Gruppe **A**

Bitte tragen Sie **SOFORT** und **LESERLICH** Namen und Matrikelnr. ein, und legen Sie Ihren Ausweis bereit.

PRÜFUNG AUS	**MUSTERLÖSUNG**	20.01.2023 **GRUPPE A**
Matrikelnr.	Familienname	Vorname

Arbeitszeit: 90 Minuten. Lösen Sie die Aufgaben auf den vorgesehenen Blättern; Lösungen auf Zusatzblättern werden nicht gewertet. **Viel Erfolg!**

Aufgabe	1	2	3	4	5	6	Σ
Max. Punkte	12	13	10	11	12	12	70
Punkte							

Bitte die Heftklammer nicht entfernen!
Matrikelnr/Namen auf jedes Blatt eintragen, es erleichtert uns die Punkte einzutragen.

Achtung!

Für sämtliche Fragen mit Ankreuzmöglichkeiten gilt: Ankreuzen alleine gibt keine Punkte!

Punkte gibt es nur in Zusammenhang mit geforderter Erklärung/Beispiel/...!

Notation:

In den Aufgaben 1 – 3 wird die folgende (aus der Vorlesung bekannte) Notation für Transaktionen T_i verwendet:

- $r_i(O)$ und $w_i(O)$: Lese- bzw. Schreibzugriff von T_i auf Objekt O.
- b_i, c_i, a_i: Beginn (`BEGIN OF TRANSACTION`), Commit (`COMMIT`) bzw. Abbruch (`ABORT`/`ROLLBACK`) von T_i.

Die Indizes $_i$ können weggelassen werden, wenn klar ist zu welcher Transaktion eine Operation gehört.

Des weiteren wird das aus der Vorlesung bekannte Format für Logeinträge verwendet:

`[LSN, TA, PageID, Redo, Undo, PrevLSN]` für "normale" Einträge, bzw.
`[LSN, TA, BOT, PrevLSN]` für `BOT` Log-Einträge und
`[LSN, TA, COMMIT, PrevLSN]` für `COMMIT` Einträge.

Kompensations Logeinträge (Compensation Log Records) haben das Format
⟨`LSN, TA, PageID, Redo, PrevLSN, UndoNextLSN`⟩ bzw.
⟨`LSN, TA, BOT, PrevLSN`⟩

Dabei stellt `LSN` die Log-Sequence Nummer dar, `TA` die Transaktion, `PageID` ist die veränderte Seite, `Redo` und `Undo` die für das Redo bzw. Undo benötigten Informationen, `UndoNextLSN` ist die LSN des nächsten Logeintrags der selben Transaktion welcher zurückgesetzt werden soll, und `PrevLSN` die LSN des vorherigen Logeintrags derselben Transaktion.

Im Falle logischer Protokollierung sollen die Änderungen zum aktuellen Datenbestand nur mittels *Addition* bzw. *Subtraktion* angegeben werden, z.B. [·, ·, ·, X+=d_1, X-=d_2, ·].

MatrikelNr: _____ **Nachname:** _____

Aufgabe 1: Eigenschaften von Transaktionen (12)

Betrachten Sie die unten angegebene Historie, welche durch eine Abfolge von Elementaroperationen der vier Transaktionen T_1, T_2, T_3 und T_4 auf den Datensätzen A, B, C und D gegeben ist.

T_1	T_2	T_3	T_4
b_1	b_2	b_3	b_4
	$r_2(A)$		
	$r_2(B)$		
$r_1(A)$			
	c_2		
$w_1(A)$			
		$r_3(A)$	
			$r_4(A)$
			$w_4(B)$
			$w_4(C)$
$r_1(B)$			
$r_1(C)$			
			$r_4(C)$
			$r_4(D)$
$r_1(B)$			
			$w_4(D)$
			$w_4(E)$
		$r_3(E)$	
		$r_3(D)$	
		c_3	
			c_4
c_1			

a) Geben Sie an, zwischen welchen Transaktionen eine Leseabhängigkeit besteht. D.h., geben Sie für jede Transaktion an, von welchen anderen Transaktionen diese Transaktion liest (falls eine Transaktion von keiner anderen Transaktion liest, streichen Sie das entsprechende Feld bitte durch). (4 Punkte)

a) Leseabhängigkeiten:

T_1 liest von T_4 ..... T_2 liest von -- .....

T_3 liest von T_1, T_4 ... T_4 liest von T_1 .....

b) Bestimmen Sie anschließend, ob die Historie kaskadierendes Rücksetzen vermeidet oder nicht, sowie ob sie strikt ist oder nicht. Geben Sie jeweils eine kurze Begründung an Hand der Historie, an.
(*Achtung:* Ankreuzen alleine ohne eine Begründung gibt keine Punkte!) (4 Punkte)

b) Eigenschaften:

Historie vermeidet kaskadierendes Rücksetzen: ◯ ja ⊗ nein

Begründung: Nicht alle Transaktionen von denen gelesen .

wird haben ihr COMMIT vor der Leseoperation. Z.B. b_1 und b_3

Historie ist strikt: ◯ ja ⊗ nein

Begründung: Eine strikte Historie muss kaskadierendes ..

