---- Nebenläufigkeit (Concurrency Control)

Prof. Dr. Ingo Claßen

Mittwoch, 16. Februar 2022 10:12

- Anforderungen an Transaktionen
- Mechanismen
- Isolationsstufen
- Snapshot Isolation
- Anomalien
- Postgres - Snapshot Isolation
- Postgres- Beispiele für Anomalien
- Postgres - weiter Beispiele

Anforderungen an Transaktionen (ACID)

Prof. Dr. Ingo Claßen

Dienstag, 8. Februar 2022 09:32

Atomarität (Atomicity)
- Alles oder Nichts
- Alle Aktionen einer Transaktion werden bestätigt oder keine

Konsistenz (Consistency)
- Eine Transaktion kann nur bestätigt werden, wenn keine Konsistenbedingungen verletzt sind
- Ist eigentlich ein Konzept der Anwendung

Isolation (Isolation)
- Nebenläufige/Parallele Transaktionen beinflussen sich nicht auf unerlaubte Art und Weise.
- Es gibt verschiedene Stufen der Isolation, deren stärkste Serialisierbarkeit ist
- Stärkere Stufe bedeutet geringere Leistung

Dauerhaftigkeit (Durability)
- Veränderungen bestätigter Transkationen haben für Folgetransaktionen Bestand
- Solange bis sie durch weitere Transaktionen bestätigt verändert werden
- Auch im Falle von Fehlern (z.B. Festplatte defekt)

Strikte 2-Phasen-Sperren (Strict two-phase locking , 2PL)

- Lesesperren (Shared Locks , Slocks) vor jedem Lesen
- Schreibsperren (Exclusive Locks, Xlocks) vor jedem Schreiben
- Sperren auf existierende Datensätze (Row Level Locks)
- Sperren basierend auf Prädikate (Predicate Locks) zur Verhinderung von Phantomen
- Sperren werden bis zum Transkationsende gehalten

Optimistic Concurrency Control (OCC)

- Lesen und Schreiben ohne Sperren (auf Kopien)
- Führen der Historie der Lese- und Schreibvorgänge
- Überprüfung der Historie auf Konflikte vor Commit
- Abbruch bei Konflikt

Multi-Version Concurrency Control (MVCC) / Snapshot Isolation

- Keine Lesesperren
- Änderungen erzeugen neue Versionen
- Transaktionen sehen Version der letzten bestätigten Änderung, d.h. Lesen und Schreiben stehen nicht in Konflikt
- Prüfung der Schreibvorgänge, Abbruch bei Konflikt

Optimistische Verfahren auf Anwendungsebene

- Explizite Versionsfelder
- Änderung nur, wenn keine Versionsänderung in Datenbank
- Bsp. OR-Mapper

SQL-Isolationsstufen

Read Uncommitted

- Lesen nichtbestätigter Änderungen möglich

Read Committed

- Lesen nur von bestätigten Änderungen möglich

Repetable Read

- Wiederholtes Lesen in der Transaktion liefert gleichen Wert

Serializable

- Nebenläufige Ausführung von Transkationen muss einer serialisierten Ausführung entsprechen

Produktspezifische Isolationsstufen

Cursor Stability (DB2, ..)

- Basiert auf Datenbank-Cursor
- Sperre bleibt solange bestehen, bis auf den nächsten Datensatz im Cursor weitergeschaltet wird

Snapshot Isolation (PostgreSQL, …)

- First Committer wins

Read Consistency (Oracle)

