	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020


1 Präsenzaufgabe: Referentielle Aktionen

Gegeben sei folgende Datendefinition:

```

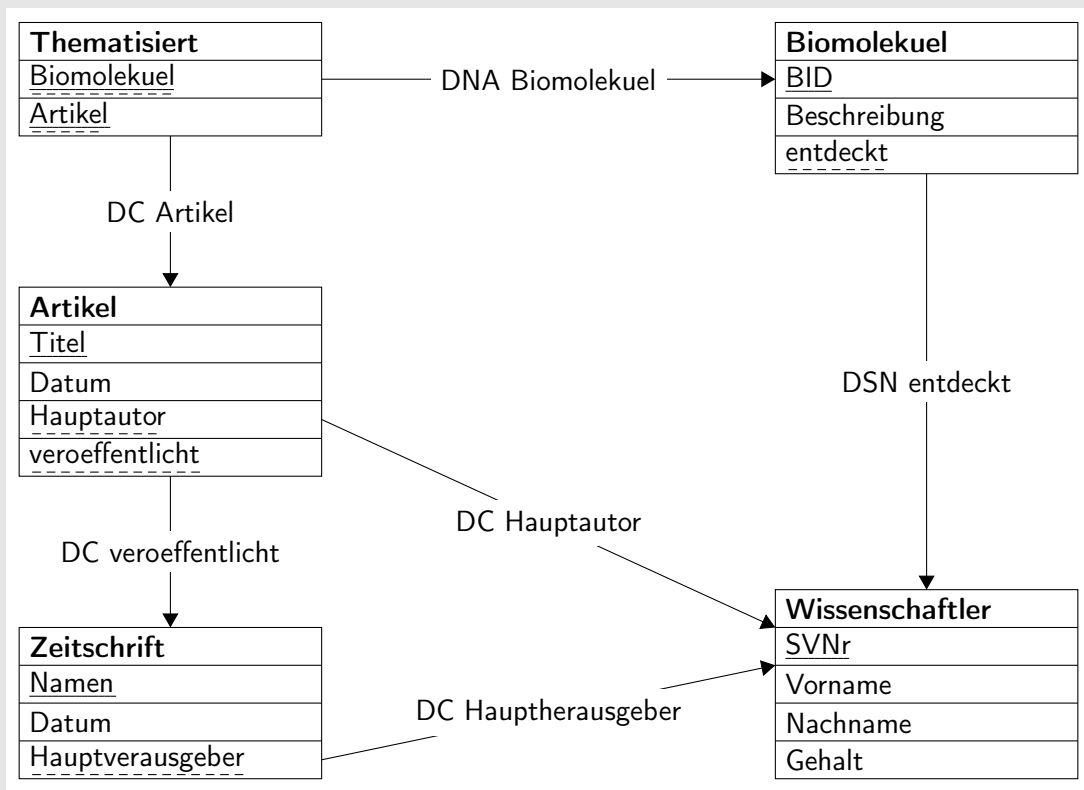
CREATE TABLE Wissenschaftler(
    SVNr int PRIMARY KEY,
    Vorname varchar(25) NOT NULL,
    Nachname varchar(25) NOT NULL,
    Gehalt int NOT NULL
);
CREATE TABLE Biomolekuel(
    BID int PRIMARY KEY,
    Beschreibung varchar(250) NOT NULL,
    entdeckt int,
    CONSTRAINT fk_biomolekuel_entdeckt FOREIGN KEY (entdeckt)
        REFERENCES Wissenschaftler(SVNr) ON DELETE SET NULL
);
CREATE TABLE Zeitschrift(
    Name varchar(25) PRIMARY KEY,
    Datum date NOT NULL,
    Hauptherausgeber int NOT NULL,
    CONSTRAINT fk_zeitschrift_hauptherausgeber FOREIGN KEY (Hauptherausgeber)
        REFERENCES Wissenschaftler(SVNr) ON DELETE CASCADE
);
CREATE TABLE Artikel(
    Titel varchar(25) PRIMARY KEY,
    Datum date NOT NULL,
    Hauptautor int NOT NULL,
    veroeffentlicht varchar(25) NOT NULL,
    CONSTRAINT fk_artikel_hauptautor FOREIGN KEY (Hauptautor)
        REFERENCES Wissenschaftler(SVNr) ON DELETE CASCADE,
    CONSTRAINT fk_artikel_veroeffentlicht FOREIGN KEY (veroeffentlicht)
        REFERENCES Zeitschrift(Name) ON DELETE CASCADE
);
CREATE TABLE Thematisiert(
    Biomolekuel int NOT NULL,
    Artikel varchar(25) NOT NULL,
    CONSTRAINT pk_thematisiert PRIMARY KEY (Biomolekuel,Artikel),
    CONSTRAINT fk_thematisiert_biomolekuel FOREIGN KEY (Biomolekuel)
        REFERENCES Biomolekuel(BID) ON DELETE NO ACTION,
    CONSTRAINT fk_thematisiert_artikel FOREIGN KEY (Artikel)
        REFERENCES Artikel(Titel) ON DELETE CASCADE