Zurücksetzen vermeiden .....

c) Betrachten Sie jeweils die unten angegebenen Paare von Transaktionen. Bestimmen Sie für jedes der beiden Paare, ob die beiden Transaktionen konfliktserialisierbar sind oder nicht.
Falls nicht, geben Sie eine Sequenz von Paaren von Konfliktoperationen der beiden Transaktionen an, welche die Existenz einer konfliktäquivalenten, seriellen Historie ausschließen.
Falls die Transaktionen konfliktserialisierbar sind, geben Sie eine konfliktäquivalente, serielle Historie (= Sequenz von Elementaroperationen!) der beiden Transaktionen an. (4 Punkte)

MatrikelNr: _____ **Nachname:** _____

Transaktionen T_1 und T_3: Konfliktserialisierbar: ⊗ ja ○ nein

Antwort: b_1, $r_1(A)$, $w_1(A)$, $r_1(B)$, $r_1(C)$, $r_1(B)$, c_1, b_3, $r_3(A)$, $r_3(E)$, $r_3(D)$, c_3 .....

.....

Transaktionen T_1 und T_4: Konfliktserialisierbar: ○ ja ⊗ nein

Antwort: z.B. $(w_1(A), r_4(A))$, $(w_4(B), r_1(B))$, ...

.....

Punkteverteilung Aufgabe 1

(1a)

T_1 korrekt	+1
T_2 korrekt	+1
T_3 korrekt	+1
T_4 korrekt	+1
$\sum$	4

(1b)

vermeidet kaskadierendes Rücksetzen: nein	+2
Historie ist strikt: nein	+2
keine/inkorrekte Begründung bei Frage 1	+-2
keine/inkorrekte Begründung bei Frage 2	+-2
$\sum$	4

(1c)

Frage 1 korrekt	+2
Frage 2 korrekt	+2
keine/inkorrekte Begründung bei Frage 1	+-2
keine/inkorrekte Begründung bei Frage 2	+-2
$\sum$	4

MatrikelNr: _____ **Nachname:** _____

Aufgabe 2: Protokollierung und Wiederanlauf (Logging und Recovery) (13)

Betrachten Sie die angegebene Historie der drei Transaktionen T_1, T_2 und T_3 in der Tabelle wobei die Datenbank nachfolgend gegebene Werte hat. Nehmen Sie dabei an, dass Feld A sich auf Seite P_A, Feld B auf Seite P_B, ... befindet.

$$A : 17,\ B : 4,\ C : 9,\ D : 8$$

a) Tragen Sie für jede Zeile der Historie, in welcher entweder der Wert eines Feldes, oder einer lokalen Variable geändert wird, den Wert des entsprechenden Feldes/Variablen nach der Operation an. (3 Punkte)

b) Geben Sie Log-Einträge der Historie an. Protokollieren Sie logisch. (4 Punkte)

	T_1	T_2	T_3	(a) Var./Feld	(b) Log-Einträge
1	$\texttt{BOT}_1$			.....	[#1, T_1, BOT,#0] .....
2		$\texttt{BOT}_2$		.....	[#2, T_2, BOT,#0] .....
3			$\texttt{BOT}_3$	.....	[#3, T_3, BOT,#0] .....
4		$r_2(A, a_2)$		$a_2 = 17$	.....
5		$w_2(B, a_2 + 3)$		$B = a_2 + 3 = 20$	[#4, T_2, P_B, $B+=16, B-=16$,#2]
6			$r_3(B, b_3)$	$b_3 = 20$	.....
7		$r_2(C, c_2)$		$c_2 = 9$	.....
8	$w_1(D, 2)$			$D = 2$	[#5, T_1, P_D, $D-=6, D+=6$,#1]
9		$w_2(D, c_2 + 1)$		$D = c_2 + 1 = 10$	[#6, T_2, P_D, $D+=8, D-=8$,#4]
10			$r_3(A, a_3)$	$a_3 = 17$	.....
11			$w_3(A, a_3 + 5)$	$A = a_3 + 5 = 22$	[#7, T_3, P_A, $A+=5, A-=5$,#3]
12		$r_2(B, b_2)$		$b_2 = 20$	.....
13		$w_2(D, b_2)$		$D = b_2 = 20$	[#8, T_2, P_D, $D-=10, D+=10$,#6]
14	$w_1(D, 4)$			$D = 4$	[#9, T_1, P_D, $D-=16, D+=16$,#5]
15			$w_3(A, b_3 + 5)$	$A = b_3 + 5 = 25$	[#10, T_3, P_A, $A+=3, A-=3$,#7]
16		$\texttt{commit}_2$		.....	[#11, T_2, COMMIT,#8] .....
17	$\texttt{abort}_1$			.....	⟨#12, T_1, P_D,$D+=16$,#9,#5⟩ ..
				.....	⟨#13, T_1, P_D,$D+=6$,#12,#1⟩ ..
				.....	⟨#14, T_1, BOT, #13⟩ .....

