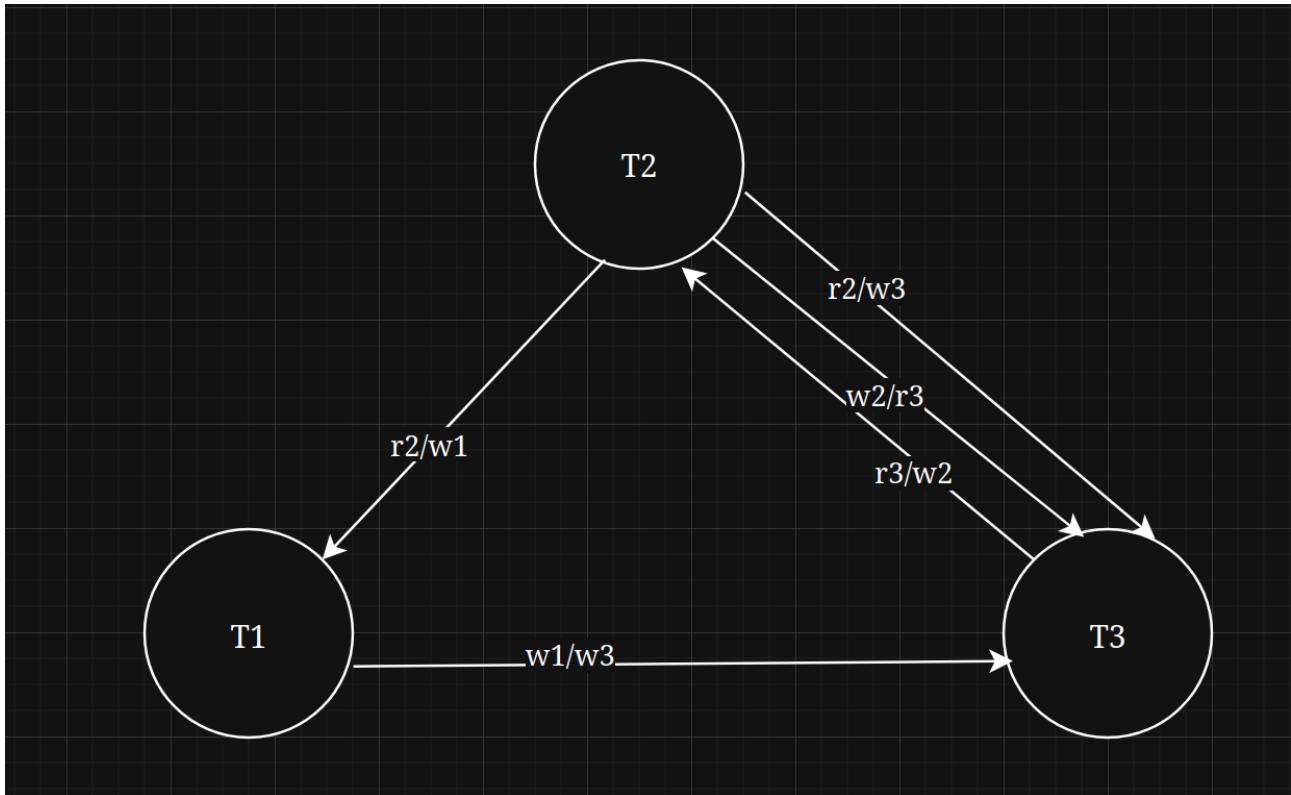


A1)



Der Konfliktgraph ist azyklisch, somit ist die Ausführung konfliktbehaftet und nicht Konfliktserialisierbar.

b)

T1	T2	T3	a		b		c		d	
85	75	100	tr(a)	tw(a)	tr(b)	tw(b)	tr(c)	tw(c)	tr(d)	tw(d)
	READ a		75	0	0	0	0	0	0	0
		READ a	100	0	0	0	0	0	0	0
	WRITE b		100	0	0	75	0	0	0	0
WRITE d			100	0	0	75	0	0	0	85
		WRITE d	100	0	0	75	0	0	0	100
	READ c		100	0	0	75	75	0	0	100
		WRITE a	100	0 (Konflikt)	0	75	75	0	0	100
WRITE c	Abort		100	0	0	75	75	85	0	100
		WRITE c	100	0	0	75	75	100	0	100
		READ b	100	0	100	75	75	100	0	100

