	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

1 Referentielle Aktionen

[10 P.]

a) Gegeben sei folgende Datendefinition:


```

CREATE TABLE ProfessorIn(
    PNR            INT            PRIMARY KEY,
    Name           VARCHAR(50)    NOT NULL,
    Fachbereich    INT            NOT NULL
);
CREATE TABLE Fachbereich(
    FBNR           INT            PRIMARY KEY,
    Name           VARCHAR(30)    NOT NULL,
    LeiterIn       INT            NOT NULL
    REFERENCES ProfessorIn (PNR) ON DELETE CASCADE
);
ALTER TABLE ProfessorIn ADD FOREIGN KEY (Fachbereich)
    REFERENCES Fachbereich (FBNR) ON DELETE NO ACTION;
CREATE TABLE StudentIn(
    MNR            INT            PRIMARY KEY,
    Name           VARCHAR(50)    NOT NULL,
    Fachbereich    INT            NOT NULL,
    MentorIn       INT            NOT NULL,
    CONSTRAINT fkFachbereich FOREIGN KEY (Fachbereich)
        REFERENCES Fachbereich (FBNR) ON DELETE RESTRICT,
    CONSTRAINT fkMentorIn FOREIGN KEY (MentorIn)
        REFERENCES ProfessorIn (PNR) ON DELETE CASCADE
);
CREATE TABLE Vorlesung(
    VNR            INT            PRIMARY KEY,
    Titel          VARCHAR(50)    NOT NULL,
    DozentIn       INT            REFERENCES ProfessorIn (PNR) ON DELETE SET NULL
);
CREATE TABLE Belegt(
    StudentIn      INT            REFERENCES StudentIn (MNR) ON DELETE RESTRICT,
    Vorlesung      INT            REFERENCES Vorlesung (VNR) ON DELETE CASCADE,
    Semester       VARCHAR(10)    NOT NULL,
    Note           DECIMAL NOT NULL,
    PRIMARY KEY (StudentIn, Vorlesung)
);

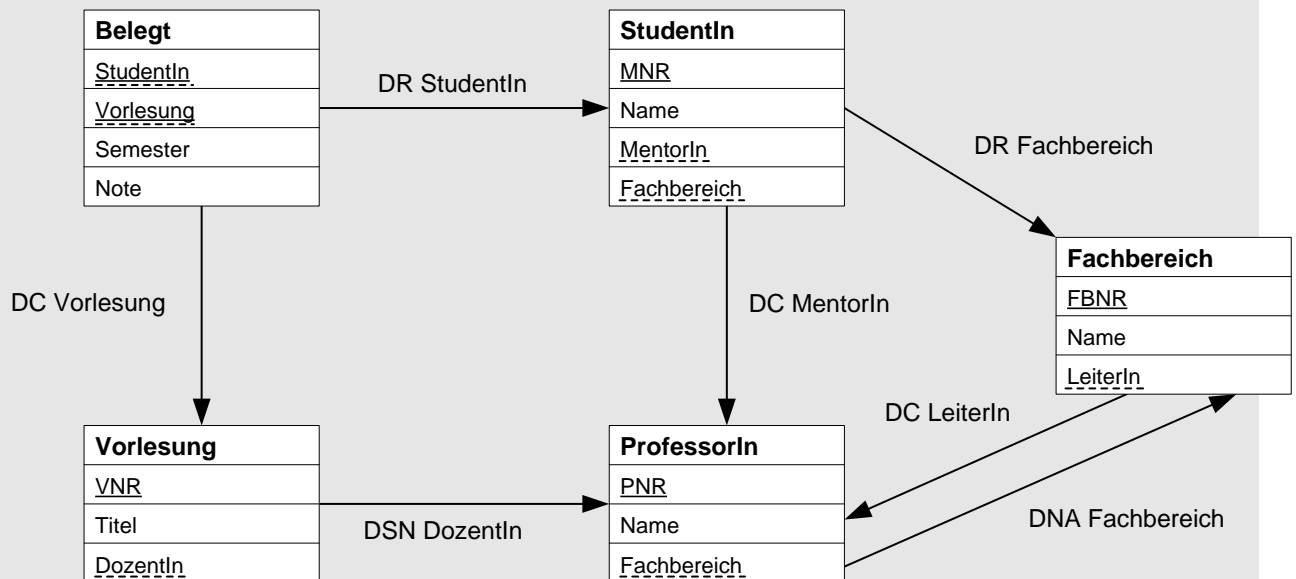
```

Zeichnen Sie den zugehörigen Referenzgraphen und beschriften Sie alle Kanten mit den entsprechenden referentiellen Aktionen.

