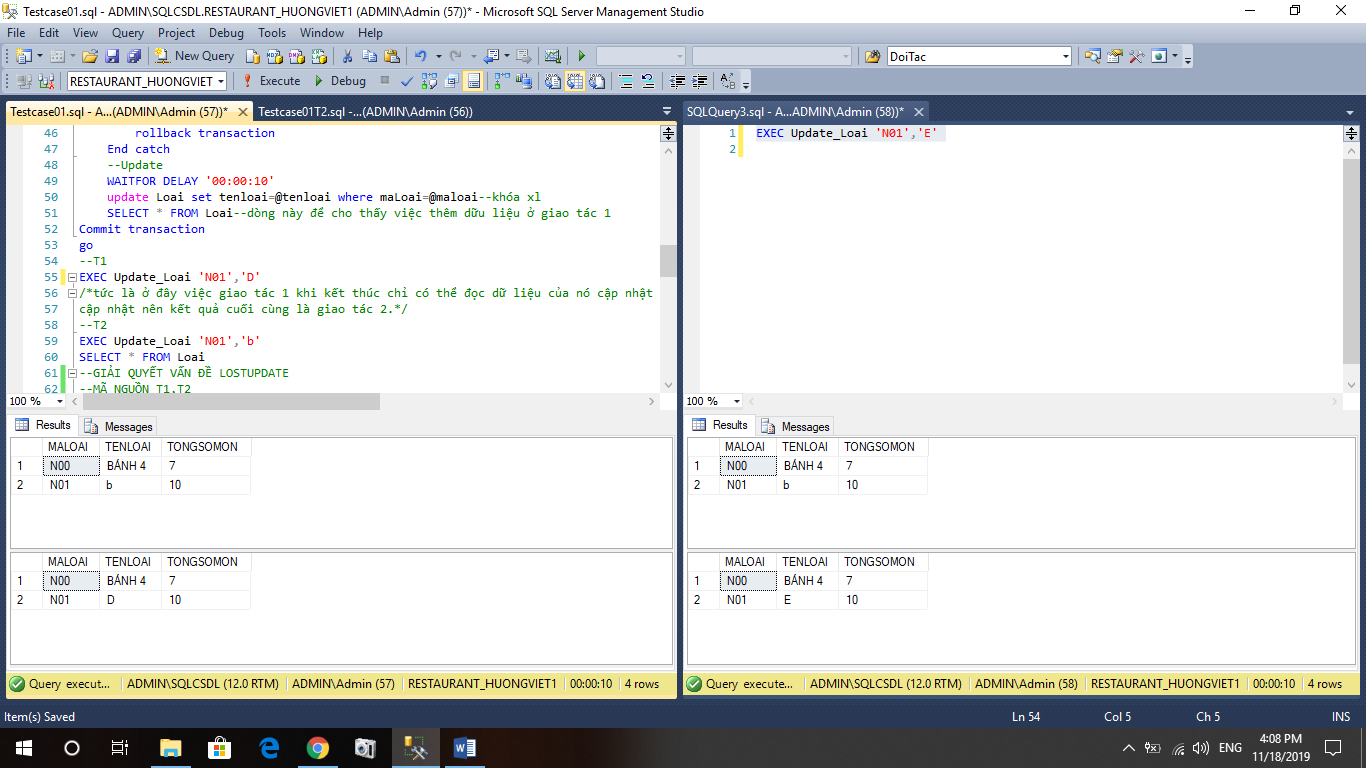
**TRƯỜNG HỢP LOST UPDATE :**

TESTCASE01:

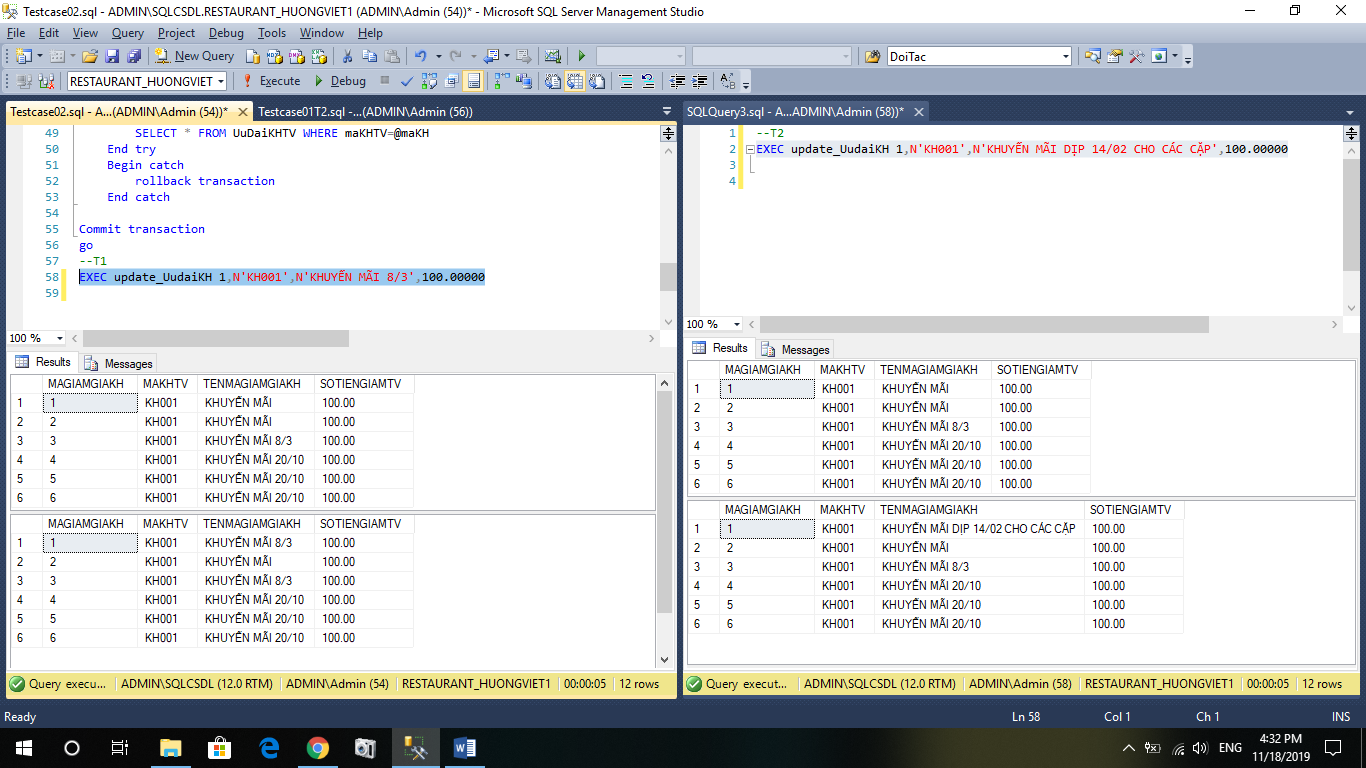


Việc 2 giao tác đọc cùng một đơn vị dữ liệu và giao tác T1 bắt đầu trước còn giao tác T2 đợi đến thao tác update của giao tác T1 commit thì giao tác T2 chạy update và commit,rõ ràng ta thấy ở đây đơn vị dữ T1 update chưa được giao tác nào đọc dẫn đến mất thông tin(lost update)

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT (LOAI) | T1:XIN SL(LOAI)  SQL:cấp SL (LOAI) và trả khóa SL |  |  |
|  |  | SELECT (LOAI) | T2:XIN SL(LOAI)  SQL:cấp SL (LOAI) và trả khóa SL |
| Update(LOAI) | T1:XIN XL(LOAI)  SQL:Cấp XL(LOAI) và giữ XL đến khi commit |  |  |
|  |  | Update(LOAI)  —  Đợi | T2:xin xl(LOAI)  SQL: không cho |
| SELECT (LOAI) | T1:XIN SL(LOAI)  SQL:cấp SL (LOAI) và trả khóa SL |  |  |
| commit |  |  |  |
|  |  | Update(LOAI) | T2:XIN XL(LOAI)  SQL:Cấp XL(LOAI) và giữ XL đến khi commit |
|  |  | SELECT (LOAI) | T2:XIN SL(LOAI)  SQL:cấp SL (LOAI) và trả khóa SL |
|  |  | commit |  |

TESTCASE02:

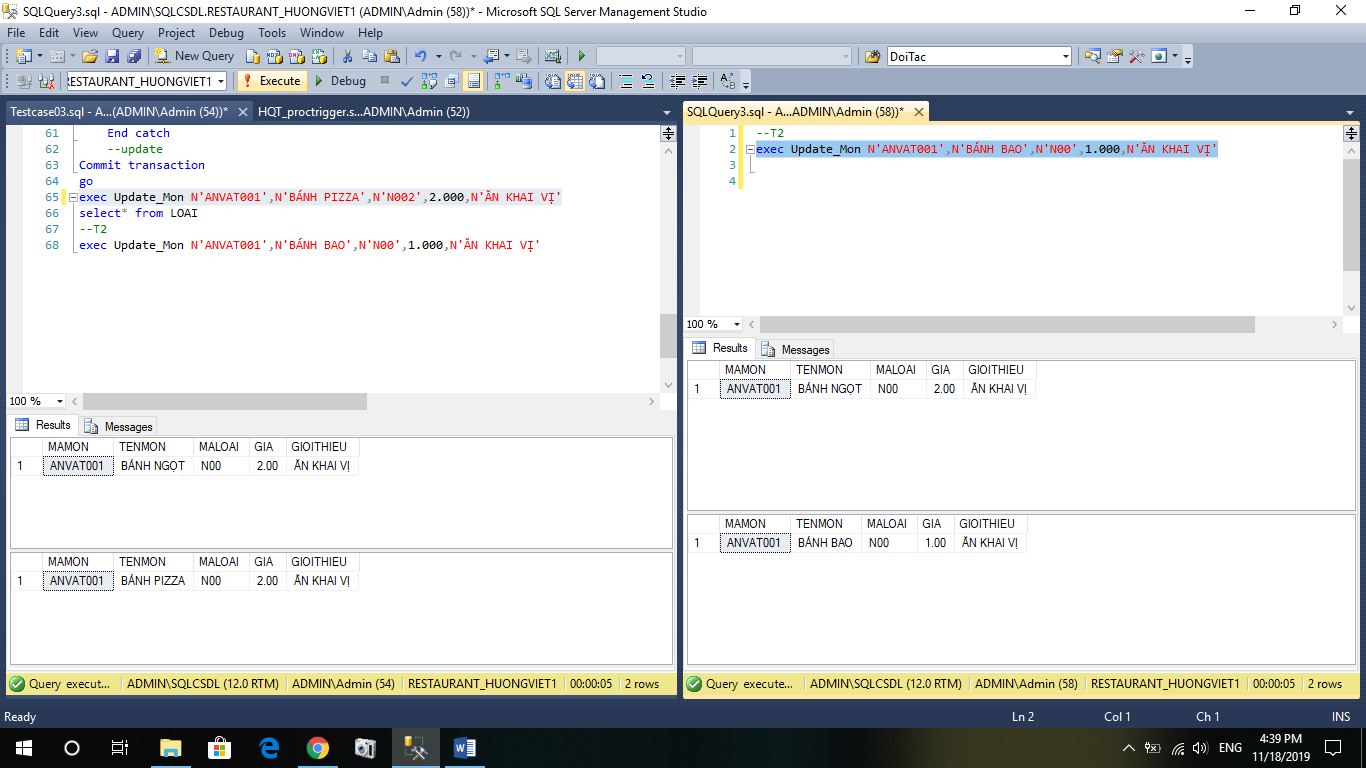


Việc 2 giao tác đọc cùng một đơn vị dữ liệu và giao tác T1 bắt đầu trước còn giao tác T2 đợi đến thao tác update của giao tác T1 commit thì giao tác T2 chạy update và commit,rõ ràng ta thấy ở đây đơn vị dữ T1 update chưa được giao tác nào đọc dẫn đến mất thông tin(lost update)

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT (UuDaiKHTV) | T1:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |  |  |
|  |  | SELECT (UuDaiKHTV) | T2:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |
| Update(UuDaiKHTV) | T1:XIN XL(UuDaiKHTV)  SQL:Cấp XL(UuDaiKHTV) và giữ XL đến khi commit |  |  |
|  |  | Update(UuDaiKHTV)  —  Đợi | T2:xin xl(UuDaiKHTV)  SQL: không cho |
| SELECT (UuDaiKHTV) | T1:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |  |  |
| commit |  |  |  |
|  |  | Update(UuDaiKHTV) | T2:XIN XL(UuDaiKHTV)  SQL:Cấp XL(UuDaiKHTV) và giữ XL đến khi commit |
|  |  | SELECT (UuDaiKHTV) | T2:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |
|  |  | commit |  |

TESTCASE03:

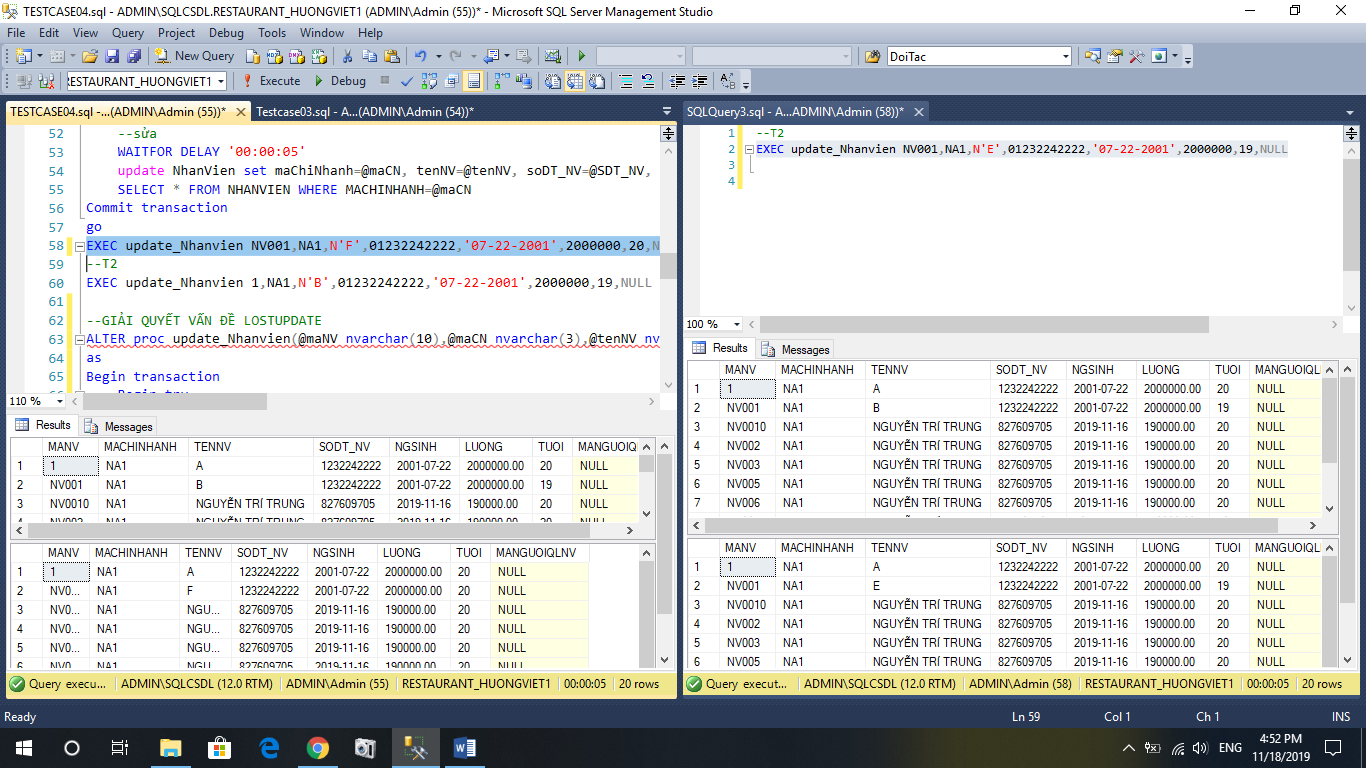


Việc 2 giao tác đọc cùng một đơn vị dữ liệu và giao tác T1 bắt đầu trước còn giao tác T2 đợi đến thao tác update của giao tác T1 commit thì giao tác T2 chạy update và commit,rõ ràng ta thấy ở đây đơn vị dữ T1 update chưa được giao tác nào đọc dẫn đến mất thông tin(lost update)