A3)

Transaktion 1	Zeitpunkt	Transaktion 2	Erklärung	
LOCK TABLE dept IN ROW SHARE MODE;	1	i Transaction Response Lock succeeded. LOCK TABLE dept IN ROW SHARE MODE; T1 2026-01-11T10:22:47.952895110Z SQL1 PTO.0399071935	Transaktion (T1) setzt einen Row Share lock auf die Tabelle dept eine Transaktion (T2) versucht den Table zu löschen Das ist nicht möglich da dieser gerade unter einem Row Share (Read) Lock steht.	
	2	DROP TABLE dept;	Verhindernung: T2 muss warten bis ein rollback oder commit von T1 ausführt wird um strukturelle Änderungen durchzuführen. T2 versucht einen exclusive lock (X) auf den Table Dept zu setzen Dieser wird sofort von Row Share (RS) von T1 Blockiert.	
	3	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT; SELECT loc FROM dept WHERE deptno=20 FOR UPDATE OF loc;	Error report - ORA-00054: resource busy and acquire with NOWAIT specific Statement Connection Created Type Executions Duration SELECT loc FROM dept WHERE deptno=20 FOR UPDATE OF loc... T2 11/01/2026 11:40... 0 00.000s All rows fetched: 1 in 0.001 seconds LOC 1 DALLAS	Verhindernung: T2 muss warten bis ein rollback oder commit von T1 ausführt wird um exclusive Locks durchzuführen. T2 selektiert das row loc und von der table dept Mit der deptno=20 und sperrt es anschließend Mit dem FOR UPDATE OF loc; mit einem Row exclusive (x) lock
	4		Konflikt: wenn eine andere Transaktion (T1) diese Zeile verändern will Muss sie warten T1 versucht loc mit „NEW YORK“ zu update Dort wo deptno=20 ist Konflikt: Da T2 einen RX lock auf diese Zeilen ausgeführt hat Und T1 kein nowait im SQL befehl hinzugefügt hat, Muss der T1 warten bis T2 den lock aufhebt oder Ein commit oder rollback durchführt T2 führt das rollback aus Und setzt die Transaktionen von T2 wieder zurück zum letzten commit	
UPDATE dept SET loc='NEW YORK' WHERE deptno=20;	5		Und T1 kann nun im anchluss update auf dept Durchführen da der rx lock aufgelöst wird	
ROLLBACK;	6	ROLLBACK;	Bis zu diesem Zeitpunkt hat T1 jedoch gewartet T1 führt das rollback aus und setzt Die Transaktionen von T1 wieder zurück zum letzten commit T1 führt einen row exclusive (x) lock auf dept durch Andere transaktionen dürfen einen rx oder rs lock Auf dept durchführen jedoch kein s,x oder sr lock durchführen ????	
LOCK TABLE dept IN ROW EXCLUSIVE MODE;	8		T2 versucht einen exclusive lock (X) auf den Table zu setzen Dieser wird von Row Exclusive (RX) von T1 Blockiert. Konflikt: RX vs X funktioniert nicht Lösung: T2 muss warten bis T1 den lock auflöst oder ein commit oder rollback durchführt	
	9	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT;	Durch das nowait wartet T2 nicht und ein fehler wird direkt geworfen Konflikt: RX vs X funktioniert nicht Lösung: T2 muss warten bis T1 den lock auflöst oder ein commit oder rollback durchführt	
	10	LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE NOWAIT;	Durch das nowait wartet T2 nicht und ein fehler wird direkt geworfen Konflikt: RX vs S funktioniert nicht Lösung: T2 muss warten bis T1 den lock auflöst oder ein commit oder rollback durchführt	
