

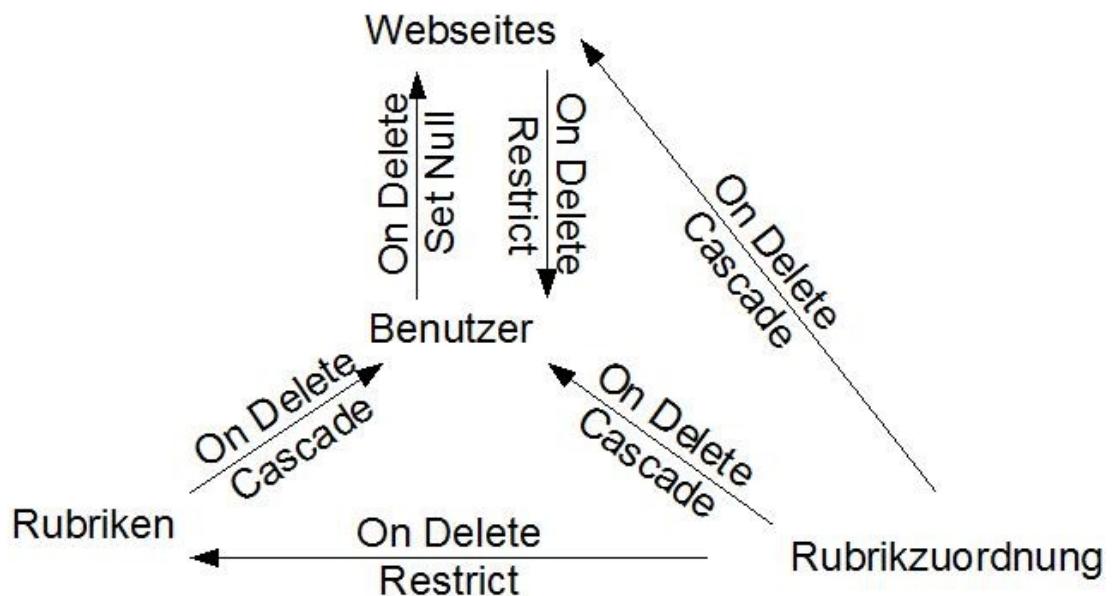
GDB Uebung 2, Gruppe 61

Arne Beer, MN 6489196

Oliver Heidmann, MN 6420331, Minh Nguyen, MN 6423136

9. Januar 2014

1. (a) Das Ergebnis muss unabhängig von der Reihenfolge der referentiellen Aktionen sein.
(b)



- (c) Das Löschen von Benutzer ist reihenfolgenabhängig. Geht man davon aus, dass der Benutzer keine Webseite eingestellt hat, so kann man entweder die Reihenfolge Benutzer-Rubriken-Rubrikzuordnung wählen, wobei der Löschvorgang abgebrochen wird, aufgrund des Restrict zwischen Rubriken und Rubrikzuordnung oder die Reihenfolge Benutzer-Rubrikzuordnung-Rubriken, die erfolgreich abläuft (Löschen in Benutzer, Löschen in Rubrikzuordnung, Löschen in Rubriken).
 - (d) Man kann das Restrict zwischen Rubriken und Rubrikzuordnung zu Cascade ändern, oder alle Eingangspfeile von Benutzer zu District ändern. Da der Löschvorgang immer zurückgesetzt wird sobald Benutzer eine Webseite eingestellt hat, ist das Ergebnis in jeder Reihenfolge gleich.
2. (a) i.

```
CREATE VIEW EnterpriseCrew AS
SELECT a.BNr, a.Name, a.Rang
FROM Besatzungsmitglieder a, Raumschiffe b.
```

```
WHERE b.Name = 'Enterprise'
AND a.Schiff = b.RNr;
```

Sicht nicht änderbar aufgrund des join

```
ii. CREATE VIEW Captains AS
SELECT Name
FROM Besatzungsmitglieder
WHERE Rang = 'Captain';
```

iii. Sicht nicht änderbar da der Primärschlüssel fehlt

```
iv. CREATE VIEW WarpFed AS
SELECT RNr, Fraktion, Baujahr
FROM Raumschiffe
WHERE Geschwindigkeit > 0
AND Fraktion = 'Förderung';
```

