

ACID

Das ACID-Prinzip beschreibt gewünschte Eigenschaften von Transaktionen in einer Datenbank. Eine Datenbanktransaktion ist eine Sequenz von Operationen, die einzigartig und durchführbar sein müssen.

Atomicity:

Eine Operation muss entweder ganz oder gar nicht ausgeführt werden. (Alles oder nichts!)

Consistency:

Man spricht von einer vorhandenen Datenkonsistenz, wenn nach einer Sequenz der Datenzustand in einem konsistenten Zustand zurückgelassen wird.

Isolation:

Die Isolation verhindert, dass sich parallele Abläufe auf Datei-Operationen gegenseitig beeinflussen können.

Durability:

Die Dauerhaftigkeit versichert eine dauerhafte Speicherung der Datei-Operationen auf einem Datenträger.

Diese Regeln sollten immer befolgt werden! Befinden sich Daten erstmal in einem redundanten Zustand, so ist die Reparatur nur schwer möglich.

Fehlerfälle

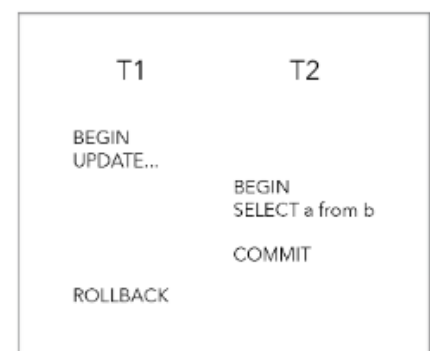
„Transaktionen geben eine Isolationsstufe an, mit der definiert wird, bis zu welchem Ausmaß eine Transaktion von Ressourcen- oder Datenänderungen isoliert sein muss, die von anderen Transaktionen durchgeführt werden. Die einzelnen Isolationsstufen werden dahingehend beschrieben, welche Parallelitätsnebenwirkungen (wie z. B. Dirty Reads oder Phantomlesezugriffe) zulässig sind.

Das Auswählen einer Isolationsstufe hat keine Auswirkungen auf die Sperren, die zum Schutz der Datenänderung eingerichtet werden!“

Isolationsstufe	Dirty Read	Non-Repeatable Read	Phantom Read
Read uncommitted	Possible	Possible	Possible
Read committed	Not possible	Possible	Possible
Repeatable Read	Not possible	Not possible	Possible
Serializable	Not possible	Not possible	Not possible

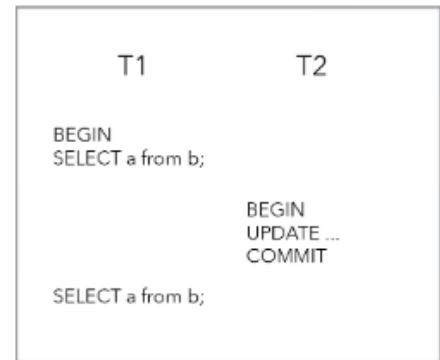
Dirty Read

Dirty Read (anders gesagt „unsauberes Lesen“) tritt auf, wenn eine Transaktion Daten aus einer Tabelle lesen darf, aber jedoch eine weitere Transaktion Daten in der Tabelle ändert, welche noch nicht committed (bestätigt) wurden.



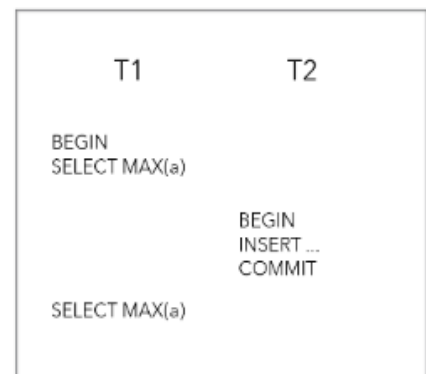
Non repeatable Read

Der non repeatable read tritt beim Auslesen von Daten auf. Die Ausgabe beim Auslesen eines Datensatzes ist nach einem Update (durch eine andere Transaktion) nicht ident, da sie verändert wurde.



Phantom Read

Eine Transaktion führt erneut eine Auslegung durch, welche ein Set von Spalten zurück liefert. Mit Hilfe einer Such-Operation wird das erforderliche Suchergebnis zurück geliefert (z.B.: MAX, MIN, SUM...)



Lost Update

Für ein Lost-Update benötigt man zwei oder mehrere Transaktionen. Transaktion A von z.B. 4 Transaktionen überschreibt den gleichen Datensatz, den Transaktion B geändert hat. Transaktion A überschreibt nun mit seinen neuen Daten für den Datensatz die Daten von Transaktion B und somit gehen die Daten von Transaktion B verloren. Diesen Verlust nennt man Lost-Update.