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT (Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và trả khóa SL |  |  |
|  |  | SELECT (Mon) | T2:XIN SL(Mon)  SQL:cấp SL (Mon) và trả khóa SL |
| Update(LOAI) | T1:XIN SL(LOAI)  SQL:cấp XL (LOAI) và giữ XL đến khi commit |  |  |
|  |  | Update(LOAI)  ĐỢI | T2:xin xl(LOAI)  SQL: không cho |
| Update(Mon) | T1:XIN XL(Mon)  SQL:Cấp XL(Mon) và giữ XL đến khi commit |  |  |
| SELECT (Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và trả khóa SL |  |  |
| commit |  |  |  |
|  |  | Update(LOAI) | T2:XIN XL(LOAI)  SQL:Cấp XL(Mon) và giữ XL đến khi commit |
|  |  | Update(Mon) | T2:XIN XL(Mon)  SQL:Cấp XL(Mon) và giữ XL đến khi commit |
|  |  | SELECT (Mon) | T2:XIN SL(Mon)  SQL:cấp SL (Mon) và trả khóa SL |
|  |  | commit |  |

TESTCASE04:

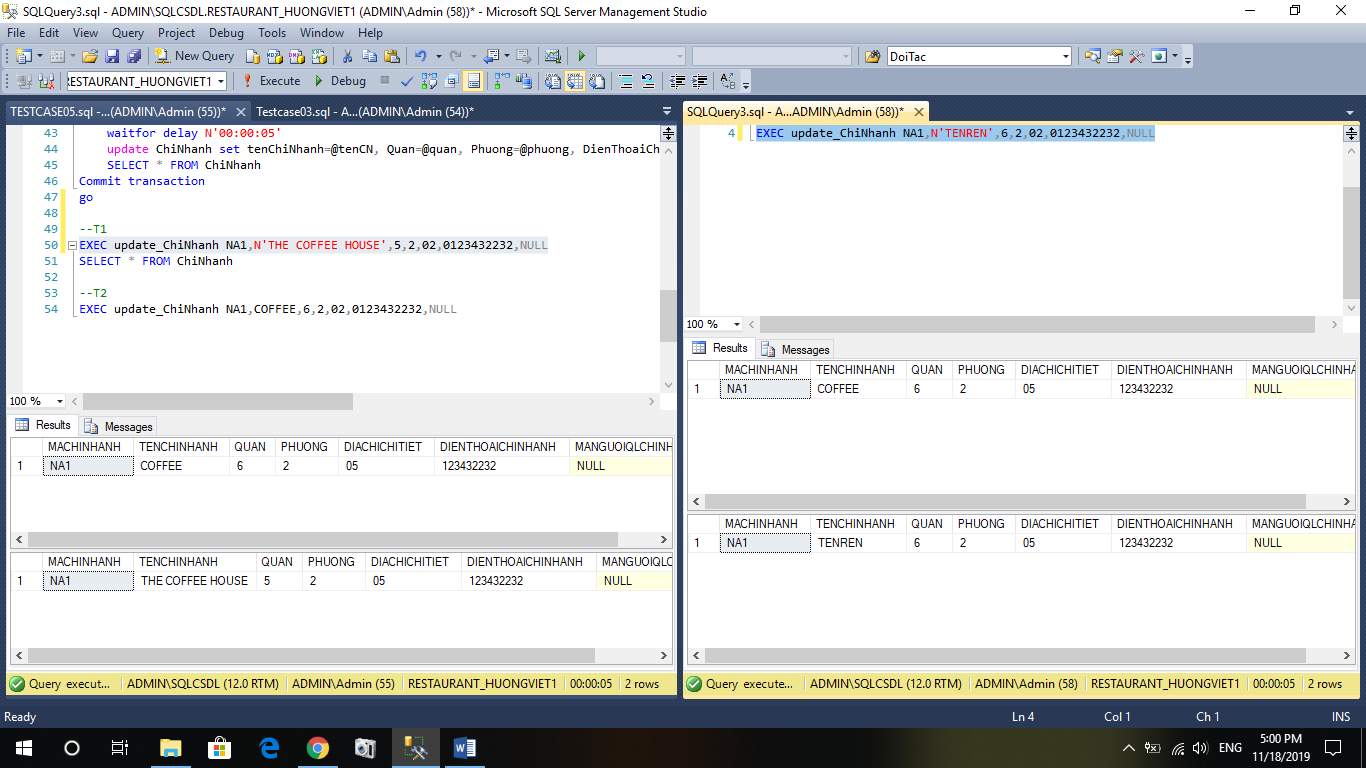


Việc 2 giao tác đọc cùng một đơn vị dữ liệu và giao tác T1 bắt đầu trước còn giao tác T2 đợi đến thao tác update của giao tác T1 commit thì giao tác T2 chạy update và commit,rõ ràng ta thấy ở đây đơn vị dữ T1 update chưa được giao tác nào đọc dẫn đến mất thông tin(lost update)

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT (NHANVIEN  ) | T1:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN  ) và trả khóa SL |  |  |
|  |  | SELECT (NHANVIEN  ) | T2:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |
| Update(NHANVIEN) | T1:XIN XL(NHANVIEN)  SQL:Cấp XL(NHANVIEN) và giữ XL đến khi commit |  |  |
|  |  | Update(NHANVIEN)  —  Đợi | T2:xin xl(NHANVIEN)  SQL: không cho |
| SELECT (NHANVIEN) | T1:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |  |  |
| commit |  |  |  |
|  |  | Update(NHANVIEN) | T2:XIN XL(NHANVIEN)  SQL:Cấp XL(NHANVIEN) và giữ XL đến khi commit |
|  |  | SELECT (NHANVIEN) | T2:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |
|  |  | commit |  |

TESTCASE05:



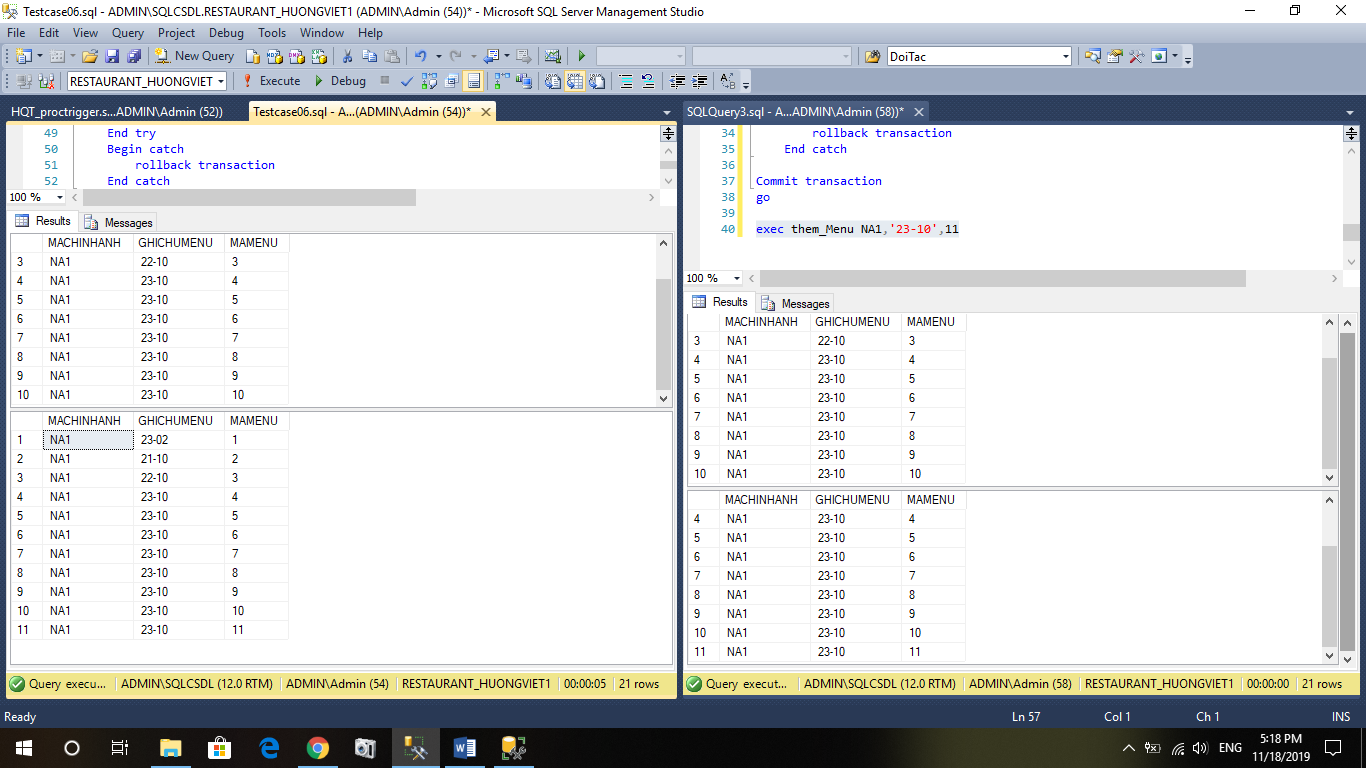
Việc 2 giao tác đọc cùng một đơn vị dữ liệu và giao tác T1 bắt đầu trước còn giao tác T2 đợi đến thao tác update của giao tác T1 commit thì giao tác T2 chạy update và commit,rõ ràng ta thấy ở đây đơn vị dữ T1 update chưa được giao tác nào đọc dẫn đến mất thông tin(lost update)

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT (ChiNhanh) | T1:XIN SL(ChiNhanh)  SQL:cấp SL (ChiNhanh  ) và trả khóa SL |  |  |
|  |  | SELECT (ChiNhanh  ) | T2:XIN SL(ChiNhanh)  SQL:cấp SL (ChiNhanh) và trả khóa SL |
| Update(ChiNhanh) | T1:XIN XL(ChiNhanh)  SQL:Cấp XL(ChiNhanh) và giữ XL đến khi commit |  |  |
|  |  | Update(ChiNhanh)  —  Đợi | T2:xin xl(ChiNhanh)  SQL: không cho |
| SELECT (ChiNhanh) | T1:XIN SL(ChiNhanh)  SQL:cấp SL (ChiNhanh) và trả khóa SL |  |  |
| commit |  |  |  |
|  |  | Update(ChiNhanh) | T2:XIN XL(ChiNhanh)  SQL:Cấp XL(ChiNhanh) và giữ XL đến khi commit |
|  |  | SELECT (ChiNhanh) | T2:XIN SL(ChiNhanh)  SQL:cấp SL (ChiNhanh) và trả khóa SL |
|  |  | commit |  |

**TRƯỜNG HỢP PHANTOM:**

TESTCASE06:

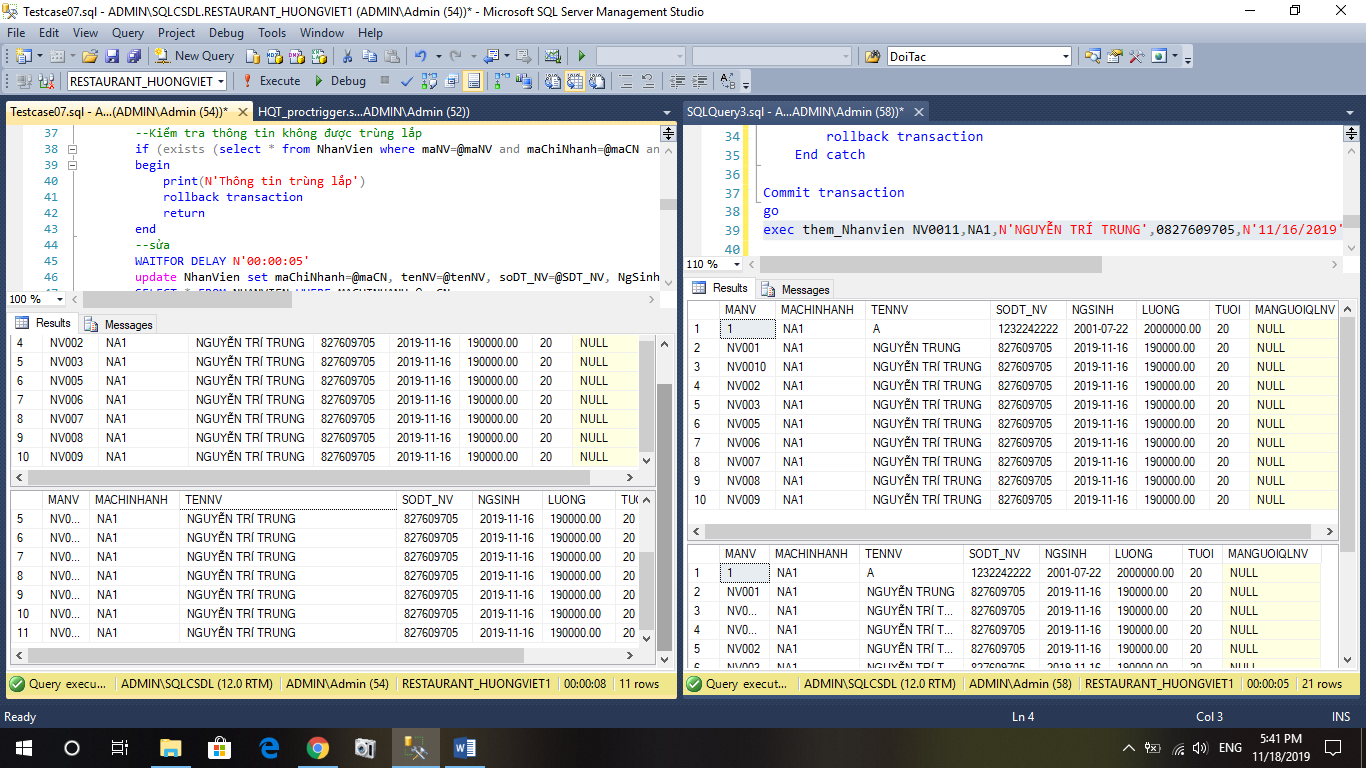


Việc 2 giao tác trong đó giao tác 1 thực hiện việc đọc 2 lần ,và ở giữa có update để xem xem việc update có thành công hay không nhưng có một giao tác khác thực hiện việc insert một đơn vị dữ liệu có điều kiện giống với đơn vị dữ liệu được đọc dẫn đến lần đọc thứ 2 của giao tác 1 thêm 1 dòng dữ liệu không rõ(giao tác 2 insert).Đây là vấn đề Phantom

