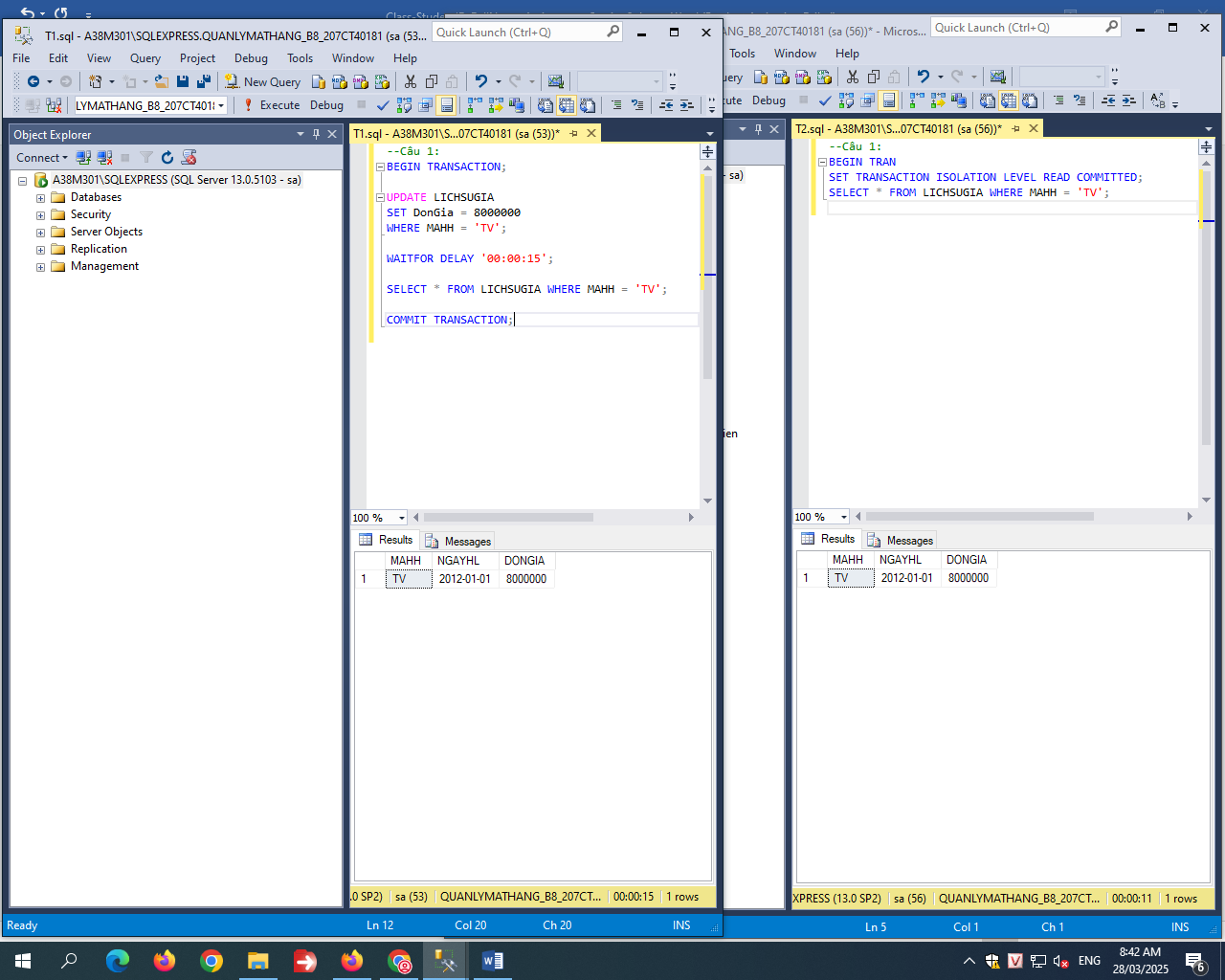
**PRACTICE - ASSIGNMENT 8  
TRANSACTION & ISOLATION**

**Question 1:**

**Result:**

Khi query 1 chạy, nó cập nhật giá trị DONGIA của mặt hàng 'TV' thành 8000000. Trong thời gian 15 giây chờ, query 2 chạy. Vì Read Committed chỉ cho phép đọc dữ liệu đã commit, nên query 2 sẽ thấy giá trị DONGIA ban đầu (trước khi query 1 commit). Sau khi query 1 commit, nếu query 2 chạy lại, nó sẽ thấy giá trị DONGIA mới (8000000).

