Übung RT: Transaktionsmanagement

Gruppe 8: Lukas Arnold, Patrick Bucher, Christopher James Christensen, Jonas Kaiser, Melvin Werthmüller

Selbststudium

Frage 1

Wie ist der Begriff der Transaktion definiert?

- Eine Transaktion ist eine Folge von Operationen, die atomar, konsistent, isoliert und dauerhaft sein muss, sprich dem ACID-Prinzip folgt.
  - atomar (**A**tomicity): eine Transaktion wird immer als Ganzes, d.g. komplett oder gar nicht durchgeführt.
  - konsistent (**C**onsitency): eine Transaktion führt die Datenbank von einem konsistenten (= widerspruchsfreien) Zustand in einen anderen kostistenten Zustand.
  - isoliert (**I**solation): eine Transaktion muss im Mehrbenutzerbetrieb zu den gleichen Resultaten führen wie im Einbenutzerbetrieb; die Transaktionen verschiedener Benutzer dürfen keine ungewollte Seiteneffekte auf Transaktionen anderer Benutzer haben.
  - dauerhaft (**D**urability): die Veränderungen korrekt abgelaufener Transaktionen werden persistent abgespeichert; die Veränderungen gescheiterter Transaktionen wirken sich nicht auf den persistenten Speicher aus.

Frage 2

Erklären Sie das Prinzip der Serialisierbarkeit.

- Transaktionen sind serialisierbar, wenn sie in beliebiger Reihenfolge ausgeführt immer zum gleichen Ergebnis führen.
- Wird das ACID-Prinzip verletzt, ist die Serialisierbarkeit nicht mehr gegeben.

Frage 3

Was ist der Unterschied zwischen pessimistischen und optimistischen Verfahren der Serialisierung?

- Bei pessimistischer Serialisierung werden Objekte, die in einer Transaktion gelesen und/oder geschrieben werden sollen, vor der Transaktion gesperrt und nach der Abarbeitung wieder freigegeben.
  - Vorteil: Transaktionen lassen sich damit einfach serialisieren.

  - Nachteil: Es können Deadlocks entstehen; die Parallelität wird stark eingeschränkt.
- Bei optimistischer Serialisierung werden Objekte, die in einer Transaktion gelesen und/oder geschrieben werden sollen, vor der Transaktion in einen Zwischenspeicher eingelesen und dort manipuliert. Anschliessend wird validiert, ob die Änderungen nicht im Konflikt zu anderen Transaktionen stehen. Bei erfolgreicher Validierung werden die Werte schliesslich geschrieben.
  - Vorteil: Keine unnötigen Sperren; weniger Einschränkungen der Parallelität.
  - Nachteil: Aufwändige Validierung.

Frage 4

Was heisst Recovery? Welchen Zusammenhang hat es mit dem ACID-Prinzip?

- Recovery bezeichnet das Wiederherstellen eines korrekten Datenbankzustandes nach einem aufgetretenem Fehler.
- Bei einem Recovery müssen sämtliche Transaktionen bis zu einem bestimmten Zeitpunkt zurück rückgängig gemacht werden. Transaktionen sind aufgrund des ACID-Prinzips serialisierbar und können daher in umgekehrter Reihenfolge ihrer Ausführung wieder rückgängig gemacht werden. Dabei gehen zwar Änderungen verloren, die Datenbank verbleibt aber immer in einem konsistenten Zustand *zu einem gewissen Zeitpunkt*.