```

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

-);
- a) Zeichnen Sie den zugehörigen Referenzgraphen und beschriften Sie alle Kanten mit den entsprechenden referentiellen Aktionen.

Lösungsvorschlag:




- b) Handelt es sich im vorliegenden Fall um ein sicheres Schema? Begründen Sie Ihre Antwort. Sollte das Schema nicht sicher sein, beschreiben Sie ein Szenario, in dem das Ergebnis einer Änderungs-/Löschoperation von der Auswertungsreihenfolge der referentiellen Aktionen abhängig ist.

Lösungsvorschlag:

Das vorliegende Schema ist sicher. Die theoretisch einzig mögliche Stelle für eine Reihenfolgenabhängigkeit im Schema existiert beim Löschen eines Wissenschaftlers, da beim Kaskadieren des Löschens auf zwei Wegen bis zum Artikel kaskadiert werden kann; Einmal über die Zeitschrift und einmal auf direktem Wege. Das vorliegende Schema ist sicher, da auf beiden Wegen keine blockierenden referentiellen Aktionen wie beispielsweise 'ON DELETE RESTRICT' verwendet werden.

- c) Für den Fremdschlüssel 'fk_artikel_veroeffentlicht' aus der Relation 'Artikel' soll nun die referentielle Aktion von 'ON DELETE CASCADE' auf 'ON DELETE RESTRICT' geändert werden. Handelt es sich

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

bei dem geänderten Fall um ein sicheres Schema? Begründen Sie Ihre Antwort. Sollte das Schema nicht sicher sein, beschreiben Sie ein Szenario, in dem das Ergebnis einer Änderungs-/Löschoperation von der Auswertungsreihenfolge der referentiellen Aktionen abhängig ist.

Lösungsvorschlag:


Die SQL-DDL-Anweisungen sind nicht gefordert, sie dienen nur der Vollständigkeit! Siehe auch die Folien zur Schemaevolution, z.B. Folie K5/F114–120:

```
ALTER TABLE Artikel DROP CONSTRAINT fk_artikel_veroeffentlicht;
```

```
ALTER TABLE Artikel ADD CONSTRAINT fk_artikel_veroeffentlicht
    FOREIGN KEY (veroeffentlicht) REFERENCES Zeitschrift(Namen) ON DELETE RESTRICT;
```

Das geänderte Schema ist jetzt nicht mehr sicher, da beim Löschen eines Wissenschaftlers die Löschoperation direkt und über 'Zeitschrift' zu 'Artikel' propagiert wird. Dies führt zu reihenfolgeabhängigen Ergebnissen, wenn ein Wissenschaftler gelöscht wurde, der Hauptautor aller Artikel in den Zeitschriften war, von denen er der Hauptherausgeber ist:

- i) Wird zunächst der direkte Weg beschritten, wird der Artikel gelöscht.
- ii) Wird der Weg über 'Zeitschrift' beschritten, wird das Löschen der Zeitschrift jedoch verweigert, da noch Artikel existieren.

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

2 Präsenzaufgabe: Serialisierbarkeit, Anomalien

Gegeben sind die folgenden Transaktionen $T_1 = r_1(B) r_1(A) w_1(B)$ und $T_2 = r_2(B) w_2(B) w_2(A)$. T_1 liest die Werte von A und B, multipliziert die diese und schreibt den neuen Wert nach B zurück.

T_2 verringert den gelesenen Wert von B um 35 und schreibt den neuen Wert nach B zurück. Außerdem erhöht T_2 den gelesenen Wert von B um 20 und schreibt ihn nach A (ohne A vorher gelesen zu haben).

