cup S3) = \infty \Rightarrow$ Es gibt keine ausgehenden Kanten aus E zu löschen.
 $t_{\min}(E, R2 \cup S2 \cup S3 \cup S4) = \infty \Rightarrow$ Keine ausgehenden Kanten aus E zu löschen.
- (vi) Weiter mit Entzug von S4 auf B:
 $t_{\min}(B, R2 \cup S2 \cup S3 \cup S4) = \infty \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
- (vii) Weiter mit Entzug von S2 auf C:
 $t_{\min}(D, R2 \cup S2) = \infty \Rightarrow$ Keine ausgehenden Kanten aus C zu löschen.
 $t_{\min}(D, R2 \cup S2 \cup S3) = \infty \Rightarrow$ Keine ausgehenden Kanten aus C zu löschen.
 $t_{\min}(D, R2 \cup S2 \cup S3 \cup S4) = \infty \Rightarrow$ Keine ausgehenden Kanten aus C zu löschen.

Zugriffstabelle:

	E	G	O	ZM
A	-	R1	0	
G1:	B	A	R1	10
G4:	E	B	S1	40
G5:	D	E	S1	50
G10:	E	A	R1	100
G11:	B	E	S1	110



b) Autorisierungsgraph mit zeitunabhängiger Interpretation



Zugriffstabelle:

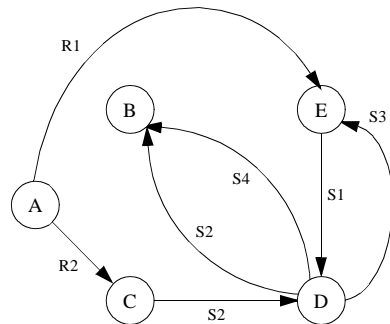
	E	G	O
	A	-	R1
G1:	B	A	R1
G2:	C	A	R2
G3:	D	C	S2
G4:	E	B	S1
G5:	D	E	S1
G6:	B	D	S2
G7:	C	B	S2
G8:	E	D	S3
G9:	B	D	S4
G10:	E	A	R1
G11:	B	E	S1

G7 (B -> C) ist nicht erlaubt wegen Zyklus B -> C -> D -> B bzgl. S2
 G11 (E -> B) ist nicht erlaubt wegen Zyklus E -> B -> E bzgl. S1

(1) REVOKE1: B, A, R1

Sei $N(E, O)$ die Anzahl der Geber, die dem Empfänger E jeweils das gleiche Zugriffsrecht auf das Objekt O gegeben haben.

- (i) G1 aus der Zugriffstabelle entfernen (A -> B)
- (ii) Weiter mit Entzug von R1 auf B:
 $N(B, R1) = 0 \Rightarrow$ Es gibt keine ausgehenden Kanten aus B zu löschen.
 $N(B, R1 \cup S1) = 0 \Rightarrow$ Lösche G4: E, B, S1
 $N(B, R1 \cup S1 \cup S3) = 0 \Rightarrow$ Es gibt keine ausgehenden Kanten aus B zu löschen.
 $N(B, R1 \cup S1 \cup S3 \cup S4) = 1 > 0 \Rightarrow$ Ausgehende Kanten bzgl. S4 werden nicht gelöscht.
- (iii) Weiter mit Entzug von S1 auf E:
 $N(E, R1 \cup S1) = 1 > 0 \Rightarrow$ Ausgehende Kanten bzgl. S4 werden nicht gelöscht.

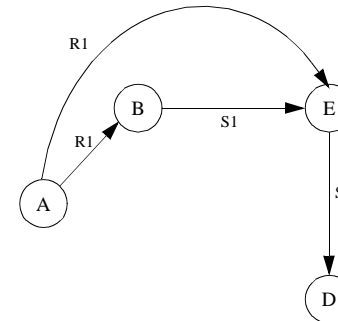


Zugriffstabelle:

	E	G	O
	A	-	R1
G2:	C	A	R2
G3:	D	C	S2
G5:	D	E	S1
G6:	B	D	S2
G8:	E	D	S3
G9:	B	D	S4
G10:	E	A	R1

(2) REVOKE2: C, A, R2

- (i) G2 aus der Zugriffstabelle entfernen (A -> C)
- (ii) Weiter mit Entzug von R2 auf C:
 $N(C, R2) = 0 \Rightarrow$ Es gibt keine ausgehenden Kanten aus C zu löschen.
 $N(C, R2 \cup S2) = 0 \Rightarrow$ Lösche G3: D, C, S2
 $N(C, R2 \cup S2 \cup S3) = 0 \Rightarrow$ Es gibt keine ausgehenden Kanten aus C zu löschen.
 $N(C, R2 \cup S2 \cup S3 \cup S4) = 0 \Rightarrow$ Keine ausgehenden Kanten aus C zu löschen.
- (iii) Weiter mit Entzug von S2 auf D:
 $t_{\min}(D, R2 \cup S2) = 0 \Rightarrow$ Lösche G6: B, D, S2
 $t_{\min}(D, R2 \cup S2 \cup S3) = 0 \Rightarrow$ Lösche G8: E, D, S3
 $t_{\min}(D, R2 \cup S2 \cup S3 \cup S4) = 0 \Rightarrow$ Lösche G9: B, D, S4
- (iv) Weiter mit Entzug von S2 auf B:
 $t_{\min}(B, R2 \cup S2) = 0 \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
 $t_{\min}(B, R2 \cup S2 \cup S3) = 0 \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
 $t_{\min}(B, R2 \cup S2 \cup S3 \cup S4) = 0 \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.
- (v) Weiter mit Entzug von S3 auf E:
 $t_{\min}(E, R2 \cup S2 \cup S3) = 0 \Rightarrow$ Es gibt keine ausgehenden Kanten aus E zu löschen.
 $t_{\min}(E, R2 \cup S2 \cup S3 \cup S4) = 0 \Rightarrow$ Keine ausgehenden Kanten aus E zu löschen.
- (vi) Weiter mit Entzug von S4 auf B:
 $t_{\min}(B, R2 \cup S2 \cup S3 \cup S4) = 0 \Rightarrow$ Keine ausgehenden Kanten aus B zu löschen.



Zugriffstabelle:

	E	G	O
	A	-	R1
G1:	B	A	R1
G4:	E	B	S1
G5:	D	E	S1
G10:	E	A	R1