Converson Deadlock:

Test case 1:

Mô tả tình huống tranh chấp:

Giả sử khi đơn hàng có mã *X* xuất một kho (Ví dụ tên kho là: Mê linh, Hà Nội), lúc này đối tác tìm đơn hàng có mã đơn hàng là *X*  đó để kiểm tra đơn hàng có trong kho không, sau đó đối tác sẽ cập nhật tình trạng đơn hàng *X*  thành “Đơn hàng đã xuất kho Mê Linh, Hà Nội”. Tuy nhiên cùng lúc đó nhận hàng, thì tài xế sẽ đọc lên thông tin giao hàng để lên kế hoạch đi giao đồng thời cập nhật tình trạng đơn hàng thành “Tài xế nhận hàng”.

|  |  |  |
| --- | --- | --- |
| T1 \_ Đối tác | T2 \_ Tài xế | Ghi Chú |
| SET TRAN ISOLATION LEVEL SERIALIZABLE | | Gán mức cô lập Serializable cho cả hai giao tác. |
| Shared\_Lock(DON\_HANG) |  | Đối tác thành công xin khóa đọc trên bảng đơn hàng vì bảng đó chưa có khóa nào chiếm. |
| IF EXISTS (Select \* From DON\_HANG Where MaDT = @madt AND MaDH = @madh); |  | Đối tác đọc trên bảng đơn hàng thành công vì đã có đã chiếm được shared lock trước đó. |
|  | Shared\_lock(DON\_HANG) | Tài xế thành công xin khóa đọc trên bảng đơn hàng. Vì bảng đơn hàng đã bị chiếm với khóa shared lock nên có thể chiếm giữ khóa S cùng lúc được. |
|  | Select TenDuong From DON\_HANG Where MaDH = @madh | Tài xế đọc đơn hàng thành công vì đã chiếm khóa đọc trước đó. |
|  | Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG (TinhTrangDH)  SET TinhTrangDH = “Tài xế đã nhận đơn hàng”  WHERE MaDH = @madh | Tài xế không thể xin được khóa ghi vì bảng đơn hàng đã bị chiếm khóa đọc trước bởi T1 và cả T2 trước đó. Mà do 2 transaction đều đang ở mức cô lập Serializable nên hai khóa đều được giữ đến cuối giao tác. Lúc này giao tác 2 chưa hoàn thành nên chưa trả khóa đọc. |
| Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG (TinhTrangDH)  SET TinhTrangDH = “Đã xuất kho Mê Linh Hà Nội”  WHERE MaDH = @madh |  | Đối tác cũng không thể cập nhật được tình trạng đơn hàng vì không xin được khóa ghi do bảng đơn hàng đã bị chiếm bởi khóa đọc. |

Cycle Deadlock:

TEST CASE 2:

Mô tả tình huống tranh chấp:

Giả sử khi tài xế chọn một đơn hàng để giao, tài xế sẽ thêm đơn hàng đó vào bảng giao hàng đồng thời tài xế cũng sẽ cập nhật tình trạng đơn hàng là “Đã nhận hàng”. Cùng lúc đó thì đối tác cũng cập nhật tình trạng đơn hàng là “Đã xuất kho” đồng thời cũng đọc bảng giao hàng để biết được tài xế nào đang giao đơn hàng đó.

|  |  |  |
| --- | --- | --- |
| T1\_ tài xế | T2\_đối tác | Ghi chú |
| SET TRAN ISOLATION LEVEL SERIALIZABLE | | Gán mức cô lập Serializable cho cả hai giao tác. |
| Exclusive\_Lock(GIAO\_HANG) |  | Tài xế xin khóa ghi thành công vì bảng giao hàng chưa bị chiếm bởi khóa nào. |
| INSERT INTO GIAO\_HANG (MaDH, MaTX)  VALUES (@madh,@matx) |  | Tài xế thêm thành công đơn hàng mình chọn vào bảng giao hàng vì đã chiếm được khóa ghi trước đó |
|  | Exclusive\_Lock(DON\_HANG) | Đối tác chiếm thành công khóa đọc trên bảng đơn hàng vì bảng đơn hàng chưa bị chiếm bởi khóa nào. |
|  | UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã xuất kho’  WHERE MaDH = @madh | Đối tác cập nhật thành công tình trạng đơn hàng vì đã chiếm được khóa độc quyền trước đó. |
|  | Shared\_Lock(GIAO\_HANG)  SELECT MaTX  FROM GIAO\_HANG  WHERE MaDH = @madh | Đối tác xin khóa chia sẽ trên bảng Giao\_Hàng không thành công vì bảng giao hàng đã bị T1 chiếm khóa độc quyền trước đó và T1 vẫn chưa trả khóa do giao tác T1 vẫn chưa hoàn thành xong |
| Exclusive\_Lock(DON\_HANG)  UPDATE DON\_HANG  SET TinhTrangDH = “Đã nhận đơn”  WHERE MaDH = @madh |  | Tài xế xin khóa độc quyền trên bảng đơn hàng thất bại vì bảng đơn hàng đã bị chiếm khóa đọc quyền bởi T2 và do T2 chưa hoàn thành nên T2 vẫn chưa trả khóa độc quyền trên bàng đơn hàng |