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(Menu) | T1:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |  |  |
|  |  | SELECT(Menu) | T2:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Insert(Menu) | T2:XIN XL(Menu)  SQL:cấp XL (Menu) và giữ đến lúc commit |
|  |  | SELECT(Menu) | T2:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |
|  |  | commit |  |
| Update(Menu) | T1:XIN XL(Menu)  SQL:Cấp XL(Menu) và giữ XL đến khi commit |  |  |
| SELECT(Menu)  🡪Đọc dòng insert của T2 | T1:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE07:

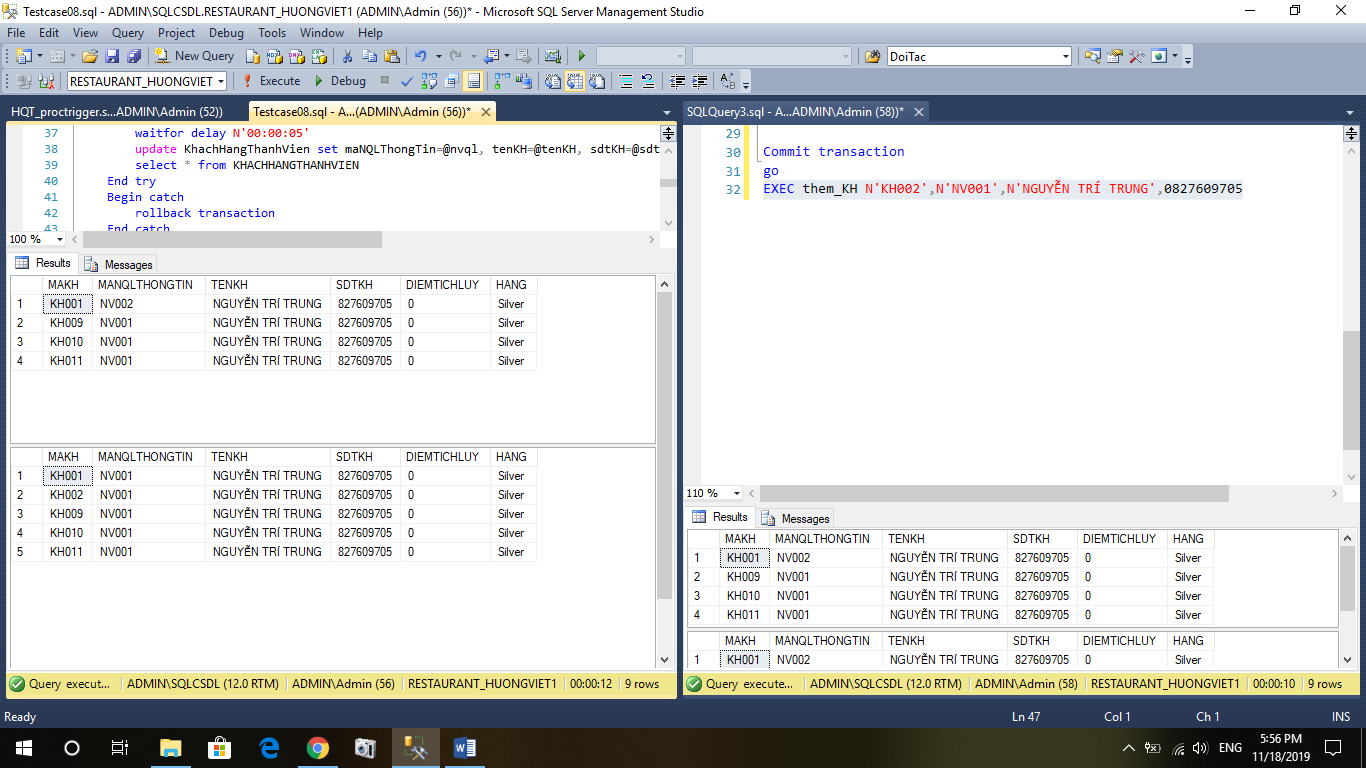


Việc 2 giao tác trong đó giao tác 1 thực hiện việc đọc 2 lần ,và ở giữa có update để xem xem việc update có thành công hay không nhưng có một giao tác khác thực hiện việc insert một đơn vị dữ liệu có điều kiện giống với đơn vị dữ liệu được đọc dẫn đến lần đọc thứ 2 của giao tác 1 thêm 1 dòng dữ liệu không rõ(giao tác 2 insert).Đây là vấn đề Phantom

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(NHANVIEN) | T1:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |  |  |
|  |  | SELECT(NHANVIEN) | T2:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Insert(NHANVIEN) | T2:XIN XL(NHANVIEN)  SQL:cấp XL (NHANVIEN) và giữ đến lúc commit |
|  |  | SELECT(NHANVIEN) | T2:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |
|  |  | commit |  |
| Update(NHANVIEN) | T1:XIN XL(NHANVIEN)  SQL:Cấp XL(NHANVIEN) và giữ XL đến khi commit |  |  |
| SELECT(NHANVIEN)  🡪Đọc dòng insert của T2 | T1:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE08:

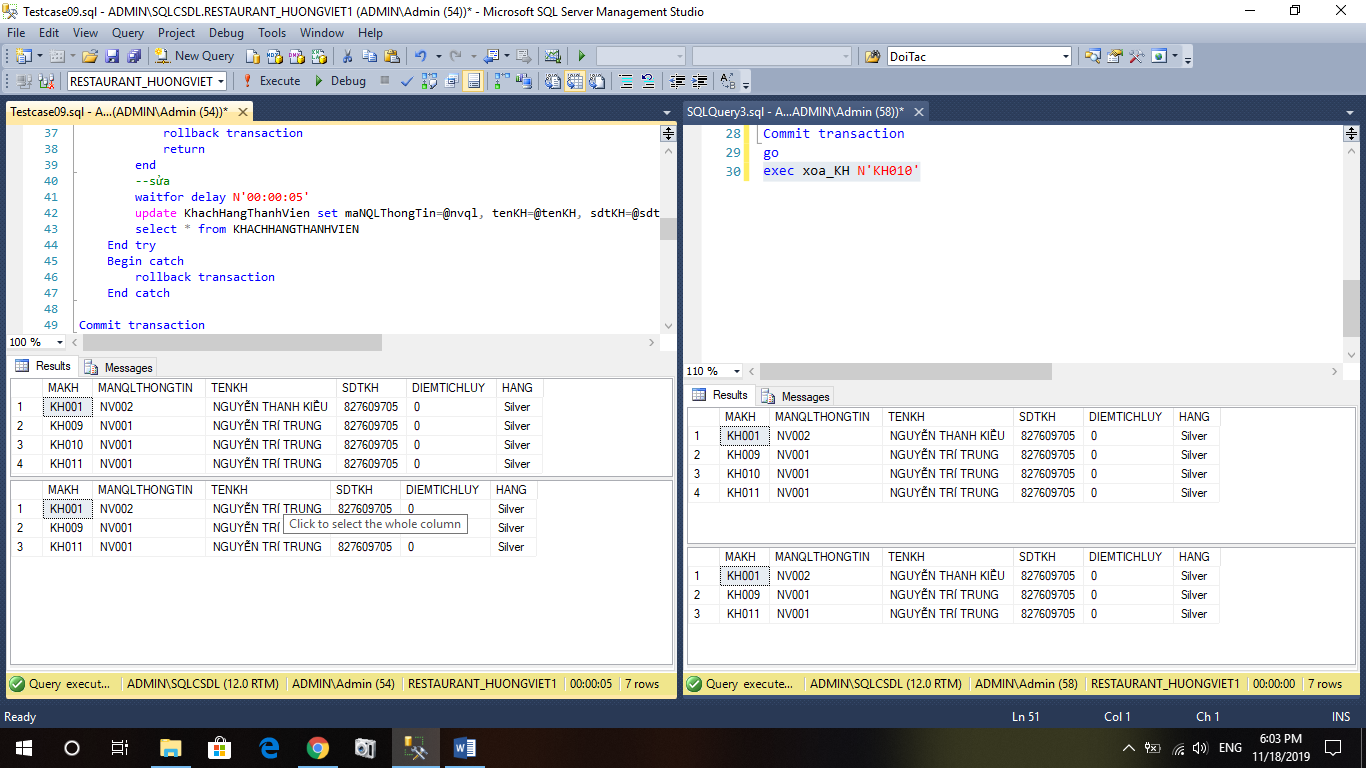


Việc 2 giao tác trong đó giao tác 1 thực hiện việc đọc 2 lần ,và ở giữa có update để xem xem việc update có thành công hay không nhưng có một giao tác khác thực hiện việc insert một đơn vị dữ liệu có điều kiện giống với đơn vị dữ liệu được đọc dẫn đến lần đọc thứ 2 của giao tác 1 thêm 1 dòng dữ liệu không rõ(giao tác 2 insert).Đây là vấn đề Phantom

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(KHACHHANGTHANHVIEN) | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Insert(KHACHHANGTHANHVIEN) | T2:XIN XL(KHACHHANGTHANHVIEN)  SQL:cấp XL (KHACHHANGTHANHVIEN) và giữ đến lúc commit |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
|  |  | commit |  |
| Update(KHACHHANGTHANHVIEN) | T1:XIN XL(KHACHHANGTHANHVIEN)  SQL:Cấp XL(KHACHHANGTHANHVIEN) và giữ XL đến khi commit |  |  |
| SELECT(KHACHHANGTHANHVIEN)  🡪Đọc dòng insert của T2 | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE09:

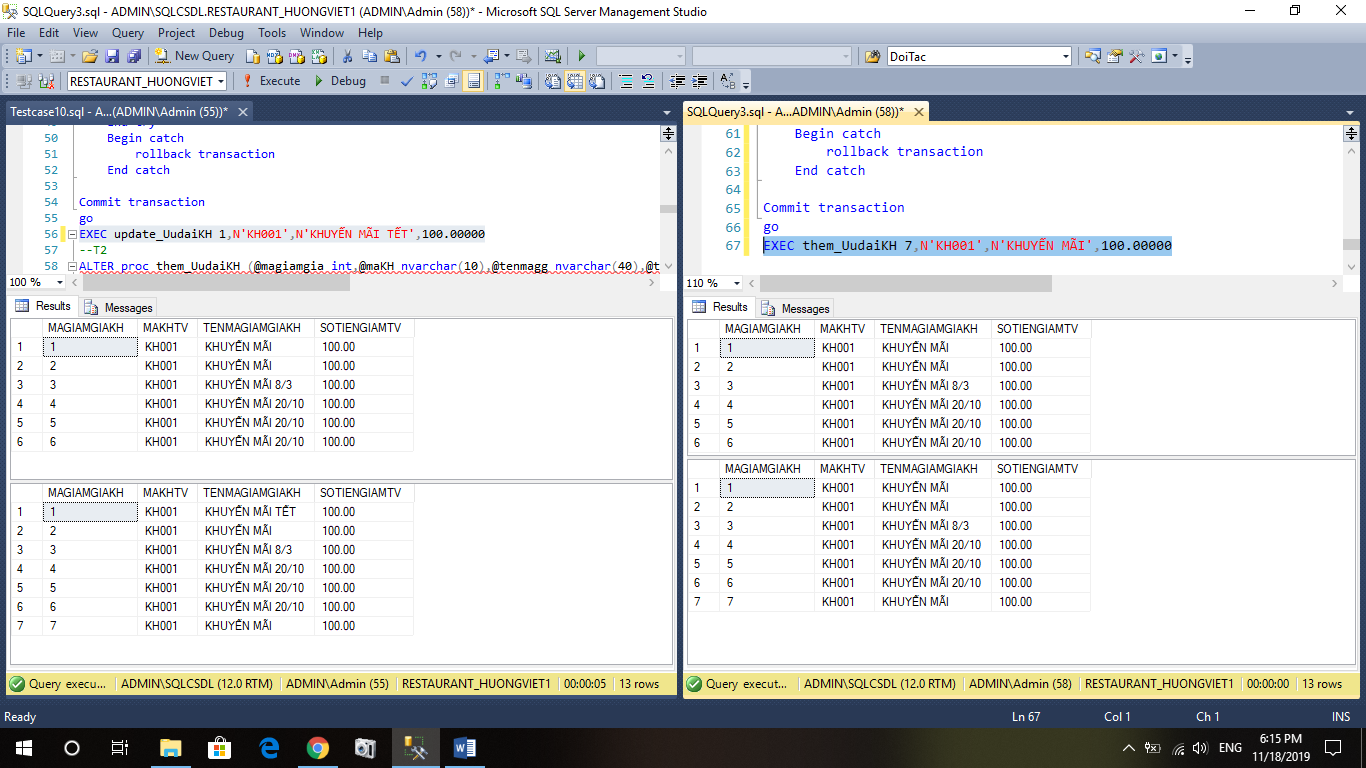


Việc 2 giao tác trong đó giao tác 1 thực hiện việc đọc 2 lần ,và ở giữa có update để xem xem việc update có thành công hay không nhưng có một giao tác khác thực hiện việc DELETE một đơn vị dữ liệu có điều kiện giống với đơn vị dữ liệu được đọc dẫn đến lần đọc thứ 2 của giao tác 1 mất 1 dòng dữ liệu không rõ(giao tác 2 delete).Đây là vấn đề Phantom

