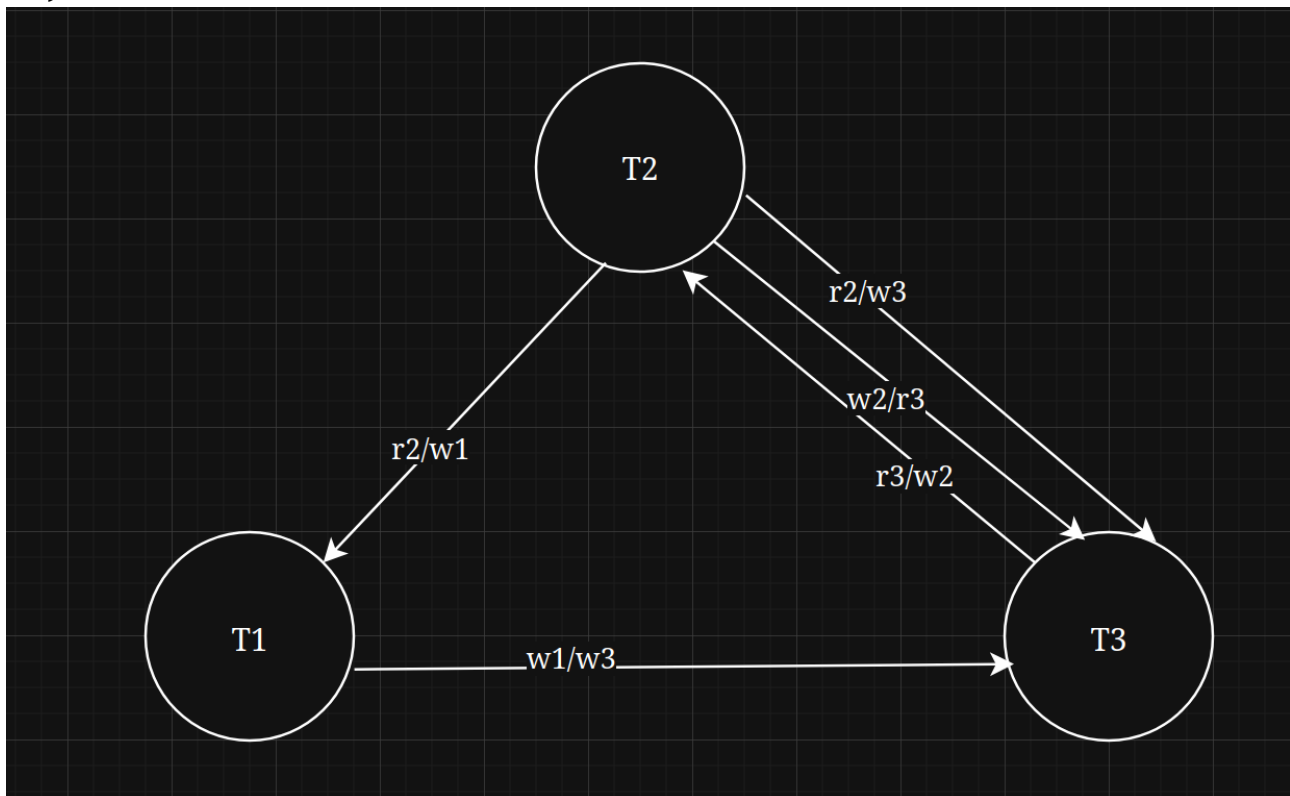


A1)



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

T1	T2	T3	a		b		c		d	
80	70	65	tr(a)	tw(a)	tr(b)	tw(b)	tr(c)	tw(c)	tr(d)	tw(d)
	READ a		70	0	0	0	0	0	0	0
		READ a	65	0	0	0	0	0	0	0
	WRITE b		65	0	0	70	0	0	0	0
WRITE d			65	0	0	70	0	0	0	80
		WRITE d	65	0	0	70	0	0	0	80 (Konflikt)
	READ c	Abort	65	0	0	70	70	0	0	80
	WRITE a		65	70	0	70	70	0	0	80
WRITE c			65	70	0	70	70	80	0	80
		WRITE c Abort	65	70	0	70	70	80 (Konflikt)	0	80
		READ b	65	70	0 (Konflikt)	70	70	80	0	80
		Abort								

A3)

Transaktion 1	Zeitpunkt	Transaktion 2	Transaction Response	Erklärung
LOCK TABLE dept IN ROW SHARE MODE;	1		Lock succeeded. LOCK TABLE dept IN ROW SHARE MODE; T1 2026-01-11T10:22:47.952895110Z SQL 1 PT0.039907193S	Transaktion (T1) setzt einen Row Share lock of 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;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired	Verhinderung: T2 muss warten bis ein rollback oder commit von T1 ausgeführt wird um strukturelle Änderungen durchzuführen. T2 versucht einen exklusiven 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 specified or timeout expired	Verhinderung: T2 muss warten bis ein rollback oder commit von T1 ausgeführt wird um exklusive Locks durchzuführen. T2 versucht einen exklusiven lock (X) auf den table dept Mit der deptno=20 und sperrt es anschließend Mit dem FOR UPDATE OF loc; mit einem Row exclusive (rx) lock
	4		All rows fetched: 1 in 0.037 seconds LOC 1 DALLAS	Konflikt: wenn eine andere Transaktion (T1) diese Zeile verändern will Muss sie warten T1 versucht loc mit „NEW YORK“ zu updaten Dort wo deptno=20 ist
UPDATE dept SET loc='NEW YORK' WHERE deptno=20;	5		Streams Table 0 ue6_a3.t1.sql: Script Runner...	Konflikt: Da T1 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
	6	ROLLBACK;	Rollback complete. 1 row updated.	Und T1 kann nun im anschluss update auf dept Durchführen da der rx lock aufgelöst wurde Bis zu diesem Zeitpunkt hat t1 jedoch gewartet t1 führt das rollback aus und setzt