3. Read Uncommited

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
```

Frage 1

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 1 nach Schritt T1-A?

$900 + 2000 = 2900$

Frage 2

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-A?

$900 + 2000 = 2900$

Frage 3

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T1-B?

$900 + 2100 + 4000 = 7000$

Frage 4

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-B?

$900 + 2100 + 4000 = 7000$

Frage 5

Wo sehen Sie eine fachliche Inkonsistenz? Welcher der in Tabelle 1 dargestellten Fehler entstand dabei?

- Dirty read: Es sind Werte von Transaktionen ersichtlich, die noch nicht bestsätigt wurden (bei T2-A).
- Non-repeatable read: Die gleiche Abfrage innerhalb einer Transaktion ergibt verschiedene Saldi (bei T2A und T2B).
- Phantom read: In der Transaktion T2 taucht plötzlich eine neuer Eintrag mit dem Saldo 4000 auf (zwischen T2-A und T1-B).

4. Read Commited

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

Frage 1

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 1 nach Schritt T1-A?

$900 + 2000 = 2900$

Frage 2

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-A?

$1000 + 2000 = 3000$

Frage 3

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T1-B?

$900 + 2100 + 4000 = 7000$

Frage 4

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-B?

$900 + 2100 + 4000 = 7000$

Frage 5

Was ist dabei anders als bei `READ UNCOMMITED`? Erklären Sie.

- Im Gegensatz zu `READ UNCOMMITED` erscheinen nach Ausführung von T2-A noch die "alten" Werte, d.h. die Änderungen von T1-A werden noch nicht berücksichtigt, da die Transaktion noch nicht bestätigt wurde.

Frage 6

Gibt es immer noch Inkonsistenzen? Wenn ja, welche (gemäss Tabelle 1)?

- Der Non-repeatable read und der Phantom read treten immer noch auf.

5. Repeatable Read

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
```

Frage 1

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 1 nach Schritt T1-A?

$900 + 2000 = 2900$

Frage 2

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-A?

$1000 + 2000 = 3000$

Frage 3

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T1-B?

$1000 + 2000 + 3000$

Frage 4

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-B?

$1000 + 2000 + 3000$

Frage 5

Was ist anders als bei den vorherigen Isolation Levels? Erklären Sie.

- Nach T1-B sieht man im Fenster 2 immer noch die Ausgangslage.

Frage 6

Welche Inkonsistenz ist nun behoben worden?

- Der Non repeatable read tritt nicht mehr auf.

Frage 7

Warum ist die Transaktion immer noch nicht serialisierbar?

- Falls die in T2-A gelesenen Saldi für Berechnungen zwischengespeichert und später wieder in die Tabelle geschrieben werden, tritt eine Inkonsistenz auf.

6. Serializable

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

Frage 1

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 1 nach Schritt T1-A?

$900 + 2000 + 2900$

Frage 2

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-A?

- Man sieht erst einmal gar nichts.

Frage 3

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T1-B?

$900 + 2100 + 4000 + 7000$

Frage 4

Welche Konto-Saldi und welche Summe sehen Sie im Fenster 2 nach Schritt T2-B?

$900 + 2100 + 4000 + 7000$

Frage 5

Was ist anders als bei den vorherigen Isolation Levels? Erklären Sie. Inwiefern sind die beiden Transaktionen serialisierbar?

- Der `SELECT` in Fenster T2 "hängt" zunächst, da noch eine schreibende Transaktion auf der Tabelle aktiv ist. Der Wert kann erst gelesen werden, sobald die hängige Transaktion abgeschlossen oder zurückgenommen wurde.

7. Zusammenfassung

Fassen Sie ihre Erkenntnisse und die Unterschiede der vier Isolation Levels bei der Ausführung des Beispiels kurz zusammen.

- `READ UNCOMMITED`:
  - Verändert Transaktion 1 Werte in einer Tabelle, sieht Transaktion 2 diese Änderungen, bevor Transaktion 1 abgeschlossen ist.
  - Innerhalb von Transaktion 2 können sich Werte, die mehrmals hintereinander gelesen wurden, voneinander unterscheiden.
  - Es treten folgende Anomalien auf: Dirty read, Non repeatable read, Phantom Read
- `READ COMMITED`
  - Verändert Transaktion 1 Werte in einer Tabelle, sieht Transaktion 2 zunächst noch die alten Werte.
  - Erst nach Abschluss von Transaktion 1 sind deren Änderungen in Transaktion 2 sichtbar.
  - Es treten folgende Anomalien auf: Non repeatable read, Phantom
- `REPEATABLE READ`
  - Verändert Transaktion 1 Werte in einer Tabelle, sieht Transaktion 2 während ihrer ganzen Laufzeit, selbst wenn Transaktion 1 mittlerweile beendet wurde, noch die alten Werte.
  - Es tritt folgende Anomalie auf: Phantom

- `SERIALIZABLE`
  - Verändert Transaktion 1 Werte in einer Tabelle, kann Transaktion 2 nicht lesend auf diese Tabelle zugreifen, solange Transaktion 1 aktiv ist.
  - Sobald Transaktion 1 abgeschlossen wurde, sieht Transaktion 2 die neuen Werte.
  - Es treten keine Anomalien mehr auf.