Report #2

# Problem description

In this case the problem was to create simple application which simulates real traffic in database. In given programming language. I chose C# for this. App should call only stored procedures on database.

Application repository

[**https://github.com/mpxx24/database\_programming\_2**](https://github.com/mpxx24/database_programming_2)

And here is the file with simple CRUD methods (so you can see that only stored procedures were used)

[**https://github.com/mpxx24/database\_programming\_2/blob/master/DatabaseProgramming2-report2/Helper.cs**](https://github.com/mpxx24/database_programming_2/blob/master/DatabaseProgramming2-report2/Helper.cs)

# Application

Application consists of two parts. First is Main() method in which the login happens, and a Helper class which contains four methods

* UpdateFirstRow(int number)
* InsertData(int number)
* DeleteData(int howMany)
* SelectData(int howMany)

I think that method names are clear enough so I don’t have to describe what each method do.

# Tests

Test were running with 678 ‘users’ (threads) at the same time trying to execute given scenario.

**Scenario #1**

**Description:** 678 users trying to update first table row with different data at the same time

**What happened:** Everything worked fine. Data updated each time I tried to execute program

**Scenario #2**

**Description:** 678 users trying to insert data to the table at the same time

**What happened:** Everything worked fine. Data was added each time. No errors occurred

**Scenario #3**

**Description:** 678 users trying to delete last 678 rows from the table

**What happened:** In most cases it worked fine. But sometimes application threw an error

*"Transaction (Process ID 61) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction."*

After this all data from table were deleted.

**Scenario #4**

**Description:** 678 users trying to select 678 rows from table

**What happened:** Everything worked fine. I didn’t return any values to console, but didn’t throw any errors.

**Scenario #5**

**Description:** 678 users running all scenarios at the same time

**What Happened:** In all cases application stopped working and threw and error.

Eg.

*"Transaction (Process ID 54) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction."*

Data from table was deleted.

*Note: during tests, deadlock was the only error that occurred, I didn’t have any problem with too many connections or timeouts. Even after increasing users number to 1000 or 1500*

*Note 2: I’m not really sure how many threads my machine could handle, and how many were actually working properly*

# Solution

**Locks**

In C# while using lock keyword we are making sure that we allow one method (for example DeleteData(int howMany)) to be called only once at the same time. Regardless of the number of threads.

What it means is that in my example with 678 users (threads) they would run 1 at a time. Making sure that for example two threads do not try to delete same row at the same time.