- First Updater wins

---- Anomalien

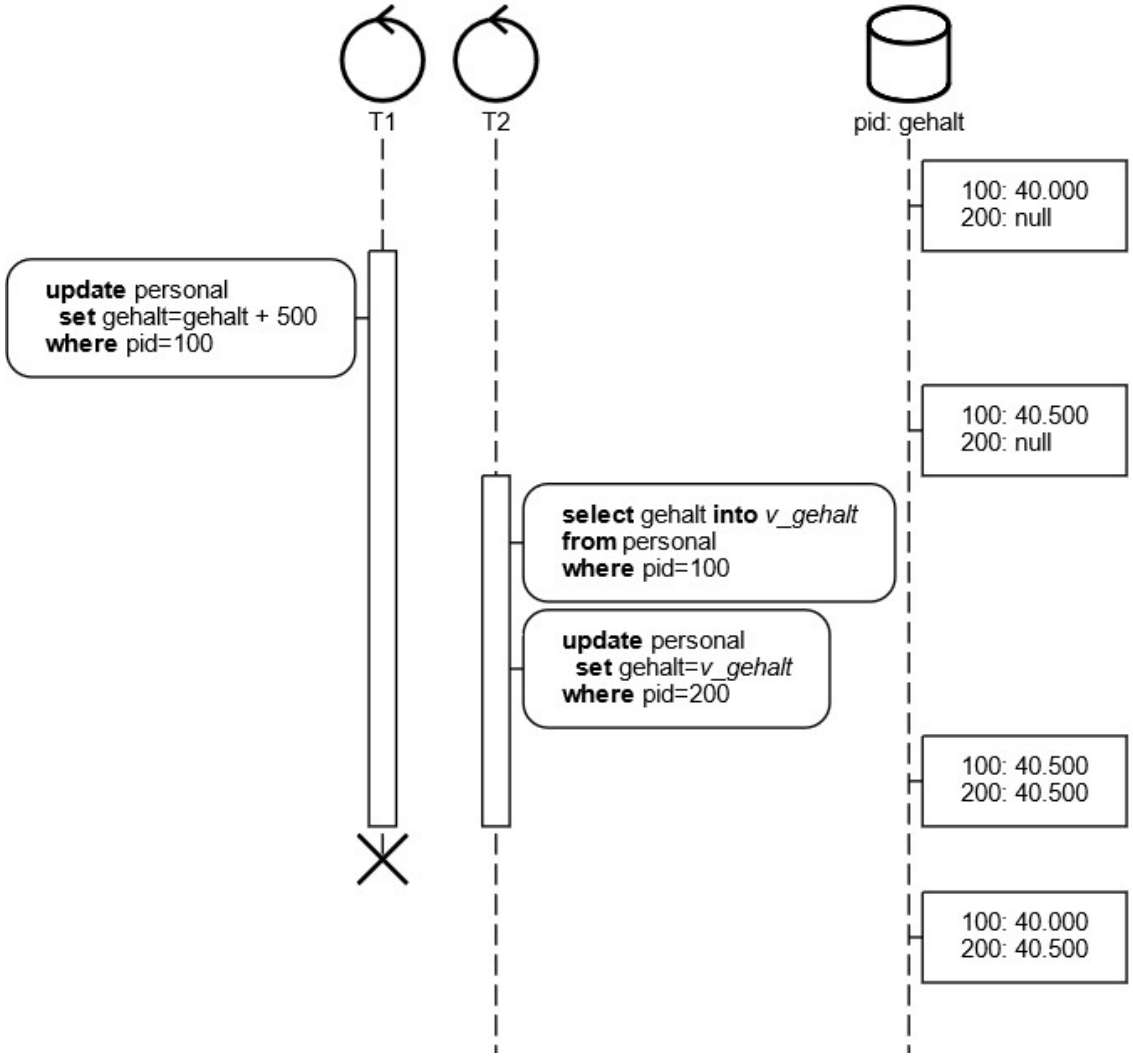
Prof. Dr. Ingo Claßen

Dienstag, 12. Juli 2022 17:23

Dirty Read

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18



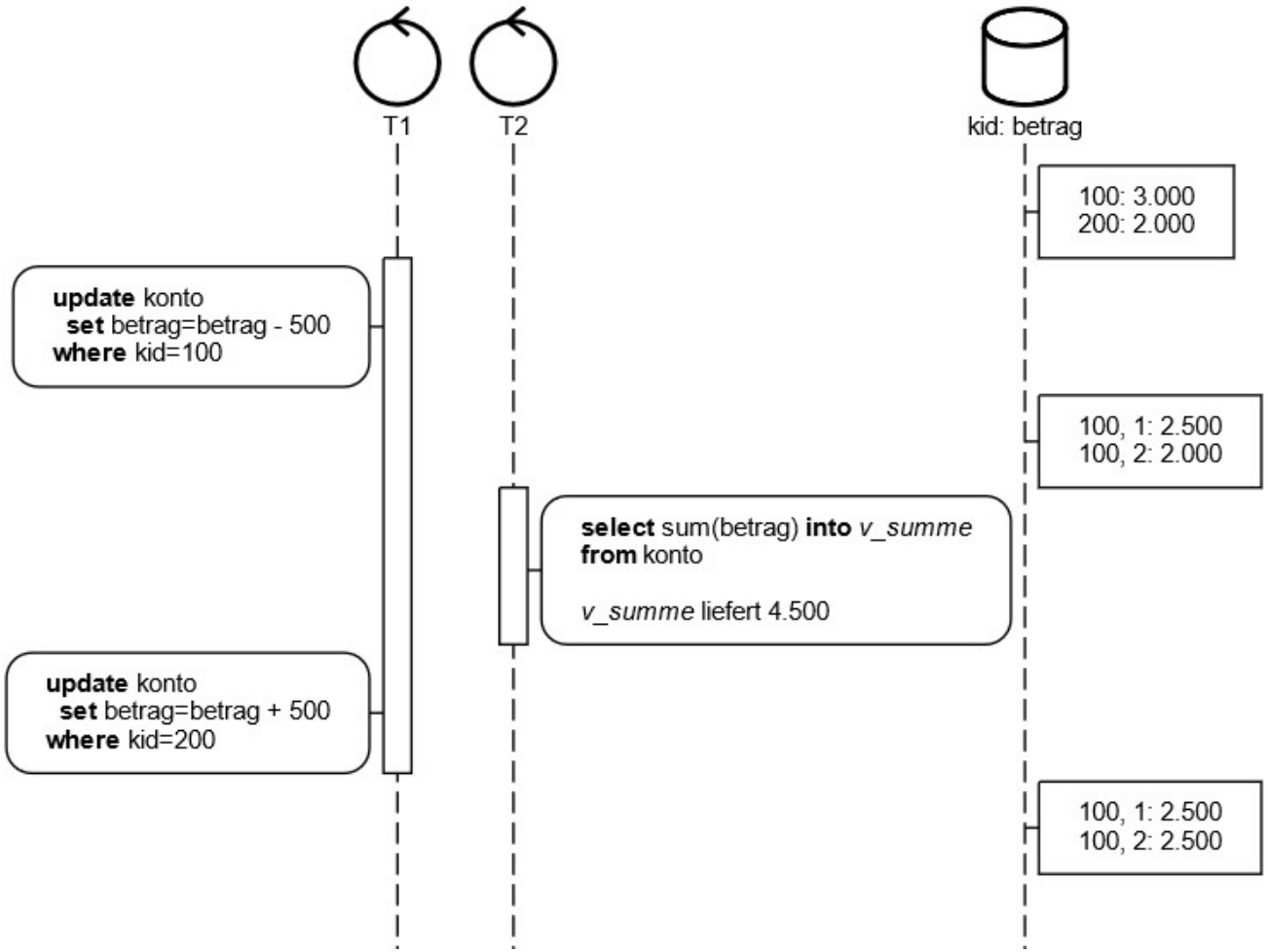
Mitarbeiterin mit pid=200 soll dasselbe Gehalt haben wie die mit pid=100