[5 P.]

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

Lösungsvorschlag:



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

[2 P.]

Lösungsvorschlag:


Das vorliegende Schema ist nicht sicher, da beim Löschen einer ProfessorIn die Löschoperation direkt und über 'Fachbereich' zu 'StudentIn' propagiert wird. Dies führt zu **reihenfolgeabhängigen Ergebnissen**, wenn die zu löschende ProfessorIn MentorIn von allen StudentInnen aus den von ihr geleiteten Fachbereichen ist, diese StudentInnen alle keine Vorlesung belegen (sonst würde deren Löschung durch die referentielle Aktion von 'Belegt' zurückgewiesen werden) und diesen Fachbereichen keine anderen ProfessorInnen angehören (sonst würde deren Löschung durch die referentielle Aktion von 'ProfessorIn' zurückgewiesen werden):

- Wird zunächst der direkte Weg beschritten, werden die StudentInnen gelöscht und einem anschließenden Löschen der geleiteten Fachbereiche steht somit nichts mehr im Wege.
- Wird der Weg über 'Fachbereich' beschritten, wird das Löschen der Fachbereiche jedoch verweigert, da die StudentInnen noch existieren.

- c) Für den Fremdschlüssel 'fkFachbereich' aus der Relation 'StudentIn' soll nun die referentielle Aktion von 'ON DELETE RESTRICT' auf „ON DELETE CASCADE“ geändert werden:

[2 P.]

Geben Sie eine Folge von **SQL-DDL**-Anweisungen an, welche diese Schemaänderung realisiert und dabei

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

die Löschung der Relation selbst vermeidet.

Lösungsvorschlag:

Siehe auch die Folien zur Schemaevolution, z.B. Folie K5/F106:

```
ALTER TABLE StudentIn DROP CONSTRAINT fkFachbereich;
```

```
ALTER TABLE StudentIn ADD CONSTRAINT fkFachbereich
FOREIGN KEY (Fachbereich)
REFERENCES Fachbereich(FBNR) ON DELETE CASCADE;
```

d) Handelt es sich bei dem geänderten Schema um ein sicheres Schema?


[1 P.]

Lösungsvorschlag:

Ja

Nein

Das vorliegende Schema ist jetzt sicher, da das Löschen einer ProfessorIn nun immer das Löschen der StudentInnen welche durch diese ProfessorIn betreut werden und das Löschen der von ihr geleiteten Fachbereiche nach sich zieht (egal in welcher Reihenfolge die referentiellen Aktionen ausgeführt werden).

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

2 Änderbarkeit von Sichten

[7 P.]

a) Gegeben seien die folgenden Basisrelationen:

Schriftsteller(SNr, Vorname, Nachname, Geburtsjahr, Nationalität)

Buch(ISBN, Titel, Erscheinungsjahr, Genre, Verlag, Sprache, Autor → Schriftsteller.SNr)

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.

[3 P.]

i) *GesamtwerkBöll*: ISBN und Titel aller Bücher, die von Heinrich Böll geschrieben wurden.

Lösungsvorschlag:

```
CREATE VIEW GesamtwerkBöll
AS SELECT b.ISBN, b.Titel
FROM Schriftsteller s, Buch b
WHERE s.SNr = b.Autor
AND s.Vorname = 'Heinrich'
AND s.Nachname = 'Böll';
```

Die Sicht ist aufgrund der Verbundoperationen nicht änderbar.

ii) *AlteBritischeAutoren*: SNr, Vorname und Nachname aller Schriftsteller, deren Nationalität 'britisch' ist und die vor 1960 geboren wurden.

Lösungsvorschlag:

```
CREATE VIEW AlteBritischeAutoren
AS SELECT SNr, Vorname, Nachname
FROM Schriftsteller
WHERE Geburtsjahr < 1960
AND Nationalität = 'britisch';
```


Die Sicht ist änderbar.

iii) *BücherHeyne*: Die Titel aller Bücher, die vom Verlag 'Heyne' veröffentlicht wurden.