Der Anfangswert von A sei 65 und der von B sei 120. Allgemein bezeichnet $w_i(x)$ den Schreibzugriff der Transaktion i auf das Objekt x und $r_i(x)$ den Lesezugriff der Transaktion i auf x. Gegeben sind die folgenden Schedules:

$$S_1 = r_2(B) w_2(B) w_2(A) r_1(B) r_1(A) w_1(B)$$

$$S_2 = r_2(B) w_2(B) r_1(B) w_2(A) r_1(A) w_1(B)$$

$$S_3 = r_1(B) r_2(B) r_1(A) w_2(B) w_1(B) w_2(A)$$

Beantworten Sie für jeden der Schedules die folgenden drei Fragestellungen:

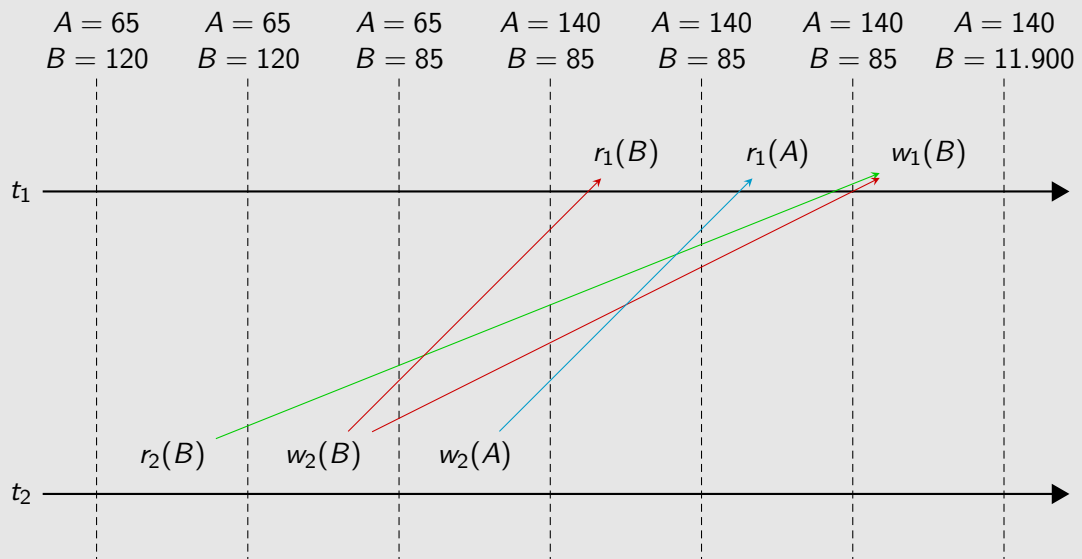
- Wie lautet nach Ausführung des Schedules die Belegung für die Variablen A und B? [3 P.]
- Welche Abhängigkeiten existieren zwischen den Operationen der beiden Transaktionen innerhalb des Schedules? [6 P.]
- Ist der Schedule seriell, serialisierbar oder nicht serialisierbar? Erläutern sie zusätzlich bei einem nicht-serialisierbaren Schedule die auftretenden Datenanomalien. Begründen Sie die Antworten mit Hilfe der Abhängigkeiten! [6 P.]



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	30		
Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

Lösungsvorschlag:

- Schedule S_1 :



a) Endwert für A ist 140, Endwert für B ist 11.900.

b) Abhängigkeiten

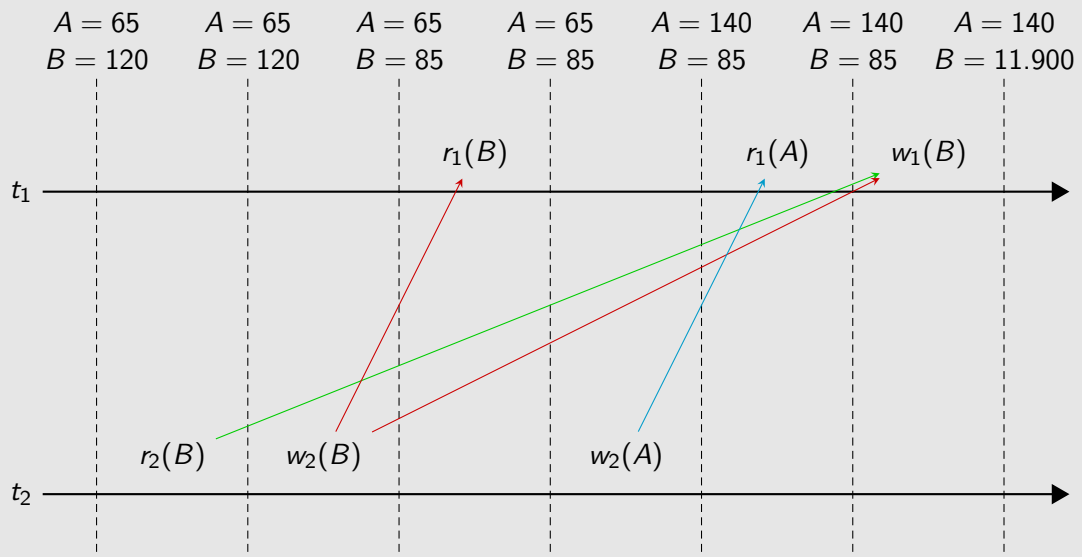
- $r_2(B) \rightarrow w_1(B)$
- $w_2(B) \rightarrow r_1(B)$
- $w_2(B) \rightarrow w_1(B)$
- $w_2(A) \rightarrow r_1(A)$

