**d**

BÁO CÁO ĐỒ ÁN THỰC HÀNH

MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

GVHD: TIẾT GIA HỒNG

# **THÔNG TIN NHÓM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Công việc** | **% Hoàn thành** |
| 1 | 18424002 | Nguyễn Thái Anh | Làm việc với bảng Admin |  |
| 2 | 18424024 | Huỳnh Văn Hậu | Làm việc với bảng chủ nhà thuê |  |
| 3 | 18424031 | Thái Quang Khải | Làm việc với bảng khách hàng |  |
| 4 | 18424077 | Ngụy Minh Trọng | Làm việc với bảng dùng chung |  |
| 5 | 18424079 | Phạm Minh Tuấn | Làm việc với bảng nhân viên |  |

# CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

# Sinh viên thực hiện: Ngụy Minh Trọng

## Tình huống 1: Không thể đọc lại thông tin tài khoản sau khi sửa

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty read, Read Committed***  T1 (Người dùng): thực hiện sửa 1 thông tin tài khoản.  T2 (Người dùng): thực hiện đăng nhập tài khoản. | | | |
| **sp\_Sua1TaiKhoan** | **Khóa** | **sp\_DangNhapTaiKhoan** | **Khóa** |
| ***Input: thông tin tài khoản***  ***Output***: **tên đăng nhập, mật khẩu** | ***Input***: username, password  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMIITED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin tài khoản  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN    END | **R(TAIKHOAN)** |  |  |
| B3: Sửa thông tin vào bảng Tài khoản  Update TAIKHOAN  SET MATKHAU = @MATKHAU | Xin MATK(TAIKHOAN) SQL: cấp MATK(TAIKHOAN) và giữ MATK đến khi Commit |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin TAIKHOAN  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN  END |  |
|  |  | B2: Đăng nhập tài khoản  SELECT MATK, MATKHAU  FROM TAIKHOAN  WHERE MATK = @MATK  AND MATKHAU = @MATKHAU | **R(TAIKHOAN): Không xin khóa MATK** |
|  |  | COMMIT |  |
| IF @error <> 0  Begin  ROLLBACK Trans  Return  End |  |  |  |

## Tình huống 2: Mất dữ liệu cập nhật

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR02: *Dirty read, Read Committed***  T1 (Admin): thực hiện xóa 1 thông tin Tài Khoản.  T2 (Người dùng): thực hiện đăng nhập tài khoản | | | |
| **sp\_Xoa1TaiKhoan** | **Khóa** | **sp\_DangNhapTaiKhoan** | **Khóa** |
| ***Input:*** *MATK*  ***Output***: @MATK | ***Input***: MATK, MATKHAU  ***Output***: MATK, MATKHAU |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** |  | SET TRANSACTION ISOLATION  LEVEL **UNREPEATEABLE READ** |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin tài khoản  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN    END |  |  |  |
| B3: Xóa thông tin tài khoản  Delete TAIKHOAN  Where MATK = @MATK |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin TAIKHOAN  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN  END |  |
|  |  | B2: Đăng nhập tài khoản  SELECT MATK, MATKHAU  FROM TAIKHOAN  WHERE MATK = @MATK  AND MATKHAU = @MATKHAU | **R(TAIKHOAN): Không xin khóa MATK** |
|  |  | COMMIT |  |
| IF @error <> 0  Begin  ROLLBACK Trans  Return  End |  |  |  |

## Tình huống 3: Đọc dữ liệu rác

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR03: *Dirty read***  T1 (Người dùng): thực hiện xem thong tin nhà thuê.  T2 (Người dùng): thực hiện tìm kiếm nhà thuê | | | |
| **sp\_Xemthongtinthuenha** | **Khóa** | **sp\_TimKiemNhaThue** | **Khóa** |
| ***Input:*** *Thông tin nhà thuê*  ***Output***: Thông tin nhà thuê | ***Input***: Thông tin nhà thuê  ***Output***: @TimKiem |
| SET TRANSACTION ISOLATION  LEVEL ………………………….. | SET TRANSACTION ISOLATION  LEVEL **UNREPEATEABLE READ** |
| BEGIN TRAN |  |  |  |
|  |  |  |  |
|  |  |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  |  |  |
|  |  |  |  |
|  |  | COMMIT |  |
|  |  |  |  |

