	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>5</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Fr. 14.12.2018</b>	Abgabe	<b>Fr. 11.01.2019</b>

## 1 Referentielle Aktionen


[10 P.]

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,
    Haupterausgeber int NOT NULL,
    CONSTRAINT fk_zeitschrift_haupterausgeber FOREIGN KEY (Haupterausgeber)
        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	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>5</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Fr. 14.12.2018</b>	Abgabe	<b>Fr. 11.01.2019</b>

- a) Zeichnen Sie den zugehörigen Referenzgraphen und beschriften Sie alle Kanten mit den entsprechenden referentiellen Aktionen. [4 P.]
- 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. [2 P.]
- 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: [2 P.]  
Geben Sie eine Folge von **SQL-DDL**-Anweisungen an, welche diese Schemaänderung realisiert und dabei die Löschung der Relation selbst vermeidet.
- d) Handelt es sich 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. [2 P.]

## 2 Änderbarkeit von Sichten

[7 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, Erscheinungsjahr, Sprache, LaengelnMin, Filmstudio, 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. [3 P.]

- i) *EnglischeDisneyFilme*: Die Titel aller Filme, die vom Filmstudio 'Disney' stammen und auf englisch sind.
- ii) *AlteKurzfilme*: FNr, Titel und Sprache von Filmen, die vor 1978 erschienen und nicht länger als 30 min sind.
- iii) *LebenswerkBradPitt*: Titel und Erscheinungsjahr aller Filme, in denen der Schauspieler Brad Pitt mitspielt.

- b) Auf der Film-Relation seien folgende Sichten definiert: [4 P.]

```
CREATE VIEW ModerneFilme
AS SELECT * FROM Film
WHERE Erscheinungsjahr > 2007;
```

```
CREATE VIEW ModerneAbenteuerFilme
```

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>5</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Fr. 14.12.2018</b>	Abgabe	<b>Fr. 11.01.2019</b>

```
AS SELECT * FROM ModerneFilme
WHERE Genre = 'Abenteuer';
```

```
CREATE VIEW ModerneKurzfilme
AS SELECT * FROM ModerneFilme
WHERE LaengeInMin < 30
WITH CASCADED CHECK OPTION;
```

```
CREATE VIEW ModerneLucasFilmAbenteuerFilme
AS SELECT * FROM ModerneAbenteuerFilme
WHERE Filmstudio = 'LucasFilm'
WITH CASCADED CHECK OPTION;
```

```
CREATE VIEW ModerneFranzoesischeKurzfilme
AS SELECT * FROM ModerneKurzfilme
WHERE Sprache = 'franzoesisch';
```

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.


- i) UPDATE ModerneAbenteuerFilme
 

```
SET Filmstudio = 'Disney'
WHERE Filmstudio = 'LucasFilm';
```
- ii) UPDATE ModerneLucasFilmAbenteuerFilme
 

```
SET Genre = 'Horror'
WHERE Genre = 'Abenteuer'
AND Studio = 'LucasFilm';
```
- iii) INSERT INTO ModerneFilme VALUES
 

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

```
(116155618, 'Life of Pi', 2012, 'englisch', 127,
'20th Century Fox', 'Abenteuer', 151854);
```

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>5</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Fr. 14.12.2018</b>	Abgabe	<b>Fr. 11.01.2019</b>

### 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, 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:

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

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.]


### 4 2PL-Synchronisation mit R/X-Sperren

[8 P.]

Gegeben sind die vier Objekte v, 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  $S_7$  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.

$$S_7 = r_1(x) r_2(z) r_1(y) w_1(z) r_3(v) w_1(y) c_1 r_3(z) r_2(v) c_2 w_3(z) c_3$$

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2018/19
	Aufgabenzettel	<b>5</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Fr. 14.12.2018</b>	Abgabe	<b>Fr. 11.01.2019</b>

Bei der Ausführung von  $S_7$  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	<b>Grundlagen von Datenbanken</b>		WS 2018/19
Aufgabenzettel	<b>5</b>		
Gesamtpunktzahl	<b>40</b>		
Ausgabe	<b>Fr. 14.12.2018</b>	Abgabe	<b>Fr. 11.01.2019</b>

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	Bemerkung
0				
1	lock(x,R)			
2	read(x)	lock(z,R)		
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				