c) Der Schedule ist seriell.



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	30		
Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

• Schedule S_2 :



a) Endwert für A ist 140, Endwert für B ist 11.900.

b) Abhängigkeiten

- $r_2(B) \rightarrow w_1(B)$
- $w_2(B) \rightarrow r_1(B)$
- $w_2(B) \rightarrow w_1(B)$
- $w_2(A) \rightarrow r_1(A)$

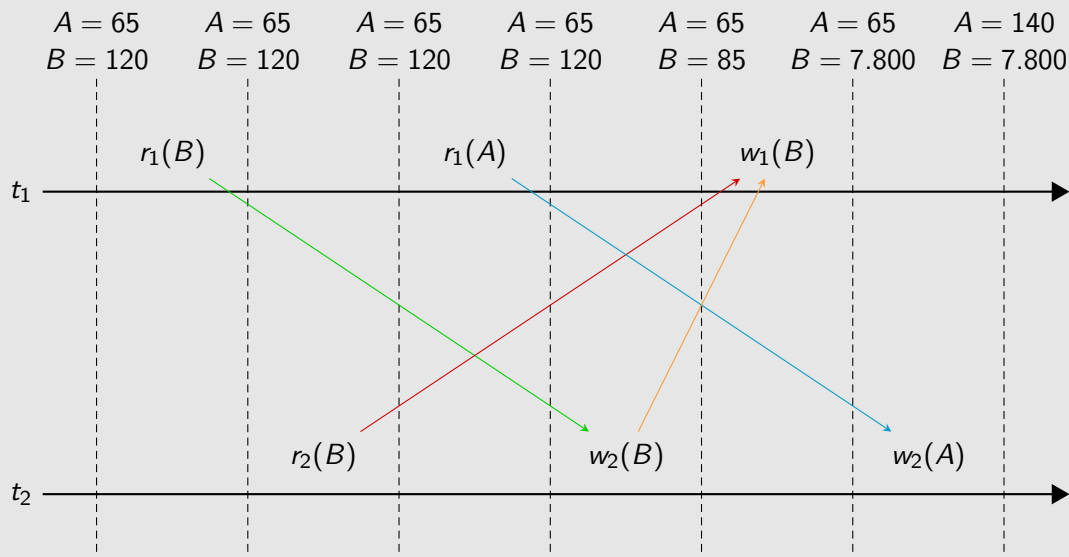
c) - Wegen der Abhängigkeiten $r_2(B) \rightarrow w_1(B)$, $w_2(B) \rightarrow r_1(B)$, $w_2(B) \rightarrow w_1(B)$ und $w_2(A) \rightarrow r_1(A)$ gilt: t_2 vor t_1 .

⇒ S_2 ist serialisierbar (Schedule S_1 erzielt das identische Ergebnis für die Variablen A und B)



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	30		
Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

• Schedule S_3 :



a) Endwert für A ist 140, Endwert für B ist 7.800.


b) Abhängigkeiten

- $r_1(B) \rightarrow w_2(B)$
- $r_2(B) \rightarrow w_1(B)$
- $r_1(A) \rightarrow w_2(A)$
- $w_2(B) \rightarrow w_1(B)$

c) - Wegen der Abhängigkeiten $r_1(B) \rightarrow w_2(B)$ und $r_1(A) \rightarrow w_2(A)$ gilt: t_1 vor t_2 .