Lösungsvorschlag:

```
CREATE VIEW BücherHeyne
AS SELECT Titel
FROM Buch
WHERE Verlag = 'Heyne';
```

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

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

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

[4 P.]

```
CREATE VIEW Klassiker
AS SELECT * FROM Buch
WHERE Erscheinungsjahr < 1970;
```

```
CREATE VIEW SciFiKlassiker
AS SELECT * FROM Klassiker
WHERE Genre = 'SciFi'
WITH CASCADED CHECK OPTION;
```

```
CREATE VIEW KlassikerDeutsch
AS SELECT * FROM Klassiker
WHERE Sprache = 'Deutsch'
WITH CASCADED CHECK OPTION;
```

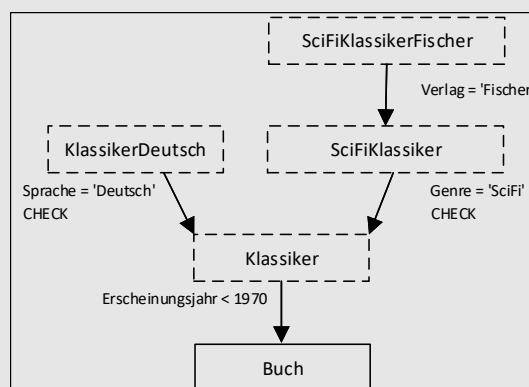
```
CREATE VIEW SciFiKlassikerFischer
AS SELECT * FROM SciFiKlassiker
WHERE Verlag = 'Fischer';
```

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.

Lösungsvorschlag:

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



i) **UPDATE** SciFiKlassikerFischer
SET Genre = 'Horror'
WHERE Genre = 'SciFi'
AND Sprache = 'Deutsch';

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

Lösungsvorschlag:

Die Operation wird zurückgewiesen, da die geänderten Tupel die Bedingung Genre = 'SciFi' der View SciFiKlassiker nicht mehr erfüllen und diese mit einer CHECK-Option versehen ist.

```
ii) UPDATE SciFiKlassiker
     SET Verlag = 'Rowohlt'
     WHERE Verlag = 'Fischer';
```

Lösungsvorschlag:

Die Operation ist zulässig. Geänderte Tupel sind in den Views Klassiker, KlassikerDeutsch (wenn Sprache 'Deutsch' war/ist) und SciFiKlassiker sichtbar.

```
iii) INSERT INTO KlassikerDeutsch
      VALUES ('3453209613', 'Per Anhalter durch die Galaxis', 1981,
              'SciFi', 'Heyne', 'Deutsch', 15);
```


Lösungsvorschlag:

Die Operation wird zurückgewiesen, da das einzufügende Tupel die Bedingung Erscheinungsjahr < 1970 nicht erfüllt, die View KlassikerDeutsch aber mit einer CHECK-Option versehen ist.

```
iv) INSERT INTO Klassiker
     VALUES ('3453209613', 'Per Anhalter durch die Galaxis', 1981,
             'SciFi', 'Heyne', 'Deutsch', 15);
```

Lösungsvorschlag:

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

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

3 Serialisierbarkeit, Anomalien

[15 P.]

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, addiert diese und schreibt den neuen Wert nach B zurück. T_2 erhöht den gelesenen Wert von B um 100 und schreibt den neuen Wert nach B zurück. Außerdem schreibt T_2 den alten Wert von B nach A (ohne A vorher gelesen zu haben). Der Anfangswert von A sei 20 und der von B sei 10. 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:

$$\begin{aligned}
 S_1 &= r_1(B) r_1(A) w_1(B) r_2(B) w_2(B) w_2(A) \\
 S_2 &= r_2(B) w_2(B) w_2(A) r_1(B) r_1(A) w_1(B) \\
 S_3 &= r_1(B) r_1(A) r_2(B) w_2(B) w_1(B) w_2(A) \\
 S_4 &= r_2(B) w_2(B) r_1(B) w_2(A) r_1(A) w_1(B) \\
 S_5 &= r_1(B) r_2(B) w_2(B) w_2(A) r_1(A) w_1(B) \\
 S_6 &= r_1(B) r_1(A) r_2(B) w_1(B) w_2(B) w_2(A)
 \end{aligned}$$

Beantworten Sie für jeden der Schedules die folgenden drei Fragestellungen:

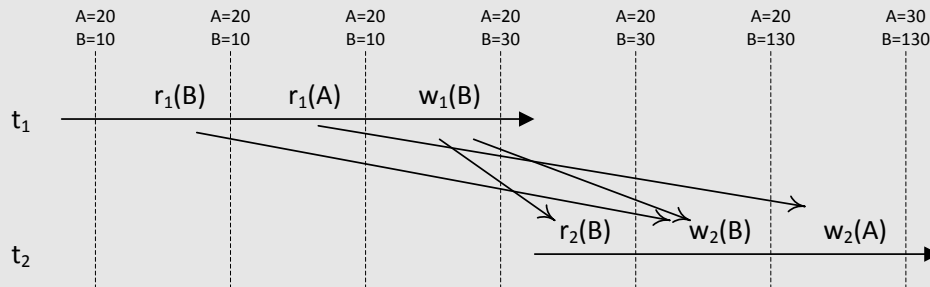
- Wie lautet nach Ausführung des Schedules die Belegung für die Variablen A und B? [3 P.]
- Welche Abhängigkeiten existieren zwischen den Operationen der beiden Transaktionen innerhalb des Schedules? [6 P.]
- Ist der Schedule seriell, serialisierbar oder nicht serialisierbar? Erläutern sie zusätzlich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



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	40		
Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

Lösungsvorschlag:

- Schedule S_1 :



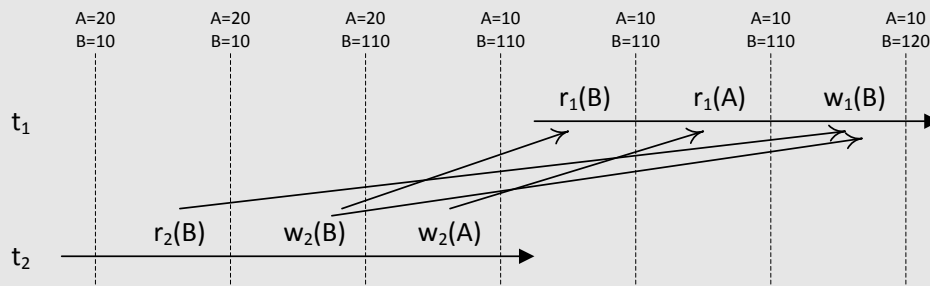
a) Endwert für A ist 30, Endwert für B ist 130.

b) Abhängigkeiten

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

c) Der Schedule ist seriell.

- Schedule S_2 :



a) Endwert für A ist 10, Endwert für B ist 120.

b) Abhängigkeiten

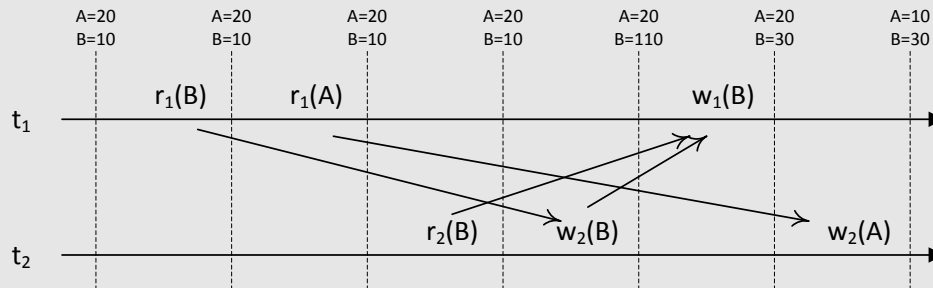
- $w_2(B) \rightarrow r_1(B)$
- $w_2(A) \rightarrow r_1(A)$
- $r_2(B) \rightarrow w_1(B)$
- $w_2(B) \rightarrow w_1(B)$



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	40		
Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

c) Der Schedule ist seriell.

• Schedule S_3 :



a) Endwert für A ist 10, Endwert für B ist 30.

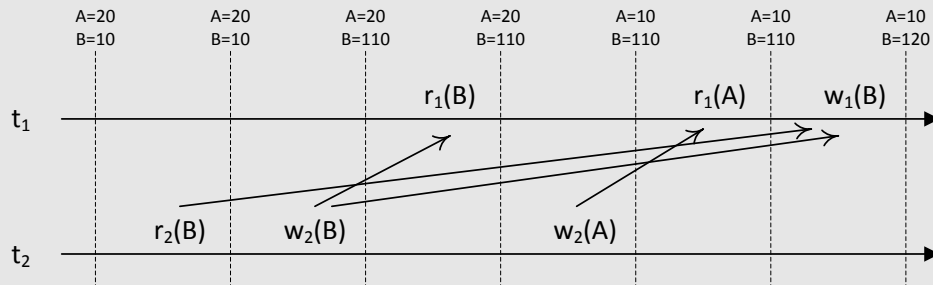
b) Abhängigkeiten

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

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

$\Rightarrow S_3$ 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).