## Tình huống 4: Đọc dữ liệu rác

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty read, Read Committed***  T1 (Người dùng): thực hiện đổi mật khẩu tài khoản.  T2 (Người dùng): thực hiện đăng nhập tài khoản | | | |
| **sp\_DoiMatKhau** | **Khóa** | **sp\_DangNhapTaiKhoan** | **Khóa** |
| ***Input: MATKHAU***  ***Output***: @MATKHAU | ***Input***: MATK, MATKHAU  ***Output***: @MATK, @MATKHAU |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMIITED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin tài khoản  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN    END | **R(TAIKHOAN)** |  |  |
| B3: Sửa mật khẩu vào bảng Tài khoản  Update TAIKHOAN  SET MATKHAU = @MATKHAU | Xin MATK(TAIKHOAN) SQL: cấp MATK(TAIKHOAN) và giữ MATK đến khi Commit |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin TAIKHOAN  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN  END |  |
|  |  | B2: Đăng nhập tài khoản  SELECT MATK, MATKHAU  FROM TAIKHOAN  WHERE MATK = @MATK  AND MATKHAU = @MATKHAU | **R(TAIKHOAN): Không xin khóa** |
|  |  | COMMIT |  |
| IF @error <> 0  Begin  ROLLBACK Trans  Return  End |  |  |  |

## Tình huống 5: Bóng ma

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty Insert, Read Uncommited***  T1 (User = quản lý): thực hiện thêm 1 tài khoản.  T2 (User = tổ trưởng): thực hiện đăng nhập tài khoản | | | |
| **sp\_Them1TaiKhoan** | **Khóa** | **sp\_DangNhapTaiKhoan** | **Khóa** |
| ***Input:*** *MATK, MATKHAU*  ***Output***: @MATK, @MATKHAU | ***Input***: MATK, MATKHAU  ***Output***: MATK, MATKHAU |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **UNREPEATEABLE READ** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin tài khoản  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN    END | **R(TAIKHOAN)**  Xin MATK(TAIKHOAN) SQL: cấp MATK(TAIKHOAN) |  |  |
| B3: Thêm thông tin vào bảng TAIKHOAN  INSERT INTO TAIKHOAN (MATK, MATKHAU)  VALUES (@MATK, MATKHAU) | **X(TAIKHOAN)** |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin TAIKHOAN  IF EXISTS (SELECT \* FROM TAIKHOAN WHERE MATK=@MATK AND MATKHAU = @MATKHAU)  BEGIN  SELECT \*  FROM TAIKHOAN  END |  |
|  |  | B2: Đăng nhập tài khoản  SELECT MATK, MATKHAU  FROM TAIKHOAN  WHERE MATK = @MATK  AND MATKHAU = @MATKHAU | **R(TAIKHOAN): Không xin khóa** |
|  |  | COMMIT |  |
| IF @error <> 0  Begin  ROLLBACK Trans  Return  End |  |  |  |

# Sinh viên thực hiện: Nguyễn Thái Anh

## 1.Tình huống 1: Admin xóa tài khoản và nhân viên đăng nhập.

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Phantom**  T1 (User = admin): Xóa tài khoản của nhân viên.  T2 (User = nhân viên): thực hiện đăng nhập tài khoản. | | | |
| **sp\_XoaTaiKhoan** | **Khóa** | **sp\_DangNhapTK** | **Khóa** |
| ***Input: @MaNhanVien,***  ***Output***: | ***Input***: @MaNhanVien, @MatKhau  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL ………………………….. | SET TRANSACTION ISOLATION  LEVEL **UNREPEATEABLE READ** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin của nhân viên  IF NOT EXISTS (SELECT \* FROM NHAN\_VIEN WHERE Ma\_Nhan\_Vien=@MaNhanVien)  BEGIN  Select N’Khong co tai khoan nay’  END | **R(NHAN\_VIEN)** |  |  |
| B2: Xóa thông tin nhân viên  DELETE FROM NHAN\_VIEN WHERE Ma\_Nhan\_Vien=@MaNhanVien |  |  |  |
|  | **X(NHAN\_VIEN)** |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin đăng nhập  IF NOT EXISTS (SELECT \* FROM NHAN\_VIEN WHERE Ma\_Nhan\_Vien and @MaNhanVien )  BEGIN  Select N’Khong co tai khoan nay’  END | **R(NHAN\_VIEN)** |
|  |  |  | **X(NHAN\_VIEN)** |
|  |  | COMMIT |  |

# Sinh viên thực hiện: Thái Quang Khải

## Tình huống 1: Chủ nhà xóa thông tin nhà thuê, người thuê nhà xem danh sách nhà đang cho thuê

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Phan Tom***  T1 (User = Chủ nhà): thực hiện xóa thông tin nhà cho thuê.  T2 (User = Người thuê nhà): thực hiện xem danh sách nhà đang cho thuê. | | | |
| **sp\_Xoa-NhaChoThue** | **Khóa** | **sp\_XemDS-NhaDangChoThue** | **Khóa** |
| ***Input:*** *Các thông tin nhà cho thuê*  ***Output***: Đã sửa được hay không | ***Input***:  ***Output***: Danh sách nhà đang cho thuê |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMIITED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin nhà thuê  IF Not EXISTS (SELECT \* FROM NHA\_THUE nt WHERE nt.Ma\_Nha\_Thue=@Ma\_Nha\_Thue)  BEGIN  Select N’Nhà thuê không tồn tại’  END | **R(NHA\_THUE)** |  |  |
| B2: Thêm thông tin vào bảng Chuyến Đi ……  DELETE NHA\_THUE nt set nt.Ma\_Nha\_Thue =@Ma\_Nha\_Thue | **X(NHA\_THUE)** |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM NHA\_THUE nt WHERE nt.Loai=@Loai)  BEGIN  Select N’Không còn nhà nào đang cho thuê’  END | **R(NHA\_THUE)** |
|  |  | COMMIT |  |
| IF @error ….  Begin  ROLLBACK Trans  Return  End |  |  |  |