- Wegen der Abhängigkeiten $r_2(B) \rightarrow w_1(B)$ und $w_2(B) \rightarrow w_1(B)$ gilt: t_2 vor t_1 . ⚡

⇒ S_3 ist nicht serialisierbar. Es gibt keine serielle Abfolge der beiden Transaktionen, die ein identisches Resultat für die Variablen A und B erzielt. Dabei überschreibt Transaktion t_1 die Änderung an Variable B die von t_2 getätigt wurde (Lost-Update).

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

3 Übungsaufgabe: Änderbarkeit von Sichten

[7 P.]

[3 P.]

a) Gegeben seien die folgenden Basisrelationen:

Person(PNr, Vorname, Nachname, Geburtsjahr, Nationalitaet)

Schauspieler(PNr → Person.PNr, Schauspielschule)

Regisseur(PNr → Person.PNr, Lieblingsgenre)

Film(FNr, Titel, Jahr, Sprache, Dauer, Filmstudio, Bewertung, Genre, Regisseur → Regisseur.PNr)

SpieltIn(Film → Film.FNr, Schauspieler → Schauspieler.PNr)

Geben Sie die SQL-Anweisungen an, die notwendig sind, um die folgenden Sichten zu erstellen. Geben Sie zu jeder dieser Sichten an, ob sie Änderungsoperationen auf den in ihr enthaltenen Tupeln erlaubt. Begründen Sie Ihre Antwort, falls dies nicht der Fall ist.

- i) *DisneysAbenteuerFilme*: FNr und Titel aller Filme des Genres 'Abenteuer', die vom Filmstudio 'Disney' stammen.

Lösungsvorschlag:

```
CREATE VIEW DisneysAbenteuerFilme
AS SELECT FNr, Titel
FROM Film
WHERE Filmstudio = 'Disney'
AND Genre = 'Abenteuer';
```


Die Sicht ist änderbar.

- ii) *Genres*: Sämtliche Genres (ohne Duplikate), welche seit 1980 erschienen sind.

Lösungsvorschlag:

```
CREATE VIEW Genres
AS SELECT DISTINCT Genre
FROM Film
WHERE Jahr >= 1980;
```

Die Sicht ist nicht änderbar, da sie nicht den Primärschlüssel der Film-Relation enthält.

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

iii) *SchauspielerProFilm*: FNr und die Anzahl der beteiligten Schauspieler pro Film.

Lösungsvorschlag:

```
CREATE VIEW SchauspielerProFilm
AS SELECT Film, count(*) as Anzahl
FROM SpieltIn
Group by Film;
```

Die Sicht ist sowohl aufgrund der Aggregation nicht änderbar.

b) Auf der Film-Relation seien folgende Sichten definiert:

[4 P.]

```
CREATE VIEW AbenteuerFilme
AS SELECT * FROM Film
WHERE Genre > 'Abenteuer';


CREATE VIEW AktuelleAbenteuerFilme
AS SELECT * FROM AbenteuerFilme
WHERE Jahr >= 2018
WITH CASCADED CHECK OPTION;

CREATE VIEW AktuelleAbenteuerKurzFilme
AS SELECT * FROM AktuelleAbenteuerFilme
WHERE Dauer <= 30
WITH CASCADED CHECK OPTION;

CREATE VIEW BeliebteAktuelleAbenteuerFilme
AS SELECT * FROM AktuelleAbenteuerFilme
WHERE Bewertung >= 5;
```

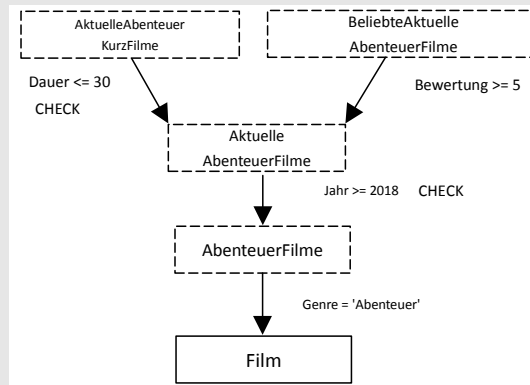
Es handelt sich bei allen obigen Sichtdefinitionen um änderbare Sichten. Bestimmen Sie, ob die folgenden SQL-Anweisungen auf diesen Sichtdefinitionen durchgeführt werden können. Für die Fälle, in denen die Änderung bzw. das Einfügen zulässig ist, geben Sie außerdem an, in welchen Sichten **auf jeden Fall alle** geänderten/eingefügten Tupel nach Abschluss der Operation sichtbar werden.