MatrikelNr: | **Nachname:**

Punkteverteilung Aufgabe 2

(2a)	Reads (a,b,c) richtig je 0.5	+1.5
	Writes (a,b,d) richtig je 0.5	+1.5
		$\sum$ 3
	Var/Felder nicht unterscheidbar	-1
(2b)	LR korrekt	+2
	CLR korrekt	+2
		$\sum$ 4
	Log Record Format fehlerhaft	-0.5
	LastLSN fehlerhaft	-1
	UndoNextLSN fehlerhaft	-1
	Fehlerhafte Wert redo/undo	-0.5

MatrikelNr: _____ **Nachname:** _____

c) *Betrachten Sie die unten angegebenen Logeinträge der zwei Transaktionen T_1 und T_2 und Datenbestand:*

P_A	*LSN: #8*
$A = 40$	

P_B	*LSN: #6*
$B = 23$	

Führen Sie anhand dieser Log-Einträge ein Recovery (nach dem ARIES Verfahren) durch. Beantworten Sie dazu die unten gestellten Teilfragen.

Logeinträge (Angabe)

$[\#1, T_1, \texttt{BOT}, \#0]$
$[\#2, T_1, P_A, A+=20, A-=20, \#1]$
$[\#3, T_2, \texttt{BOT}, \#0]$
$[\#4, T_1, P_A, A+=4, A-=4, \#2]$
$[\#5, T_1, P_B, B+=4, B-=4, \#4]$
$[\#6, T_1, P_B, B+=8, B-=8, \#5]$
$[\#7, T_2, P_A, A+=10, A-=10, \#6]$
$[\#8, T_2, P_A, A+=10, A-=10, \#7]$
$[\#9, T_2, P_B, B+=2, B-=2, \#8]$
$[\#10, T_2, \texttt{COMMIT}, \#9]$
$\langle \#11, T_1, P_B, B-=8, \#6, \#5 \rangle$
$\langle \#12, T_1, P_B, B-=4, \#11, \#4 \rangle$

c1) Bestimmen Sie die Werte im Datenbestand und PageLSN nach der Redo-Phase (2 Punkte)

P_A	LSN: `#8`
$A =$ `40`	

P_B	LSN: `#12`
$B =$ `13`	

c2) Ergänzen Sie die CLR Records. (3 Punkte)
$\langle$`LSN, TA, PageID, Redo, PrevLSN, UndoNextLSN`$\rangle$
$\langle$`LSN, TA, BOT, PrevLSN`$\rangle$

$\langle$`#13, `T_1`, `P_A`, A-=4, #12, #2 `$\rangle$ .....

$\langle$`#14, `T_1`, `P_A`, A-=20, #13, #1`$\rangle$ .....

$\langle$`#15, `T_1`, BOT,#14`$\rangle$ .....

.....

.....

c3) Bestimmen Sie die Werte im Datenbestand und PageLSN nach dem Wiederanlauf. (1 Punkt)

P_A	LSN: `#14`
$A =$ `16`	

P_B	LSN: `#12`
$B =$ `13`	

Punkteverteilung

(c1) $\sum$ +2 davon LSN, Wert je +0.5:

(c3) $\sum$ +1 davon Wert je +0.5 (keine Punkte für LSN, da sie direkt oben abgelesen werden können)

(c2) $\sum$ +3

- 1 Punkt pro CLR
- -1 Punkt pro zusätzlichem CLR
- -1 Falsches format
- 0 Punkte falls undo für committed Transaktion

Total (c) $\sum$ +6

MatrikelNr: _____ **Nachname:** _____

Aufgabe 3: Sperrprotokolle (Locking) (10)

(3a) 2-Phasen Sperrprotokoll (2PL)

Gegeben ist die untenstehende Folge von Exclusive- und Share-Sperren ($X_i(O)$, $S_i(O)$), Freigaben von Sperren ($relX_i(O)$, $relS_i(O)$), sowie Lese- und Schreiboperationen. Bei Einträgen in der selben Zeile spielt die zeitliche Anordnung der betroffenen Operationen keine Rolle, es kann eine beliebe Reihenfolge angenommen werden.

