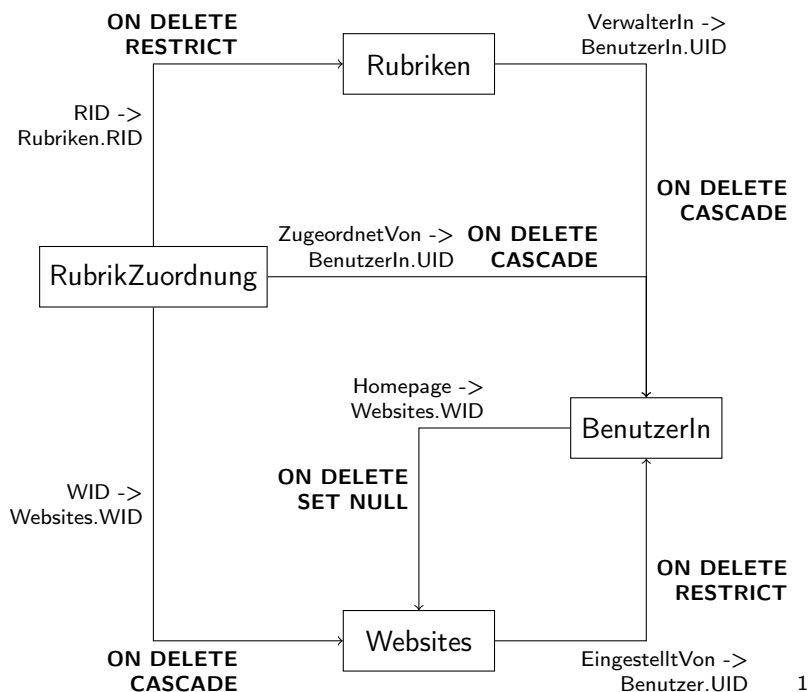
	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	5			
	STiNE-Gruppe 11	Kobras, Pöhlmann, Tsiamis			
	Ausgabe	Mi. 09.12.2015	Abgabe	Fr. 08.01.2016	

1 Referentiell Aktionen

1.1 Teilaufgabe a

Ein bezüglich der referentiellen Aktion sicheres Schema darf nicht Reihenfolge-abhängig sein. Dies gewährleistet die referentielle Integrität.

1.2 Teilaufgabe b




1.3 Teilaufgabe c

Das Schema ist in folgenden Fällen unsicher bezüglich referentiellen Aktionen:

Ein Eintrag der Tabelle "BenutzerIn" wird gelöscht.

Aktualisiert man zuerst die Tabelle "RubrikZuordnung" und danach erst die Tabelle "Rubriken", kann dies dazu führen, dass sich Rubriken jetzt löschen lassen, die auf anderem Wege abgelehnt worden wären.

¹Zwischen "Rubriken", "RubrikZuordnung" und "BenutzerIn" ist keine Kreuzung, hier überlagern sich nur 2 Linien, weil ich es nicht besser zeichnen konnte.

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	5			
	STiNE-Gruppe 11	Kobras, Pöhlmann, Tsiamis			
	Ausgabe	Mi. 09.12.2015	Abgabe	Fr. 08.01.2016	

1.4 Teilaufgabe d

Dieser Missstand lässt sich z.B. dadurch beheben, dass man das **ON DELETE RESTRICT** zwischen "Rubriken" und "RubrikZuordnung" in ein **ON DELETE CASCADE** umändert.

Andernfalls könnte man auch die beiden **ON DELETE CASCADE** zwischen "BenutzerIn" und "Rubriken" und zwischen "BenutzerIn" und "RubrikZuordnung" jeweils in ein **ON DELETE RESTRICT** umändern.

2 Änderbarkeit von Sichten

2.1 Teilaufgabe a

2.1.1 i

```
CREATE VIEW EnterpriseCrew
AS SELECT BNr, Name, Rang
FROM Besatzungsmitglieder
WHERE IN
(SELECT * FROM Besatzungsmitglieder, Raumschiffe
WHERE Besatzungsmitglieder.Schiff = Raumschiffe.RNr
AND Raumschiffe.Name = 'Enterprise')
WITH CASCADED CHECK OPTION;
```

Diese Relation ist änderbar. Deshalb wurde eine **CHECK OPTION** eingefügt.