Hinweis: Im Falle von UPDATE-Operationen ist davon auszugehen, dass die zu ändernden Tupel vor der Änderung die die Sicht definierenden Prädikate erfüllen.

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

Lösungsvorschlag:

In der folgenden Abbildung ist die Vererbungshierarchie des vorliegenden Beispiels abgebildet:



- i) `UPDATE AktuelleAbenteuerFilme`
`SET Bewertung = 4`
`WHERE Titel = 'Jener Film';`

Lösungsvorschlag:

Die Operation ist zulässig. Die geänderten Tupel tauchen in AbenteuerFilme, AktuelleAbenteuerFilme und sofern sie maximal 30 Minuten lang sind auch in AktuelleAbenteuerKurzFilme auf.

- ii) `UPDATE AktuelleAbenteuerKurzFilme`
`SET Jahr = 2017`
`WHERE Titel = 'Dieser andere Film';`

Lösungsvorschlag:


Die Operation ist nicht zulässig, da der einzufügende Tupel die Bedingung `Jahr >= 2018` verletzt.

- iii) `UPDATE BeliebteAktuelleAbenteuerFilme`
`SET Jahr = 2017`
`WHERE Titel = 'Dieser andere Film';`

Lösungsvorschlag:

Auch hier ist die Operation nicht zulässig, da der einzufügende Tupel die Bedingung `Jahr > 2018` verletzt.

- iv) `INSERT INTO AbenteuerFilme VALUES`
`(3453209613, "The Hitchhiker's Guide to the Galaxy", 2005, 'Englisch', 109,`
`'Buena Vista Pictures', 5, 'SciFi comedy', 48564685);`

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

Lösungsvorschlag:

Die Operation ist zulässig. Das eingefügte Tupel ist in keiner der Views sichtbar.

4 Übungsaufgabe: Transaktionen

[15 P.]

Gegeben sind die drei Objekte x , y und z , welche von den Transaktionen T_1 , T_2 und T_3 gelesen bzw. geschrieben werden. Dabei bezeichnet $w_i(x)$ den Schreibzugriff der Transaktion T_i auf das Objekt x , $r_i(x)$ den Lesezugriff der Transaktion T_i auf x und c_i das Commit der Transaktion T_i .


Bestimmen Sie bei den folgenden fünf Schedules S_1 bis S_5 jeweils, ob diese serialisierbar sind oder nicht. Begründen Sie Ihre Antwort, indem Sie alle Abhängigkeiten zwischen den Operationen der einzelnen Transaktionen und dann entweder alle äquivalenten, seriellen Transaktionsreihenfolgen (z.B. $T_1T_2T_3$) oder einen auftretenden Konfliktzyklus angeben. Um die Abhängigkeiten anzugeben benutzen Sie folgende Notation: Sind $w_i(a)$ und $r_j(a)$ voneinander abhängig und wird $w_i(a)$ zeitlich vor $r_j(a)$ ausgeführt, so wird dies durch $w_i(a) \rightarrow r_j(a)$ notiert

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

Lösungsvorschlag:

- $w_2(x) \rightarrow r_1(x)$
- $w_2(x) \rightarrow r_3(x)$
- $w_2(z) \rightarrow r_3(z)$
- $r_3(y) \rightarrow w_1(y)$

Serialisierbar \Rightarrow mögliche serielle Abfolge: $T_2T_3T_1$

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

b) $S_2 = w_2(x) r_1(x) r_1(y) r_3(x) w_1(y) c_1 r_2(z) r_3(y) r_3(z) w_2(z) c_2 c_3$

Lösungsvorschlag:

- $w_2(x) \rightarrow r_1(x)$
- $w_2(x) \rightarrow r_3(x)$
- $r_3(z) \rightarrow w_2(z)$
- $w_1(y) \rightarrow r_3(y)$

Konfliktzyklus:

- Wegen $w_2(x) \rightarrow r_3(x)$ gilt T_2 vor T_3
- Wegen $r_3(z) \rightarrow w_2(z)$ gilt T_3 vor T_2

⇒ Nicht serialisierbar

c) $S_3 = r_3(x) r_1(x) w_2(x) r_1(y) w_1(y) r_3(y) r_2(z) r_3(z) c_3 w_2(z) c_2 c_1$