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(KHACHHANGTHANHVIEN) | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Delete(KHACHHANGTHANHVIEN) | T2:XIN XL(KHACHHANGTHANHVIEN)  SQL:cấp XL (KHACHHANGTHANHVIEN) và giữ đến lúc commit |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
|  |  | commit |  |
| Update(KHACHHANGTHANHVIEN) | T1:XIN XL(KHACHHANGTHANHVIEN)  SQL:Cấp XL(KHACHHANGTHANHVIEN) và giữ XL đến khi commit |  |  |
| SELECT(KHACHHANGTHANHVIEN)  🡪Đọc mất 1 dòng do delete của T2 | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE10:



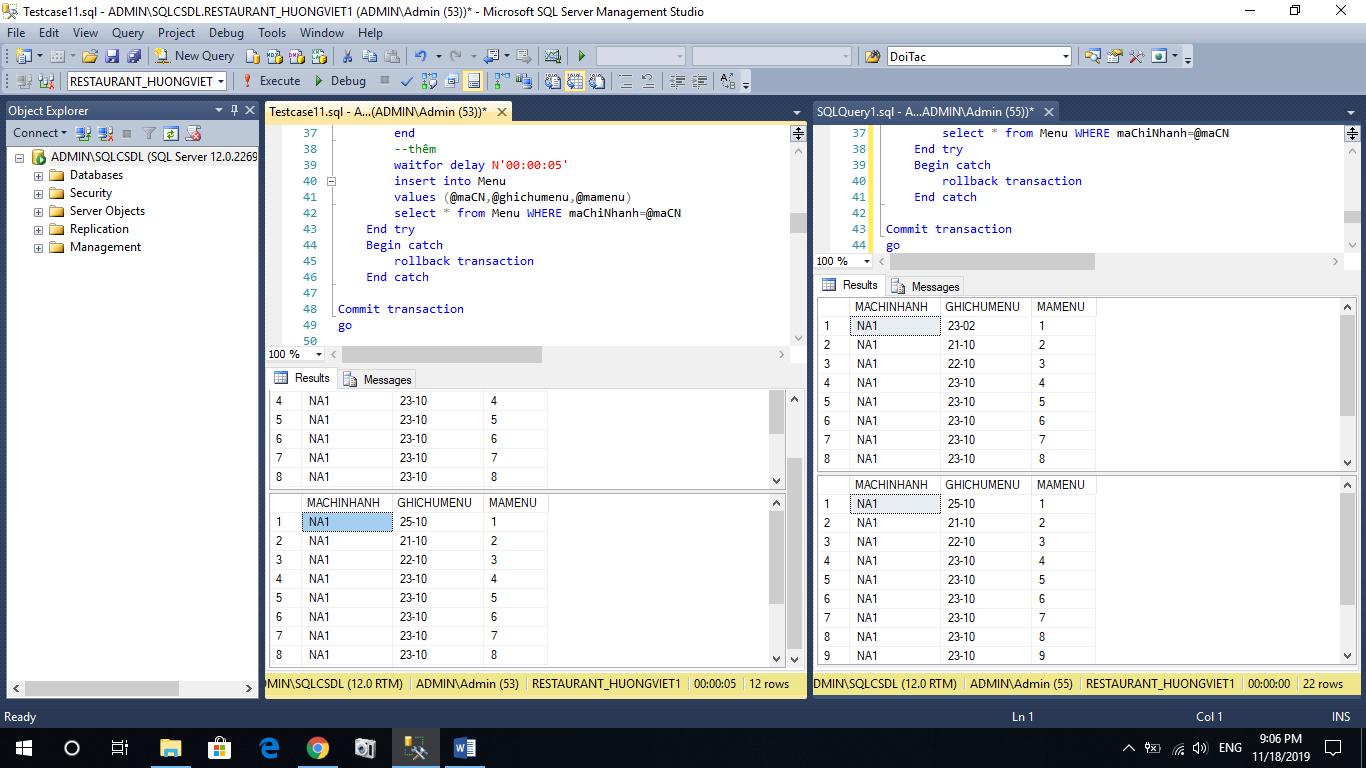
Việc 2 giao tác trong đó giao tác 1 thực hiện việc đọc 2 lần ,và ở giữa có update để xem xem việc update có thành công hay không nhưng có một giao tác khác thực hiện việc insert một đơn vị dữ liệu có điều kiện giống với đơn vị dữ liệu được đọc dẫn đến lần đọc thứ 2 của giao tác 1 thêm 1 dòng dữ liệu không rõ(giao tác 2 insert).Đây là vấn đề Phantom

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(UuDaiKHTV) | T1:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |  |  |
|  |  | SELECT(UuDaiKHTV  ) | T2:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Insert(UuDaiKHTV) | T2:XIN XL(UuDaiKHTV)  SQL:cấp XL (UuDaiKHTV) và giữ đến lúc commit |
|  |  | SELECT(UuDaiKHTV) | T2:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |
|  |  | commit |  |
| Update(UuDaiKHTV) | T1:XIN XL(UuDaiKHTV)  SQL:Cấp XL(UuDaiKHTV) và giữ XL đến khi commit |  |  |
| SELECT(UuDaiKHTV)  🡪Đọc đơn vị dữ liệu insert của T2 | T1:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |  |  |
| commit |  |  |  |

**TRƯỜNG HỢP UNREPEATABLE READ:**

TESTCASE11:

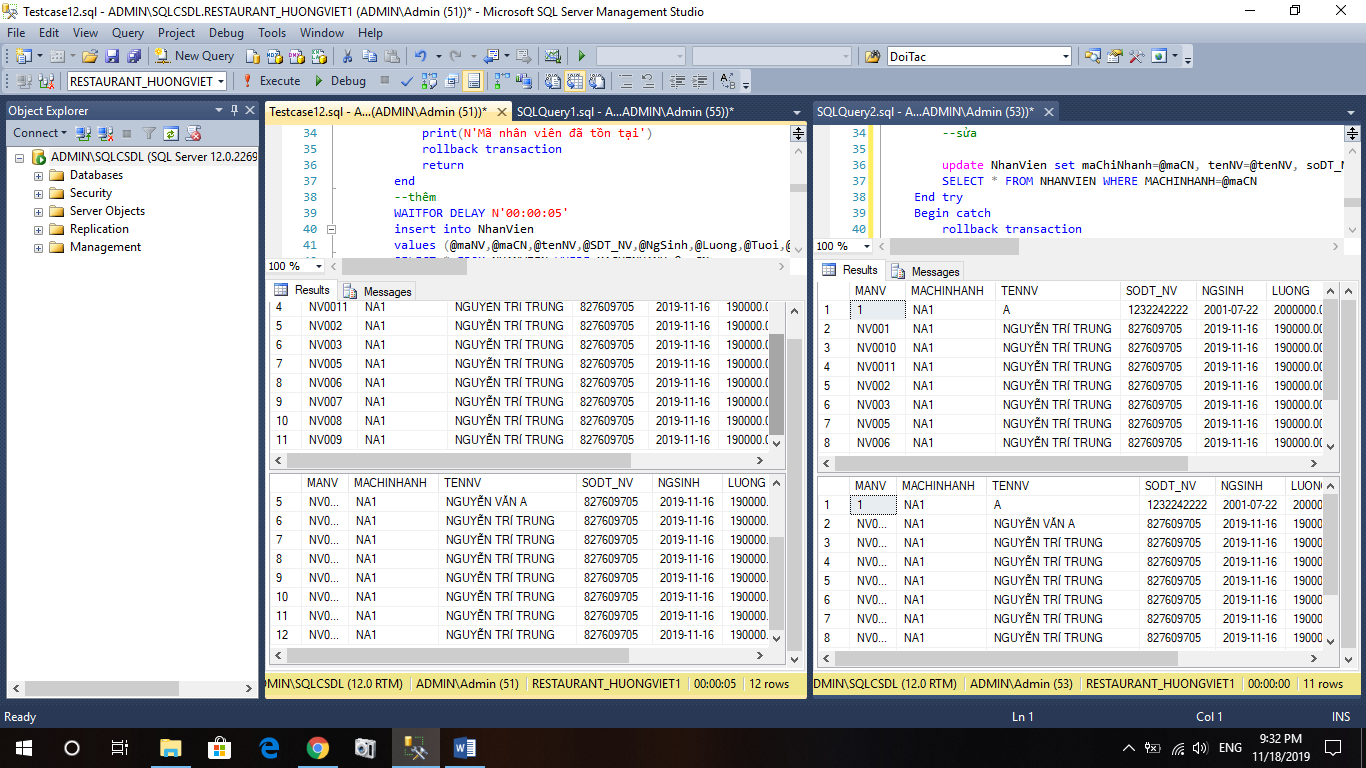


Việc 2 giao tác trong đó 1 giao tác insert và đọc 2 lần (lần đầu để xem dữ liệu trước insert và lần 2 để đọc dữ liệu insert thành công),nhưng việc insert bị delay và đợi giao tác 2 thực hiện update xong và commit mới insert và đọc dữ liệu dẫn đến việc đọc lần 2 có dữ liệu insert và dữ liệu liệu của giao tác T2 update.Đây là trường hợp Unrepeatable read.

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(Menu) | T1:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |  |  |
|  |  | SELECT(Menu  ) | T2:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Update(Menu) | T2:XIN XL(Menu)  SQL:cấp XL (Menu) và giữ đến lúc commit |
|  |  | SELECT(Menu) | T2:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |
|  |  | commit |  |
| Insert(Menu) | T1:XIN XL(Menu)  SQL:Cấp XL(Menu) và giữ XL đến khi commit |  |  |
| SELECT(Menu)  🡪Đọc đơn vị dữ liệu có update của T2 | T1:XIN SL(Menu)  SQL:cấp SL (Menu) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE12:

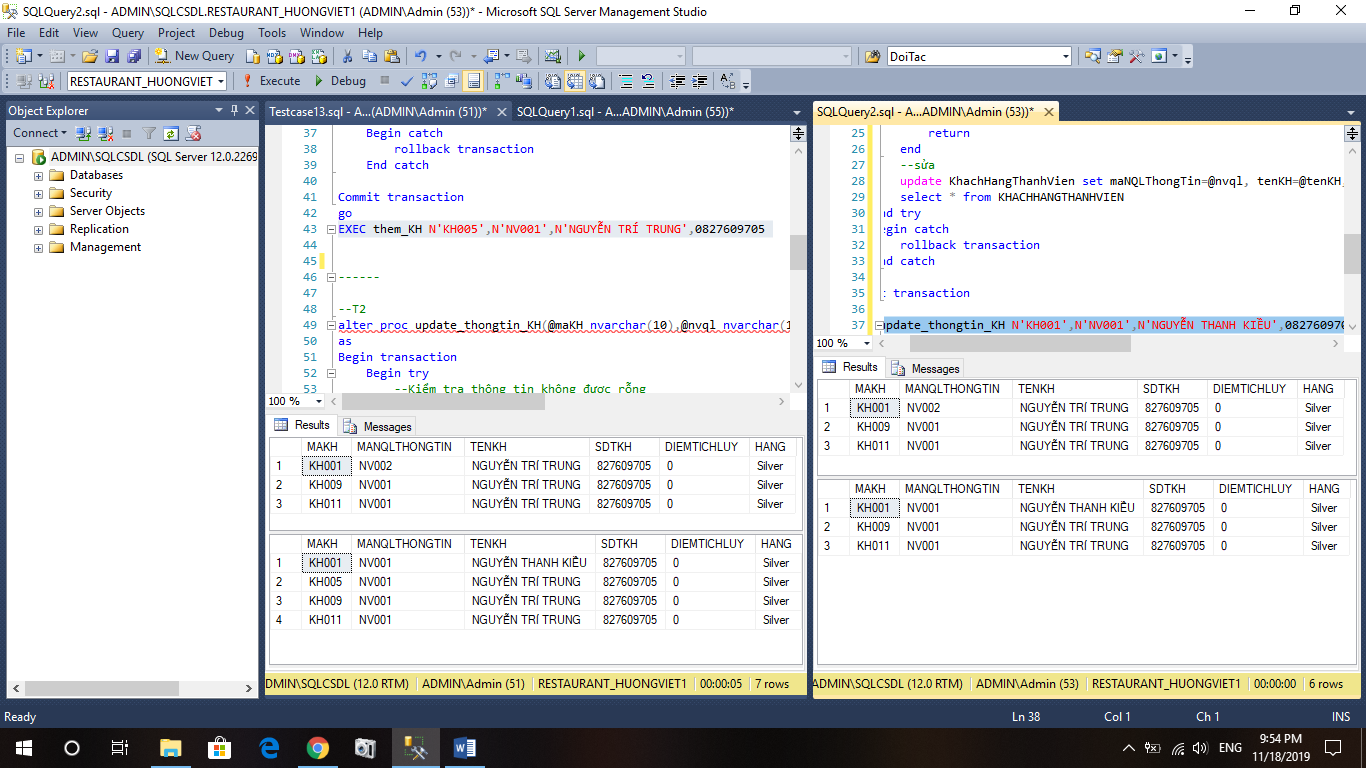


Việc 2 giao tác trong đó 1 giao tác insert và đọc 2 lần (lần đầu để xem dữ liệu trước insert và lần 2 để đọc dữ liệu insert thành công),nhưng việc insert bị delay và đợi giao tác 2 thực hiện update xong và commit mới insert và đọc dữ liệu dẫn đến việc đọc lần 2 có dữ liệu insert và dữ liệu liệu của giao tác T2 update.Đây là trường hợp Unrepeatable read.