Geben Sie für jede der fünf Transaktionen T_1, T_2, T_3, T_4 und T_5 an, ob sie das *2-Phasen Sperrprotokoll (2PL)* befolgt. Falls nicht, beschreiben Sie wodurch das 2PL verletzt wird. (5 Punkte)

T_1	T_2	T_3	T_4	T_5
BOT	BOT	BOT	BOT	BOT
$X_1(A)$	$S_2(C)$	$X_3(E)$		
	$S_2(D)$		$X_4(G)$	
$r_1(A)$	$X_2(F)$		$w_4(G)$	$S_5(B)$
$w_1(A)$			$relX_4(G)$	$X_5(A)$
	$r_2(D)$		$S_4(B)$	$r_5(B)$
	$r_2(C)$	$w_3(E)$		
$S_1(B)$				
	$relS_2(D)$			$w_5(A)$
$w_1(B)$	$relS_2(C)$	$X_3(D)$		
	$w_2(F)$	$r_3(D)$		$X_5(C)$
$relS_1(B)$	$relX_2(F)$	$w_3(D)$		$relX_5(A)$
$relX_1(A)$	c_2			$w_5(C)$
		$X_3(A)$		
c_1		$w_3(A)$	$r_4(B)$	
		$relX_3(A)$	$relS_4(B)$	$relS_5(B)$
		$relX_3(D)$		
		$relX_3(E)$		$relX_5(C)$
		c_3	c_4	c_5

T_1: schreibt auf B, aber nur SL. .....

T_2: korrektes 2PL. .....

T_3: korrektes 2PL. .....

T_4 sperrt B (SL(B)), nachdem es die ...
Sperre XL(D) bereits freigegeben hat. .

T_5 schreibt auf A obwohl T_1 .....
noch eine Exclusive Lock (XL(A)) hält.

.....

.....

.....

.....

.....

.....

Hinweis: Betrachten Sie nicht nur jede Transaktion für sich.

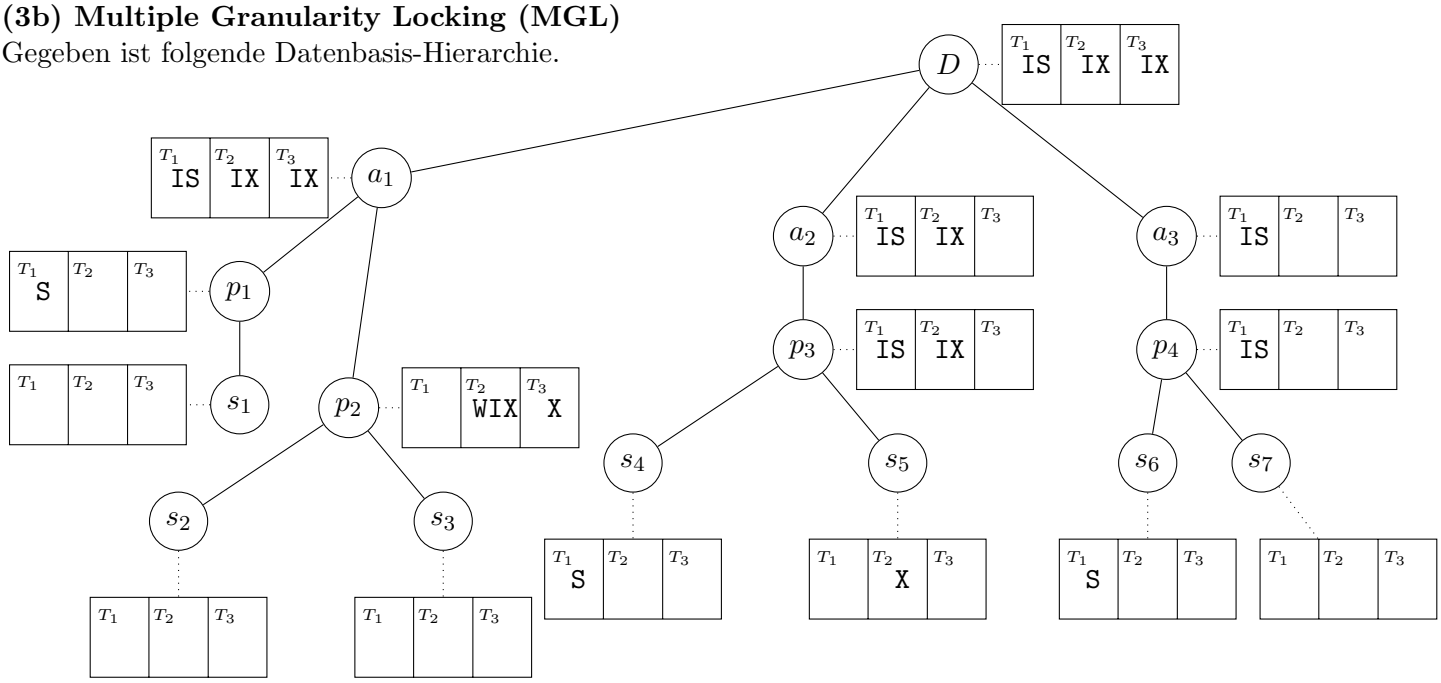
Pro korrekter Antwort je Transaktion. +1

$\sum$ 5

MatrikelNr: _____ **Nachname:** _____

(3b) Multiple Granularity Locking (MGL)

Gegeben ist folgende Datenbasis-Hierarchie.