ROLLBACK;	7		Rollback complete.	Die Transaktionen von t1 wieder zurück zum letzten commit T1 führt einen row exclusive (rx) lock auf dept durch Andere transaktionen dürfen einen rx oder rs lock Auf dept durchführen jedoch kein s.x oder sr.x lock durchführen ??? T2 versucht einen exklusiven lock (X) auf den Table dept 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
LOCK TABLE dept IN ROW EXCLUSIVE MODE;	8		Statement LOCK TABLE dept IN ROW EXCLUSIVE MODE T1 11/01/2026 SQL 1 00.052s	Durch das nowait wartet t2 nicht und ein fehler wird direkt geworfen Konflikt: RX vs SRX funktioniert ebenfalls nicht Da row exclusive den schreibzugriff für einige Zeilen sperrt Und Share Row Exclusive den Lesezugriff für die ganze Tabelle und den Schreibzugriff für ein paar Felder sperrt Das Problem ist der Lesezugriff der gesamten table RX vs RX solange es unterschiedliche Reihen sind funktioniert Lösung: T2 muss warten bis T1 den lock auflöst oder ein commit oder rollback durchführt
	9	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT;	ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired	Durch das nowait wartet t2 nicht und ein fehler wird direkt geworfen Konflikt: RX vs SRX funktioniert ebenfalls nicht Da row exclusive den schreibzugriff für einige Zeilen sperrt Und Share Row Exclusive den Lesezugriff für die ganze Tabelle und den Schreibzugriff für ein paar Felder sperrt Das Problem ist der Lesezugriff der gesamten table RX vs RX solange es unterschiedliche Reihen sind funktioniert 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;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired	Durch das nowait wartet t2 nicht und ein fehler wird direkt geworfen Konflikt: RX vs S T2 möchte die gesamte Tabelle mit einem S (Lese) lock belegen Funktioniert nicht da die Tabelle schon mit RX von T1 belegt ist Lösung: T2 muss warten bis T1 den lock auflöst oder ein commit oder rollback durchführt
	11	LOCK TABLE dept IN SHARE MODE NOWAIT; UPDATE dept SET loc='NEW YORK' WHERE deptno = 20;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired	Durch das nowait wartet t2 nicht und ein fehler wird direkt geworfen Die Transaktion funktioniert da UPDATE Table Einen RX lock erzeugt und RX vs RX funktionieren (wenn verschiedene Reihen betroffen sind)