Lesen eines nicht bestätigten Datenbankzustands

Anomalie - Read Skew 1

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18

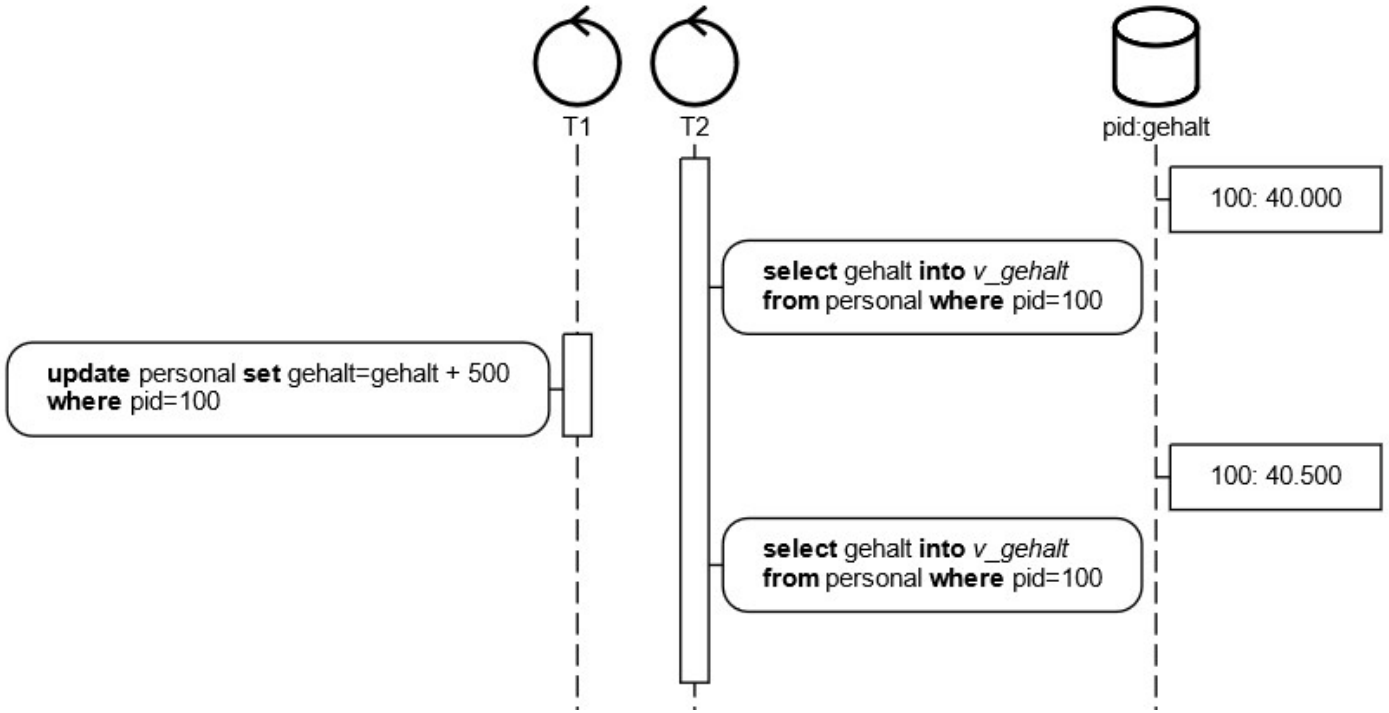


Lesen eines nicht bestätigten Datenbankzustands
Ähnlich wie Dirty Read, nur kein Rollback

