**UNREPEATABLE READ**

**Test case 1:**

**Hướng xử lý:** ta đặt khóa độc quyền (XLOCK) khi đọc dữ liệu trong bảng DON\_HANG lần đầu tiên cho cả tài xế và đối tác để ngăn không cho các thao tác khác can thiệp vào giữa giai đoạn cập nhật làm thay đổi dữ liệu ở bảng DON\_HANG(cụ thể ở đây là tình trạng đơn hàng).

|  |  |  |
| --- | --- | --- |
| **T1 – Tài xế** | **T2 – Đối tác** | **Ghi Chú** |
| SELECT \* FROM GIAO\_HANG WHERE MaDH = ‘123’ AND MaDT = ‘xyz’ |  | T1 kiểm tra xem mã đơn hàng có khớp với mã đối tác không |
| SELECT TinhTrangDH FROM DON\_HANG **WITH(XLOCK)** WHERE MaDH = ‘123’ |  | **T1 yêu cầu khóa độc quyền trên bảng DON\_HANG**  T1 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa  **Tình trạng đơn hàng được select lúc này là: “Đang giao hàng”** |
|  | SELECT \* FROM DON\_HANG **WITH(XLOCK)** WHERE MaDH = ‘123’ AND MaDT = ‘xyz’  **=> CHỜ** | **T2 yêu cầu khóa độc quyền trên bảng DON\_HANG nhưng do T1 đang giữ khóa nên T2 phải chờ** |
| SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ |  | T1 kiểm tra tình trạng đơn hàng xem có phải là “Đã giao hàng” không, nếu phải thì chỉ được cập nhật là “Đã hoàn trả hàng”  **Tình trạng đơn hàng được select lúc này vẫn là: “Đang giao hàng”** |
| UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã giao hàng’  WHERE MaDH = ‘123’ |  | T1 cập nhật tình trạng đơn hàng thành “Đã giao hàng” |
| COMMIT |  | **T1 commit và nhả khóa**  **Tình trạng đơn hàng sau khi T1 commit là: “Đã giao hàng”** |
|  | SELECT \* FROM DON\_HANG **WITH(XLOCK)** WHERE MaDH = ‘123’ AND MaDT = ‘xyz’ | **T2 yêu cầu khóa độc quyền trên bảng DON\_HANG => lấy được khóa do T1 đã nhả khóa**  T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa |
|  | SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ | T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã hoàn trả hàng” không, nếu phải thì không cập nhật nữa.  **Tình trạng đơn hàng được select lúc này là: “Đã giao hàng”** |
|  | SELECT TinhTrangDH FROM DON\_HANG WHERE MaDH = ‘123’ | T2 kiểm tra tình trạng đơn hàng xem có phải là “Đã giao hàng” không, nếu phải thì chỉ được cập nhật là “Đã hoàn trả hàng”  **Tình trạng đơn hàng được select lúc này là: “Đã giao hàng”** |
|  | UPDATE DON\_HANG  SET TinhTrangDH = ‘Đã hoàn trả hàng’  WHERE MaDH = ‘123’ | T2 cập nhật tình trạng đơn hàng thành “Đã hoàn trả hàng” |
|  | COMMIT | T2 commit  **Tình trạng đơn hàng sau khi T2 commit là: “Đã hoàn trả hàng”** |

**Test case 2:**

**Hướng xử lý:** ta đặt khóa độc quyền (XLOCK) khi đọc dữ liệu trong bảng HOP\_DONG lần đầu tiên để ngăn không cho các thao tác khác can thiệp vào giữa giai đoạn cập nhật làm thay đổi dữ liệu ở bảng HOP\_DONG(cụ thể ở đây là thời gian hiệu lực hợp đồng).