Sicht änderbar

- (b) i. Anweisung durchführbar. Änderungen in der Sicht Förderungsschiffe.
- ii. Anweisung nicht durchführbar, da es die Bedingung Fraktion = 'Förderung' in Förderungsschiffe nicht erfüllt (WITH CASCADED CHECK OPTION)
- iii. Anweisung durchführbar. Änderungen in der Sicht Forschungsschiffe, Förderungsschiffe, GalaxyKlasse
- iv. Anweisung nicht durchführbar, aufgrund der Verletzung der Bedingung Baujahr \geq 2365 in NebulaKlasse (WITH CASCADED CHECK OPTION)
- v. Anweisung durchführbar. Änderungen in der Sicht Forschungsschiffe und Förderungsschiffe
3. S1: i. A = 305 ; B = 195
- ii. $r_1(B) \rightarrow w_2(B)$
 $r_1(A) \rightarrow w_2(A)$
 $w_1(A) \rightarrow r_2(A)$
 $w_1(A) \rightarrow w_2(A)$
- iii. Schedule ist seriell.
- S2: i. A = 195 ; B = 10
- ii. $r_2(A) \rightarrow w_1(A)$
 $r_1(B) \rightarrow w_2(B)$
 $r_1(A) \rightarrow w_2(A)$
 $w_2(A) \rightarrow w_1(A)$
- iii. Schedule ist nicht serialisierbar
- S3: i. A = 300 ; B = 5
- ii. $r_2(A) \rightarrow w_1(A)$
 $w_2(B) \rightarrow r_1(B)$
 $w_2(A) \rightarrow w_1(A)$
 $w_2(A) \rightarrow r_1(A)$
- iii. Schedule ist serialisierbar, da alle Abhängigkeiten T_1 vor T_2 festlegen.

S4: i. $A = 190$; $B = 5$

- ii. $r_2(A) \longrightarrow w_1(A)$
 $w_2(B) \longrightarrow r_1(B)$
 $r_1(A) \longrightarrow w_2(A)$
 $w_2(A) \longrightarrow w_1(A)$

iii. Schedule ist nicht serialisierbar

S5: i. $A = 115$; $B = 5$

- ii. $r_2(A) \longrightarrow w_1(A)$
 $r_1(B) \longrightarrow w_2(B)$
 $r_1(A) \longrightarrow w_2(A)$
 $w_1(A) \longrightarrow w_2(A)$

iii. Schedule ist nicht serialisierbar

S6: i. $A = 305$; $B = 5$

- ii. $r_2(A) \longrightarrow w_1(A)$
 $w_2(B) \longrightarrow r_1(B)$
 $w_2(A) \longrightarrow r_1(A)$
 $w_2(A) \longrightarrow w_1(A)$

iii. Schedule ist seriell.

4.

$$S = w_1(x)r_2(y)r_3(z)w_3(y)r_2(z)w_3(z)w_1(z)r_2(y)c_3c_1c_2$$

Zeitschritt	T_1	T_2	T_3	x	y	z	Bemerkung
0				NL	NL	NL	
1	lock(x,X)			X_1	NL	NL	
2	write(x)	lock(y,R)		X_1	R_2	NL	
3		read(y)	lock(z,R)	X_1	R_2	R_3	
4			read(z)	X_1	R_2	R_3	
5			lock(y,X)	X_1	R_2	R_3	T_3 wartet auf Freigabe von y
6		lock(z,R)		X_1	R_2	R_3 R_2	
7		read(z)	lock(z,X)	X_1	R_2	R_3 R_2	T_3 wartet auf Freigabe von z
8	lock(z,X)			X_1	R_2	R_3 R_2	T_1 wartet auf Freigabe von z
9		read(y)		X_1	R_2	R_3 R_2	
10		unlock(y)		X_1	X_3	R_3 R_2	T_3 wird benachrichtigt
11		unlock(z)	write(y)	X_1	X_3	X_3	T_3 wird benachrichtigt
12			write(z)	X_1	X_3	X_3	
13			unlock(y)	X_1	NL	X_3	
14			unlock(z)	X_1	NL	X_1	T_1 wird benachrichtigt
15	write(z)		commit	X_1	NL	X_1	
16	unlock(x)			NL	NL	X_1	
17	unlock(z)			NL	NL	NL	
18	commit						
19		commit					