Nehmen Sie an, die Transaktionen T_1, T_2 und T_3 möchten folgenden Sperren in der angegeben Reihenfolge erhalten

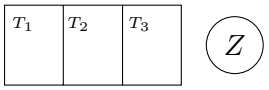
$$\mathrm{X}_3(p_2), \mathrm{X}_2(s_5), \mathrm{S}_1(p_1), \mathrm{S}_1(s_4), \mathrm{X}_2(s_2), \mathrm{X}_2(s_7), \mathrm{S}_1(s_6)$$

Nehmen Sie an, dass die Transaktionen zum Erhalt dieser Sperren nach dem Sperrprotokoll des MGL vorgehen (und dieses auch korrekt einhalten; beachten Sie, dass obige Angabe nur die Sperren beinhaltet, welche die Transaktionen haben wollen, und dass zum Erhalt dieser zusätzliche Sperren nötig sind). Dabei bezeichnet $S_i(o)$ den Wunsch von Transaktion T_i eine Lesesperre auf das Objekt o zu erhalten, und $X_i(o)$ den Wunsch eine Schreibsperre zu erhalten.

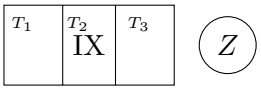
Beschreiben Sie den Zustand am Ende dieser Sperranforderungen, indem Sie in der Grafik oben zu jedem Knoten notieren welche Transaktionen welche Sperren auf diesem Knoten halten. Tragen Sie dazu S, X, IS und IX in das Feld (passend zum Knoten und Transaktionsnummer) ein um auszudrücken, dass die Transaktion die entsprechende Sperre hält.
Sollte eine Transaktion eine Sperre angefordert aber nicht erhalten haben, tragen Sie bitte WS, WX, WIS oder WIX ein. Sollte eine Transaktion blockieren, ignorieren Sie alle weiteren Anforderungen dieser Transaktion.

(5 Punkte)

Beispiel:



Die Transaktion T_2 will am Knoten Z ein Intention Exclusive Lock (IX) der Transaktion T_2 erhalten und bekommt diesen. Für den Zustand tragen Sie dann IX in das zugehörige Feld am Knoten ein.



T_1 korrekt	+1
T_2 korrekt	+3
T_3 korrekt	+1
$\sum$	5
T_2 wartet nicht	-3
pro falschem Eintrag	-0.5

Seite bleibt zum besseren Bearbeiten der Folgeaufgabe leer. Bitte hier keine Lösungen notieren.

MatrikelNr: _____ **Nachname:** _____

Für die Aufgaben 4 – 6 gilt die Datenbankbeschreibung auf diesem Blatt.

Aufgabe 4: Erstellen eines Datenbankschemas mittels SQL und Aufgaben zu Schlüsselabhängigkeiten (11)

a) Folgendes Schema sei gegeben:

`readers`	(name, sid, country, mostBooks: *deliveries.id*)
`deliveries`	(id, date, toName: *readers.name*, toSid: *readers.sid*)
`books`	(id: *deliveries.id*, itemNumber, pages, cost, fromAuthor: *authors.name*)
`authors`	(name, printedPages, country)

Ein Universitätsassistent muss für eine Aufgabe in einer Datenbanksysteme Prüfung ein relationales Schema finden, und entscheidet sich mangels neuer Ideen für ein generisches Schema aus Leser:innen, Buchlieferungen, Büchern und Autor:innen.

Eine Leser:in (`readers`) ist eindeutig gekennzeichnet durch Name und Sozialversicherungs-ID (`sid`). Zusätzlich wird das Land (`country`) gespeichert aus dem die Leser:in stammt.

Jede der Buchlieferungen (`deliveries`) ist eindeutig durch eine ID gekennzeichnet. Die ID muss eine Sequence sein, beginnend mit 10, und in 5er Schritten fortlaufen. Daneben wird auch das Datum (`date`) festgehalten, an dem die Bücher ausgeliefert wurden. Es wird ebenfalls gespeichert welche Leser:in beliefert wurde (`toName, toSid`). Daneben wird bei jeder Leser:in auch die Lieferung mit den meisten Büchern festgehalten (`mostBooks`) (Für die Lösung dieser Aufgabe muss nicht geprüft werden ob `mostBooks` tatsächlich auf die Lieferung mit den meisten Büchern verweist).

Jede Lieferung enthält Bücher (`books`). Jedes Buch ist eindeutig gekennzeichnet durch die Kombination der ID der zugehörigen Lieferung und einer Bestellpositionsnummer (`itemNumber`). Jedes Buch hat eine Seitenzahl (`pages`), Kosten (`cost`) und eine Autor:in (`fromAuthor`). Die Seitenanzahl muss mindestens 1 sein. `cost` soll als eine Dezimalzahl mit maximal 2 Kommastellen implementiert werden.