2.1.2 ii

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

Diese Relation ist nicht änderbar, da der Primärschlüssel fehlt.

2.1.3 iii

```
CREATE VIEW WarpFed
AS SELECT RNr, Fraktion, Baujahr
FROM Raumschiffe
WHERE Raumschiffe.Geschwindigkeit >= 1
AND Raumschiffe.Fraktion = 'Föderation'
WITH CASCADED CHECK OPTION;
```

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	5			
	STiNE-Gruppe 11	Kobras, Pöhlmann, Tsiamis			
	Ausgabe	Mi. 09.12.2015	Abgabe	Fr. 08.01.2016	

Diese Relation ist änderbar. Deshalb wurde eine **CHECK OPTION** eingefügt.

2.2 Teilaufgabe b

2.2.1 i

Diese SQL-Anweisung kann durchgeführt werden.

Die geänderten Tupel sind in folgenden Sichten auf jeden Fall alle zu sehen:

- Föderationsschiffe

2.2.2 ii

Diese SQL-Anweisung kann durchgeführt werden.

Die geänderten Tupel sind in folgenden Sichten auf jeden Fall alle zu sehen:

- Föderationsschiffe
- Forschungsschiffe

2.2.3 iii

Diese SQL-Anweisung kann durchgeführt werden.

Die geänderten Tupel sind in folgenden Sichten auf jeden Fall alle zu sehen:

- Föderationsschiffe
- Forschungsschiffe
- GalaxyKlasse

2.2.4 iv


Diese SQL-Anweisung kann nicht durchgeführt werden.

2.2.5 v

Diese SQL-Anweisung kann durchgeführt werden.

Die geänderten Tupel sind in folgenden Sichten auf jeden Fall alle zu sehen:

- Föderationsschiffe
- Forschungsschiffe


	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	5		
	STiNE-Gruppe 11	Kobras, Pöhlmann, Tsiamis		
	Ausgabe	Mi. 09.12.2015	Abgabe	Fr. 08.01.2016

3 Serialisierbarkeit und Anomalien

3.1 Teilaufgabe a

Die Bindung für A und B nach den folgenden Zeitplänen:

- S_1
 - A = 305
 - B = 195
- S_2
 - A = 195
 - B = 5
- S_3
 - A = 300
 - B = 5
- S_4
 - A = 190
 - B = 5
- S_5
 - A = 115
 - B = 5
- S_6
 - A = 300
 - B = 5

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	5		
	STiNE-Gruppe 11	Kobras, Pöhlmann, Tsiamis		
	Ausgabe	Mi. 09.12.2015	Abgabe	Fr. 08.01.2016

3.2 Teilaufgabe b

Es existieren folgende Abhängigkeiten:

- T_1 liest A und T_2 schreibt A.
- T_2 liest A und T_1 schreibt A.
- T_1 liest B und T_2 schreibt B.

Daraus folgt, dass verschiedene Werte für A und B raus kommen, je nachdem welche dieser 6 Aktionen an welcher Stelle kommt.

So greifen z.B:

$w_1(A)$, $r_1(A)$, $w_2(A)$, $r_2(A)$

alle auf das erste Feld zu.


Und

$w_1(B)$, $r_2(B)$

beide auf das zweite Feld.

