## Tình huống 1: Kiểm tra hơp đồng

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Unrepeatable Data**  T1 (User = Nhân viên A): Kiểm tra Hợp đồng đã hết hạn  T2 (User = Nhân viên B): Gia hạn 1 hợp đồng | | | |
| **sp\_HopDongHetHan** | **Khóa** | **sp\_GiaHanHopDong** | **Khóa** |
| ***Input:***  ***Output***: | ***Input***: @*MaHopDong varchar(10), @new date*  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra những hợp đồng đã hết hạn  SELECT \*  FROM HopDong  WHERE ThoiGianHieuLuc < GETDATE() | // Không cần xin khoá |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1:Kiểm tra hợp đồng có tồn tại hay không  IF NOT EXISTS(select \* from HopDong where MaHopDong = @MaHopDong)  BEGIN  PRINT N'HopDong' + CAST(@MaHopDong AS VARCHAR(10)) + N' Không Tồn Tại'  ROLLBACK TRAN  RETURN 1  END |  |
|  |  | B2: Gia hạn hợp đồng  UPDATE HopDong SET ThoiGianHieuLuc = @newWHERE MaHopDong = @MaHopDong | **X(HopDong)**  // Xin khoá ghi trên bảng Hợp đồng |
| B2: Xuất số lượng hợp đồng hết hạn  SELECT @HDHH = COUNT(\*)  FROM HopDong  WHERE ThoiGianHieuLuc < GETDATE() | **S(HopDong)**  // Xin khoá đọc trên bảng Hợp đồng |  |  |
| COMMIT |  |  |  |
|  |  | COMMIT |  |

## Tình huống 2: Đọc thông báo và lập hợp đồng

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Deadlock Cycle**  T1 (User = Nhân viên): Gửi thông báo gia hạn tới một đối tác và gia hạn hợp đồng  T2 (User = Đối tác): Lập hợp đồng và đọc thông báo | | | |
| **Sp\_GuiThongBao\_DuyetHopDong** | **Khóa** | **sp\_LapHopDong\_DocThongBao** | **Khóa** |
| ***Input:*** @MaHopDong varchar(10), @NoiDung nvarchar(50), @MaSoThue varchar(10), @MaThongBao varchar(10), @ThoiHan date  ***Output***: | ***Input***: @MaSoThue varchar(10), @MaHopDong varchar(10)  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** |
| BEGIN TRAN |  |  |  |
| INSERT INTO ThongBao VALUE (@MaThongBao, @NoiDung, @MaSoThue) | **X(ThongBao)**  // Xin khoá ghi trên bảng thông báo |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | INSERT INTO HopDong(MaHopDong, MaSoThue)  VALUES (@MaHopDong, @MaSoThue) | **X(HopDong)**  // Xin khoá ghi trên bảng Hợp đồng |
|  |  | WAITFOR DELAY '00:00:05' |  |
| IF NOT EXISTs(select \* from HopDong where MaHopDong = @MaHopDong)  BEGIN  PRINT N'Hợp đồng' + CAST(@MaHopDong AS VARCHAR(10)) + N' Không Tồn Tại'  ROLLBACK TRAN  RETURN 1  END | **S(HopDong)**  // Xin khoá đọc trên bảng Hợp đồng |  |  |
| UPDATE HopDong SET TinhTrang = N’Đã Duyệt’WHERE MaHopDong = @MaHopDong  UPDATE HopDong SET ThoiGianHieuLuc = N’Đã Duyệt’WHERE MaHopDong = @MaHopDong | **X(HopDong)**  // Xin khoá ghi trên bảng Hợp đồng |  |  |
|  |  | SELECT NoiDung  FROM ThongBao  WHERE MaSoThue = @MaSoThue | **S(ThongBao)**  // Xin khoá đọc trên bảng thông báo |
| COMMIT TRAN |  |  |  |
|  |  | COMMIT TRAN |  |