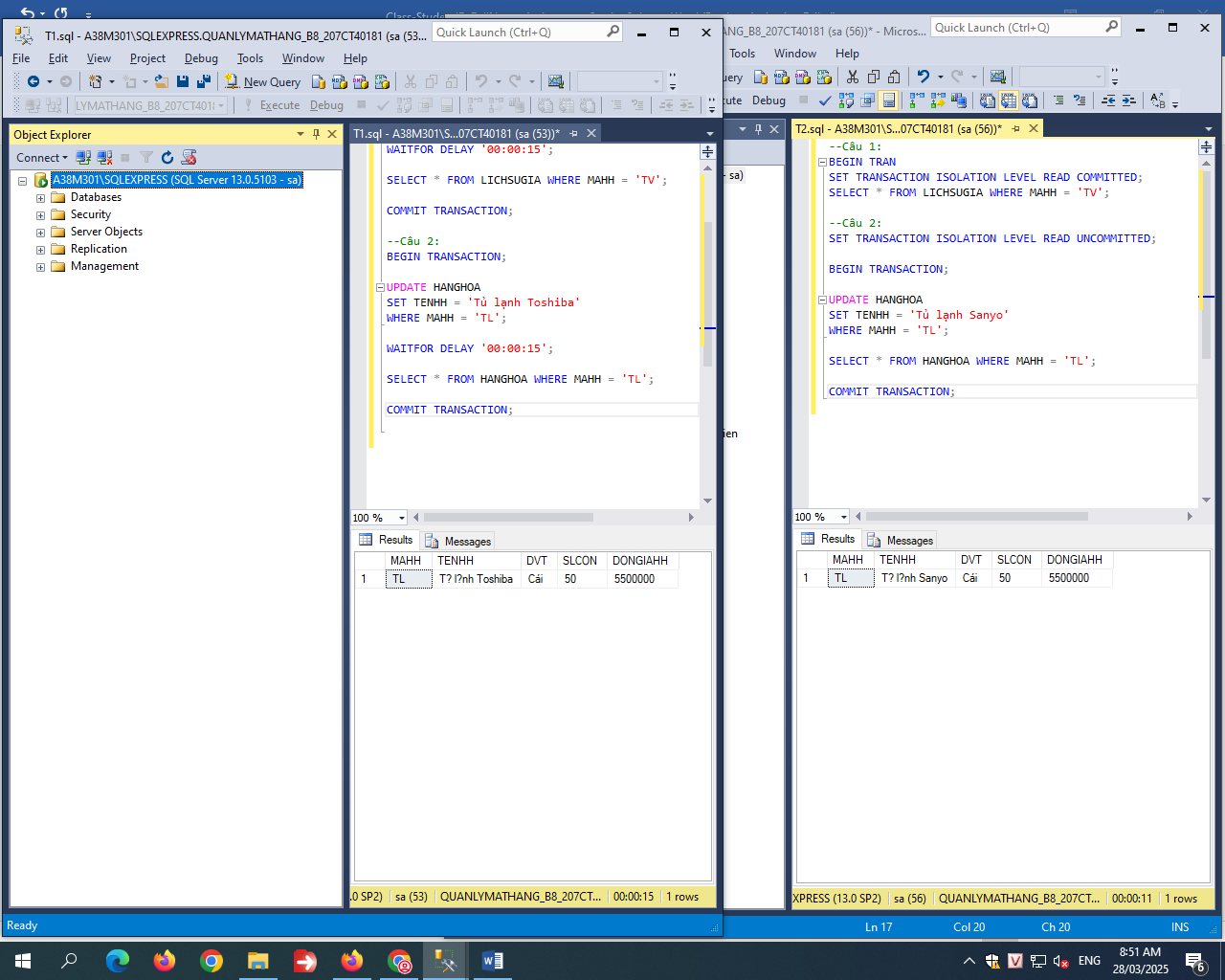
Kết quả: Query 2 sẽ không thấy dữ liệu chưa commit của query 1, do đó vấn đề Dirty Read không xảy ra trong trường hợp này.

