

GDB Aufgabenblatt 5

Abgegeben von Timon Back, Fabian Behrendt, Nicolai Stäger

Aufgabe 1: Referentielle Aktionen

1.A

Welche Anforderung erfüllt ein (bzgl. der referentiellen Aktionen) sicheres Schema?

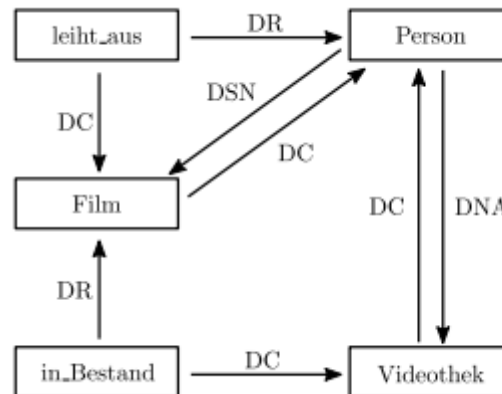
Bei einem sicheren Schema ist das Ergebnis der Operation reihenfolge-unabhängig. Wenn also eine Operation durchgeführt wird und diese mehrere Folgeoperationen auslöst (Cascade, Restrictions), dann ist das Resultat identisch, egal in welcher Reihenfolge diese Folgeoperation ausgeführt werden.

1.B

Handelt es sich im vorliegenden Fall um ein sicheres Schema?

In diesem Fall handelt es sich um kein sicheres Schema.

Sobald ein Eintrag in Person gelöscht wird, wird durch DC auch die entsprechenden Einträge in Film und Videothek gelöscht. Wenn nun Videothek per DC zuerst an in_Bestand weiterleitet, dann kann sich ein anderes Ergebnis ergeben als wenn Film zuerst über DR an in_Bestand weiterleitet.



Erklärung:
 DC = On Delete Cascade
 DR = On Delete Restrict
 DSN = On Delete Set Null
 DNA = On Delete No Action

Ein zweites Problem ergibt sich ebenfalls wenn die Person gelöscht wird. Denn wenn sofort geprüft wird ob die Person etwas ausgeliehen hat (DR), dann bricht die Operation unter Umständen ab, was sie allerdings nicht tun würden, wenn zuerst per DC erst in Film und dann per DC in leih_aus gelöscht wird.

Ein ähnliches Szenario ergibt sich beim Löschen eines Films. Wird das Löschen erst an in_Bestand weitergeleitet, so wird das Löschen verhindert. Wird das Löschen jedoch erst an leih_aus weitergeleitet, dann wird diese Verknüpfung erst gelöscht.

Aufgabe 2: Änderbarkeit von Sichten

2.A

FerrariMechaniker: Vorname und Nachname aller Mechaniker, die Ferrari-Rennwagen warten.

```

CREATE VIEW FerrariMechaniker
AS SELECT Vorname, Nachname FROM Mechaniker
WHERE wartet IN (SELECT RNr FROM Rennwagen WHERE Rennstall = 'Ferrari')
  
```

Änderungsoperationen sind zulässig.

reicheMechaniker : Vorname und Nachname aller Mechaniker, die ein Gehalt von mehr als 2 Millionen Geldeinheiten haben

```
CREATE VIEW reicheMechaniker
AS SELECT Vorname, Nachname FROM Mechaniker
WHERE Gehalt >= 2000000
WITH CASCADED CHECK OPTION
```

Änderungsoperationen sind zulässig.

alteRennserien: Die Namen aller Rennserien, die es schon vor 1950 gab.

```
CREATE VIEW alteRennserien
AS SELECT DISTINCT Rennserie FROM Rennwagen
WHERE Jahr < 1950
```

Änderungsoperationen sind nicht zulässig. Es fehlt der Primärschlüssel.

FerrariWagen: RNr, Typ, Rennserie und Jahr aller Rennwagen aus dem Stall 'Ferrari'.

```
CREATE VIEW FerrariWagen
AS SELECT RNr, Typ, Rennserie, Jahr FROM Rennwagen
WHERE Rennstall = 'Ferrari'
WITH CASCADED CHECK OPTION
```

Änderungsoperationen sind zulässig.

2.B

Bestimmen Sie, ob die folgenden SQL-Anweisungen auf diesen Sichtdefinitionen durchgeführt werden können.