Anomalie - Non Repeatable Read

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18

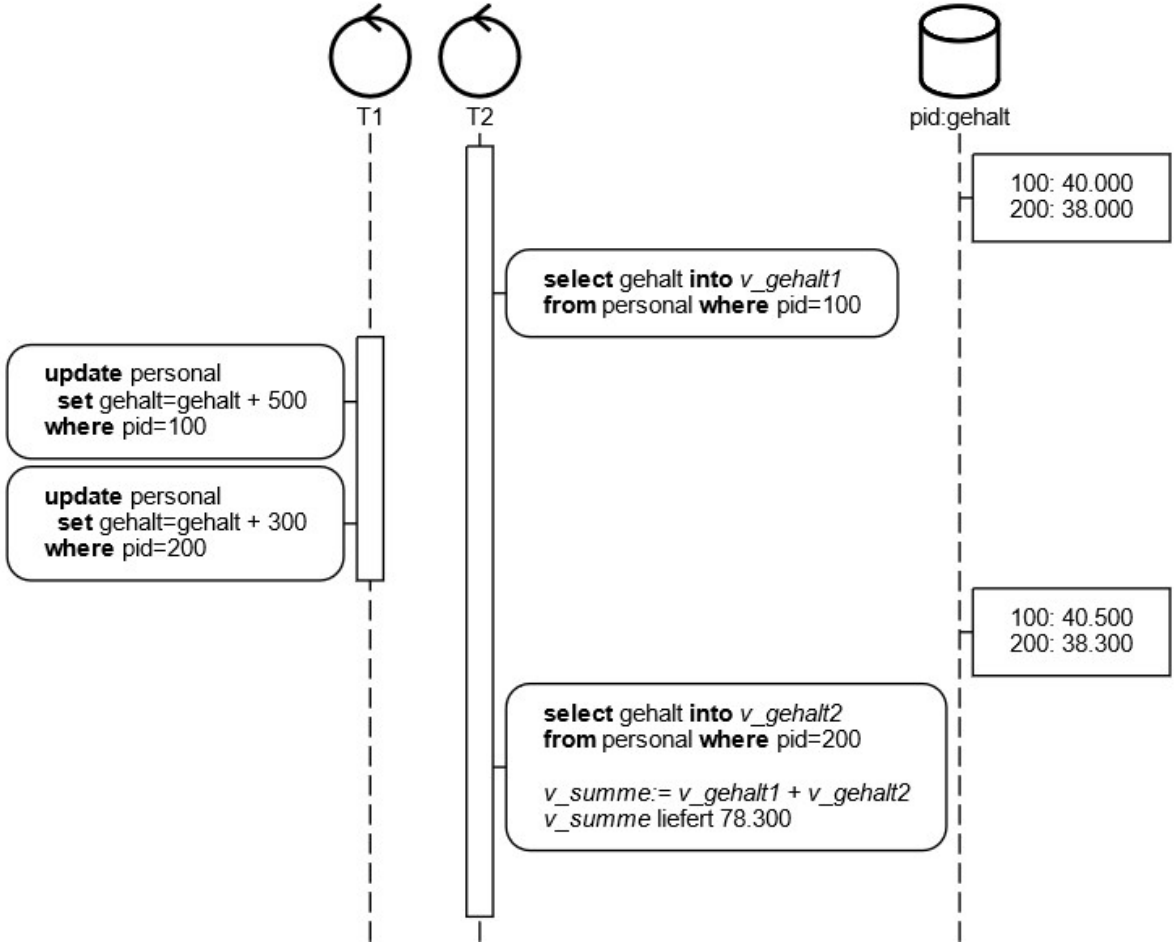


Wiederholtes Lesen des gleichen Datensatzes innerhalb einer Transktion führt zu unterschiedlichen Werten