**Picture:** 

**Question 2:**

**Result:**

Query 1 cập nhật tên hàng hóa của 'TL' thành 'Tủ lạnh Toshiba' và chờ 15 giây. Query 2 cập nhật tên hàng hóa của 'TL' thành 'Tủ lạnh Sanyo' và commit ngay lập tức. Vì Read Uncommitted cho phép đọc dữ liệu chưa commit, nếu Query 2 chạy trong khoảng 15 giây chờ của Query 1, nó sẽ thấy tên hàng hóa là "Tủ lạnh Toshiba", sau đó nó cập nhật thành "Tủ lạnh Sanyo". Sau khi Query 1 commit, giá trị cuối cùng của TENHH trong bảng HANGHOA sẽ là 'Tủ lạnh Toshiba'. Điều này dẫn đến vấn đề Lost Update, vì thay đổi của Query 2 đã bị mất.

**Picture:** 

**Question 3:**

**Result:**

Query 1 bắt đầu giao dịch với mức cô lập Serializable, đọc toàn bộ bảng HANGHOA, chờ 15 giây, và đọc lại bảng HANGHOA. Serializable sẽ đảm bảo rằng trong suốt quá trình giao dịch của Query 1, không có giao dịch nào khác có thể thay đổi dữ liệu mà nó đã đọc. Vì vậy, nếu Query 2 chạy trong thời gian chờ của Query 1, nó sẽ bị block cho đến khi Query 1 commit hoặc rollback. Do đó, Query 1 sẽ thấy cùng một tập kết quả trong cả hai lần SELECT, và vấn đề Phantom (thêm hàng mới) sẽ không xảy ra.

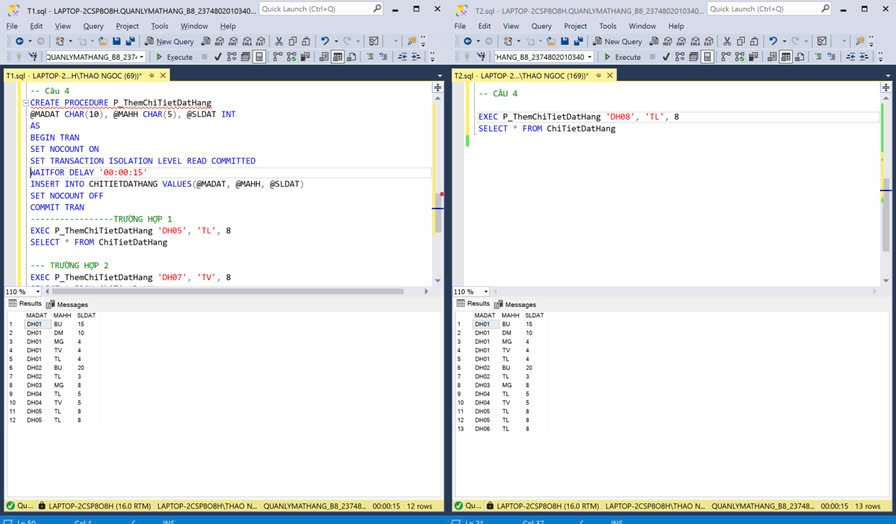
Kết quả: Query 2 phải chờ Query 1 thực hiện song, nó mới có thể thực hiện được việc thêm dữ liệu vào bảng.