Dies bedeutet für die folgenden Schedules:

- S_1
 - Abhängigkeit($w_1(A)$ vor $r_2(A)$)
- S_2
 - Abhängigkeit($r_1(A)$ vor $w_2(A)$)
 - Abhängigkeit($r_1(B)$ vor $w_2(B)$)
 - Abhängigkeit($w_2(A)$ vor $w_1(A)$)
- S_3
 - Abhängigkeit($w_2(B)$ vor $r_1(B)$)
 - Abhängigkeit($w_2(A)$ vor $r_1(A)$)
- S_4
 - Abhängigkeit($w_2(B)$ vor $r_1(B)$)
 - Abhängigkeit($r_1(A)$ vor $w_2(A)$)
 - Abhängigkeit($w_2(A)$ vor $w_1(A)$)
- S_5
 - Abhängigkeit($r_2(A)$ vor $w_1(A)$)
 - Abhängigkeit($w_1(A)$ vor $w_2(A)$)
- S_6


	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	5		
	STiNE-Gruppe 11	Kobras, Pöhlmann, Tsiamis		
	Ausgabe	Mi. 09.12.2015	Abgabe	Fr. 08.01.2016

- Abhängigkeit($w_2(B)$ vor $r_1(B)$)
- Abhängigkeit($w_2(A)$ vor $r_1(A)$)

3.3 Teilaufgabe c

Die Zeitpläne sind:

- S_1
 - seriell, da erst alle Operationen von T_1 und danach alle Operationen von T_2 ausgeführt werden (da Abhängigkeit($w_1(A)$ vor $r_2(A)$) gilt).
- S_2
 - nicht serialisierbar, da T_1 den Grundwert liest und danach das Ergebnis von T_2 überschreibt (da Abhängigkeit($w_2(A)$ vor $w_1(A)$) und Abhängigkeit($r_1(A)$ vor $w_2(A)$) gilt).
 - Datenanomalie: T_2 hat in den Daten nicht statt gefunden.
- S_3
 - serialisierbar, da T_1 erst liest, nachdem T_2 sie nicht mehr weiter bearbeitet (da Abhängigkeit($w_2(B)$ vor $r_1(B)$) und Abhängigkeit($w_2(A)$ vor $r_1(A)$) gilt).
- S_4
 - nicht serialisierbar, da die Veränderung des Wertes A von T_2 nicht in der Datenbank auftaucht (da Abhängigkeit($r_1(A)$ vor $w_2(A)$) und Abhängigkeit($w_2(A)$ vor $w_1(A)$) gilt).
 - Datenanomalie: T_2 verändert in den Daten nur B nicht jedoch A.
- S_5
 - nicht serialisierbar, da die Veränderung des Wertes A von T_1 nicht in der Datenbank auftaucht (da Abhängigkeit($r_2(A)$ vor $w_1(A)$) und Abhängigkeit($w_1(A)$ vor $w_2(A)$) gilt).
 - Datenanomalie: T_1 hat in den Daten nicht statt gefunden.
- S_6
 - seriell, da erst T_2 und danach T_1 ausgeführt wird (da Abhängigkeit($w_2(B)$ vor $r_1(B)$) und Abhängigkeit($w_2(A)$ vor $r_1(A)$) gilt).

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	5			
	STiNE-Gruppe 11	Kobras, Pöhlmann, Tsiamis			
	Ausgabe	Mi. 09.12.2015	Abgabe	Fr. 08.01.2016	

4 Transaktionen

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	z hat nun 2 Lesesperren auf sich
7	lock(z, X)	read(z)		X ₁	R ₂	R ₃ ∧ R ₂	T ₁ wartet auf Freigabe von z
8		read(y)		X ₁	R ₂	R ₃ ∧ R ₂	
9		unlock(y)		X ₁	X ₃	R ₃ ∧ R ₂	Benachrichtigung von T ₃
10		unlock(z)	write(y)	X ₁	X ₃	R ₃	erste Lesesperre wird von z entfernt
11		commit	lock(z, X)	X ₁	X ₃	X ₃	
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	Benachrichtigung von T ₁
15	write(z)		commit	X ₁	NL	X ₁	
16	unlock(x)			NL	NL	X ₁	
17	unlock(z)			NL	NL	NL	alle Objekte sind wieder freigegeben
18	commit			NL	NL	NL	alle Transaktionen sind fertig
19				NL	NL	NL	
20				NL	NL	NL	
21				NL	NL	NL	
22				NL	NL	NL	
23				NL	NL	NL	
24				NL	NL	NL	
25				NL	NL	NL	