Anomalie - Read Skew 2

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18

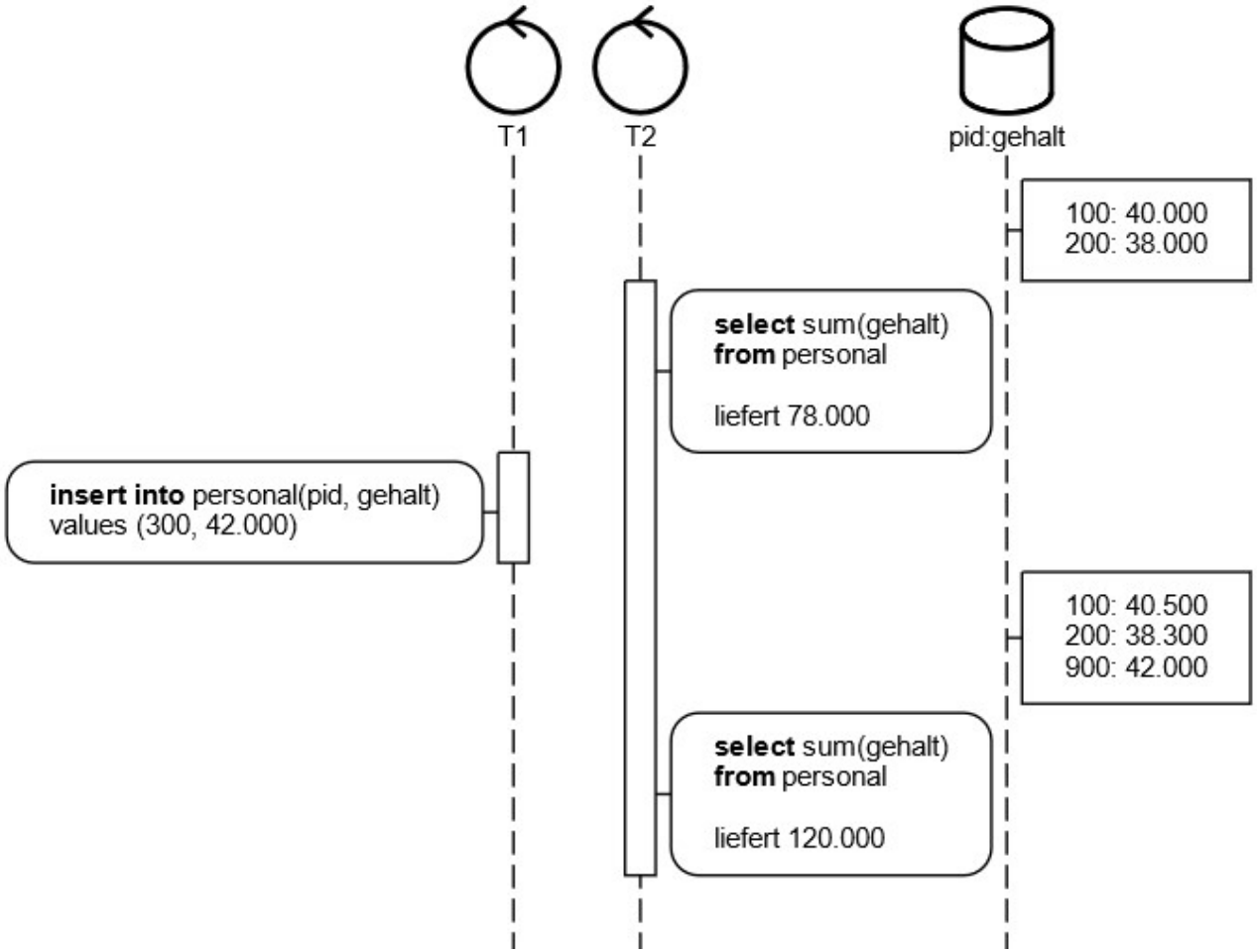


Ähnlich zu Non Repeatble Read
Nur nicht zweimaliges Lesen des gleichen Datensatzes
Der Gesamtzustand der Datenbank hat sich beim zweiten Lesen geändert