Jede Autor:in (`authors`) hat einen eindeutigen Namen. Daneben wird die gesamte Anzahl der gedruckten Seiten in allen ausgelieferten Büchern der Autor:in (`printedPages`) und das Land (`country`) der Autor:in in der Datenbank festgehalten. Die Anzahl der gedruckten Seiten muss bei jedem Eintrag in der Tabelle zwischen 0 und 9999 liegen. (Für die Lösung dieser Aufgabe muss nicht geprüft werden ob `printedPages` tatsächlich die gesamte Seitenanzahl aller assoziierten Lieferungen festhält.)

Geben Sie die nötigen SQL Statements an, um obiges Schema (mit allen Konsistenzbedingungen) anzulegen. Wählen Sie passende Typen für Attribute. **Die Abkürzung VC statt VARCHAR(100) ist erlaubt.**

(7 Punkte)

```
CREATE SEQUENCE seq_id INCREMENT BY 5 MINVALUE 10 NO CYCLE;

CREATE TABLE readers(
   name         VARCHAR(100),
   sid          INTEGER,
   country              VARCHAR(100) NOT NULL,
   mostBooks    INTEGER NOT NULL,
   PRIMARY KEY (name,sid)
);

CREATE TABLE deliveries(
   id           INTEGER DEFAULT nextval('seq_id') PRIMARY KEY,
   date         DATE NOT NULL,
   toName       VARCHAR(100),
   toSid        INTEGER,
   FOREIGN KEY (toName,toSid) REFERENCES readers(name,sid)
);

CREATE TABLE authors(
   name         VARCHAR(100) PRIMARY KEY,
   printedPages INTEGER NOT NULL CHECK(printedPages BETWEEN 0 AND 9999),
   country      VARCHAR(100) NOT NULL
);

CREATE TABLE books(
   id   INTEGER REFERENCES deliveries(id),
   itemNumber   INTEGER,
   pages    INTEGER NOT NULL CHECK(pages > 0),
   cost         NUMERIC(5,2) NOT NULL,
   fromAuthor   VARCHAR(100) REFERENCES authors(name),
   PRIMARY KEY (id,itemNumber)
);

ALTER TABLE readers ADD CONSTRAINT c_mostBooks
   FOREIGN KEY (mostBooks) REFERENCES deliveries(id)
   DEFERRABLE INITIALLY DEFERRED;
```

Punkteverteilung Aufgabe 4

(4a)	Create Table statement fehlerfrei je 1	+1
	Create Sequence statement fehlerfrei	+1
	Falsche Reihenfolge	-1
	Foreign Key Constraint (mostBooks) fehlerfrei	+2
	$\sum$ 7, min 0	
(4b)(i/ii)	Ein falscher/fehlender Eintrag -1	-1
	Zwei oder mehr falsche/fehlende Einträge -2	-2
	$\sum$ 4, min 0	

Hinweis: Achten Sie bei den Statements auf die Reihenfolge.

b) Für diese Aufgabe müssen Sie sich mit verschiedenen Schlüsselabhängigkeiten auseinandersetzen.

(4 Punkte)

i) Gegeben ist folgendes Schema:

`room` (house: *house.number*, name)
`house` (number: *room.house*, largestRoom: *room.name*)

Es ist gefragt, dass Sie folgende Tabellen so erweitern, **dass die resultierende Datenbank-Instanz das Schema erfüllt**. Hierbei stellen die 2 Tabellen den vollständigen Inhalt der Datenbank dar. Um Punkte für diesen Teil zu erhalten, müssen Sie die Tabellen vollständig ausfüllen.

room

house	**name**
13	Living Room
22	Bathroom
43	Kitchen
52	Bedroom
10	Garage

house

number	**largestRoom**
22	Bathroom
43	Kitchen
10	Garage
13	Living Room
52	Bedroom

Achtung: Die Lösung ist nicht einzigartig, es wird nur eine beliebige korrekte Instanz verlangt. **Achtung:** Die Lösung ist nicht einzigartig, es wird nur eine mögliche Lösung gezeigt.

ii) Gegeben ist folgendes Schema:

`X` (id, Z: *Z.id*)
`Y` (id, X: *X.id*, Z: *Z.id*)
`Z` (id, Y: *Y.id*)

Es ist gefragt, dass Sie folgende Tabellen so erweitern, **dass die resultierende Datenbank-Instanz das Schema erfüllt**. Hierbei stellen die 3 Tabellen den vollständigen Inhalt der Datenbank dar. Um Punkte für diesen Teil zu erhalten, müssen Sie die Tabellen vollständig ausfüllen.

X

id	**Z**
1	3
2	2
3	1

Y

id	**X**	**Z**
1	2	1
2	3	1
3	1	2

Z

id	**Y**
1	1
2	2
3	3

Achtung: Die Lösung ist nicht einzigartig, es wird nur eine beliebige korrekte Instanz verlangt. **Achtung:** Die Lösung ist nicht einzigartig, es wird nur eine mögliche Lösung gezeigt.

