

Aufgabe 3

Bekommen in L4

Abgabe in L6

Bemerkungen.

1. Für alle Szenarien müsst ihr eine **Log-Datei** haben, um die durchgeführten Operationen wiederfinden zu können.
2. Um Fehler einfach zu finden, ist es empfohlen try-catch zu benutzen sowohl in der Anwendung, als auch in SQL.
3. Benutze **keine IDs als Input Parametern** für gespeichert Prozeduren und **überprüfe die Gültigkeit aller Parametern** (versuche Funktionen zu benutzen, wo nötig).
4. Ihr müsst **Testfälle** (test cases) vorbereiten sowohl für den „**Happy Flow**“, als auch für den „**Error Flow**“

Implementiere folgende Szenarien für eure Datenbank:

- A. erstelle eine gespeicherte Prozedur (stored procedure), die Daten in Tabellen einfügt, die in einer M:M Beziehung sind. Falls ein Teil der Operation abgebrochen wird (nicht erfolgreich ist), dann muss die ganze Prozedur zurückgestellt werden (rollback) **(2 Punkte)**

- B. erstelle eine gespeicherte Prozedur (stored procedure), die Daten in Tabellen einfügt, die in einer M:M Beziehung sind. Falls ein Teil der Operation abgebrochen wird (nicht erfolgreich ist), dann muss so viel wie möglich der Operation bewahrt werden.

z.B. Man versucht Studenten und Vorlesungen einzufügen. Wenn ein Student eingefügt wurde, und das Einfügen einer Vorlesung nicht erfolgreich ist, dann muss das Erstellen der Vorlesung zurückgestellt, aber nicht die Erstellung des Studenten. **(2 Punkte)**

- C. Schreibe vier Szenarien, um die Nebenläufigkeitsprobleme darzustellen:

- Dirty Reads
- Unrepeatable Reads
- Phantomprobleme
- Deadlock

Diese können mit Hilfe von gespeicherten Prozeduren oder Anfragen dargestellt werden.

Für jedes Szenario muss auch eine Lösung (workaround) gefunden werden. **(3 Punkte)**

- D. Schreibe eine .NET Anwendung, die mit Hilfe von Multithreading eine Deadlock Situation darstellt. Man muss zwei gespeicherte Prozeduren/Anfragen in verschiedenen Threads ausführen. Die Prozedur/Anfrage, die wegen dem Deadlock scheitert, muss wiederholt werden. Ihr könnt selber entscheiden wie viele Male ein Prozedur/Anfrage neu gestartet wird bis man entscheidet, dass diese fehlgeschlagen ist und abgebrochen wird. **(2 Punkte)**

Mehrere Informationen:

- Seminar 3 und Vorlesung 2
- <https://docs.microsoft.com/en-us/sql/t-sql/statements/set-transaction-isolation-level-transact-sql>
- <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/transactions-transact-sql>
- <http://www.albahari.com/threading/>