

Isolation Level

Ergänzungen zum Skriptum

Einige Punkte zur Ergänzung von Kapitel 14.7 des Skriptums:

- Der Isolation-Level gibt an, welche der Phänomene, die beim *unkontrollierten Mehrbenutzerbetrieb* auftreten können, aus Performance-Gründen (verbesserte Parallelität) *bewusst* in Kauf genommen werden.
- Beim SQL-Server ist die Voreinstellung READ COMMITTED, d.h. die Phänomene *Nonrepeatable-Read* und *Phantom-Read* werden damit in Kauf genommen.
- Der Isolation-Level ist eine Session-Einstellung (*keine* datenbankweite Einstellung), d.h. jede Session (Verbindung zum Server) kann entscheiden, welche Phänomene zu Lasten oder Gunsten der Parallelität in Kauf genommen werden.
- Egal welcher Isolation-Level gewählt ist: beim *Schreiben* wird *immer* gesperrt.
- Der Isolation-Level ändert nur die Art und Weise, wie beim *Lesen* innerhalb der Session gesperrt wird:
 - READ UNCOMMITTED: die Transaktion holt sich vor dem Lesen keine Sperre.
 - READ COMMITTED: die Transaktion holt sich eine Sperre (S-Lock) auf die zu lesenden Datensätze, gibt die Sperren aber sofort wieder frei.
 - REPEATABLE READ: die Transaktion holt sich eine Sperre (S-Lock) auf die zu lesenden Datensätze und hält sie bis Transaktionsende (S2PL).
 - SERIALIZABLE: die Transaktion holt sich eine Sperre (S-Lock) auf die *gesamte Tabelle* und hält die Sperre bis Transaktionsende (S2PL).

Video

Watch the following video on YouTube:

- [MSSQL - Overview of the Isolation Level Videos](#)

Übung 1

- a) Erklären Sie in knapper Form für jedes der Phänomene *Dirty Read*, *Nonrepeatable-Read* und *Phantom-Read*, warum diese beim Isolation-Level READ UNCOMMITTED auftreten können.

Die Transaktion holt sich vor dem Lesen keine Sperre. D.h. im weiteren Verlauf kann es dazu kommen, dass die Werte von anderen Transaktionen verändert werden.

- b) Erklären Sie in knapper Form für jedes der Phänomene *Dirty Read*, *Nonrepeatable-Read* und *Phantom-Read*, warum diese beim Isolation-Level READ COMMITTED auftreten bzw. nicht auftreten können.

Die Transaktion holt sich vor dem Lesen eine Sperre. D.h. im weiteren Verlauf werden nur Daten gelesen die auch committed wurden. Dadurch wird das Dirty Read behoben. Die anderen bleiben bestehen, da zwischen Sperren committed werden kann.

- c) Erklären Sie in knapper Form für jedes der Phänomene *Dirty Read*, *Nonrepeatable-Read* und *Phantom-Read*, warum diese beim Isolation-Level REPEATABLE READ auftreten bzw. nicht auftreten können.

Repeatable Read wird behoben da sich die Datensätze bis zum Ende der Transaktion nicht mehr verändern können. Phantom Read bleibt bestehen, da neue Zeilen weiterhin eingefügt werden können.

- d) Erklären Sie in knapper Form für jedes der Phänomene *Dirty Read*, *Nonrepeatable-Read* und *Phantom-Read*, warum diese beim Isolation-Level SERIALIZABLE auftreten bzw. nicht auftreten können.

Phantom Read wird behoben da die ganze Tabelle bis zum Transaktionsende gesperrt wird.

- e) Geben Sie für alle fünf beim SQL Server möglichen Isolation-Levels die entsprechenden Kommandos an, die in einer Session diese Isolation-Level einstellen.

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;  
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
SET TRANSACTION ISOLATION LEVEL SNAPSHOT;  
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

- f) Mit welchen Kommandos können Sie beim SQL-Server abfragen, welcher Isolation-Level aktuell in Ihrer Session eingestellt ist? Geben Sie das Kommando samt Ausführungsergebnis an.

DBCC USEROPTIONS;

13 isolation level read committed

Übung 2

- Sehen Sie sich das Video [MSSQL - Understanding Isolation Level by Example \(Read Uncommitted\)](#) an.
- Kopieren Sie den Beispiel-Code aus dem Video, der sich bei YouTube unterhalb des Videos befindet (auf **MEHR ANSEHEN** klicken), in eine Datei.
- Führen Sie folgende Änderungen am Code durch:
 - Erstellen Sie eine Beispieldatenbank mit Ihrem Nachnamen.
 - Die Beispieltabelle soll ebenso statt SampleTable heißen.
 - Formatieren Sie den Code so, dass er unseren Codierungs-Richtlinien im DBI-Unterricht entspricht.
- Vollziehen Sie nun anhand dieses Beispiel-Codes alle Experimente, die in dem Video gezeigt werden, nach.
- Erstellen Sie einen Screen-Shot einer interessanten Phase Ihrer Experimente:
 - Layout der SQL Server Management Studio-Fenster wie im Video.
 - Auf dem Screen-Shot muss Ihr Nachname erkennbar sein (ist ja bereits durch Tabellennamen gewährleistet).
 - Das Bild muss im Format *.png sein und darf nicht mehr als 1 MB Speicherplatz benötigen.

Übung 3

- Sehen Sie sich das Video [MSSQL - Understanding Isolation Level by Example \(Serializable\)](#) an.
- Kopieren Sie den Beispiel-Code aus dem Video, der sich bei YouTube unterhalb des Videos befindet (auf **MEHR ANSEHEN** klicken), in eine Datei.
- Führen Sie folgende Änderungen am Code durch:
 - Erstellen Sie eine Beispieldatenbank mit Ihrem Nachname.
 - Die Beispieltabelle soll ebenso statt SampleTable heißen.
 - Formatieren Sie den Code so, dass er unseren Codierungs-Richtlinien im DBI-Unterricht entspricht.
 - Verwenden Sie bereits vorhandenen Code von Übung 2 wieder.
- Vollziehen Sie nun anhand dieses Beispiel-Codes alle Experimente, die in dem Video gezeigt werden, nach.
- Erstellen Sie folgende Screen-Shots:
 - Je einen Screen-Shot jeweils nach dem Test mit READ COMMITTED, REPEATABLE READ und SERIALIZABLE.
 - Das Layout der vier SQL Server Management Studio-Fenster muss wie im Video sein.
 - Auf jedem Screen-Shot muss Ihre Matrikelnummer erkennbar sein (bereits durch Tabellennamen gewährleistet).

DBI - Isolation Level

- Jedes Bild muss im Format *.png sein und darf nicht mehr als 1 MB Speicherplatz benötigen.