MatrikelNr: _____ **Nachname:** _____

Aufgabe 5: Rekursive Abfragen (12)

Gegeben ist die folgende Rekursive Abfrage auf dem Datenbank-Schema von Aufgabe 4 a):

```
WITH RECURSIVE tmp(id, toName) AS
(
 SELECT  d.id, d.toName
 FROM    deliveries d
 WHERE   d.toName = 'Anna Schweiger' AND d.toSid = '22'
UNION
 SELECT  d.id, d.toName
 FROM    tmp t, deliveries d, books b1, books b2
 WHERE   t.id = b1.id
         AND d.id = b2.id
         AND b1.fromAuthor = b2.fromAuthor
)
SELECT  t.id, t.toName
FROM    tmp t
```

Werten Sie diese Abfrage auf der Datenbank-Instanz, die auf der Seite 16 (letztes Blatt der Prüfung) angegeben ist, aus. Wir empfehlen das letzte Blatt abzutrennen.

id	name
10	Anna Schweiger
30	Hubert Stoll
40	Elisabeth Stark
35	Theodor Riegler

Punkteverteilung Aufgabe 5

'Anna Schweiger' richtig (nicht rekursiv)	+2
Andere Zeile richtig (rekursiv)	+3
Pro komplett falsche Zeile	-2
Leichter Fehler (z.B. eine falsche id)	-1
Schwerer Fehler (z.B. statt id toSid verwendet)	-2
Duplikate (Ausführung ca. wie UNION ALL)	-2
Behauptete Lösung ist leer da Endlosschleife	-2
$\sum$ 11, min 0	
Richtiges Schema	+1

MatrikelNr: _____ **Nachname:** _____

Aufgabe 6: PL/SQL Trigger (12)

Betrachten Sie die Beispielinstanz auf **Seite 16** (letztes Blatt der Prüfung). Wir empfehlen das Blatt abzutrennen.

In jeder der folgenden Teilaufgaben ist ein SQL Statement gegeben, das **über die Beispielinstanz** ausgeführt wird. Nehmen Sie jeweils nur den Bestand entsprechend des letzten Blattes an, d.h., Inserts in Aufgabe a haben keinen Einfluss auf Aufgabe b usw. Auch die Funktionen und Trigger beziehen sich nur auf die jeweiligen Teilaufgaben.

Geben Sie die Ausgabe der `SELECT`-Statements an. Falls ein Fehler auftreten würde, geben Sie an welcher Fehler auftritt.

a) (4 Punkte)

```
CREATE OR REPLACE FUNCTION fInsertBooksTwo() RETURNS TRIGGER AS $$
BEGIN

        UPDATE authors SET printedPages = printedPages + NEW.pages
        WHERE NEW.fromAuthor = name;

        RETURN NEW;

END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER fInsertBooksTwo AFTER INSERT ON books
       FOR EACH ROW EXECUTE PROCEDURE fInsertBooksTwo();
```

```
INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (40, 3, 100, 50.00, 'Wells');
INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (35, 2, 50, 48.99, 'Swift');

SELECT * FROM authors WHERE country = 'United Kingdom';
```

```
name | printedpages | country
---------+--------------+----------------
Christie | 1360 | United Kingdom
Tolkien | 550 | United Kingdom
Wells | 100 | United Kingdom
Swift | 375 | United Kingdom
```

b)

(4 Punkte)

```
CREATE OR REPLACE FUNCTION fInsertBooks() RETURNS TRIGGER AS $$
DECLARE
        curr_del INTEGER;
        curr_sid INTEGER;
        curr_name VARCHAR(255);
        largest_del INTEGER;
BEGIN

        SELECT COUNT(itemNumber), d.toSid, d.toName
               INTO curr_del, curr_sid, curr_name
        FROM deliveries d, books b
        WHERE NEW.id = d.id AND d.id = b.id
        GROUP BY d.toSid, d.toName;


        SELECT COUNT(itemNumber) INTO largest_del
        FROM readers r, deliveries d, books b
        WHERE curr_name = r.name AND curr_sid = r.sid AND
              r.mostBooks = d.id AND d.id = b.id;

        IF curr_del > largest_del THEN
                UPDATE readers SET mostBooks = NEW.id
                WHERE curr_name = name AND curr_sid = sid;
        END IF;

        RETURN NEW;

END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER fInsertBooks AFTER INSERT ON books
       FOR EACH ROW EXECUTE PROCEDURE fInsertBooks();
```

```
INSERT INTO deliveries(id, date, toName, toSid) VALUES (50, '24.11.2022', 'Anna Schweiger', 22);
INSERT INTO deliveries(id, date, toName, toSid) VALUES (55, '24.11.2022', 'Elisabeth Stark', 25);

INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (50, 1, 475, 50.00, 'Goethe');
INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (50, 2, 325, 30.00, 'Twain');
INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (50, 3, 250, 40.00, 'Jelinek');
INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (55, 1, 322, 45.50, 'Twain');

SELECT name, mostBooks FROM readers;
```

```
 name | mostbooks
------------------+-----------
Thomas Mueller | 25
Johanne Mueller | 20
Theodor Riegler | 35
Hubert Stoll | 30
Elisabeth Stark | 40
Lieschen Mueller | 15
Anna Schweiger | 50
```