Anomalie - Phantom

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18

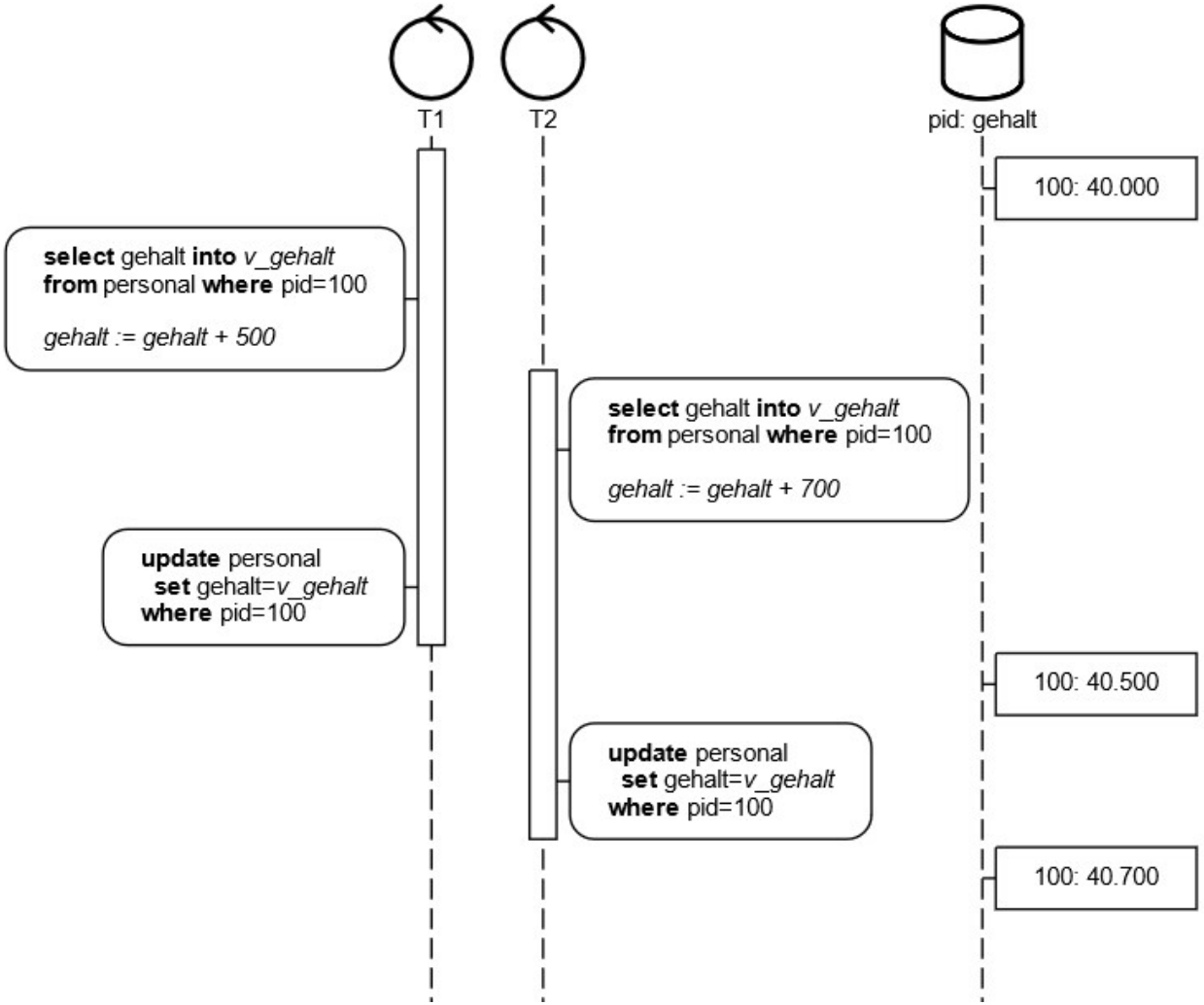


Ähnlich zu Read Skew 2
Der Gesamtzustand der Datenbank hat sich beim zweiten Lesen geändert

Anomalie - Lost Update

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18

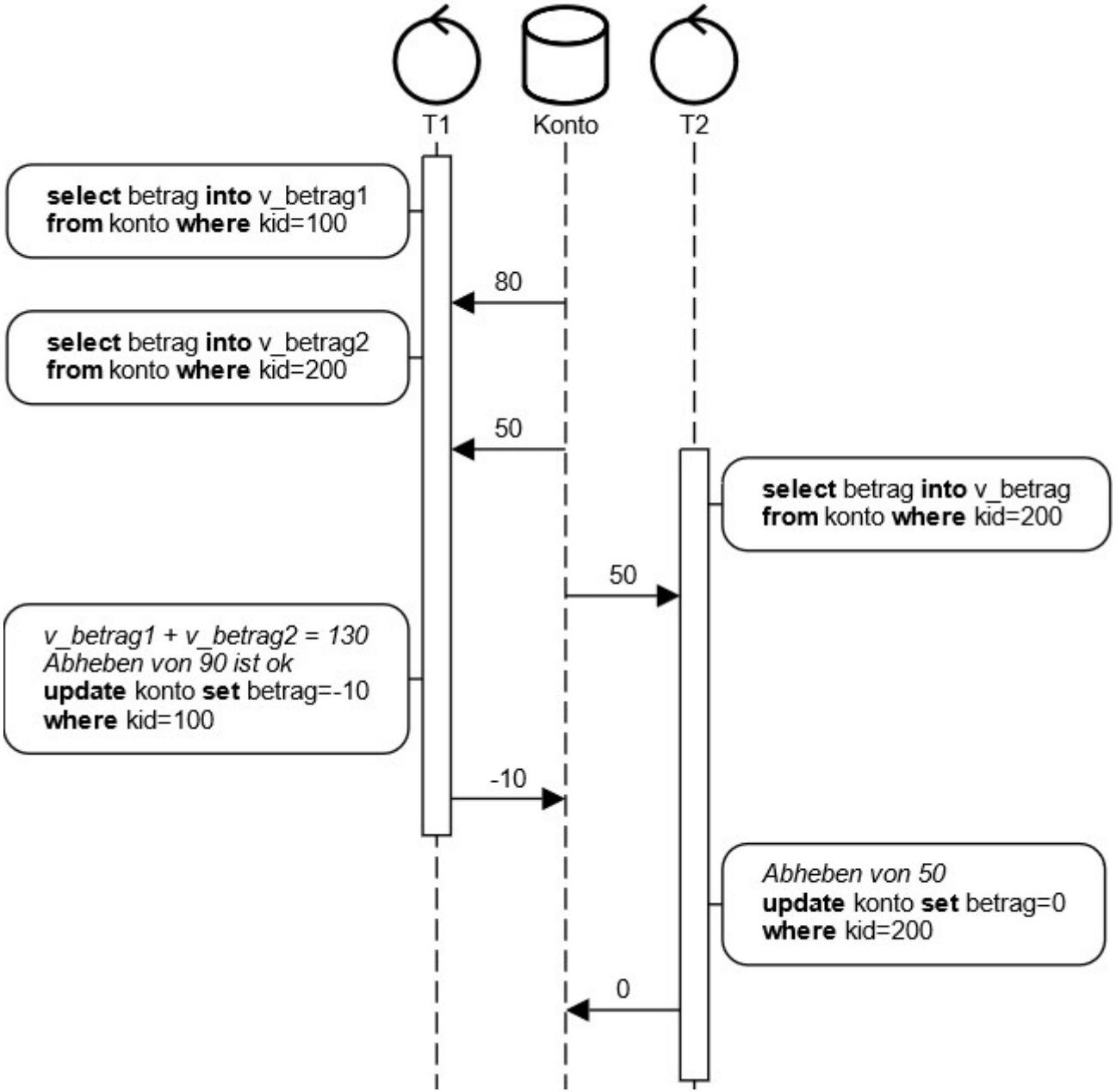


Unerlaubtes Überschreiben einer Änderung

Anomalie - Write Skew

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18



Invariante: Summe auf beiden Konten darf nicht negativ werden

T1 prüft die Summe (130) und stellt fest, dass ein Abheben von 90 von kid=100 ok ist, da Deckung durch kid=200 gegeben