	12	UPDATE dept SET loc='NEW YORK' WHERE deptno = 20;	Statement UPDATE dept T1 11/01/2026 SQL 1 00.050s	
	13	ROLLBACK;	Rollback complete.	
SELECT loc FROM dept WHERE deptno=20 FOR UPDATE OF loc;	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 (rx) lock
	15	UPDATE dept SET loc='NEW YORK' WHERE deptno=20;	Table 0 ue6_a3.t2.sql: Script Runner...	T2 versucht loc mit „NEW YORK“ zu updaten 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
ROLLBACK;	16		Rollback complete. 1 row updated.	T1 führt ein Rollback aus und somit den RX lock rückgängig Dadurch kann T2 das Update durchführen T2 führt das rollback aus und setzt
	17	ROLLBACK;	Rollback complete.	Die Transaktionen von t2 wieder zurück zum letzten commit T1 führt einen share mode (s) lock (read lock führt die ganze table dept)
LOCK TABLE dept IN SHARE MODE;	18		Statement LOCK TABLE T1 11/01/2026 SQL 2 00.055s	Konsequenz: Es kann nur ein s oder rs lock Von einer anderen Transaktion durchgeführt werden T2 versucht einen X lock durchzuführen funktioniert nicht Funktioniert erst nach auflösen, rollback oder commit von r1
	19	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired	T2 versucht einen SRX lock durchzuführen funktioniert nicht Funktioniert erst nach auflösen, rollback oder commit von r1
	20	LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired	T2 versucht einen SRX lock durchzuführen funktioniert nicht Funktioniert erst nach auflösen, rollback oder commit von r1
	21	LOCK TABLE dept IN SHARE MODE;	Lock succeeded.	T2 führt einen S lock auf dept durch
	22	SELECT loc FROM dept WHERE deptno=20;	All rows fetched: 1 in 0.054 seconds LOC 1 DALLAS	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.
	23	SELECT loc FROM dept WHERE deptno=20 FOR UPDATE OF loc;	Table 0 ue6_a3.t2.sql: Script Runner...	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)
ROLLBACK;	24		Rollback complete. Lock succeeded.	T1 führt einen Rollback aus (T1 wird bis zum letzten commit zurückgesetzt) RX lock von T2 wird somit anschließend ausgeführt Der RX lock von T2 hat genau diese rows gesperrt Die nun von diesem Statement geupdated werden Funktioniert
	25	UPDATE dept SET loc='NEW YORK' WHERE deptno=20;	1 row updated.	T2 führt das rollback aus und setzt Die Transaktionen von t2 wieder zurück zum letzten commit
	26	ROLLBACK;	Rollback complete.	T1 führt einen share row exclusive mode (sr.x) lock (read lock führt die ganze table dept)
LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE;	27		Statement LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE T1 11/01/2026 SQL 2 00.052s	Konsequenz: nur kompatibel mit rs lock Konflikt: SRX vs X S vs X und RX vs X würde nicht funktionieren Aufgrund von nowait direkte Fehlermeldung Ansonsten würde die Transaktion warten
	28	LOCK TABLE dept IN EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired	Bis der SRX lock von T1 aufgelöst werden würde

					<p>Konflikt: SRX vs SRX funktioniert nicht</p> <p>SRX erzeugt eine lese und row schreibsperre</p> <p>Und eine lesesperre der ganzen Tabelle und eine</p> <p>Row schreibsperre stehen im konflikt</p> <p>Aufgrund von nowait direkte Fehlermeldung</p> <p>Ansonsten würde die Transaktion warten</p> <p>Bis der SRX lock von T1 aufgelöst werden würde</p> <p>Konflikt: SRX vs S</p> <p>S vs RX würde nicht funktionieren</p> <p>Aufgrund von nowait direkte Fehlermeldung</p> <p>Ansonsten würde die Transaktion warten</p> <p>Bis der SRX lock von T1 aufgelöst werden würde</p>