c) (4 Punkte)

```
CREATE OR REPLACE FUNCTION fCheapPocketBooks() RETURNS TRIGGER AS $$
BEGIN

        IF NEW.pages < 40 AND NEW.cost > 20 THEN
                INSERT INTO books VALUES (NEW.id, NEW.itemNumber,
                                          NEW.pages, NEW.cost / 2, NEW.fromAuthor);
                RETURN NULL;
        ELSE
                RETURN NEW;
        END IF;

END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER fCheapPocketBooks BEFORE INSERT ON books
       FOR EACH ROW EXECUTE PROCEDURE fCheapPocketBooks();
```

```
INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (40, 3, 10, 60.00, 'Tolkien');
INSERT INTO books(id, itemNumber, pages, cost, fromAuthor) VALUES (15, 4, 30, 26.00, 'Wells');

SELECT fromAuthor, itemNumber, cost FROM books WHERE itemNumber >= 3;
```

```
fromauthor | itemNumber | cost
------------+------------+-------
Jelinek | 3 | 50.99
Tolkien | 3 | 15.00
Wells | 4 | 13.00
```

Punkteverteilung Aufgabe 6

(6a)

für korrekten Eintrag (Wells | 100 | UK) - ruft Trigger einmal auf	+1.5
für korrekten Eintrag (Swift | 375 | UK) - ruft Trigger einmal auf	+1.5
für korrekten Eintrag (Christie | 1360 | UK) - ruft Trigger nicht auf	+0.5
für korrekten Eintrag (Tolkien | 550 | UK) - ruft Trigger nicht auf	+0.5
	$\sum$ 4

(6b)

für korrekten Eintrag (Elisabeth Stark | 40) - ruft Trigger nicht auf	+1.5
für korrekten Eintrag (Anna Schweiger | 50) - ruft Trigger nicht auf	+1.5
für korrekte restliche Einträge - rufen Trigger nicht auf	+1
	$\sum$ 4
Für falsches Schema	-0.5

(6c)

für korrekten Eintrag (Tolkien | 3 | 15.00) - ruft Trigger zweimal auf	+2
für korrekten Eintrag (Wells | 4 | 13.00) - ruft Trigger einmal auf	+1.5
für korrekte restliche Einträge - rufen Trigger nicht auf	+0.5
	$\sum$ 4
Wenn nur (Tolkien | 3 | 30.00) hinzugefügt wurde	-1
wenn wenn sowohl (Tolkien | 3 | 60) als auch (Tolkien | 3 | 30) hinzugefügt wurden	-1

Gesamtpunkte: 70

Viel Erfolg

MatrikelNr: _____ **Nachname:** _____

Beispielinstanz für Aufgabe 5 und Aufgabe 6:

Sie können diesen Zettel abtrennen und brauchen ihn nicht abgeben!

Diesen Zettel daher bitte nicht beschriften! (Lösungen auf diesem Zettel werden nicht gewertet!)

readers

name	sid	country	mostBooks
Thomas Mueller	21	Austria	25
Johanne Mueller	23	United States	20
Anna Schweiger	22	Germany	10
Theodor Riegler	24	Germany	35
Hubert Stoll	26	Brazil	30
Elisabeth Stark	25	Denmark	40
Lieschen Mueller	27	Japan	15

books

id	itemNumber	pages	cost	fromAuthor
10	1	475	50.00	Goethe
10	2	305	60.50	Christie
15	1	255	75.00	Twain
15	2	660	40.50	Jelinek
15	3	220	50.99	Jelinek
30	1	430	60.00	Christie
35	1	95	44.50	Swift
20	1	115	50.00	Tolkien
25	1	435	60.95	Tolkien
40	1	230	70.00	Swift
40	2	625	75.90	Christie

deliveries

id	date	toName	toSid
10	24.11.2022	Anna Schweiger	22
15	15.11.2022	Lieschen Mueller	27
20	27.11.2022	Johanne Mueller	23
25	12.10.2020	Thomas Mueller	21
30	14.09.2021	Hubert Stoll	26
35	01.12.2022	Theodor Riegler	24
40	12.12.2020	Elisabeth Stark	25

authors

name	printedPages	country
Swift	325	United Kingdom
Goethe	475	Germany
Christie	1360	United Kingdom
Twain	255	United States
Wells	0	United Kingdom
Jelinek	880	Austria
Tolkien	550	United Kingdom