**ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**Khoa Công Nghệ Thông Tin**

**BÁO CÁO CUỐI KỲ**

**RESTAURANT SYSTEM**

**Giảng viên hướng dẫn:**

***Hoàng Anh Tú***

**Các thành viên trong nhóm 40:**

***Nguyễn Bá Ngọc – MSSV : 1141085***

***Bùi Trung Tân – MSSV :*** ***1141129***

***Nguyễn Minh Nghĩa – MSSV : 1141083***

***Nguyễn Minh Vũ – MSSV : 1141187***

## Phụ Lục

[1. Thông tin nhóm 1](#_Toc325069036)

[1.1 Thông tin thành viên: 1](#_Toc325069037)

[2. Các tranh chấp đồng thời đã phát hiện 1](#_Toc325069038)

[2.1 1141085 – Nguyễn Bá Ngọc 1](#_Toc325069039)

[2.1.1 Các trường hợp tranh chấp 1](#_Toc325069040)

[2.1.2 Xử lý các tranh chấp 2](#_Toc325069041)

[2.2 1141187 – Nguyễn Minh Vũ 2](#_Toc325069042)

[2.2.1 Các trường hợp tranh chấp 2](#_Toc325069043)

[2.2.2 Xử lý các tranh chấp 3](#_Toc325069044)

[2.3 1141083 – Nguyễn Minh Nghĩa 3](#_Toc325069045)

[2.3.1 Các trường hợp tranh chấp 3](#_Toc325069046)

[2.3.2 Xử lý các tranh chấp 3](#_Toc325069047)

[2.4 1141129 – Bùi Trung Tân 4](#_Toc325069048)

[2.4.1 Các trường hợp tranh chấp 4](#_Toc325069049)

[2.4.2 Xử lý các tranh chấp 4](#_Toc325069050)

# Thông tin nhóm

## Thông tin thành viên:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Nhóm trưởng** | **Đánh giá** |
| 1 | 1141085 | Nguyễn Bá Ngọc | X | 100% |
| 2 | 1141187 | Nguyễn Minh Vũ |  |  |
| 3 | 1141083 | Nguyễn Minh Nghĩa |  |  |
| 4 | 1141129 | Bùi Trung Tân |  |  |

# Các tranh chấp đồng thời đã phát hiện

## 1141085 – Nguyễn Bá Ngọc

### Các trường hợp tranh chấp

#### Drity read: Nghiệp vụ Đặt Bàn Ăn.

* ***Kịch bản:***

|  |  |
| --- | --- |
| T1 | T2 |
| * Thực hiện thao tác Thêm bàn đặt |  |
|  | * **Xem danh sách đặt bàn** |
| * Hủy thao tác Thêm bàn đặt |  |

* Khi T1 đang thực hiện thao tác Thêm 1 bàn đặt mà chưa Commit, cùng lúc đó T2 giữ khóa Read Uncommitted (đọc không cần xin khóa) đọc danh sách các bàn đã được đặt trong đó có bàn mà T1 mới đặt. Lỗi Drity Read xảy ra khi T1 hủy bản vừa mới đặt.
* **Sữa lỗi:**

|  |  |
| --- | --- |
| T1 | T2 |
| * begin tran * insert into DatBan (maNH,MaBan,MaKhachHang,   ThoiGianDat,TrangThai,ThoiGianDen,sdt) values(1,3,123654789,'05/12/2012',0,'05/12/2012',0) |  |
|  | * **SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED** * **begin tran** * **select \* from datban with (rowlock,readcommitted)** * **commit tran** |
| * waitfor delay '0:0:5' * declare @sdt int * select @sdt=sdt from DatBan where maban=3 * if @@error>0 or @sdt=0   rollback tran   * else commit tran |  |

* Khi T1 đang thực hiện thao tác Thêm 1 bàn đặt mà chưa Commit, cùng lúc đó T2 đọc danh sách bàn đã được đặt “With (rowlock,readcommitted)” (xin khóa đọc những dòng dữ liệu cần đọc trong 1 bảng và nhả khóa sau khi đọc xong). Như vậy: T2 đọc dữ liệu, nếu trong dữ liệu đó có dòng của T1 đã thêm thì T2 sẽ phải chờ T1 thực hiện xong mới thực hiện được việc đọc dữ liệu.