Sperren

Die meisten DBS unterstützen 2 Arten von Sperren:

- shared Locks (Lesesperren)
 - ermöglichen paralleles Lesen
 - verhindern Änderungen
- exclusive Locks (Schreibsperren)
 - erlauben nur dem Sperrinhaber zu lesen und zu ändern

Erklärung:

Stellen Sie sich vor ein lockable object ist eine Tafel (lockable) in einer Klasse mit einem Lehrer (writer) und einigen Schülern (reader).

Während der Lehrer etwas auf die Tafel schreibt (exclusive lock) ...

- 1) ... kann keiner der Schüler etwas von der Tafel lesen, da der Lehrer darauf schreibt und sie somit verdeckt.

Wenn ein Objekt exclusive gelockt ist, können shared locks dieses Objekt weder lesen noch erreichen.

- 2) ... können andere Lehrer auch nicht herkommen und etwas auf die Tafel schreiben, weil diese dann unleserlich wird und das verwirrt die Schüler.

Wenn ein Objekt exclusive gelockt ist können andere exclusive locks nicht darauf zugreifen.

Wenn alle Schüler das lesen was auf der Tafel steht (shared lock) ...

- 1) ... können alle Schüler die Information gleichzeitig lesen.

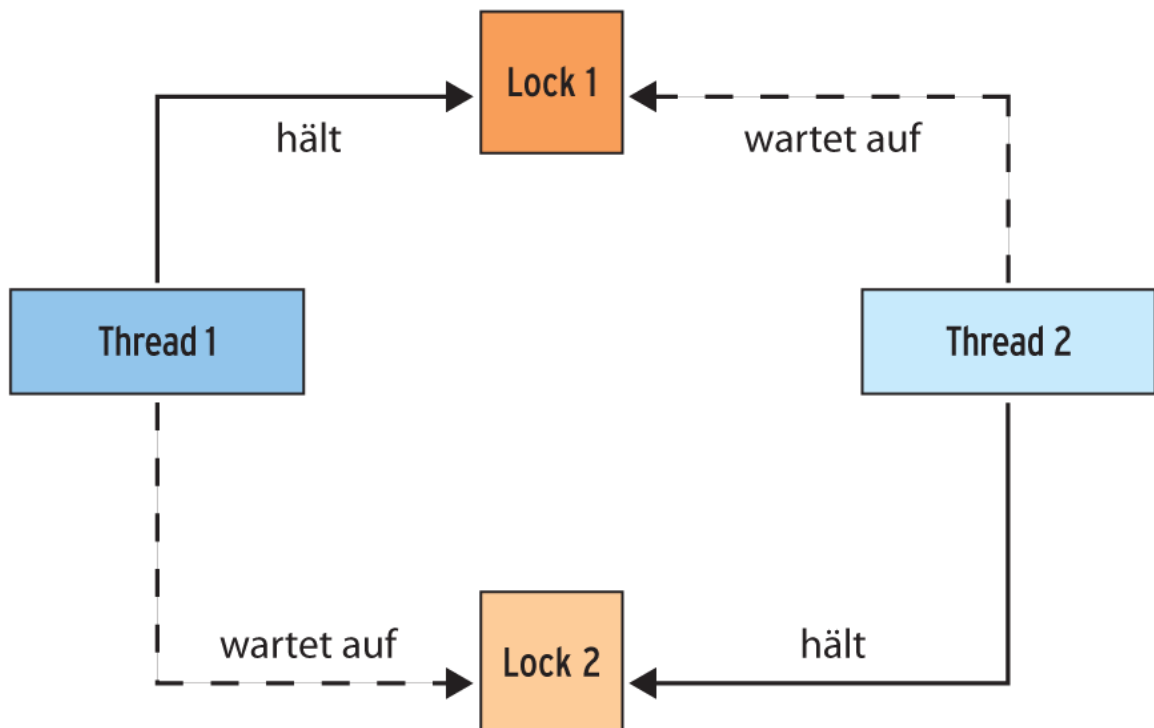
Mehrere shared locks können gleichzeitig ein Objekt auslesen.

- 2) ... der Lehrer wartet bis die Schüler fertig mit dem lesen sind, bevor er die Tafel löscht, damit er mehr darauf schreiben kann.

Wenn ein oder mehrere shared locks bereits existieren, kann der exclusive lock nicht das Objekt verändern.

Dead-Locks

Bei einem Dead-Lock wartet jede Transaktion auf eine Sperre, welche eine andere Transaktion hält. In solch einer Warteaktion können beliebig viele Transaktionen involviert sein.



Lösung: Rollback bei einer beliebigen Transaktion

Live-Locks

Dieser Lock tritt beispielsweise auf, wenn kleineren Prozesse höhere Prioritäten bei den Abarbeitungen gesetzt werden und diese dann zu oft auftreten. Die größeren, nicht abgearbeiteten Prozesse werden dabei nicht beachtet, wenn die größeren Prozesse länger als 60 Sekunden warten, bekommen sie einen Time-Out und werden anschließend gekillt.