SELECT loc FROM dept WHERE deptno=20 FOR UPDATE OF loc;	11	LOCK TABLE dept IN SHARE MODE NOWAIT;	Lösung: T2 muss warten bis T1 den lock auflöst oder ein commit oder rollback durchführt Durch das nowait wartet T2 nicht und ein fehler wird direkt geworfen Die Transaktion funktioniert da UPDATE Table Einne RX lock erzeugt und RX vs X funktionieren (wenn verschiedene Reihen betroffen sind)	
	12	UPDATE dept SET loc='NEW YORK' WHERE deptno= 20;		
	13	ROLLBACK;		
	14	All rows fetched: 1 in 0.526 seconds LOC 1 DALLAS	T1 selektiert das Feld loc und von der table dept Mit der deptno=20 und sperrt es anschließend Mit dem FOR UPDATE OF loc; mit einem Row exclusive (x) lock	
	15	UPDATE dept SET loc='NEW YORK' WHERE deptno=20;	T2 versucht loc mit „NEW YORK“ zu update Dort wo deptno=20 ist Konflikt: Da T1 einen RX lock auf diese Zeilen ausgeführt hat Und T2 kein nowait im SQL befehl hinzugefügt hat, Muss der T2 warten bis T1 den lock aufhebt oder Ein commit oder rollback durchführt T1 führt ein Rollback aus und somit den RX lock rückgängig Dadurch kann T2 das Update durchführen	
ROLLBACK;	16		T2 führt das rollback aus und setzt Die Transaktionen von T2 wieder zurück zum letzten commit	
	17	ROLLBACK;	T1 führt einen share mode (s) lock (read lock für die ganze tabelle dept)	
LOCK TABLE dept IN SHARE MODE;	18		Konsequenz: Es kann nur ein s oder rs lock Von einer anderen Transaktion durchgeführt werden T2 versucht einen X lock durchzuführen funktioniert nicht	
	19	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT;	Funktioniert erst nach auflösen, rollback oder commit von r1	
	20	LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE NOWAIT;	T2 versucht einen SRX lock durchzuführen funktioniert nicht Funktioniert erst nach auflösen, rollback oder commit von r1	
	21	LOCK TABLE dept IN SHARE MODE;	T2 führt einen S lock auf dept durch	
	22	SELECT loc FROM dept WHERE deptno=20;	T2 führt eine select Abfrage auf dept durch Funktioniert da ein reines select keinen lock beinhaltet T2 führt ebenfalls eine select Abfrage auf dept Durch. Zusätzlich wird aber noch die For Update Of Klausel angegeben Diese verursacht einen RX lock Da diese Klausel ohne NOWAIT angegeben wurde, wartet T2 solange bis der S lock von T1 aufgelöst wird (oder rollback/commit)	
	23	SELECT loc FROM dept WHERE deptno=20 FOR UPDATE OF loc;	T1 führt einen Rollback aus (T51 wird bis zum letzten commit zurückgesetzt) RX lock von T52 wird somit anschließend ausgeführt Der RX lock von T52 hat genau diese rows gesperrt Die nun von diesem Statement updated werden Funktioniert T2 führt das rollback aus und setzt Die Transaktionen von T2 wieder zurück zum letzten commit T1 führt einen share row exclusive mode (sr) lock (read lock für die ganze tabelle dept)	
ROLLBACK;	24		Konsequenz: nur kompatibel mit rs lock	
	25	UPDATE dept SET loc='NEW YORK' WHERE deptno=20;	Konflikt: SRX vs X S vs X und RX vs X würde nicht funktionieren Aufrund von nowait direkte Fehlermeldung Ansonsten würde die Transaktion warten	
	26	ROLLBACK;		
LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE;	27			
	28	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specific Bis der SRX lock von T1 aufgelöst werden würde	