- i) Diese Operation kann durchgeführt werden. Nur in Formel1_Wagen und Auto-Union-Rennwagen können neue Tupel auftreten.
- ii) Diese Operation kann nicht durchgeführt werden. Die Check-Option wird nicht erfüllt
- iii) Diese Operation kann ausgeführt werden. Alle Änderungen werden in Formel1_Wagen und Ferrari_F1_Wagen sichtbar.
- iv) Diese Operation kann nicht ausgeführt werden, da der neue Eintrag die Rennserie AVUS haben würde, welches die CHECK-OPTION von Formel1_Wagen nicht zulässt.

Aufgabe 3: Serialisierbarkeit, Anomalien

3.A

Wie lautet nach Ausführung des Schedules die Belegung für die Variablen A und B?

Aufgabe	A	B		Aufgabe	A	B
S1	320	10		S4	215	10
S2	315	215		S5	110	10
S3	315	10		S6	320	10

3.B

Welche Abhängigkeiten existieren zwischen den Operationen der beiden Transaktionen innerhalb des Schedules?

Es entsteht immer dann ein Problem, wenn eine Transaktionen einen Wert liest und verarbeitet und dann bevor der neue Wert geschrieben wird, der ‚alte‘ Wert von einer anderen Transaktion ausgelesen wird.

Für diese Transaktionen bedeutet dies, dass die r und w Operation einer Transaktion vor der r Operation einer anderen Transaktion geschehen muss, wie dies z.B. bei S1 der Fall ist. Die Beiden Transaktionen sind also voneinander abhängig.

3.C

Ist der Schedule seriell, serialisierbar oder nicht serialisierbar? Erläutern sie zusätzlich bei einem nichtserialisierbaren Schedule die auftretenden Datenanomalien. Begründen Sie die Antworten mit Hilfe der Abhängigkeiten!

S1	Seriell (Serielle Abarbeitung der Transaktionen)
S2	Seriell (Transaktionen werden nacheinander abgearbeitet)
S3	Nicht serialisierbar. Der Wert b wird von T2 gelesen, bevor T1 b speichert.
S4	Nicht serialisierbar. Der Wert von a wird von T2 gelesen, bevor T1 a speichert.
S5	Nicht serialisierbar. Der Wert von a und b wird von T2 gelesen, bevor T1 a und b speichert.
S6	Serialisierbar (Keine serielle Abarbeitung, jedoch identisches Ergebnis)

Aufgabe 4: 2PL-Synchronisation mit R/X-Sperren

$S_1 = w_1(x) \ r_2(z) \ r_1(x) \ r_3(y) \ w_2(y) \ w_1(y) \ c_1 \ r_2(z) \ c_2 \ w_3(y) \ c_3$

	T1	T2	T3	x	y	z	Bemerkung
0				NL	NL	NL	
1	lock(x, X)			X1	NL	NL	
2	write(x)	lock(z, R)		X1	NL	R2	
3	lock(x, R)	read(z)		R1	NL	R2	
4	read(x)		lock(y, R)	R1	R3	R2	
5		lock(y, X)	read(y)	R1	R3	R2	T2 muss warten, da y von T3 blockiert ist
6	lock(y, X)			R1	R3	R2	T1 muss warten, da y von T3 blockiert ist
7				R1	R3	R2	Der Commit für T1 kann noch nicht ausgeführt werden
8		read(z)		R1	R3	R2	Der Lesevorgang auf z von T2 wird sofort durchgeführt, da T2 die Ressource bereits reserviert hat
9							Der Commit für T2 kann noch nicht ausgeführt werden
10			lock(y, X)	R1	X3	R2	T3 kann y sofort exklusiv reservieren
11			write(y)	R1	X3	R2	
12			unlock(y)	R1	X2	R2	T3 hat alle Transaktionen abgeschlossen und

							gibt Y frei, somit kann T2 auf y zugreifen
13		write(y)	commit	R1	X2	R2	
14		unlock(y)		R1	X1	R2	T2 hat alle Transaktionen abgeschlossen und beginnt mit dem Freigeben. Sofort kann T1 auf y zugreifen.
15	write(y)	unlock(z)		R1	X1	NL	
16	unlock(x)	commit		NL	X1	NL	T1 hat alle Transaktionen abgeschlossen und beginnt mit dem Freigeben.
17	unlock(y)			NL	NL	NL	
18	commit			NL	NL	NL	