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(NHANVIEN) | T1:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |  |  |
|  |  | SELECT(NHANVIEN  ) | T2:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Update(NHANVIEN) | T2:XIN XL(NHANVIEN)  SQL:cấp XL (NHANVIEN) và giữ đến lúc commit |
|  |  | SELECT(NHANVIEN) | T2:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |
|  |  | commit |  |
| Insert(NHANVIEN) | T1:XIN XL(NHANVIEN)  SQL:Cấp XL(NHANVIEN) và giữ XL đến khi commit |  |  |
| SELECT(NHANVIEN)  🡪Đọc đơn vị dữ liệu có update của T2 | T1:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE13:

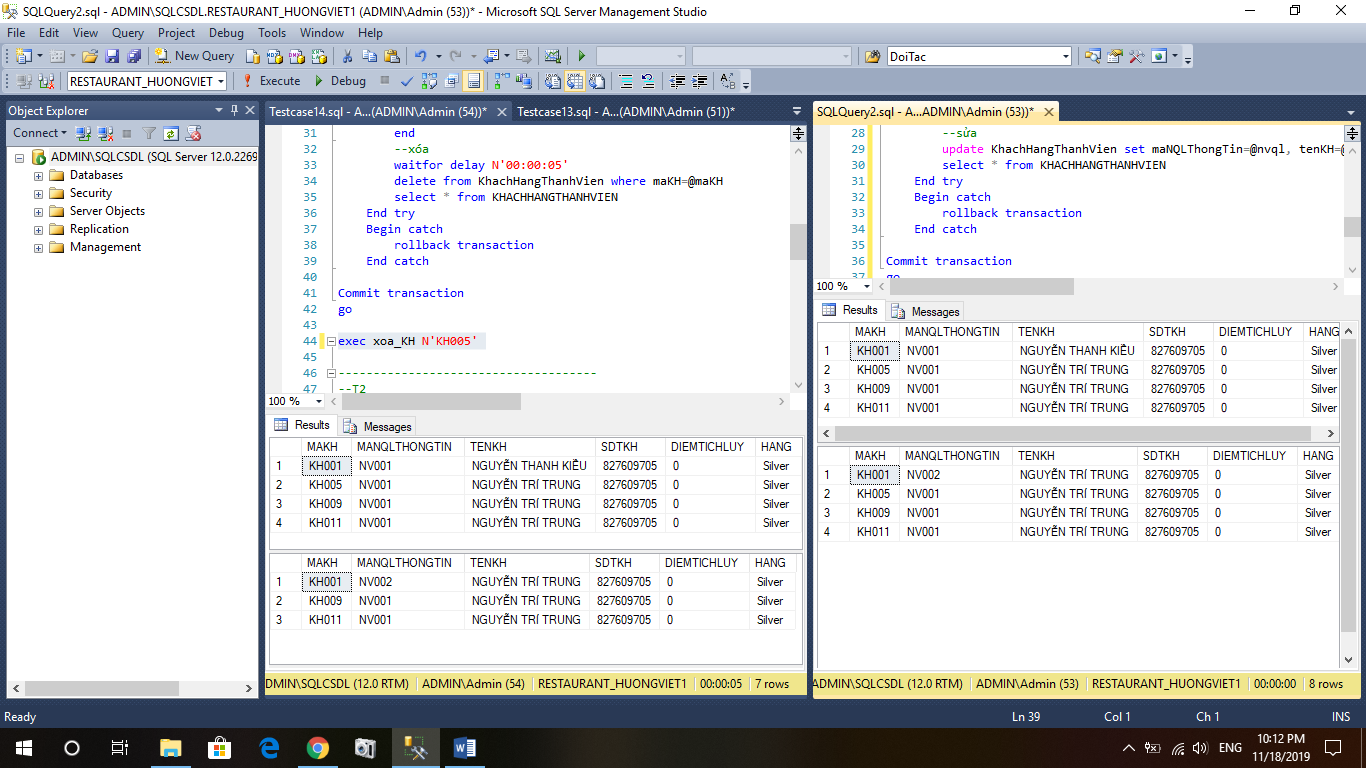


Việc 2 giao tác trong đó 1 giao tác insert và đọc 2 lần (lần đầu để xem dữ liệu trước insert và lần 2 để đọc dữ liệu insert thành công),nhưng việc insert bị delay và đợi giao tác 2 thực hiện update xong và commit mới insert và đọc dữ liệu dẫn đến việc đọc lần 2 có dữ liệu insert và dữ liệu liệu của giao tác T2 update.Đây là trường hợp Unrepeatable read.

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(KHACHHANGTHANHVIEN) | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Update(KHACHHANGTHANHVIEN) | T2:XIN XL(KHACHHANGTHANHVIEN)  SQL:cấp XL (KHACHHANGTHANHVIEN) và giữ đến lúc commit |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
|  |  | commit |  |
| Insert(KHACHHANGTHANHVIEN) | T1:XIN XL(KHACHHANGTHANHVIEN)  SQL:Cấp XL(KHACHHANGTHANHVIEN) và giữ XL đến khi commit |  |  |
| SELECT(KHACHHANGTHANHVIEN)  🡪Đọc đơn vị dữ liệu có update của T2 | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE14:

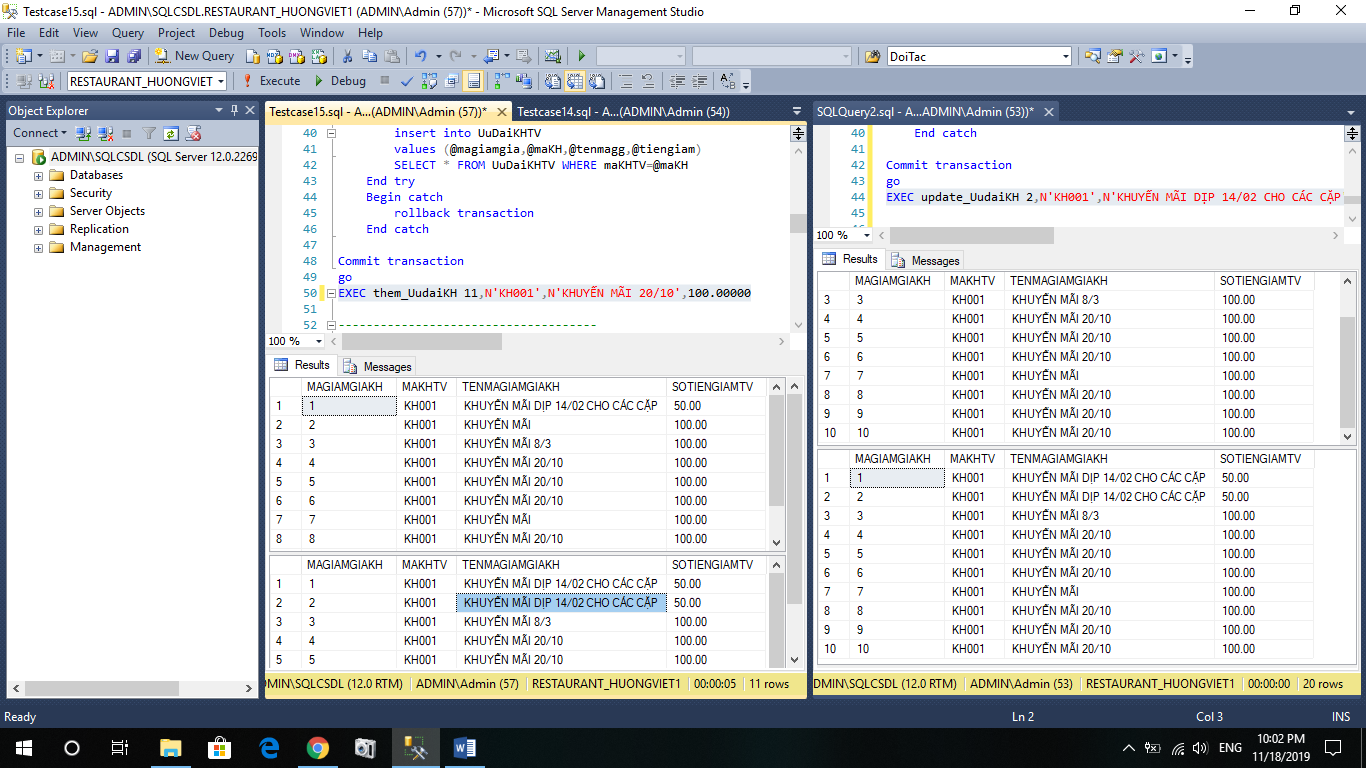


Việc 2 giao tác trong đó 1 giao tác DELETE và đọc 2 lần (lần đầu để xem dữ liệu trước DELETE và lần 2 để đọc dữ liệu DELETE thành công),nhưng việc DELETE bị delay và đợi giao tác 2 thực hiện update xong và commit mới DELETE và đọc dữ liệu dẫn đến việc đọc lần 2 mất dữ liệu bị DELETE và dữ liệu liệu của giao tác T2 update.Đây là trường hợp Unrepeatable read.

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(KHACHHANGTHANHVIEN) | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Update(KHACHHANGTHANHVIEN) | T2:XIN XL(KHACHHANGTHANHVIEN)  SQL:cấp XL (KHACHHANGTHANHVIEN) và giữ đến lúc commit |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |
|  |  | commit |  |
| DELETE(KHACHHANGTHANHVIEN) | T1:XIN XL(KHACHHANGTHANHVIEN)  SQL:Cấp XL(KHACHHANGTHANHVIEN) và giữ XL đến khi commit |  |  |
| SELECT(KHACHHANGTHANHVIEN)  🡪Đọc đơn vị dữ liệu có update của T2 | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và trả khóa SL |  |  |
| commit |  |  |  |

TESTCASE15:



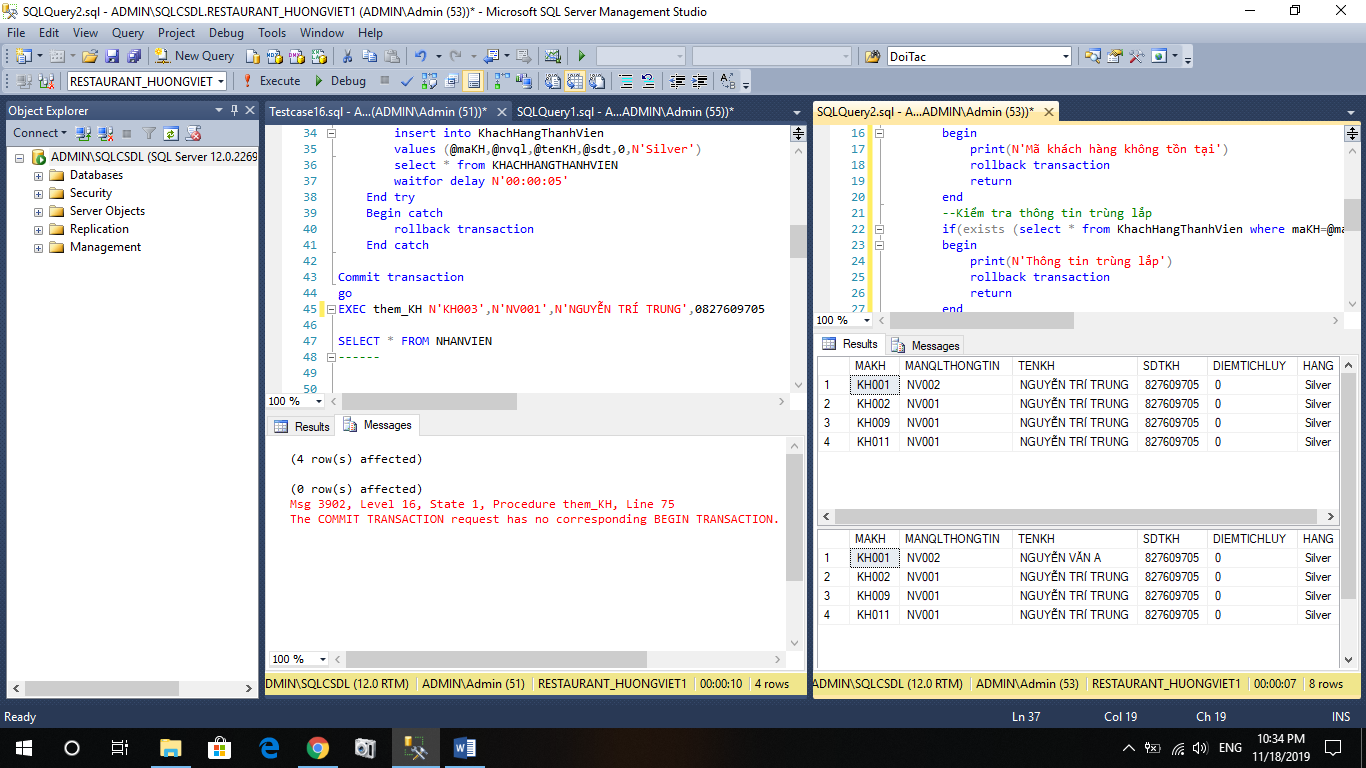
Việc 2 giao tác trong đó 1 giao tác insert và đọc 2 lần (lần đầu để xem dữ liệu trước insert và lần 2 để đọc dữ liệu insert thành công),nhưng việc insert bị delay và đợi giao tác 2 thực hiện update xong và commit mới insert và đọc dữ liệu dẫn đến việc đọc lần 2 có dữ liệu insert và dữ liệu liệu của giao tác T2 update.Đây là trường hợp Unrepeatable read.

