# Constraint 5: Geen concurrency problemen

Hieronder wordt er gekeken naar een constraint waarbij geen concurrency problemen op kunnen treden. Deze constraint voert een enkele update uit, en het is niet mogelijk dat er tijdens deze update een tweede connectie iets aan kan passen. Ondanks dat er dus twee keer dezelfde select statement wordt uitgevoerd, zal een andere connectie geen aanpassingen kunnen aanbrengen.

|  |  |
| --- | --- |
| Connection 1 | Connection 2 |
| SET TRANSACTION ISOLATION LEVEL  READ COMMITED |  |
| BEGIN TRANSACTION |  |
| (initialy)  insert into reg values (1111, 'RGDEV', '2004-09-03', 4) |  |
| (trigger)  UPDATE offr  SET status = 'CONF'  WHERE course IN (  SELECT O.course  FROM offr O JOIN reg R ON O.course = R.course AND O.starts = R.starts  WHERE O.status <> 'CONF'  GROUP BY O.course, O.starts  HAVING COUNT(\*) >= 6  ) AND starts IN (  SELECT O.starts  FROM offr O JOIN reg R ON O.course = R.course AND O.starts = R.starts  WHERE O.status <> 'CONF'  GROUP BY O.course, O.starts  HAVING COUNT(\*) >= 6  )  X-Lock requested and granted until end of transaction |  |
| COMMIT TRANSACTION |  |

Nu blijkt dat volgens verschillende bronnen, nieuwere versies van SQL Server queries kunnen onderscheppen door het principe van: interleaving. Dit kan mogelijk tot problemen zorgen binnen de uitvoering van de select statements in deze query, maar aangezien dit buiten de course valt moet dit voor de huidige situatie geen probleem zijn. Er is meer over te lezen in een artikel van Joseph Sack (2017), of van Dmitry Piliugin (2018).

# Constraint 11: Non Repeatable Read

Hieronder wordt een situatie geschetst waarin het voor kan komen dat een employee een course gaat geven terwijl deze geen trainer is, ondanks dat er gecontroleerd wordt of dat deze employee wel een trainer is. Voor deze situatie bestaat er een employee met employee nummer 1111, en job TRAINER.

|  |  |  |
| --- | --- | --- |
| Connection 1 | Connection 2 | Employee job value |
| SET TRANSACTION ISOLATION LEVEL  READ COMMITED |  |  |
| BEGIN TRANSACTION |  |  |
| (initialy)  exec usp\_InsertOffering  'J2EE', '2050-10-10', 'CONF', 6, 1111, 'Amsterdam' |  |  |
| (stored procedure)  SELECT \* FROM emp WHERE job <> 'TRAINER'  AND empno = @trainer  S-Lock requested and granted until end of select |  | Employee 1111  Is een TRAINER |
|  | **update emp set job = 'ADMIN' where empno = 1111**  X-lock request and granted until auto commit | Employee 1111  Is een ADMIN |
| SELECT 1 FROM emp e WHERE e.empno = @trainer AND ( DATEDIFF( YEAR, hired, @starts ) < 1 )  S-Lock requested and granted until end of select |  | Employee 1111  Is een ADMIN |
| SELECT 1 FROM emp e INNER JOIN reg r ON e.empno = r.stud WHERE e.empno = @trainer AND r.course = @course  S-Lock requested and granted end of select |  | Employee 1111  Is een ADMIN |
| INSERT INTO offr VALUES ( @course, @starts, @status, @maxcap, @trainer, @loc ) |  | Employee 1111  Is een ADMIN |
| COMMIT TRANSACTION |  |  |

Nadat deze transactie is uitgevoerd, is er een course offering waarbij de employee een ADMIN is, en geen TRAINER terwijl dit wel de bedoeling is volgens de constraint. Dit kan dus gebeuren wanneer er na de select controle een update gedaan wordt op dezelfde employee.

Deze situatie kan opgelost worden door de transaction isolation level te zetten naar Serializable. Wanneer dit gedaan wordt zal connectie 2 wachten tot connectie 1 zijn transactie heeft voltooid. Dit is te zien in de volgende tabel.

|  |  |  |
| --- | --- | --- |
| Connection 1 | Connection 2 | Employee job value |
| SET TRANSACTION ISOLATION LEVEL  SERIALIZABLE |  |  |
| BEGIN TRANSACTION |  |  |
| (initialy)  exec usp\_InsertOffering  'J2EE', '2050-10-10', 'CONF', 6, 1111, 'Amsterdam' |  |  |
| (stored procedure)  SELECT \* FROM emp WHERE job <> 'TRAINER'  AND empno = @trainer  S-Lock requested and granted on empno 1111 until end of transaction |  | Employee 1111  Is een TRAINER |
|  | **update emp set job = 'ADMIN' where empno = 1111**  X-lock request and not granted due to serializable isolation level in connection 1. X-lock will be granted when connection 1 has completed it’s transaction. | Employee 1111  Is een TRAINER |
| … |  |  |

Deze situatie is niet op te lossen wanneer de isolation level op repeatable read wordt gezet, aangezien de employee (1111) door de eerste select niet gevonden wordt en dus ook niet vergrendeld zal worden door de isolation level. Dit zal dus betekenen dat connectie 2 nog steeds de mogelijkheid heeft om de job van deze employee aan te passen.

# Literatuurlijst

Sack, J. (2017, 19 April). Introducing Interleaved Execution for Multi-Statement Table Valued Functions. Geraadpleegd van: <https://blogs.msdn.microsoft.com/sqlserverstorageengine/2017/04/19/introducing-interleaved-execution-for-multi-statement-table-valued-functions/>

Piliugin, D. (2018, 27 April). SQL Server 2017: Interleaved Execution for mTVF. Geraadpleegd van: <https://www.sqlshack.com/sql-server-2017-interleaved-execution-for-mtvf/>