Lösungsvorschlag:

- $r_1(x) \rightarrow w_2(x)$
- $r_3(x) \rightarrow w_2(x)$
- $r_3(z) \rightarrow w_2(z)$
- $w_1(y) \rightarrow r_3(y)$


Serialisierbar ⇒ mögliche serielle Abfolge: $T_1 T_3 T_2$

d) $S_4 = r_1(x) r_2(y) r_3(z) w_2(y) w_1(z) w_1(x) c_1 r_3(y) w_2(x) c_3 c_2$

Lösungsvorschlag:

- $r_3(z) \rightarrow w_1(z)$
- $w_2(y) \rightarrow r_3(y)$
- $r_1(x) \rightarrow w_2(x)$
- $w_1(x) \rightarrow w_2(x)$

S_4 ist nicht serialisierbar, da ein Zyklus zwischen den Abhängigkeiten

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

$r_3(z) \rightarrow w_1(z)$, $w_1(x) \rightarrow r_2(x)$ und $w_2(y) \rightarrow r_3(y)$ existiert.

e) $S_5 = r_1(x) r_2(y) w_1(z) r_3(z) w_2(y) w_1(x) c_1 r_3(y) w_2(x) c_3 c_2$

Lösungsvorschlag:


- $w_1(z) \rightarrow r_3(z)$
- $w_2(y) \rightarrow r_3(y)$
- $r_1(x) \rightarrow w_2(x)$
- $w_1(x) \rightarrow w_2(x)$

S_5 ist serialisierbar, da kein Zyklus zwischen den Abhängigkeiten besteht. Der einzige äquivalente serielle Schedule lautet $T_1 T_2 T_3$.

5 Übungsaufgabe: 2PL-Synchronisation mit RX-Sperren

[8 P.]

Bei den folgenden Schedules soll das RX-Sperrverfahren mit 2PL zum Einsatz kommen. Vervollständigen Sie die angegebenen Tabellen, indem Sie die Sperranforderungen (**lock**) und -freigaben (**unlock**) der Transaktionen, deren Lese- und Schreibzugriffe (**read bzw. write**) und Commits (**commit**) eintragen. Beachten Sie, dass eine Transaktion innerhalb eines Zeitschritts nur jeweils eine Operation durchführen kann. Nutzen Sie die Spalte **Bemerkungen für etwaige Wartebeziehungen und Benachrichtigungen an wartende Transaktionen**. Das Ändern der Operationsreihenfolge innerhalb einer Transaktion ist nicht erlaubt.

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	30		
	Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

i) $S_4 = r_2(x) w_3(y) r_1(z) r_1(x) r_2(y) w_3(z) w_1(x) c_1 c_2 r_3(x) c_3$

	T ₁	T ₂	T ₃	Bemerkung
0				
1		lock(x,R)		
2		read(x)	lock(y,X)	
3				
4				
5				
6				
7				
8				
9				

Lösungsvorschlag:

4 Punkte

Zeitschritt	T ₁	T ₂	T ₃	Bemerkung
0				
1		lock(x,R)		
2		read(x)	lock(y,X)	
3	lock(z,R)		write(y)	
4	read(z)			
5	lock(x,R)			
6	read(x)	lock(y,R)		T ₂ wartet auf Freigabe von y
7			lock(z,X)	T ₃ wartet auf Freigabe von z
8	lock(x,X)			T ₁ wartet auf Freigabe von x
9				Deadlock



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	30		
Ausgabe	Mi. 11.12.2019	Abgabe	Fr. 10.01.2020

ii) $S_5 = r_1(y) w_2(x) r_3(y) w_3(y) c_2 r_1(z) c_3 c_1$

	T ₁	T ₂	T ₃	Bemerkung
0				
1	lock(y,R)			
2	read(y)	lock(x,X)		
3				
4				
5				
6				
7				
8				
9				
10				
11				

Lösungsvorschlag:

Zeitschritt	T ₁	T ₂	T ₃	Bemerkung
0				
1	lock(y,R)			
2	read(y)	lock(x,X)		
3		write(x)	lock(y,R)	
4		unlock(x)	read(y)	
5		commit	lock(y,X)	T ₃ wartet auf Freigabe von y
6	lock(z,R)			
7	read(z)			
8	unlock(y)			T ₃ wird benachrichtigt
9	unlock(z)		write(y)	
10	commit		unlock(y)	
11			commit	