|  |  |  |
| --- | --- | --- |
| **T1 – Nhân viên A** | **T2 – Nhân viên B** | **Ghi Chú** |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON **WITH(XLOCK)** (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ |  | **T1 yêu cầu khóa độc quyền trên bảng HOP\_DONG**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T1 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON **WITH(XLOCK)** (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’  **=> CHỜ** | **T2 yêu cầu khóa độc quyền trên bảng HOP\_DONG nhưng phải chờ do T1 đang giữ khóa** |
| SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ |  | T1 so sánh xem thời gian hiệu lực truyền vào từ input có hợp lệ không bằng cách so với thời gian hiệu lực hiện tại trong cơ sở dữ liệu  **Lúc này, thời gian hiệu lực được select từ cơ sở dữ liệu vẫn giống với lần select trước đó của T1** |
| SET @doanhsoban = (SELECT SUM(dh.TongPhiSP) FROM DON\_HANG dh JOIN DOI\_TAC dt ON (dh.MaDT = dt.MaDT) WHERE dt.MaSoThue = ‘123’) |  | T1 tính toán doanh số bán để cập nhật phần trăm hoa hồng |
| UPDATE HOP\_DONG  SET TG\_HieuLucHD = @tg\_hlhd, PhanTramHoaHong = (@pthh \* @doanhsoban) / 100  WHERE MaDT IN (SELECT MaDT FROM DOI\_TAC WHERE MaSoThue = ‘123’) |  | T1 cập nhật thời gian hiệu lực và phần trăm hoa hồng |
| COMMIT |  | **T1 commit và nhả khóa** |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON **WITH(XLOCK)** (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ | **T2 yêu cầu khóa độc quyền trên bảng HOP\_DONG => lấy được khóa do T1 vừa nhả khóa**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T2 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.TG\_HieuLucHD FROM DOI\_TAC d JOIN HOP\_DONG h ON (d.MaDT = h.MaDT) WHERE d.MaSoThue = ‘123’ | T2 so sánh xem thời gian hiệu lực truyền vào từ input có hợp lệ không bằng cách so với thời gian hiệu lực hiện tại trong cơ sở dữ liệu  **=> T2 rollback do T1 vừa cập nhật lại thời gian hiệu lực hợp đồng thêm 3 tháng nhưng T2 lại muốn cập nhật chỉ thêm 2 tháng** |
|  | T2 ROLLBACK |  |

**PHANTOM READ**

**Test case 1:**

**Hướng xử lý:** ta đặt khóa độc quyền (XLOCK) khi đọc dữ liệu trong bảng HOP\_DONG lần đầu tiên để ngăn không cho các thao tác khác can thiệp vào giữa chừng làm thay đổi dữ liệu ở bảng HOP\_DONG (cụ thể ở đây là ngăn thao tác xóa của T2 can thiệp vào giữa giai đoạn xóa của T1 làm cho 2 lần SELECT ra số dòng dữ liệu không giống nhau).

|  |  |  |
| --- | --- | --- |
| **T1 – Nhân viên A** | **T2 – Nhân viên B** | **Ghi Chú** |
| SELECT h.MaHD FROM HOP\_DONG h **WITH(XLOCK)** WHERE MaDT = ‘abc’ |  | **T1 yêu cầu khóa độc quyền trên bảng HOP\_DONG**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T1 kiểm tra thì thấy select ra được 1 dòng dữ liệu => hợp đồng có tồn tại |
|  | SELECT h.MaHD FROM HOP\_DONG h **WITH(XLOCK)** WHERE MaDT = ‘abc’  **=> CHỜ** | **T2 yêu cầu khóa độc quyền trên bảng HOP\_DONG nhưng phải chờ do T1 đang giữ khóa** |
| SET @mahd = (SELECT h.MaHD FROM HOP\_DONG h WHERE MaDT = ‘abc’) |  | T1 select để lấy mã hợp đồng ứng với mã đối tác được truyền vào |
| DELETE FROM CT\_HOPDONG  WHERE MaHD = @mahd |  | T1 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng chi tiết hợp đồng |
| DELETE FROM HOP\_DONG  WHERE MaHD = @mahd |  | T1 xóa các dòng dữ liệu ứng với mã hợp đồng vừa select ở trên trong bảng hợp đồng |
| COMMIT |  | **T1 commit và nhả khóa** |
|  | SELECT h.MaHD FROM HOP\_DONG h **WITH(XLOCK)** WHERE MaDT = ‘abc’ | **T2 yêu cầu khóa độc quyền trên bảng HOP\_DONG thành công do T1 vừa nhả khóa trước đó**  Kiểm tra xem hợp đồng có tồn tại hay không, nếu không thì rollback  T2 kiểm tra thì thấy select ra được số dòng dữ liệu là 0 => hợp đồng không tồn tại do T1 đã xóa trước đó => T2 rollback |
|  | T2 ROLLBACK |  |