T2 hebt ebenfalls Geld ab

Dadurch wird die Invariante verletzt

---- Postgres - Snapshot Isolation

Prof. Dr. Ingo Claßen

Mittwoch, 13. Juli 2022 07:39

Snapshot Isolation - Serializable

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

konto	kid|betrag| ---+------+ 100| 2| 200| 6|	T1	T2	T3	T
T1	begin transaction isolation level serializable; -- T1 update konto set betrag = 3 where kid=100; select * from konto order by kid; commit;	kid|betrag| ---+------+ 100| 3| 200| 6|			
T	begin transaction isolation level serializable; -- T				
T2	begin transaction isolation level serializable; -- T2 update konto set betrag = 7 where kid=200; select * from konto order by kid; commit;		kid|betrag| ---+------+ 100| 3| 200| 7|		
T3	begin transaction isolation level serializable; -- T3 insert into konto values (300, 1); select * from konto order by kid; commit;			kid|betrag| ---+------+ 100| 3| 200| 7| 300| 1|	
T	select * from konto order by kid; commit;				kid|betrag| ---+------+ 100| 3| 200| 6|

Snapshot Isolation - Read Committed 1

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

konto	kid|betrag| ---+------+ 100| 2| 200| 6|	T1	T2	T3	T
T1	begin transaction isolation level serializable; -- T1 update konto set betrag = 3 where kid=100; select * from konto order by kid; commit;	kid|betrag| ---+------+ 100| 3| 200| 6|			
T	begin transaction isolation level read committed; -- T				
T2	begin transaction isolation level serializable; -- T2 update konto set betrag = 7 where kid=200; select * from konto order by kid; commit;		kid|betrag| ---+------+ 100| 3| 200| 7|		
T3	begin transaction isolation level serializable; -- T3 insert into konto values (300, 1); select * from konto order by kid; commit;			kid|betrag| ---+------+ 100| 3| 200| 7| 300| 1|	
T	select * from konto order by kid; commit;				kid|betrag| ---+------+ 100| 3| 200| 7| 300| 1|

Snapshot Isolation - Read Committed 2

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

konto	kid|betrag| ---+------+ 100| 2| 200| 6|	T1	T2	T3	T
T1	begin transaction isolation level serializable; -- T1 update konto set betrag = 3 where kid=100; select * from konto order by kid; commit;	kid|betrag| ---+------+ 100| 3| 200| 6|			
T	begin transaction isolation level read committed; -- T				
T2	begin transaction isolation level serializable; -- T2 update konto set betrag = 7 where kid=200; select * from konto order by kid;		kid|betrag| ---+------+ 100| 3| 200| 7|		
T3	begin transaction isolation level serializable; -- T3 insert into konto values (300, 1); select * from konto order by kid;			kid|betrag| ---+------+ 100| 3| 200| 7| 300| 1|	
T	select * from konto order by kid; commit;				kid|betrag| ---+------+ 100| 3| 200| 6|
T2	commit;				
T3	commit;				

---- Postgres- Beispiele für Anomalien

Prof. Dr. Ingo Claßen

Montag, 28. Februar 2022 15:49

```
drop table konto;
create table konto (
    kid integer not null primary key,
    betrag integer not null
);
delete from konto;
insert into konto values (100, 0);
insert into konto values (200, 0);
select * from konto order by kid;
```

```
drop table personal;
create table personal (
    pid integer not null primary key,
    gehalt integer not null
);
delete from personal;
insert into personal values (100, 0);
insert into personal values (200, 0);
select * from personal order by pid;
```

Postgres - Read Skew 2 bei Read Committed

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

personal	pid|gehalt| ---+------+ 100| 40000| 200| 38000|	T1	T2
T2	begin transaction isolation level read committed; -- *T2* select * from personal where pid=100;		pid|gehalt| ---+------+ 100| 40000|
T1	begin transaction isolation level read committed; -- *T1* update personal set gehalt=gehalt + 500 where pid=100; update personal set gehalt=gehalt + 300 where pid=200; select * from personal order by pid; commit;	pid|gehalt| ---+------+ 100| 40500| 200| 38300|	
T2	select * from personal where pid=200; commit;		pid|gehalt| ---+------+ 200| 38300|

Postgres - Kein Read Skew 2 bei Repeatable Read

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

personal	pid|gehalt| ---+------+ 100| 40000| 200| 38000|	T1	T2
T2	begin transaction isolation level repeatable read; -- T2 select * from personal where pid=100;		pid|gehalt| ---+------+ 100| 40000|
T1	begin transaction isolation level read committed; -- T1 update personal set gehalt=gehalt + 500 where pid=100; update personal set gehalt=gehalt + 300 where pid=200; select * from personal order by pid; commit;	pid|gehalt| ---+------+ 100| 40500| 200| 38300|	
T2	select * from personal where pid=200; commit;		pid|gehalt| ---+------+ 200| 38000|