	29	LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified		
	30	LOCK TABLE dept IN SHARE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified		<p>Konflikt: SRX vs RX</p> <p>S vs RX würde nicht funktionieren</p> <p>Aufgrund von nowait direkte Fehlermeldung</p> <p>Ansonsten würde die Transaktion warten</p> <p>Bis der SRX lock von T1 aufgelöst werden würde</p>
	31	LOCK TABLE dept IN ROW EXCLUSIVE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified		<p>Konflikt: SRX vs RX</p> <p>S vs RX würde nicht funktionieren</p> <p>Aufgrund von nowait direkte Fehlermeldung</p> <p>Ansonsten würde die Transaktion warten</p> <p>Bis der SRX lock von T1 aufgelöst werden würde</p>
	32	LOCK TABLE dept IN ROW SHARE MODE;	Lock succeeded.		<p>Funktioniert da es ein share mode lock ist</p> <p>Und nur den Lesezugriff von T2 zulässt</p>
	33	ROLLBACK;	Rollback complete.		<p>T2 führt das rollback aus und setzt</p> <p>Die Transaktionen von T2 wieder zurück zum letzten commit</p>
ROLLBACK; SELECT loc FROM dept WHERE deptno=20 FOR UPDATE;	34		Rollback complete.		<p>T1 führt das rollback aus und setzt</p> <p>Die Transaktionen von T1 wieder zurück zum letzten commit</p> <p>T1 führt ein Select mit einem FOR UPDATE durch</p> <p>Konsequenz:</p> <p>Das ist ein RX lock und lässt somit nur mehr</p> <p>RS und RX locks von anderen Transaktion zu</p>
	35	SELECT loc FROM dept WHERE deptno=40 FOR UPDATE;	All rows fetched: 1 in 0.046 seconds LOC 1 BOSTON		
	36				<p>T2 führt ebenfalls ein select mit einem FOR UPDATE durch</p> <p>RX vs RX und T2 greift auf andere ROWs zu</p> <p>Somit funktioniert es</p> <p>T2 führt noch einmal ein select mit</p> <p>Einem FOR UPDATE durch</p>
	37	SELECT loc FROM dept WHERE deptno=20 FOR UPDATE;	ue6_a3_t2.sql: Script Runner...		<p>RX vs RX aber T2 will die gleichen ROWs updaten</p> <p>Wie T1</p> <p>Da T2 kein NOWAIT hinzugefügt hat, wartet die</p> <p>Transaktion bis T1 seinen lock auflöst</p> <p>T1 möchte nun ebenfalls ein FOR UPDATE auf</p> <p>Die selben Zeilen wie T2 in 36 ausführen</p>
SELECT loc FROM dept WHERE deptno=40 FOR UPDATE;	38		ue6_a3_t1.sql: Script Runner...		<p>RX vs RX auf dieselben Zeilen</p> <p>T1 wartet ebenfalls bis der lock aufgelöst wird</p>
	39	ROLLBACK;	Rollback complete.		<p>T2 macht ein rollback RX auf Rows where deptno=40</p> <p>Wird aufgelöst</p> <p>Und T1 kann diesen RX lock setzen</p> <p>T1 macht ein Rollback und setzt sich zurück auf</p> <p>Den letzten commit</p>
ROLLBACK;	40		ORA-00060: deadlock detected while waiting for resource https://docs.oracle.com/		<p>T1 setzt X lock auf dept</p> <p>Auswirkung: Andere Transaktionen dürfen keine</p> <p>Anderen locks mehr setzen</p> <p>T2 versucht ebenfalls X lock</p>
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 specified		<p>T2 muss warten bis X lock aufgelöst wird</p>
LOCK TABLE dept IN SHARE ROW EXCLUSIVE MODE NOWAIT;	43		ue6_a3_t2.sql: Script Runner...		<p>T1 erstellt SRX lock</p> <p>X lock ist die restriktivere Sperre und diese Sperre</p> <p>Gilt immer noch</p> <p>T2 versucht S lock</p>
	44	LOCK TABLE dept IN SHARE MODE NOWAIT;	Error report - ORA-00054: resource busy and acquire with NOWAIT specified		<p>Konflikt: X vs S wird nicht zugelassen</p>
	45	LOCK TABLE dept IN ROW EXCLUSIVE MODE NOWAIT;	Error starting at line: 1 in command - LOCK TABLE dept IN ROW EXCLUSIVE MODE NOWAIT;		<p>T2 versucht RX lock</p> <p>Konflikt: X vs RX wird nicht zugelassen</p>