Koller Moritz

UE6_Datenbanken

29	LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specific	Konflikt: SRX vs SRX funktioniert nicht 'SRX erzeugt eine lese und row schreibsperrre 'Und eine lesesperrre der ganzen Tabelle und eine 'Row schreibsperrre stehen im konflikt 'Aufgrund von nowait direkte Fehlermeldung 'Ansonsten würde die Transaktion warten Bis der SRX lock von T1 aufgelöst werden würde Konflikt: SRX vs S S vs RX würde nicht funktionieren Aufgrund von nowait direkt Fehlermeldung Ansonsten würde die Transaktion warten Bis der SRX lock von T1 aufgelöst werden würde	
30	LOCK TABLE dept IN SHARE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specific	Konflikt: SRX vs RX S vs RX würde nicht funktionieren Aufgrund von nowait direkt Fehlermeldung Ansonsten würde die Transaktion warten Bis der SRX lock von T1 aufgelöst werden würde	
31	LOCK TABLE dept IN ROW EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specific	Funktioniert da es ein share mode lock ist 'Und nur den Lesezugriff von T2 zulässt t2 führt das rollback aus und setzt Die Transaktionen von t2 wieder zurück zum letzten commit t1 führt das rollback aus und setzt	
32	LOCK TABLE dept IN ROW SHARE MODE;	Lock succeeded.	'Die Transaktionen von t1 wieder zurück zum letzten commit t1 führt ein Select mit einem FOR UPDATE durch Konsequenz: 'Das ist ein RX lock und lässt somit nur mehr RS und RX locks von anderen Transaktion zu	
33	ROLLBACK;	Rollback complete.		
ROLLBACK; SELECT loc FROM dept WHERE deptno=20 FOR UPDATE;	34	Rollback complete.		
	35	All rows fetched: 1 in 0.005 seconds 		
	36	All rows fetched: 1 in 0.034 seconds 	T2 führt ebenfalls ein select mit einem FOR UPDATE durch RX vs RX und T2 greift auf andere ROWS zu Somit funktioniert es t2 führt hoch einmar ein select mit Einem FOR UPDATE durch RX vs RX aber T2 will die gleichen ROWS updaten Wie T1 Da T2 kein NOWAIT hinzugefügt hat, wartet die Transaktion bis T1 seinen lock auflöst T1 möchte nun ebenfalls ein FOR UPDATE auf Die selben Zeilen wie T2 in 36 ausführen	
SELECT loc FROM dept WHERE deptno=40 FOR UPDATE;	37	ue6_a3_t2.sql: Script Runner... 	RX vs RX auf dieselben Zeilen T1 wartet ebenfalls bis der lock aufgelöst wird	
SELECT loc FROM dept WHERE deptno=40 FOR UPDATE;	38	All rows fetched: 1 in 104.647 seconds 		
	39	ROLLBACK; 1 BOSTON All rows fetched: 1 in 0.001 seconds 	T2 macht ein rollback RX auf Rows where deptno=40 Wird aufgelöst Und T1 kann diesen RX lock setzen T1 macht ein Rollback und setzt sich zurück auf Den letzten commit T1 setzt X lock auf dept Auswirkung: Andere Transaktionen dürfen keine Anderen locks mehr setzen T2 versucht ebenfalls X lock	
ROLLBACK;	40	ROLLBACK complete. ORA-00060: deadlock detected while waiting for resource https://docs.oracle.com/cd/B19306_01/doc/server.102/b14225/detect.htm#REFRN10001		
LOCK TABLE dept IN EXCLUSIVE MODE;	41	Lock succeeded.		
	42	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specific T2 muss warten bis X lock aufgelöst wird	
LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE NOWAIT;	43	All rows fetched: 1 in 0.001 seconds 	T1 erstellt SRX lock X lock ist die restriktivere Sperre und diese Sperre Gilt immer noch T2 versucht S lock	
	44	LOCK TABLE dept IN SHARE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specific Konflikt: X vs S wird nicht zugelassen T2 versucht RX lock	
	45	LOCK TABLE dept IN ROW EXCLUSIVE MODE NOWAIT;	Error starting at line : 1 in command - LOCK TABLE dept IN ROW EXCLUSI Konflikt: X vs RX wird nicht zugelassen T2 versucht RS lock	
	46	LOCK TABLE dept IN ROW SHARE MODE NOWAIT;	Error starting at line : 1 in command - LOCK TABLE dept IN ROW SHARE MODE Konflikt: X vs RS wird nicht zugelassen T2 führt ein normales select aus Dies verursacht keine Sperre und kann somit Ausgeführt werden Mit FOR UPDATE verursacht das Select jedoch Eine RX Sperre (von T2 ausgeführt)	
	47	SELECT loc FROM dept WHERE deptno=40; 	T2 versucht R lock Das Update soll auf der loc spalte ausgeführt werden, An den Spalten an denen deptno=40 ist T2 wartet bis beide locks freigegeben werden	