## Tình huống 3: Hai tài xế cùng nhận chung một đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Lost update**  T1 (User = Tài xế A): Xem và nhận đơn hàng  T2 (User = Tài xế B): Xem và nhận đơn hàng | | | |
| **sp\_NhanDonHang** | **Khóa** | **sp\_NhanDonHang** | **Khóa** |
| ***Input:*** *@MaTaiXe varchar(10), @MaDonHang varchar(10)*  ***Output***: | ***Input***: *@MaTaiXe varchar(10), @MaDonHang varchar(10)*  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **READ COMMITTED** | SET TRANSACTION ISOLATION  LEVEL **READ COMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra tài xế có tồn tại hay không  IF NOT EXISTS (select \* from TaiXe where MaTaiXe = @MaTaiXe)  BEGIN  PRINT N'Tài Xế' + CAST(@MaTaiXe AS VARCHAR(10)) + N' Không Tồn Tại'  ROLLBACK TRAN  RETURN 1  END | **S(Tài Xế)**  //Xin khoá đọc trên bảng Tài xế |  |  |
| B2: Kiểm tra đơn hàng có thuộc khu vực của tài xế  IF NOT EXISTS  (select \* from DonHang,TaiXe  where DonHang.KhuVuc = KhuVucHoatDong  and Taixe.MaTaiXe = @MaTaiXe  and TinhTrang = N'Chờ'  and MaDonHang = @MaDonHang)  BEGIN  PRINT N'Đặt hàng không khả thi'  ROLLBACK TRAN  RETURN 1  END | **S(Đơn hàng)**  //Xin khoá đọc trên bảng Tài xế |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra tài xế có tồn tại hay không  IF NOT EXISTS (select \* from TaiXe where MaTaiXe = @MaTaiXe)  BEGIN  PRINT N'Tài Xế' + CAST(@MaTaiXe AS VARCHAR(10)) + N' Không Tồn Tại'  ROLLBACK TRAN  RETURN 1  END | **S(DonHang)**  //Xin khoá đọc trên bảng đơn hàng |
|  |  | B2: Kiểm tra đơn hàng có thuộc khu vực của tài xế  IF NOT EXISTS  (select \* from DonHang,TaiXe  where DonHang.KhuVuc = KhuVucHoatDong  and Taixe.MaTaiXe = @MaTaiXe  and TinhTrang = N'Chờ'  and MaDonHang = @MaDonHang)  BEGIN  PRINT N'Đặt hàng không khả thi'  ROLLBACK TRAN  RETURN 1  END | **S(Đơn hàng)**  //Xin khoá đọc trên bảng Tài xế |
|  |  | WAITFOR DELAY '00:00:05' |  |
| B3: Đặt hàng  update DonHang set TinhTrang = N'Đang Giao' where @MaDonHang = MaDonHang  update DonHang set MaTaiXe = @MaTaiXe where @MaDonHang = MaDonHang | **X(DonHang)**  //Xin khoá ghi trên bảng đơn hàng |  |  |
|  |  | B3: Đặt hàng  update DonHang set TinhTrang = N'Đang Giao' where @MaDonHang = MaDonHang  update DonHang set MaTaiXe = @MaTaiXe where @MaDonHang = MaDonHang | **X(DonHang)**  //Xin khoá ghi trên bảng đơn hàng |
| COMMIT |  |  |  |
|  |  | COMMIT |  |

## Tình huống 4: Khách hàng đặt hàng thất bại nhưng đối tác vẫn thấy đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Dirty Read**  T1 (User = Khách Hàng): Đặt hàng  T2 (User = Đối tác): Xem số lượng đơn hàng trong ngày | | | |
| **sp\_DatHang** | **Khóa** | **sp\_XemSoLuongDonHang** | **Khóa** |
| ***Input:*** @MaSoThue varchar(10), @MaKhachHang varchar(10),  @MaDonHang varchar(10), @KhuVuc nvarchar(50)  ***Output***: |  | ***Input***: @MaSoThue  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **READ COMMITTED** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin về đối tác  IF NOT EXISTS (select \* from DoiTac where MaSoThue = @MaSoThue)  BEGIN  PRINT N'Đối tác' + CAST(@MaSoThue AS VARCHAR(10)) + N' Không Tồn Tại'  ROLLBACK TRAN  RETURN 1  END |  |  |  |
| INSERT INTO DonHang (MaDonHang, MaSoThue, MaKhachHang, TinhTrang, KhuVuc, NgayDat)  values  (@MaDonHang, @MaSoThue, @MaKhachHang, N'Chờ', @KhuVuc, getdate()) | **X(Donhang)**  //xin khoá ghi trên bảng đơn hàng |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | IF NOT EXISTS(select \* from DoiTac where MaSoThue = @MaSoThue)  BEGIN  PRINT N'Đối tác' + CAST(@MaSoThue AS VARCHAR(10)) + N' Không Tồn Tại'  ROLLBACK TRAN  RETURN 1  END | //Không xin khoá |
|  |  | select \* from DonHang where MaSoThue = @MaSoThue  and DAY(DonHang.NgayDat) = DAY(getdate())  and MONTH (DonHang.NgayDat) = MONTH(getdate())  and YEAR(DonHang.NgayDat) = YEAR(getdate()) |  |
| IF @@error<>0  //Khi insert đến bảng DonHang, Thì số lượng đơn hàng quá số lượng đơn mà đối tác có thể cung cấp, nên buộc phải huỷ đơn của khách hàng  ROLLBACK |  |  |  |
|  |  | COMMIT TRAN |  |
| COMMIT TRAN |  |  |  |

