**4.5.** **Triggers**

One of the main focuses in the development of this project was the database and information management performed in it. Since we chose to work on relation database throughout the extent of this project we wanted to use as many of the features provided by SQL Server. Triggers being one of the most important and used procedure, were a necessity to be implemented.

**4.5.1.** **What are triggers**

Triggers in relational databases are special [stored procedures](http://www.sqlservertutorial.net/sql-server-stored-procedures/) that are executed automatically in response to event occurring in the database server.

SQL Server provides three type of triggers:

1. Data manipulation language (DML) triggers which are invoked automatically in response to events against tables or views ([INSERT](http://www.sqlservertutorial.net/sql-server-basics/sql-server-insert/), [UPDATE](http://www.sqlservertutorial.net/sql-server-basics/sql-server-update/), and [DELETE](http://www.sqlservertutorial.net/sql-server-basics/sql-server-delete/)). They fire when any valid event fires, whether table rows are affected or not.
2. Data definition language (DDL) triggers fire in response to statements changing the data behaviour ([CREATE](http://www.sqlservertutorial.net/sql-server-basics/sql-server-create-table/), ALTER, and [DROP](http://www.sqlservertutorial.net/sql-server-basics/sql-server-drop-table/)) and certain system stored procedures that perform DDL-like operations.
3. Logon triggers fire when a user's session is being established.

**4.5.2.** **How use of triggers was chosen**

The use of triggers was determined based on the requirements provided by the GTL. Although higher number and different types of triggers could have been used when developing solution, we decided to focus more on basic requirements of the library system, thus growing our project slowly with future development and usability in mind. Furthermore, it was not determined that more extensive use of triggers was not requested or information necessary to create trigger was not provided.

**4.5.3.** **What triggers were used for**

In solution provided alongside this document it was decided to implement only one trigger, that being for inserting new entry into “loan” table. We chose to implement this trigger since we wanted to be sure that the book being borrowed would be available for borrowing (book not being loaned out already) and so that member would not be able to borrow more than the allowed limit of materials. If either of the two conditions were violated, then the transaction would be rolled back and no changes to the database would be made. However, if none of the previously mentioned rules were broken the entry would be saved in the database updating “FromDate” (See Figure 1), so that we ensure wrong date, which user could have entered to modify the expected results, would not be saved.

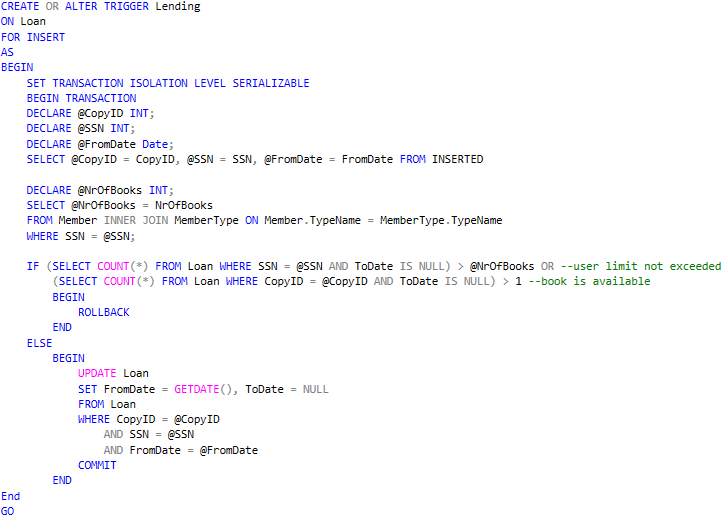


Figure 1

Used:

<https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-2017>

<http://www.sqlservertutorial.net/sql-server-triggers/>