#### Lost update: Nghiệp vụ Đặt bàn ăn.

* ***Kịch bản:***

|  |  |
| --- | --- |
| T1 | T2 |
| * Đọc danh sách đặt bàn |  |
|  | * **Đọc danh sách đặt bàn.** |
| * Cập nhật tên khách hàng đã đặt bàn. |  |
|  | * **Cập nhật tên khách hàng đã đặt bàn.** |
| * Thực hiện xong |  |
|  | * **Thực hiện xong** |

* T1 đọc danh sách bàn đã được đặt, và cập nhật tên khách hàng của bàn số 3, cùng lúc đó T2 đọc danh sách bàn đã được đặt và cập nhật tên khách hàng khác cũng trên cùng bàn số 3, cùng bàn mà T1 đã sữa. Như vậy sau khi thực hiện xong, dữ liệu của T1 không giống như dữ liệu mà mình đã thay đổi trước đó. Lý do: vì T2 đã ghi đè lên dữ liệu của T1, nên dữ liệu của T1 bị mất khi cập nhật.
* ***Sữa lỗi:***

|  |  |
| --- | --- |
| T1 | T2 |
| * begin tran * select \* from DatBan with(rowlock,xlock)   where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,GETDATE(),101)  and TrangThai=0 and maBan=3   * waitfor delay '0:0:05' | * **begin tran** |
|  | * **select \* from DatBan**   **where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,GETDATE(),101) and TrangThai=0 and maBan=3**   * **waitfor delay '0:0:5'** |
| * update Datban set maKhachHang=123654789 where MaBan=3 |  |
|  | * **update Datban set maKhachHang=234234 where MaBan=3** |
| commit tran |  |
|  | **commit tran** |

* T1 đọc danh sách bàn đã được đặt với khóa *(rowlock,xlock)* nghĩa là đặt khóa ghi cho các dòng mà T1 muốn đọc, và cập nhật tên khách hàng của bàn số 3, cùng lúc đó T2 đọc bàn mà T1 đang đọc, T2 sẽ không được đọc, sẽ phải chờ cho đến khi T1 hoàn thành xong thì T2 mới thực hiện.

#### Unrepeatable read: Nghiệp vụ Đặt bàn ăn.

* ***Kịch bản:***

|  |  |
| --- | --- |
| T1 | T2 |
| * Đọc danh sách đặt bàn |  |
|  | * **Sửa tình trạng bàn đã đặt bàn thành bàn đang sử dụng** |
| * Đọc danh sách đặt bàn |  |

* T1 đọc danh sách đặt bàn lần 1, T2 cập nhật tình trạng bàn từ được đặt thành đang sử dụng, tiếp đó T1 đọc lại danh sách đặt bàn lần 2. Thì ta thấy rằng đọc lần 1 của T1 sẽ có kết quả khác với lần đọc thứ 2. Lần đọc 1 có số dòng nhiều hơn lần đọc 2.
* ***Sữa lỗi:***

|  |  |
| --- | --- |
| T1 | T2 |
| * begin tran * select MaBan from DatBan *with (rowlock,repeatableread)*   where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,  GETDATE(),101) and TrangThai=0   * waitfor delay '0:0:05' |  |
|  | * **begin tran** * **update DatBan set TrangThai=1**   **where MaNH=1 and MaBan=4 and ThoiGianDen=CONVERT(nvarchar,GETDATE(),101)**   * **commit tran** |
| * select MaBan from DatBan   where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,  GETDATE(),101) and TrangThai=0   * commit tran |  |

* T1 đọc danh sách đặt bàn lần 1 với khóa *(rowlock,repeatableread)* nghĩa là khi T1 đọc sẽ giữ khóa đọc những dòng T1 cần đọc cho đến hết transaction.Lúc này T2 thực hiện việc cập nhật dòng dữ liệu mà T1 đọc, T1 đang giữ khóa đọc nên T2 sẽ không cấp được khóa Ghi. T2 muốn thực hiện phải chờ T1 thực hiện xong thì T2 mới thực hiện được.