## Tình huống 5: Khách hàng đang xem sản phẩm thì thêm 1 sản phẩm khác vào

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR03: *Phantom read***  T1 (User = Khách hàng A): thực hiện xem danh sách sản phẩm của đối tác B  T2 (User = Đối tác B): thực hiện thêm 1 sản phẩm vào danh sách sản phẩm | | | |
| **sp\_XemSanPham** | **Khóa** | **sp\_ThemSanPham** | **Khóa** |
| ***Input:*** @MaSoThue varchar(10)  ***Output***: | ***Input***: @MaSP varchar(10),  @MaChiNhanh varchar(10),  @SoLuongTon int,  @GiaCa int,  @TenSP varchar(50)  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL READ COMMITTED | SET TRANSACTION ISOLATION  LEVEL REPEATEABLE READ |
| BEGIN TRAN |  |  |  |
| B1: Xem danh sách các sản phẩm do đối tác B cung cấp  SELECT \* FROM SanPham  WHERE MaChiNhanh IN (SELECT MaChiNhanh FROM ChiNhanh WHERE MaSoThue=@MaSoThue) | **S(SanPham)**  **S(ChiNhanh)**  //Xin khoá đọc trên bảng sản phẩm và bảng chi nhánh |  |  |
|  |  | B1: Thêm 1 sản phẩm vào danh sách sản phẩm  INSERT SanPham(MaSP, MaChiNhanh, SoLuongTon, GiaCa, TenSP) VALUES (@MaSP, @MaChiNhanh, @SoLuongTon, @GiaCa, @TenSP) | //Không khoá ghi trên bảng sản phẩm |
| WAITFOR DELAY '00:00:20' |  |  |  |
| B2: Xem lại danh sách các sản phẩm do đối tác B cung cấp  SELECT \* FROM SanPham  WHERE MaChiNhanh IN (SELECT MaChiNhanh FROM ChiNhanh WHERE MaSoThue=@MaSoThue) | **S(SanPham)**  **S(ChiNhanh)**  //Xin khoá đọc trên bảng sản phẩm và bảng chi nhánh |  |  |
| COMMIT |  | COMMIT |  |

## Tình huống 6: Hai tài xế cùng nhận 1 đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR04: *Converson Deadlock***  T1 (User = Tài xế A): thực hiện xem danh sách đơn hàng được nhận, nhận đơn X  T2 (User = Tài xế B): thực hiện xem danh sách đơn hàng được nhận, nhận đơn X | | | |
| **sp\_NhanDonHang** | **Khóa** | **sp\_ NhanDonHang** | **Khóa** |
| ***Input:*** @KhuVuc nvarchar(50),  @MaDonHang varchar(10),  @MaTaiXe varchar(10),  ***Output***: | ***Input***: @KhuVuc nvarchar(50),  @MaDonHang varchar(10),  @MaTaiXe varchar(10),  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL SERIALIZABLE | SET TRANSACTION ISOLATION  LEVEL SERIALIZABLE |
| BEGIN TRAN |  |  |  |
| B1: Xem danh sách các đơn hàng có thể nhận  SELECT \* FROM DonHang  WHERE KhuVuc = @KhuVuc | **S(DonHang)**  //Xin khoá đọc trên bảng đơn hàng |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | B1: Xem danh sách các đơn hàng có thể nhận  SELECT \* FROM DonHang  WHERE KhuVuc = @KhuVuc | **S(DonHang)**  //Xin khoá đọc trên bảng đơn hàng |
|  |  | WAITFOR DELAY '00:00:10' |  |
| B2: Nhận 1 đơn hàng trong danh sách  UPDATE DonHang  SET MaTaiXe = @MaTaiXe  WHERE MaDonHang = @MaDonHang | **X(DonHang)**  //Xin khoá ghi trên bảng đơn hàng |  |  |
|  |  | B2: Nhận 1 đơn hàng trong danh sách  UPDATE DonHang  SET MaTaiXe = @MaTaiXe  WHERE MaDonHang = @MaDonHang | **X(DonHang)**  //Xin khoá ghi trên bảng đơn hàng |
| COMMIT |  | COMMIT |  |