Phần 1: **Mức ưu tiên có sẵn trong cơ sở dữ liệu**

Exercise 1: Acquire Locks by Using the Read Committed Isolation Level

1. Open Microsoft SQL Server Management Studio, and connect to an instance of SQL Server 2005.

2. In a new query window, which will be referred to as Connection 1, type and execute the following SQL statements to create the TestDB database, the Test schema, and the table that you will use in this exercise:

Graphical user interface, table

Description automatically generated with medium confidence

3. Open another query window, which will be referred to as Connection 2, and type and execute the following SQL statement to prepare the connection

Graphical user interface, text, application, email

Description automatically generated

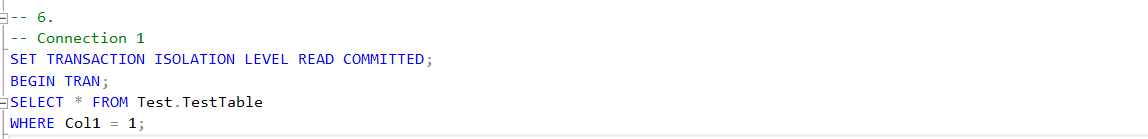
4. Open a third query window, which will be referred to as Connection 3, and type and execute the following SQL statement to prepare the connection

5. In Connection 1, execute the following SQL statements to start a transaction in the read committed transaction isolation level, and read a row from the test table (but do not commit the transaction!).

Graphical user interface, text, application

Description automatically generated

6. To see which locks have been acquired by the transaction in Connection 1, open Connection 3, and execute the following SELECT statement. In the line of code that contains @@SPID of Connection 1>, be sure to replace this with the ID value returned by the code executed in step 2 of this exercise.



Trả lời: dữ liệu trả về là dữ liệu trước khi được Update bởi giao dịch tạo ra bởi Connection 2

7. In Connection 1, execute the following SQL statement to end the started transaction

Graphical user interface, text, application, Word

Description automatically generated

8. In Connection 2, execute the following SQL statements to start a transaction, and acquire an exclusive lock on one row in the test table.

Graphical user interface, text, application

Description automatically generated

9. In Connection 1, execute the following transaction to try to read the row that has been updated (but not committed) by Connection 2. After you execute the code in this step, move on to the next step, as this connection will now be blocked

Graphical user interface, text, application

Description automatically generated

10. To see which locks have been acquired by the transaction in Connection 1, open Connection 3, and execute the following SELECT statement. In the line of code that contains @@SPID of Connection 1, be sure to replace this with the ID value returned by the code executed in step 2 of this exercise.



=> dữ liệu trả về là dữ liệu đã được thay đổi bởi giao dịch của Connection2

Câu lệnh select đầu tiên ở Connection 1 không trả về dirty reads do giao dịch tạo bởi Connection 2 chưa được commit.

11. In Connection 2, execute the following SQL statements to end the transaction started earlier

Graphical user interface, text, application, email

Description automatically generated

12. Now, first have a look in Connection 1 and note that the SELECT statement has been completed. Switch to Connection 3, and execute its SELECT statement again to see which locks are now acquired by the transaction in Connection 1. In the line of code that contains @@SPID of Connection 1>, be sure to replace this with the ID value returned by the code executed in step 2 of this exercise.

13. Close the three query windows for Connections 1, 2, and 3. Open a new query window, and execute the following SQL statement to clean up after this exercise:

Graphical user interface, application, Word

Description automatically generated