UPDATE dept SET deptno=50 WHERE deptno=40;	48	FOR UPDATE OF loc; 1 row updated. Commit complete.	T1 hat von tabelle dept deptno=40 auf deptno=50 Updated T1 hat die changes committet Die zwei locks wurden aufgelöst jedoch hat sich deptno=40 auf deptno=50 geändert T2 versucht nun ein select auf deptno=40 und Durch die FOR UPDATE Klausel einen RX lock Darauf zu setzen. Da es keine deptno=40 mehr gibt ist das Select Leer und der RX lock wird nicht durchgeführt T1 hat seine Transaktion auf read only gesetzt Das bedeutet INSERT, UPDATE oder DELETE werden Blockiert und geben eine Fehlermeldung aus Zusätzlich wird zu diesem Zeitpunkt ein Snapshot Der DB gemacht. Änderungen die nach diesem Zeitpunkt festgeschrieben Werden (committed) bleiben für T1 unsichtbar Besteht bis zum nächsten Rollback oder commit	
COMMIT;	49	All rows fetched: 0 in 374.766 seconds 	T1 hat von tabelle dept deptno=40 auf deptno=50 Updated T1 hat die changes committet Die zwei locks wurden aufgelöst jedoch hat sich deptno=40 auf deptno=50 geändert T2 versucht nun ein select auf deptno=40 und Durch die FOR UPDATE Klausel einen RX lock Darauf zu setzen. Da es keine deptno=40 mehr gibt ist das Select Leer und der RX lock wird nicht durchgeführt T1 hat seine Transaktion auf read only gesetzt Das bedeutet INSERT, UPDATE oder DELETE werden Blockiert und geben eine Fehlermeldung aus Zusätzlich wird zu diesem Zeitpunkt ein Snapshot Der DB gemacht. Änderungen die nach diesem Zeitpunkt festgeschrieben Werden (committed) bleiben für T1 unsichtbar Besteht bis zum nächsten Rollback oder commit	
SET TRANSACTION READ ONLY;	50	Transaction READ succeeded.		
SELECT loc FROM dept WHERE deptno=10;	51	All rows fetched: 1 in 0.036 seconds 	T1 führt einen normalen select durch Das ist kein Problem Es ist keine Sperre mehr auf dept T2 kann also ohne Probleme das UPDATE durchführen	
	52	LOC 1 NEW YORK		
	53	UPDATE dept SET loc='LINZ' WHERE deptno=10; 1 row updated.	Das Update ist derzeit nur für T2 sichtbar Und T2 setzt ebenfalls einen RX lock bis zum nächsten Commit auf diese Zeile Das Update von T2 ist noch nicht in T1 sichtbar Erst nachdem T2 einen commit durchgeführt hat Da ein reines Select keine Sperre durchführt Kommt es auch zu keinem Konflikt T2 hat einen Commit durchgeführt Alle seine Änderungen sind nun in der DB gespeichert Aufgrund von Read Only wird T1 repeatable read Garantiert und T1 sieht immer noch das alte Ergebnis Im Select. T1 hat seine Änderungen in die DB committed	
SELECT loc FROM dept WHERE deptno=10;	54	All rows fetched: 1 in 0.001 seconds 	Transaction read only wurde somit aufgelöst und T1 sieht den aktuellen stand der DB	
SELECT loc FROM dept WHERE deptno=10;	55	COMMIT;		
	56	Commit complete.	Transaction read only wurde durch das commit aufgelöst und T1 sieht den aktuellem stand der DB	
COMMIT;	57	Commit complete.		
SELECT loc FROM dept WHERE deptno=10;	58	All rows fetched: 1 in 0.002 seconds 	Transaction read only wurde durch das commit aufgelöst und T1 sieht den aktuellem stand der DB	

Die lange Fehlermeldung ist meist identisch.
Sie beschreibt dass ein lock einen anderen lock blockiert

```
Error starting at line : 1 in command -  
LOCK TABLE dept  
IN ROW SHARE MODE  
NOWAIT  
Error report -  
ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired  
https://docs.oracle.com/error-help/db/ora-00054/  
00054. 00000 - "Failed to acquire a lock (Type: \"%s\", Name: \"%s\", Description: \"%s\") because it is currently held by another session. The resource being locked can be identified by %s (%\"s\") and %s (%\"s\")"  
*Cause: The current session is failing to acquire the specified lock on  
the specified resource because another session holds the lock.  
*Action: Retry the operation and see if the lock has been  
released in the meantime. If not, identify which  
session holds the lock and on which resource using the information  
provided in the error message text and terminate the  
transaction of the session before retrying the operation.
```