	46	LOCK TABLE dept IN ROW SHARE MODE NOWAIT;	Error starting at line: 1 in command - LOCK TABLE dept IN ROW SHARE MODE NOWAIT;		<p>T2 versucht RS lock</p> <p>Konflikt: X vs RS wird nicht zugelassen</p>
SELECT loc FROM dept WHERE deptno=40;	47		All rows fetched: 1 in 0.031 seconds LOC 1 BOSTON		<p>T2 führt ein normales select aus</p> <p>Dies verursacht keine Sperre und kann somit</p> <p>Ausgeführt werden</p> <p>Mit FOR UPDATE verursacht das select ja noch</p> <p>Eine RX Sperre (von T2 ausgeführt)</p>
	48	SELECT loc FROM dept WHERE deptno=40 FOR UPDATE OF loc;	ue6_a3_t2.sql: Script Runner...		<p>Das Update soll auf der loc spalte ausgeführt werden,</p> <p>An den Spalten an denen deptno=40 ist</p> <p>T2 wartet bis beide locks freigegeben werden</p>
UPDATE dept SET deptno=50 WHERE deptno=40;	49		1 row updated. Commit complete.		<p>T1 hat von tabelle dept deptno=40 auf deptno=50</p> <p>Upgedated</p> <p>T1 hat die changes committet</p> <p>Die zwei locks wurden aufgelöst</p> <p>Jedoch hat sich deptno=40 auf deptno=50 geändert</p>
	50	COMMIT;	All rows fetched: 0 in 374.766 seconds LOC		<p>T2 versucht nun ein select auf deptno=40 und</p> <p>Durch die FOR UPDATE Klausel einen RX lock</p> <p>Darauf zu setzen.</p> <p>Da es keine deptno=40 mehr gibt ist das Select</p> <p>Leer und der RX lock wird nicht durchgeführt</p> <p>T1 hat seine transaktion auf read only gesetzt</p> <p>Das bedeutet INSERT, UPDATE oder DELETE werden</p> <p>Blockiert und geben eine Fehlermeldung aus</p> <p>Zusätzlich wird zu diesem Zeitpunkt ein Snapshot</p> <p>Der DB gemacht.</p> <p>Änderungen die nach diesem Zeitpunkt festgeschrieben</p> <p>Werden (committed) bleiben für T1 unsichtbar</p> <p>Besteht bis zum nächsten Rollback oder commit</p>
SET TRANSACTION READ ONLY;	51		Transaction READ succeeded.		
SELECT loc FROM dept WHERE deptno=10;	52		All rows fetched: 1 in 0.036 seconds LOC 1 NEW YORK		<p>T1 führt einen normalen select durch</p> <p>Das ist kein Problem</p> <p>Es ist keine Sperre mehr auf dept</p> <p>T2 kann also ohne Probleme das UPDATE durchführen</p>
	53	UPDATE dept SET loc='LINZ' WHERE deptno=10;	1 row updated.		<p>Das Update ist derzeit nur für T2 sichtbar</p> <p>Und T2 setzt ebenfalls einen RX lock bis zum nächsten</p> <p>Commit auf diese Zeile</p> <p>Das Update von T2 ist noch nicht in T1 sichtbar</p> <p>Erst nachdem T2 einen commit durchgeführt hat</p>
SELECT loc FROM dept WHERE deptno=10;	54		All rows fetched: 1 in 0.037 seconds LOC 1 NEW YORK		<p>Da ein reines Select keine Sperre durchführt</p> <p>Kommt es auch zu keinem Konflikt</p> <p>T2 hat einen Commit durchgeführt</p> <p>Alle seine Änderungen sind nun in der DB gespeichert</p> <p>Aufgrund von Read Only wird T1 repeatable read</p> <p>Garantiert und T1 sieht immer noch das alte Ergebnis</p> <p>Im Select.</p> <p>T1 hat seine Änderungen in die DB committed</p>
COMMIT;	55	COMMIT;	Commit complete.		<p>Transaction read only wurde somit aufgelöst und</p> <p>T1 sieht den aktuellen stand der DB</p>
SELECT loc FROM dept WHERE deptno=10;	56		All rows fetched: 1 in 0.033 seconds LOC 1 NEW YORK		<p>Transaction read only wurde durch das commit aufgelöst und</p> <p>T1 sieht den aktuellen stand der DB</p>
COMMIT;	57		Commit complete.		
SELECT loc FROM dept WHERE deptno=10;	58		All rows fetched: 1 in 0.033 seconds LOC 1 LINZ		

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.
```