• Schedule S_4 :



a) Endwert für A ist 10, Endwert für B ist 120.

b) Abhängigkeiten

- $w_2(B) \rightarrow r_1(B)$

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

– $w_2(A) \rightarrow r_1(A)$

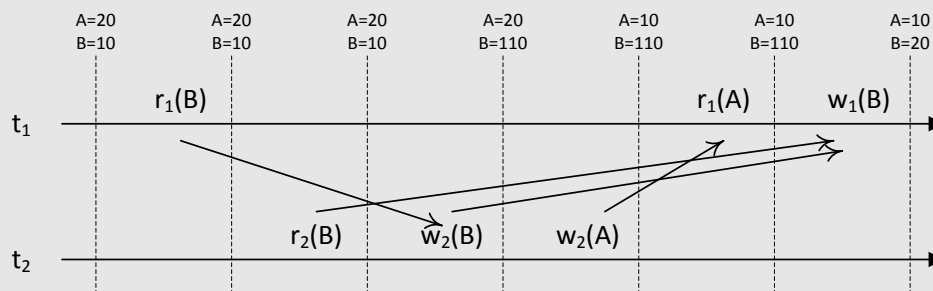
– $r_2(B) \rightarrow w_1(B)$

– $w_2(B) \rightarrow w_1(B)$

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

$\Rightarrow S_4$ ist serialisierbar (Schedule S_2 erzielt das identische Ergebnis für die Variablen A und B)

• Schedule S_5 :



a) Endwert für A ist 10, Endwert für B ist 20.

b) Abhängigkeiten

– $r_1(B) \rightarrow w_2(B)$

– $r_2(B) \rightarrow w_1(B)$

– $w_2(B) \rightarrow w_1(B)$

– $w_2(A) \rightarrow r_1(A)$

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

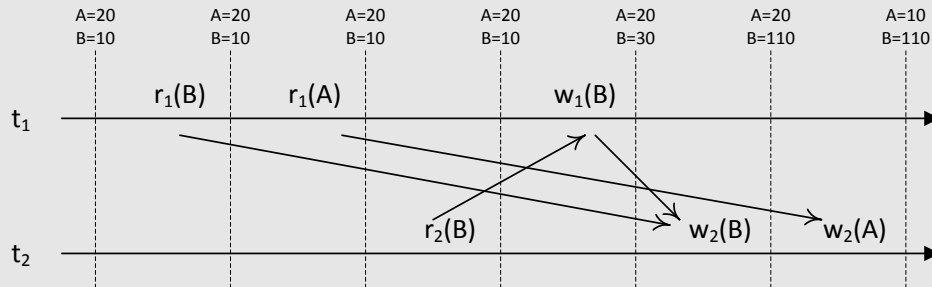
– Wegen der Abhängigkeiten $r_1(B) \rightarrow w_2(B)$ gilt: t_1 vor t_2 . ⚡

$\Rightarrow S_5$ 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).


• Schedule S_6 :



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	40		
Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018



- a) Endwert für A ist 10, Endwert für B ist 110.
- b) Abhängigkeiten
- $r_1(B) \rightarrow w_2(B)$
 - $r_1(A) \rightarrow w_2(A)$
 - $r_2(B) \rightarrow w_1(B)$
 - $w_1(B) \rightarrow w_2(B)$
- c) - Wegen der Abhängigkeiten $r_2(B) \rightarrow w_1(B)$ gilt: t_2 vor t_1 .
- Wegen der Abhängigkeiten $r_1(B) \rightarrow w_2(B)$, $r_1(A) \rightarrow w_2(A)$ und $w_1(B) \rightarrow w_2(B)$ gilt: t_1 vor t_2 . ↓
- ⇒ S_6 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_2 die Änderung an Variable B die von t_1 getätigt wurde (Lost-Update).

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
	Aufgabenzettel	5 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

4 2PL-Synchronisation mit R/X-Sperren

[8 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 und $r_i(x)$ den Lesezugriff der Transaktion T_i auf x .

Der Schedule S_1 zeigt an, in welcher Reihenfolge die Operationen der drei Transaktionen T_1 , T_2 , T_3 beim Scheduler eines Datenbanksystems eintreffen. Die Operation c_i soll das Commit der Transaktion T_i darstellen.

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

Bei der Ausführung von S_1 soll das RX-Sperrverfahren mit 2PL zum Einsatz kommen. Vervollständigen Sie die auf der nächsten Seite angegebene Tabelle, 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.



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2017/18
Aufgabenzettel	5 (Lösungsvorschläge)		
Gesamtpunktzahl	40		
Ausgabe	Mi. 13.12.2017	Abgabe	Fr. 12.01.2018

Lösungsvorschlag:

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

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