**Picture:**

**Question 4:**

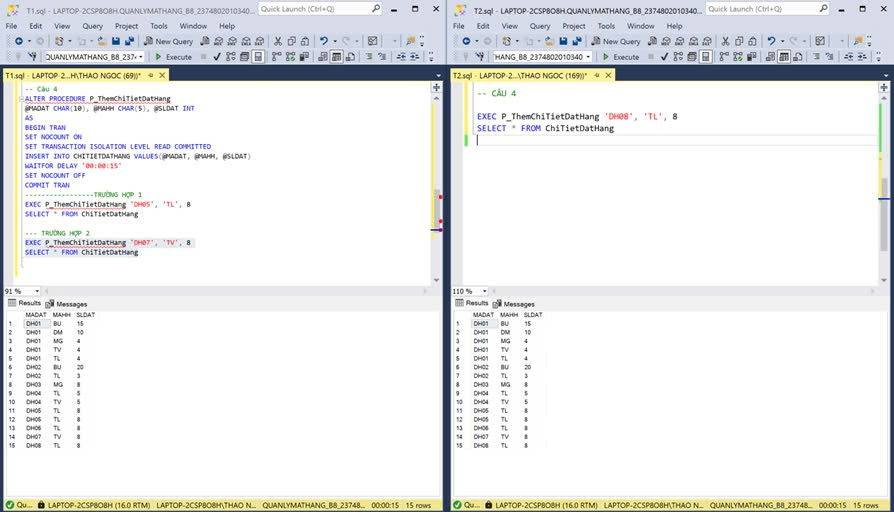
* **Case 1:**

**Result:**

**Picture: **

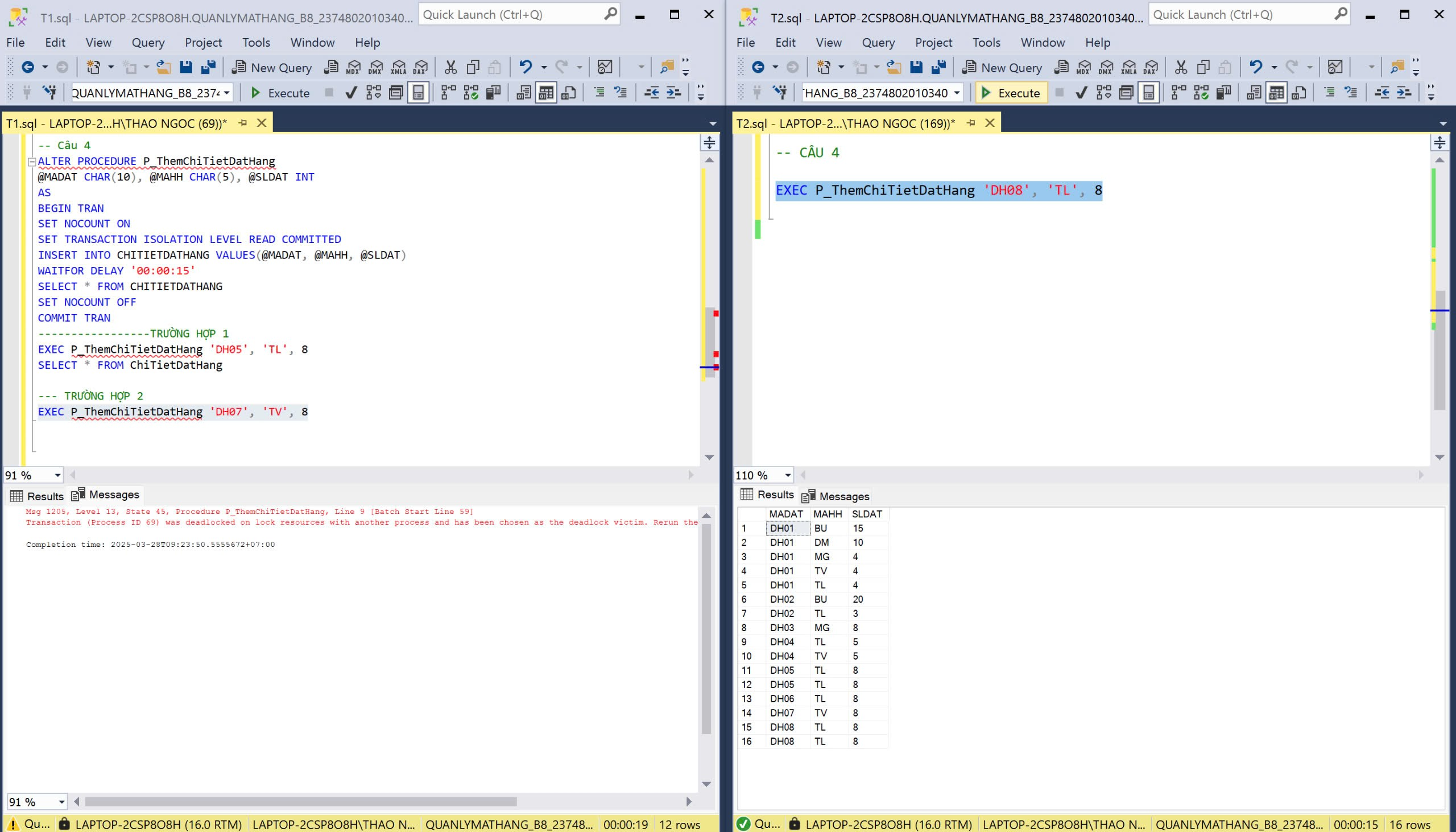
* **Case 2:**

**Result:**

**Picture: **

* **Case 3:**

**Result:**

**Picture: **