***Mức cô lập mặc định là read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(UuDaiKHTV) | T1:XIN SL(UuDaiKHTV)  SQL:cấp SL (NHANVIEN) và trả khóa SL |  |  |
|  |  | SELECT(UuDaiKHTV) | T2:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |
| Waitfor delay |  |  |  |
|  |  | Update(UuDaiKHTV) | T2:XIN XL(UuDaiKHTV)  SQL:cấp XL (NHANVIEN) và giữ đến lúc commit |
|  |  | SELECT(UuDaiKHTV) | T2:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |
|  |  | commit |  |
| Insert(UuDaiKHTV) | T1:XIN XL(UuDaiKHTV)  SQL:Cấp XL(UuDaiKHTV) và giữ XL đến khi commit |  |  |
| SELECT(UuDaiKHTV)  🡪Đọc đơn vị dữ liệu có update của T2 | T1:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và trả khóa SL |  |  |
| commit |  |  |  |

**TRƯỜNG HỢP DEADLOCK:**

TESTCASE16:



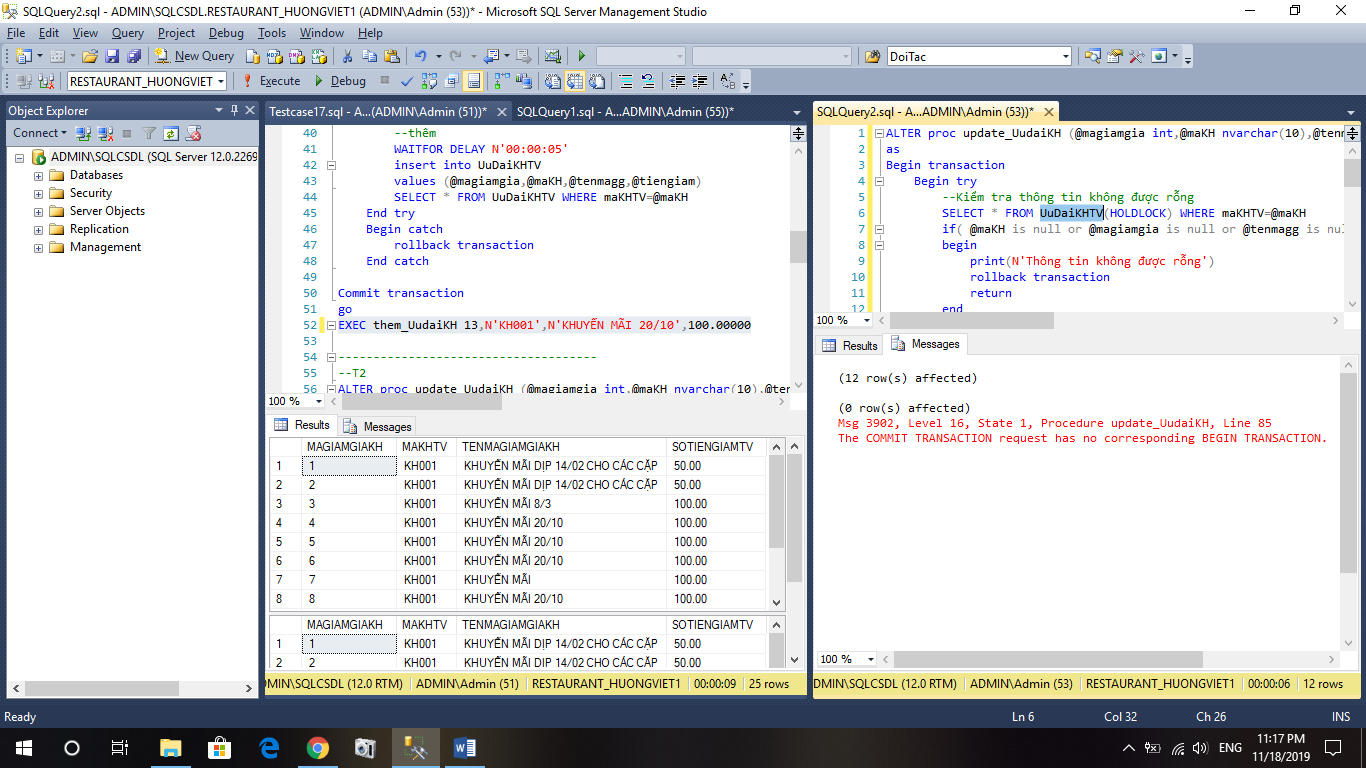
Một giao tác thực hiện việc insert dữ liệu và đọc 2 lần để kiểm tra xem việc insert đó có thành công không.Và ở đây thao tác đọc đầu tiên sẽ giữ holdlock,rowlock để tránh việc unrepeatable read ,và giao tác thứ 2 cũng có 2 thao tác đọc để xem việc update có thành công không và cũng giữ holdlock,rowlock để tránh việc unrepeatable read.Và việc giao tác thứ nhất insert mà bị đợi bởi select (giữ đến cuối commit) và giao tác thứ 2 việc update cũng đợi select(giữ đến cuối commit) trên cũng 1 đơn vị dữ liệu dẫn đến 2 giao tác đợi nhau và từ đó bị deadlock

* ở đây việc giữu như vậy cũng giống cài mức cô lập số 3 nên ta sẽ giả sử đang ở mức cô lập số 3

***Mức cô lập mặc định là repeatable read***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(KHACHHANGTHANHVIEN) | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và giữ khóa SL đến hết giao tác |  |  |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và giữ khóa SL đến hết giao tác |
| Waitfor delay |  |  |  |
|  |  | Update(KHACHHANGTHANHVIEN)  Đợi | T2:XIN XL(KHACHHANGTHANHVIEN)  SQL:không cấp |
| Insert(KHACHHANGTHANHVIEN)  Đợi | T1:XIN XL(KHACHHANGTHANHVIEN)  SQL:không cấp |  |  |
| Deadlock |  |  |  |

TESTCASE17:



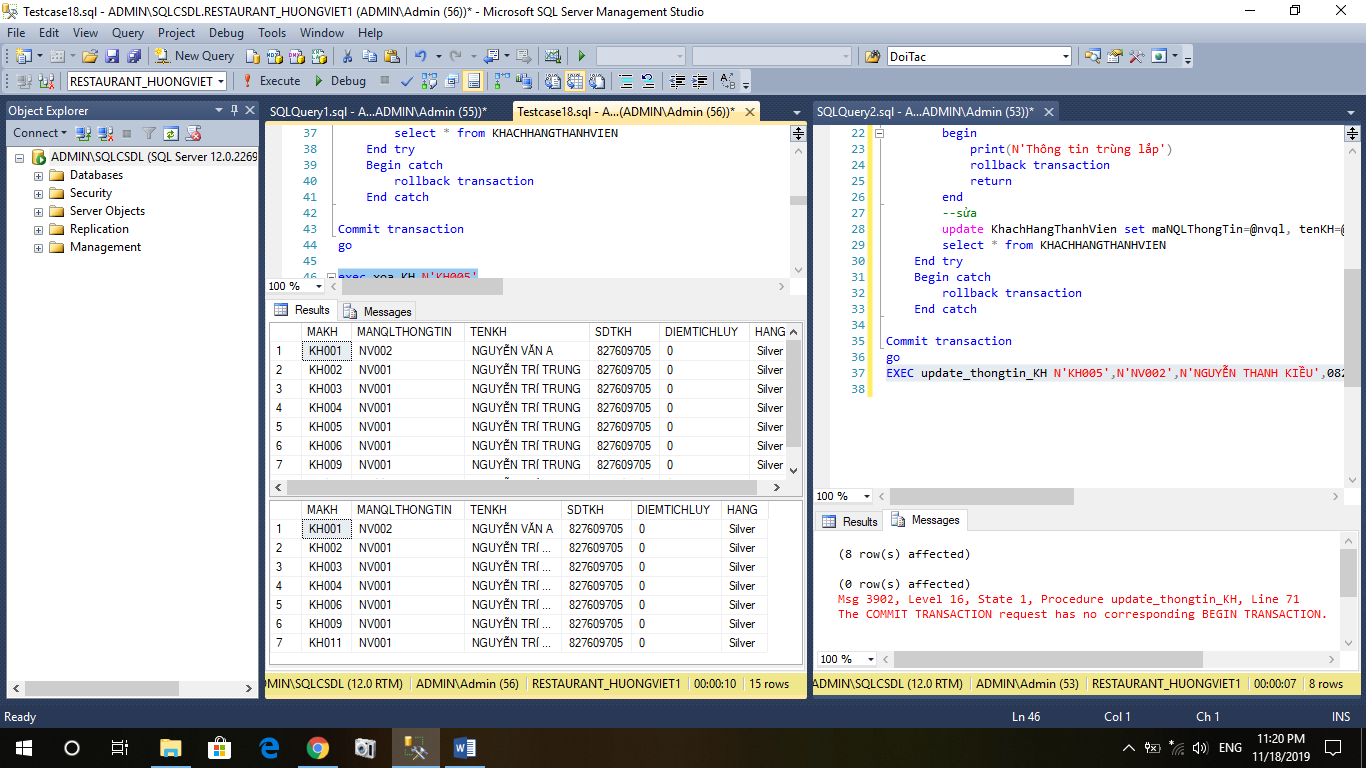
Một giao tác thực hiện việc insert dữ liệu và đọc 2 lần để kiểm tra xem việc insert đó có thành công không.Và ở đây thao tác đọc đầu tiên sẽ giữ holdlock để tránh việc unrepeatable read ,và giao tác thứ 2 cũng có 2 thao tác đọc để xem việc update có thành công không và cũng giữ holdlock,rowlock để tránh việc unrepeatable read.Và việc giao tác thứ nhất insert mà bị đợi bởi select (giữ đến cuối commit) và giao tác thứ 2 việc update cũng đợi select(giữ đến cuối commit) trên cũng 1 đơn vị dữ liệu dẫn đến 2 giao tác đợi nhau và từ đó bị deadlock

* ở đây việc giữu như vậy cũng giống cài mức cô lập số 3 nên ta sẽ giả sử đang ở mức cô lập số 3

***Mức cô lập mặc định là repeatable read***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(UuDaiKHTV) | T1:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và giữ khóa SL đến hết giao tác |  |  |
|  |  | SELECT(UuDaiKHTV) | T2:XIN SL(UuDaiKHTV)  SQL:cấp SL (UuDaiKHTV) và giữ khóa SL đến hết giao tác |
| Waitfor delay |  |  |  |
|  |  | Update(UuDaiKHTV)  Đợi | T2:XIN XL(UuDaiKHTV)  SQL:không cấp |
| Insert(UuDaiKHTV)  Đợi | T1:XIN XL(UuDaiKHTV)  SQL:không cấp |  |  |
| Deadlock |  |  |  |

TESTCASE18:

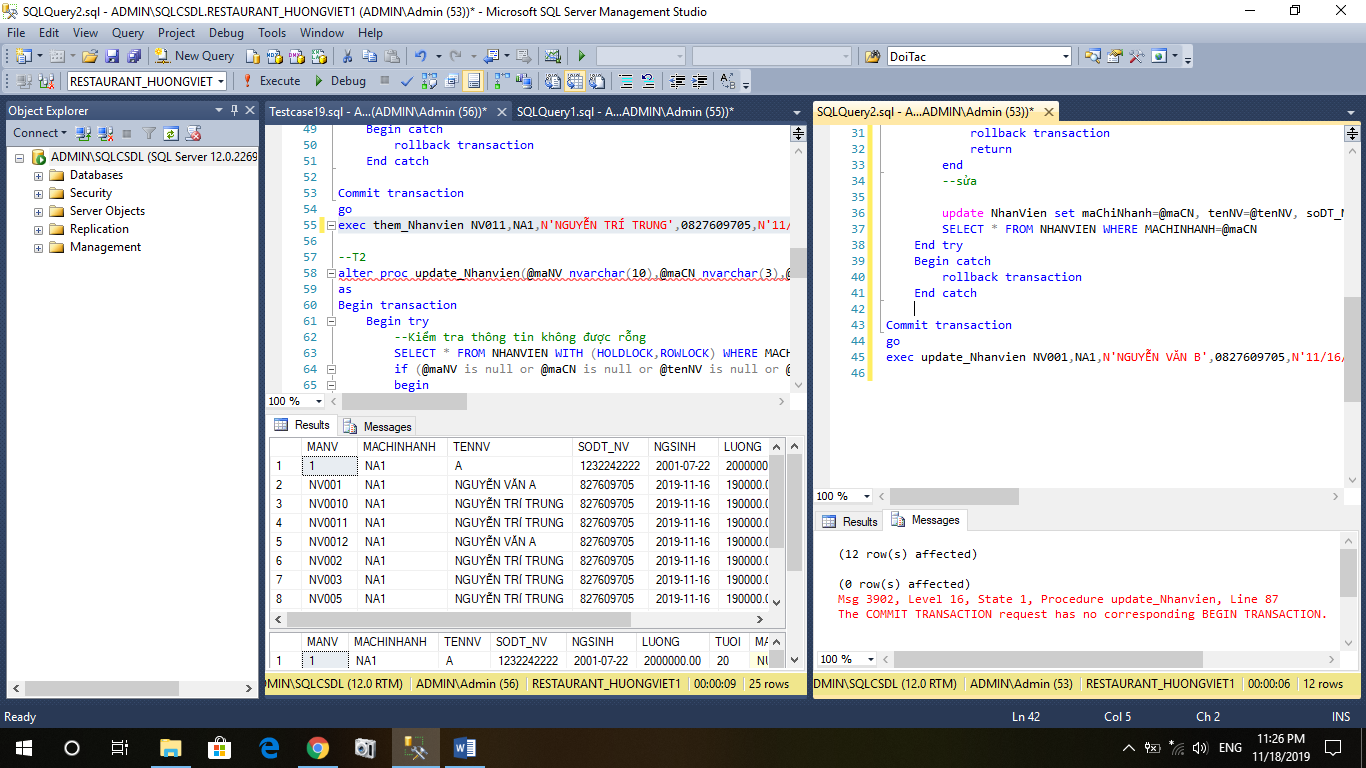


Một giao tác thực hiện việc insert dữ liệu và đọc 2 lần để kiểm tra xem việc insert đó có thành công không.Và ở đây thao tác đọc đầu tiên sẽ giữ holdlock để tránh việc unrepeatable read ,và giao tác thứ 2 cũng có 2 thao tác đọc để xem việc update có thành công không và cũng giữ holdlock,rowlock để tránh việc unrepeatable read.Và việc giao tác thứ nhất insert mà bị đợi bởi select (giữ đến cuối commit) và giao tác thứ 2 việc update cũng đợi select(giữ đến cuối commit) trên cũng 1 đơn vị dữ liệu dẫn đến 2 giao tác đợi nhau và từ đó bị deadlock

* ở đây việc giữu như vậy cũng giống cài mức cô lập số 3 nên ta sẽ giả sử đang ở mức cô lập số 3
* ***Mức cô lập mặc định là repeatable read***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(KHACHHANGTHANHVIEN) | T1:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và giữ khóa SL đến hết giao tác |  |  |
|  |  | SELECT(KHACHHANGTHANHVIEN) | T2:XIN SL(KHACHHANGTHANHVIEN)  SQL:cấp SL (KHACHHANGTHANHVIEN) và giữ khóa SL đến hết giao tác |
| Waitfor delay |  |  |  |
|  |  | Update(KHACHHANGTHANHVIEN)  Đợi | T2:XIN XL(KHACHHANGTHANHVIEN)  SQL:không cấp |
| DELETE(KHACHHANGTHANHVIEN)  Đợi | T1:XIN XL(KHACHHANGTHANHVIEN)  SQL:không cấp |  |  |
| Deadlock |  |  |  |

TESTCASE19:



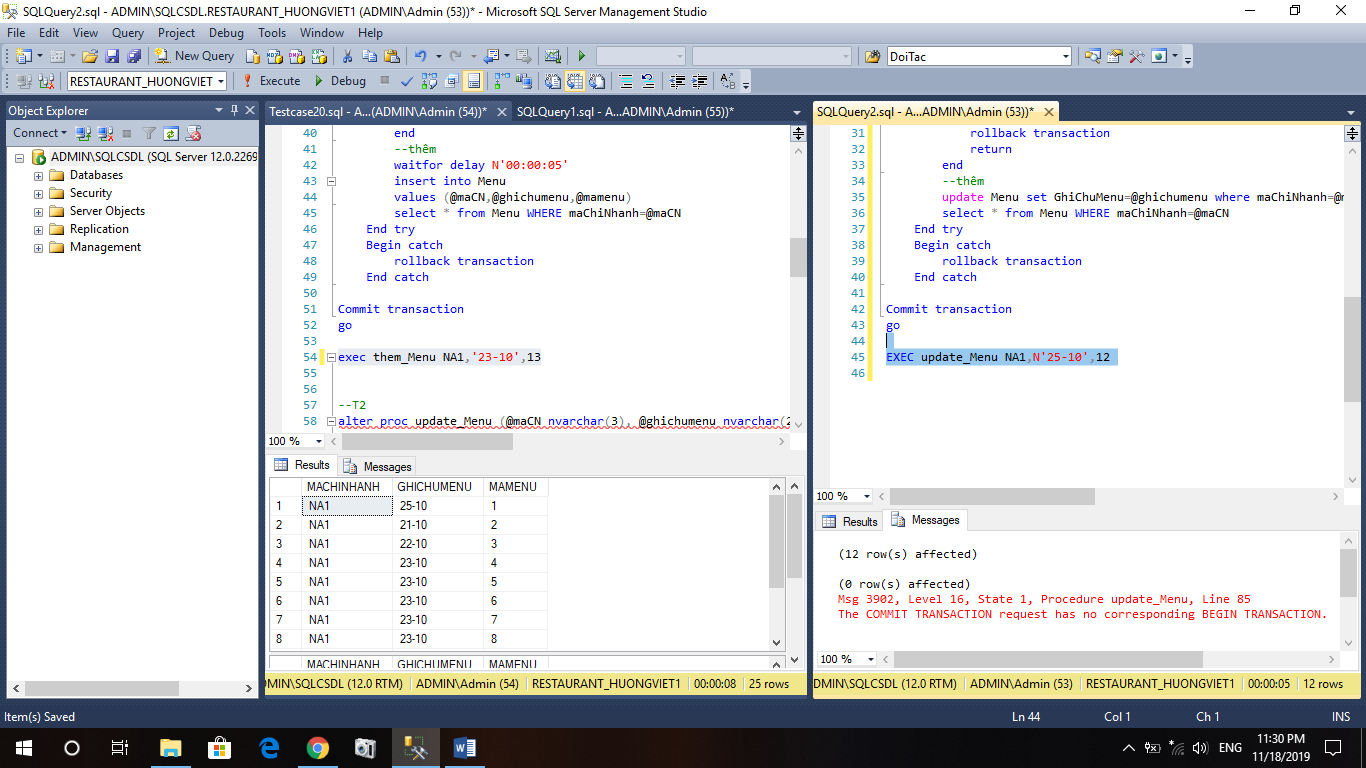
Một giao tác thực hiện việc insert dữ liệu và đọc 2 lần để kiểm tra xem việc insert đó có thành công không.Và ở đây thao tác đọc đầu tiên sẽ giữ holdlock để tránh việc unrepeatable read ,và giao tác thứ 2 cũng có 2 thao tác đọc để xem việc update có thành công không và cũng giữ holdlock,rowlock để tránh việc unrepeatable read.Và việc giao tác thứ nhất insert mà bị đợi bởi select (giữ đến cuối commit) và giao tác thứ 2 việc update cũng đợi select(giữ đến cuối commit) trên cũng 1 đơn vị dữ liệu dẫn đến 2 giao tác đợi nhau và từ đó bị deadlock

* ở đây việc giữ như vậy cũng giống cài mức cô lập số 3 nên ta sẽ giả sử đang ở mức cô lập số 3

***Mức cô lập mặc định là repeatable read***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(NHANVIEN) | T1:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và giữ khóa SL đến hết giao tác |  |  |
|  |  | SELECT(NHANVIEN) | T2:XIN SL(NHANVIEN)  SQL:cấp SL (NHANVIEN) và giữ khóa SL đến hết giao tác |
| Waitfor delay |  |  |  |
|  |  | Update(NHANVIEN)  Đợi | T2:XIN XL(NHANVIEN)  SQL:không cấp |
| Insert(NHANVIEN)  Đợi | T1:XIN XL(NHANVIEN)  SQL:không cấp |  |  |
| Deadlock |  |  |  |

TESTCASE20:



Một giao tác thực hiện việc insert dữ liệu và đọc 2 lần để kiểm tra xem việc insert đó có thành công không.Và ở đây thao tác đọc đầu tiên sẽ giữ holdlock để tránh việc unrepeatable read ,và giao tác thứ 2 cũng có 2 thao tác đọc để xem việc update có thành công không và cũng giữ holdlock,rowlock để tránh việc unrepeatable read.Và việc giao tác thứ nhất insert mà bị đợi bởi select (giữ đến cuối commit) và giao tác thứ 2 việc update cũng đợi select(giữ đến cuối commit) trên cũng 1 đơn vị dữ liệu dẫn đến 2 giao tác đợi nhau và từ đó bị deadlock

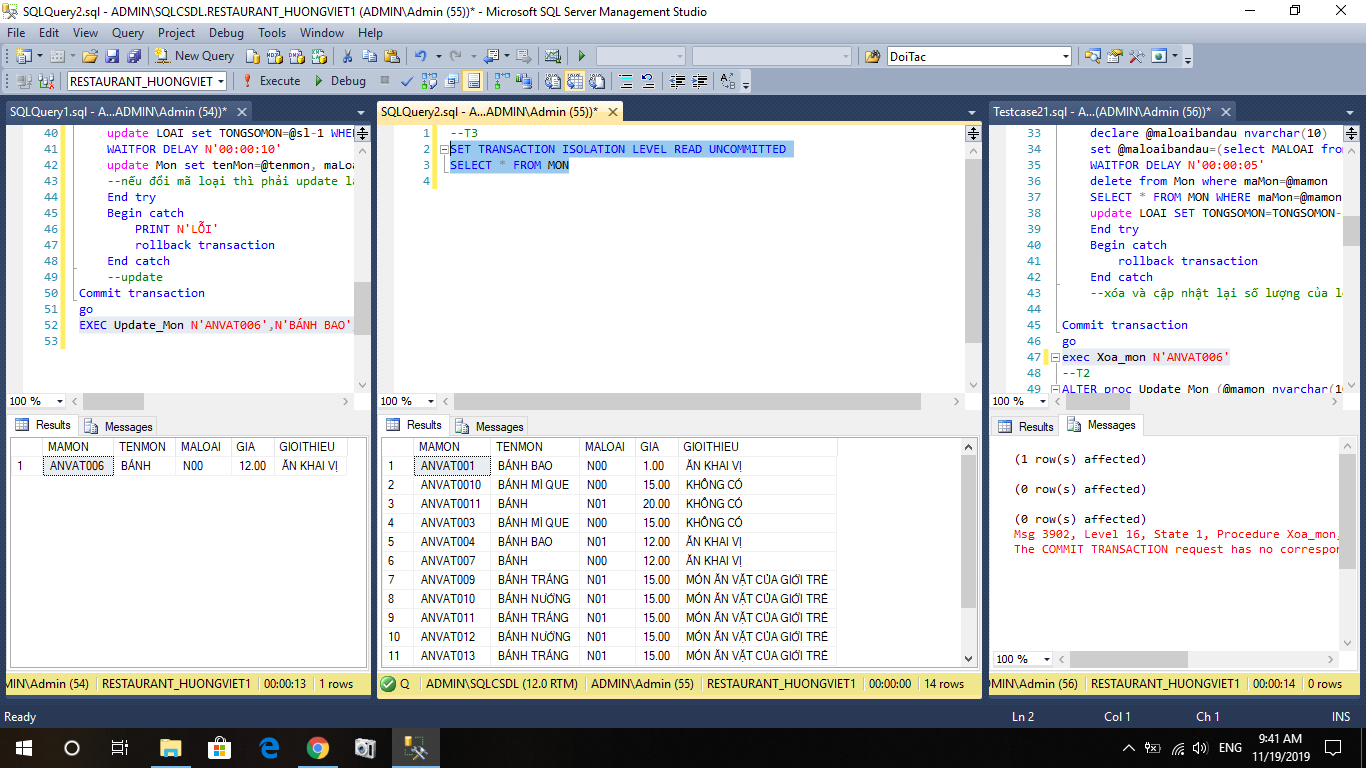
* ở đây việc giữ như vậy cũng giống cài mức cô lập số 3 nên ta sẽ giả sử đang ở mức cô lập số 3

***Mức cô lập mặc định là repeatable read***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(Menu) | T1:XIN SL(Menu)  SQL:cấp SL (Menu) và giữ khóa SL đến hết giao tác |  |  |
|  |  | SELECT(Menu) | T2:XIN SL(Menu)  SQL:cấp SL (Menu) và giữ khóa SL đến hết giao tác |
| Waitfor delay |  |  |  |
|  |  | Update(Menu)  Đợi | T2:XIN XL(Menu)  SQL:không cấp |
| Insert(Menu)  Đợi | T1:XIN XL(Menu)  SQL:không cấp |  |  |
| Deadlock |  |  |  |

**TRƯỜNG HỢP DIRTY READ:**

TESTCASE21:

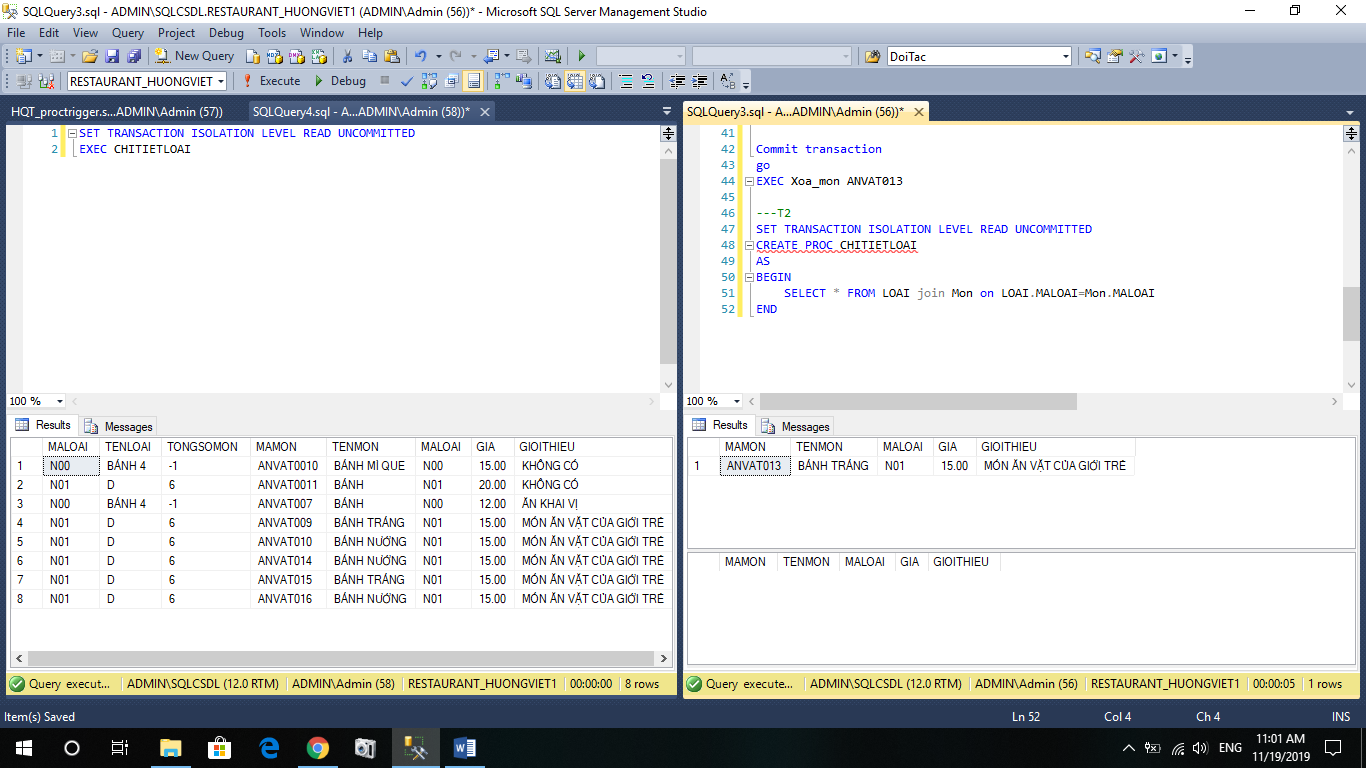


Một giao tác(T1) đang thực hiện việc delete món ăn thì bị delay mà chưa delete,một giao tác khác (T2) đang thực hiện việc update trên cùng đơn vị dữ liệu với T1.T2 thực hiện việc update Loại và bị delay ,rồi mới update món,còn T1 hết thời gian delay thì thực hiện việc delete món mà T2 sắp update nhưng chưa được vì bị delay.T1 thực hiện việc update Loại nhưng phải đợi T2 thực hiện xong,còn T2 thì đợi T1 thực hiện delete món🡪 2 giao tác đợi nhau bị Deadlock,và hủy một giao tác.Và có giao tác T3 muốn đọc ngay Món mà không đợi nên chọ mức cô lập số 1 và đọc.Nhưng cuối cùng T2 bị rollback nhưng T3 lại đọc dữ liệu đã được T2 xóa(thực ra việc xóa không diễn ra mà bị rollback)

***Mức cô lập mặc định là read commited***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| T1 | Khóa | T2 | Khóa | T3 | Khóa |
| SELECT(Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và nhả ngay khóa |  |  |  |  |
| Waitfor delay |  |  |  |  |  |
|  |  | SELECT(Mon) | T2:XIN SL(Mon)  SQL:cấp SL (Mon) và nhả ngay khóa |  |  |
|  |  | Update(Loai) | T2:XIN XL(Loai)  SQL:cấp XL (Loai) và giữ đến cuối giao tác |  |  |
|  |  | Waitfor delay |  |  |  |
| Delete(Mon) | T1:XIN XL(Mon)  SQL:cấp XL (Mon) và giữ đến cuối giao tác |  |  |  |  |
|  |  |  |  | SET ISOLATION READ UNCOMMITTED |  |
|  |  |  |  | Select(Mon)   * đọc được đơn vị dữ liệu xóa bởi T1 | T2:không cần xin vì mức cô lập này chưa có SL |
| Update(Loai)  Đợi | T1:XIN XL(Loai)  SQL:không cho do T2 đang giữ XL(Loai) |  |  |  |  |
|  |  | Update(Mon)  Đợi | T2:XIN XL(Mon)  SQL:không cho do T1 đang giữ XL(Mon) |  |  |
| T1 bị rollback |  |  |  |  |  |

TESTCASE22:

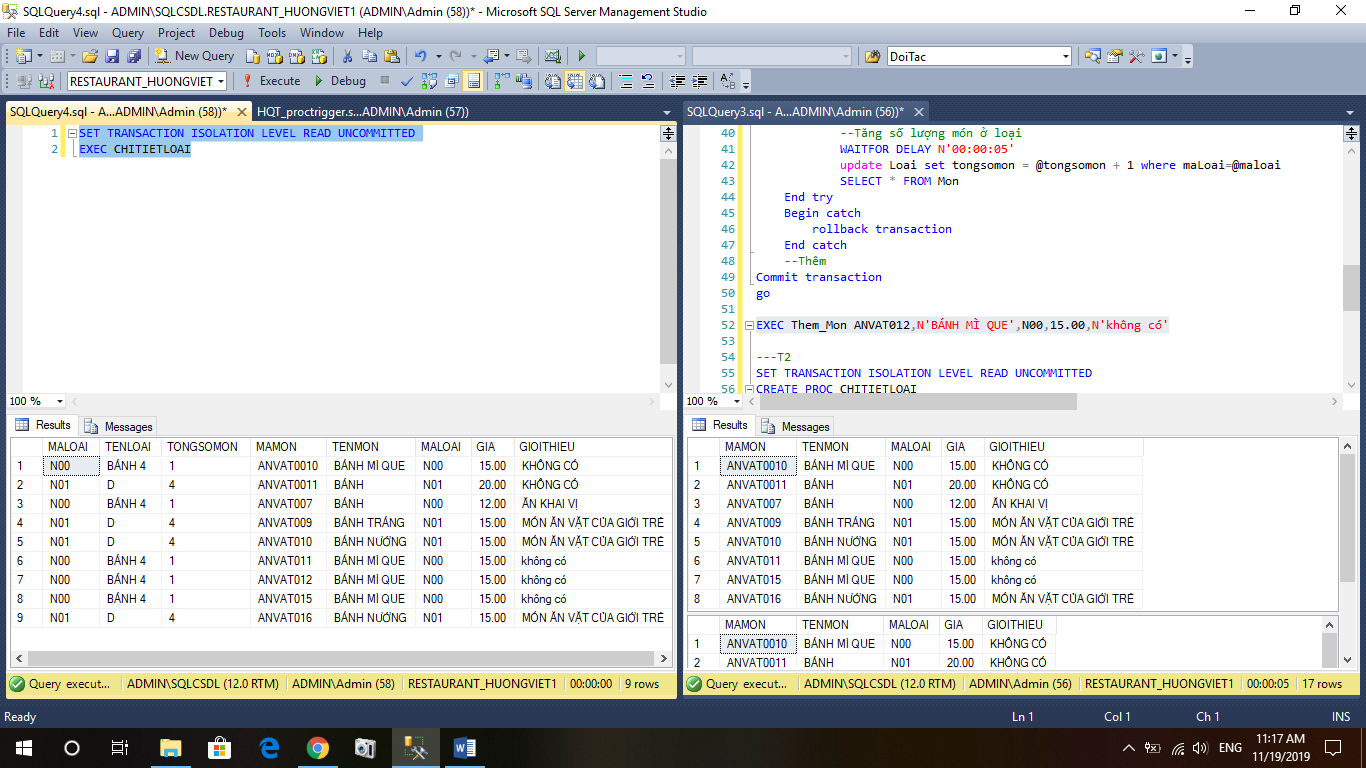


Một giao tác đang thực hiện việc xóa món ăn và update lại số lượng món ở loại nhưng chưa thực hiện update Loại thì bị delay và một giao tác khác thực hiện việc đọc dữ liệu(cùng đơn vị dữ liệu T1) ở mức cô lập số 1 ,và việc đọc tổng số món bị sai.

***Mức cô lập mặc định read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và giữ khóa SL đến hết giao tác |  |  |
| Delete(Mon) | T1:XIN XL(Mon)  SQL:cấp XL(Món ) và giữu đến cuối giao tác |  |  |
| Waitfor delay |  |  |  |
|  |  | Set isolation read uncommitted |  |
|  |  | Select(Mon,Loai) | T2:không cần xin vì mức cô lập này chưa có SL |
|  |  | commit |  |
| Update(Loai) | T1:XIN XL(Loai)  SQL:cấp XL(Loai ) và giữu đến cuối giao tác |  |  |
| SELECT(Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và giữ khóa SL đến hết giao tác |  |  |
| commit |  |  |  |

TESTCASE23:

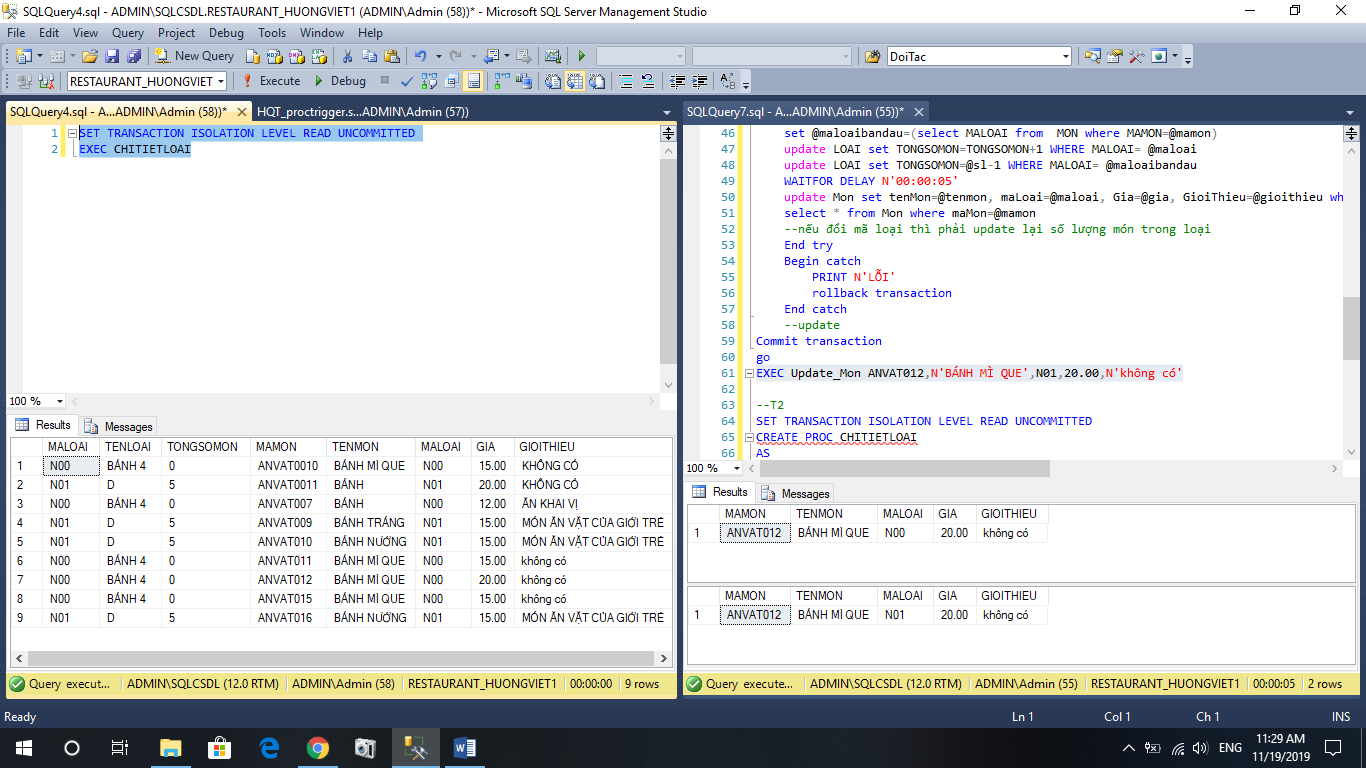


Một giao tác đang thực hiện việc thêm món ăn và update lại số lượng món ở loại nhưng chưa thực hiện update Loại thì bị delay và một giao tác khác thực hiện việc đọc dữ liệu(cùng đơn vị dữ liệu T1) ở mức cô lập số 1 ,và việc đọc tổng số món bị sai.

***Mức cô lập mặc định read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và giữ khóa SL đến hết giao tác |  |  |
| Insert(Mon) | T1:XIN XL(Mon)  SQL:cấp XL(Món ) và giữu đến cuối giao tác |  |  |
| Waitfor delay |  |  |  |
|  |  | Set isolation read uncommitted |  |
|  |  | Select(Mon,Loai) | T2:không cần xin vì mức cô lập này chưa có SL |
|  |  | commit |  |
| Update(Loai) | T1:XIN XL(Loai)  SQL:cấp XL(Loai ) và giữ đến cuối giao tác |  |  |
| SELECT(Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và giữ khóa SL đến hết giao tác |  |  |
| commit |  |  |  |

TESTCASE24:

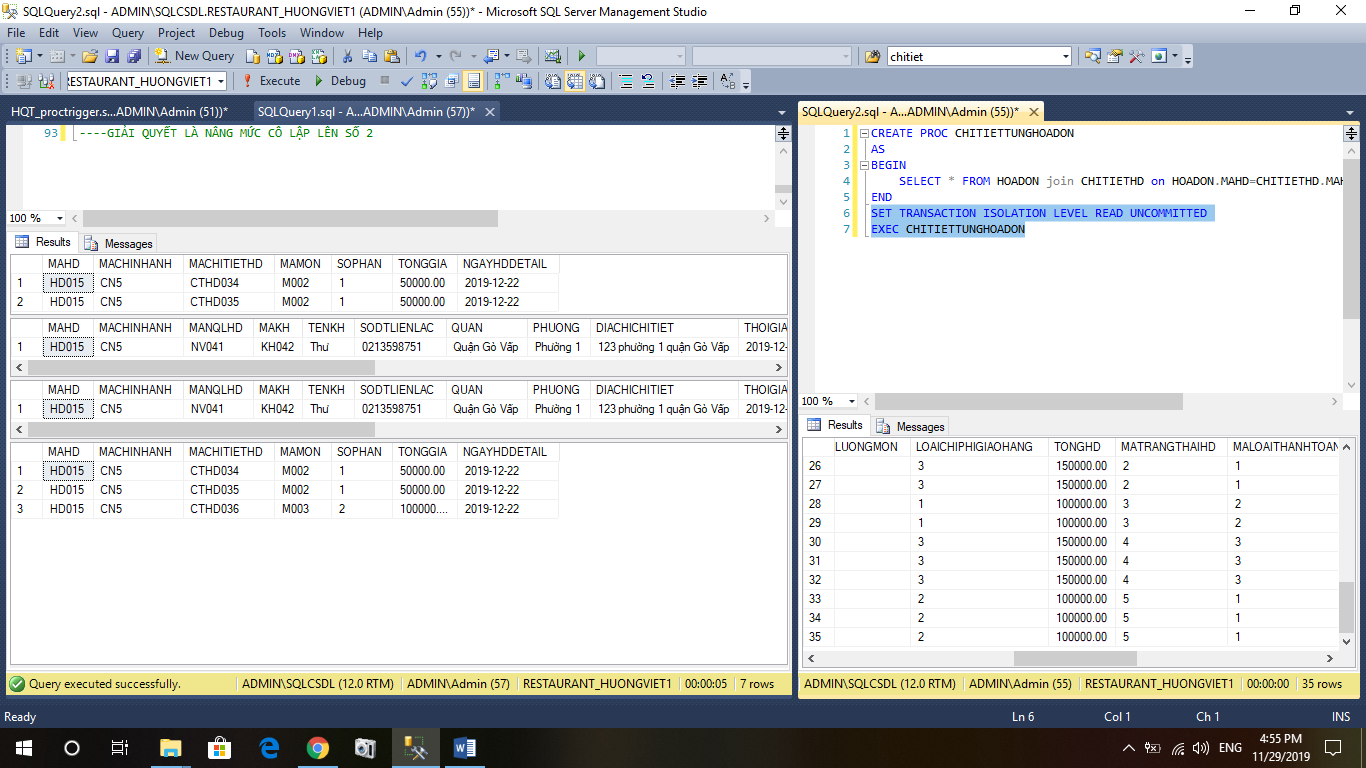


Một giao tác đang thực hiện việc update món ăn và update lại số lượng món ở loại nhưng chưa thực hiện update Mon mà mới chỉ update Loai thì bị delay và một giao tác khác thực hiện việc đọc dữ liệu(cùng đơn vị dữ liệu T1) ở mức cô lập số 1 ,và việc đọc tổng số món đúng nhưng Loại của món thì chưa được cập nhật.

***Mức cô lập mặc định read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và giữ khóa SL đến hết giao tác |  |  |
| Update(Loai) | T1:XIN XL(Loai)  SQL:cấp XL(Loai ) và giữu đến cuối giao tác |  |  |
| Waitfor delay |  |  |  |
|  |  | Set isolation read uncommitted |  |
|  |  | Select(Mon,Loai) | T2:không cần xin vì mức cô lập này chưa có SL |
|  |  | commit |  |
| Update(Mon) | T1:XIN XL(Mon)  SQL:cấp XL(Mon ) và giữ đến cuối giao tác |  |  |
| SELECT(Mon) | T1:XIN SL(Mon)  SQL:cấp SL (Mon) và giữ khóa SL đến hết giao tác |  |  |
| commit |  |  |  |

TESTCASE 25:



Một giao tác đang thực hiện việc thêm 1 chi tiết hóa đơn vào một đơn hàng và đã thêm xong nhưng bị delay ở trước khi cập nhật tổng hóa đơn và có giao tác T2 thực hiện việc kiểm tra hết tất cả hóa đơn lúc đó và thấy việc hóa đơn có 3 món tổng tiền phải là số mới cập nhật nhưng lại đọc tổng tiền trước đó dẫn tới đọc dữ liệu bị sai.Đây là lỗi Dirty read.

***Mức cô lập mặc định read committed***

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | KHÓA | T2 | KHÓA |
| SELECT(ChiTietHD  ) | T1:XIN SL(ChiTietHD  )  SQL:cấp SL (ChiTietHD  ) và nhả khóa ngay khi đọc xong |  |  |
| insert(ChiTietHD  ) | T1:XIN XL(ChiTietHD  )  SQL:cấp XL(ChiTietHD  ) và giữu đến cuối giao tác |  |  |
| Waitfor delay |  |  |  |
|  |  | Set isolation read uncommitted |  |
|  |  | Select(HoaDon, ChiTietHD) | T2:không cần xin vì mức cô lập này chưa có SL |
|  |  | commit |  |
| Update(update\_tongHD  ) | Chạy hàm cập nhật lại tổng hóa đơn |  |  |
| Update(ChiTietMenu) | T1:XIN SL(ChiTietMenu  )  SQL:cấp SL (ChiTietMenu  ) và giữ khóa SL đến hết giao tác |  |  |
| commit |  |  |  |