#### Phantom: Nghiệp vụ Đặt bàn ăn.

* ***Kịch bản:***

|  |  |
| --- | --- |
| T1 | T2 |
| * Đọc danh sách đặt bàn |  |
|  | * **Thêm bàn đặt** |
| * Đọc danh sách đặt bàn |  |

* T1 đọc danh sách đặt bàn lần 1, T2 thực hiện thao tác thêm 1 bàn đặt, tiếp đó T1 đọc lại danh sách đặt bàn lần 2. Thì ta thấy rằng đọc lần 1 của T1 sẽ có kết quả khác với lần đọc thứ 2. Lần đọc 1 có số dòng ít hơn lần đọc 2.
* ***Sữa lỗi:***

|  |  |
| --- | --- |
| T1 | T2 |
| * begin tran * select MaBan from DatBan with (rowlock,serializable)   where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,  GETDATE(),101) and TrangThai=0   * waitfor delay '0:0:05' |  |
|  | * **begin tran** * **insert into DatBan (maNH,MaBan,MaKhachHang,ThoiGianDat,**   **TrangThai,ThoiGianDen,sdt)**  **values(1,2,123456756,'05/10/2012',0,CONVERT(nvarchar,GETDATE(),101),234234)**   * **commit tran** |
| * select MaBan from DatBan   where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,  GETDATE(),101) and TrangThai=0   * commit tran |  |

* T1 đọc danh sách đặt bàn lần 1 với khóa *(rowlock,* *serializable)* nghĩa là khi T1 đọc sẽ giữ khóa đọc những dòng T1 cần đọc cho đến hết transaction.Lúc này T2 thực hiện việc Thêm dữ liệu, T1 đang giữ khóa đọc nên T2 sẽ không cấp được khóa Ghi. T2 muốn thực hiện phải chờ T1 thực hiện xong thì T2 mới thực hiện được.

#### Deadlock: Nghiệp vụ Xếp bàn ăn cho khách.

* ***Kịch bản:***

|  |  |
| --- | --- |
| T1 | T2 |
| * Đọc danh sách bàn ăn |  |
|  | * **Đọc danh sách bàn ăn** |
| * Cập nhật tình trạng bàn ăn |  |
|  | * **Cập nhật tình trạng bàn ăn** |

* T1 đọc danh sách bàn ăn và giữ khóa đọc đến hết giao tác, T2 cũng đọc danh sách bàn ăn và giữ khóa đọc đến hết giao tác. T1 cập nhật tình trạng bàn ăn nên phải chờ cho T2 nhả khóa đọc trên bảng bản ăn, cùng lúc đó T2 cũng cập nhật tình trạng bàn ăn và cũng phải chờ cho đến khi T1 nhả khóa đọc trong bảng bàn ăn. Nên cả hai cùng chờ nhau gây lỗi conversion deadlock
* ***Sữa lỗi:***

|  |  |
| --- | --- |
| T1 | T2 |
| * SET TRANSACTION ISOLATION LEVEL repeatable read * begin tran * select \* from DatBan with (rowlock,xlock)   where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,  GETDATE(),101) and TrangThai=0   * waitfor delay '0:0:05' |  |
|  | * **SET TRANSACTION ISOLATION LEVEL**   **repeatable read**   * **begin tran** * **select \* from DatBan with (rowlock,xlock)**   **where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,**  **GETDATE(),101) and TrangThai=0**   * + **waitfor delay '0:0:5'** |
| * + update Datban set TinhTrang=1 where MaBan=2 |  |
| * commit tran | * **update Datban set TinhTrang=1 where MaBan=2** |
|  | * **Commit tran** |

* T1 đọc danh sách bàn ăn và giữ khóa ghi đến hết giao tác, T2 cũng đọc danh sách bàn ăn nhưng T2 phải chờ cho đến khi T1 hoàn thành giao tác, nên sẽ không xảy ra hiện tượng conversion Deadlock.

1. **Các chức năng trong đề tài**
   1. **Nhân Viên Tiếp Tân**
   2. **Nhân Viên Thu Ngân**
   3. **Nhân Viên Quản Lý Kho**
   4. **Quản Lý Nhà Hàng**