Postgres - Lost Update bei Read Committed

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

personal	pid|gehalt| ---+------+ 100| 40000|	T1	T2	T
T1	begin transaction isolation level read committed; -- T1 select * from personal where pid=100;	pid|gehalt| ---+------+ 100| 40000|		
T2	begin transaction isolation level read committed; -- T2 select * from personal where pid=100;		pid|gehalt| ---+------+ 100| 40000|	
T1	-- erhöhe gehalt auf 40500 update personal set gehalt=40500 where pid=100; select * from personal where pid=100; commit;	pid|gehalt| ---+------+ 100| 40500|		
T2	-- erhöhe gehalt auf 40700 update personal set gehalt=40700 where pid=100; select * from personal where pid=100; commit;		pid|gehalt| ---+------+ 100| 40700|	
T	begin transaction isolation level read committed; -- T select * from personal where pid=100; commit;			pid|gehalt| ---+------+ 100| 40700|
T				
T				

Postgres - Kein Lost Update bei Serializable

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

personal	pid|gehalt| ---+------+ 100| 40000|	T1	T2
T1	begin transaction isolation level serializable; -- T1 select * from personal where pid=100;	pid|gehalt| ---+------+ 100| 40000|	
T2	begin transaction isolation level serializable; -- T2 select * from personal where pid=100;		pid|gehalt| ---+------+ 100| 40000|
T1	-- erhöhe gehalt auf 40500 update personal set gehalt=40500 where pid=100; select * from personal where pid=100; commit;	pid|gehalt| ---+------+ 100| 40500|	
T2	-- erhöhe gehalt auf 40700 update personal set gehalt=40700 where pid=100; select * from personal where pid=100; commit;		Serialisierungsfehler

Postgres - Write Skew bei Read Committed

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

konto	kid|betrag| ---+------+ 100| 80| 200| 50|	T1	T2	T
T1	begin transaction isolation level read committed; -- *T1* select * from konto where kid=100;	kid|betrag| ---+------+ 100| 80|		
T1	select * from konto where kid=200;	kid|betrag| ---+------+ 200| 50|		
T2	begin transaction isolation level read committed; -- *T2* select * from konto where kid=200;		kid|betrag| ---+------+ 200| 50|	
T1	update konto set betrag = -10 where kid=100; select * from konto where kid=100; commit;	kid|betrag| ---+------+ 100| -10|		
T2	update konto set betrag = 0 where kid=200; select * from konto where kid=200; commit;		kid|betrag| ---+------+ 200| 0|	
T	begin transaction isolation level read committed; -- *T* select * from konto order by kid; commit;			kid|betrag| ---+------+ 100| -10| 200| 0|

Postgres - Kein Write Skew bei Serializable

Prof. Dr. Ingo Claßen

Montag, 11. Juli 2022 16:40

konto	kid|betrag| ---+------+ 100| 80| 200| 50|	T1	T2
T1	begin transaction isolation level serializable; -- *T1* select * from konto where kid=100;	kid|betrag| ---+------+ 100| 80|	
T1	select * from konto where kid=200;	kid|betrag| ---+------+ 200| 50|	
T2	begin transaction isolation level serializable; -- *T2* select * from konto where kid=200;		kid|betrag| ---+------+ 200| 50|
T1	update konto set betrag = -10 where kid=100; select * from konto where kid=100; commit;	kid|betrag| ---+------+ 100| -10|	
T2	update konto set betrag = 0 where kid=200; select * from konto where kid=200; commit;		Serialisierungsfehler

```
create table kv(k integer primary key, v varchar(10));

insert into kv values(10, '10-a');
```

Unterschiedliche Snapshots bei Read Committed und Repeatble Read

kv	k |v | --+----+ 10|10-a|	T1	T2	T3
T2	`begin transaction isolation level read committed; -- T2`			
T3	`begin transaction isolation level repeatable read; --T3`			

T1	`begin transaction isolation level read committed; -- T1`
	`update kv set v = '10-b' where k=10;`
	`select * from kv;`
	`commit;`
		k |v |
		--+----+
		10|10-b|		

T2	`select * from kv;` `commit;`		k |v | --+----+ 10|10-b|	
T3	`select * from kv;` `commit;`			k |v | --+----+ 10|10-a|

Postgres - Konflikt beim Schreiben

Prof. Dr. Ingo Claßen

Montag, 28. Februar 2022 15:50

kv	k |v | --+----+ 10|10-a|	T1	T2	T
T1	begin transaction isolation level serializable; -- *T1* update kv set v = '10-b' where k=10; select * from kv;	k |v | --+----+ 10|10-b|		
T2	begin transaction isolation level serializable; -- *T2* update kv set v = '10-c' where k=10;		wartet	
T1	commit;		Serialisierungsfehler	
T	begin transaction isolation level serializable; --*T* select * from kv; commit;			k |v | --+----+ 10|10-b|

kv	k |v | --+----+ 10|10-a|	T1	T2	T
T1	begin transaction isolation level read committed; --*T1* update kv set v = '10-b' where k=10; select * from kv;	k |v | --+----+ 10|10-b|		
T2	begin transaction isolation level read committed; --*T2* update kv set v = '10-c' where k=10;		wartet	
T1	commit;		läuft weiter	
T2	commit;			
T	begin transaction isolation level read committed; --*T* select * from kv; commit;			k |v | --+----+ 10|10-c|