## Tình huống 2: Chủ nhà xóa thông tin nhà thuê, nhân viên phê duyệt nhà cho thuê

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Lost Update***  T1 (User = Chủ nhà): thực hiện xóa thông tin nhà cho thuê.  T2 (User = NhanVien): thực hiện phê duyệt nhà đang cho thuê. | | | |
| **sp\_Xoa-NhaChoThue** | **Khóa** | **sp\_CapNhat-NhaDangChoThue** | **Khóa** |
| ***Input:*** *Các thông tin nhà cho thuê*  ***Output***: Đã sửa được hay không | ***Input***:  ***Output***: Danh sách nhà đang cho thuê |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMIITED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin nhà thuê  IF Not EXISTS (SELECT \* FROM NHA\_THUE nt WHERE nt.Ma\_Nha\_Thue=@Ma\_Nha\_Thue)  BEGIN  Select N’Nhà thuê không tồn tại’  END | **R(NHA\_THUE)** |  |  |
| B2: Thêm thông tin vào bảng Chuyến Đi ……  DELETE NHA\_THUE nt set nt.Ma\_Nha\_Thue =@Ma\_Nha\_Thue | **X(NHA\_THUE)** |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin  IF NOT EXISTS (SELECT \* FROM NHA\_THUE nt WHERE nt.Loai=@Loai)  BEGIN  Select N’Không còn nhà nào đang cho thuê’  END | **R(NHA\_THUE)** |
|  |  | COMMIT |  |
| IF @error ….  Begin  ROLLBACK Trans  Return  End |  |  |  |

# Sinh viên thực hiện: Phạm Minh Tuấn

## Tình huống 1: Nhân viên chỉnh sửa giá nhà, người thuê thuê nhà

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Dirty Update**  T1 (User = NhânViên): Thực hiện chỉnh sửa giá nhà.  T2 (User = NgườiThuê): Thực hiện thao tác đặt nhà | | | |
| **Sp\_ChinhSuaGiaNhan** | **Khóa** | **Sp\_ThucHienDatNha** | **Khóa** |
| ***Input: Bảng NhanVien,NguoiThue***  ***Output***:**Bảng NhanVien đã chỉnh sửa giá** | ***Input***: **Bảng NguoiThue**  ***Output***: **Bảng NguoiThue** |
| SET TRANSACTION ISOLATION  LEVEL ………………………….. | SET TRANSACTION ISOLATION  LEVEL **UNREPEATEABLE READ** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin Nhà Cho Thuê  IF EXISTS (SELECT \* FROM NHATHUE WHERE MaNhaThue=@MaNhaThue)  BEGIN  SELECT \*  FROM NHATHUE  END | **R(NHATHUE)** |  |  |
| B2: Thêm giá tiền thuê nhà mời  SET @giaThueNhaMoi = (  SELECT giaThueNhaMoi  FROM NHATHUE  WHERE MaNT = ‘MNT001’)  SET @giaThueNhaMoi = @giaThueNhaMoi + @giaNhaThueCu | **Xin(NHATHUE)**  **Cập nhật cho đến khi commit** |  |  |
| B3: Cập nhật giá tiền thuê mới  UPDATE NHATHUE  SET giaNhaThueMoi = @giaNhaThueMoi  WHERE MaNT = ‘MNT001’ |  |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin Nhà Thuê  IF EXISTS (SELECT \* FROM NHATHUE WHERE MaNhaThue=@MaNhaThue)  BEGIN  SELECT \*  FROM NHATHUE  END |  |
|  |  | B2: Đăng kí thuê nhà giá cũ  SELECT \*  FROM NHATHUE  WHERE MaTN= ‘MNT001’ AND giaNhaThue = @giaNhaThue | **R(NHATHUE)** |
|  |  | COMMIT |  |
| IF @error ….  // Chưa Biết Cách Giải Quyết  ROLLBACK |  |  |  |

# Sinh viên thực hiện: Huỳnh Văn Hậu

## Tình huống 1: Chủ nhà cập nhật thông tin về giá của một căn nhà, khách hàng xem thông tin nhà thuê

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty read***  T1 (User = Chủ nhà): thực hiện cập nhât giá 1 nhà thuê.  T2 (User = Khách hàng): thực hiện xem thông tin nhà | | | |
| **sp\_CapNhatNhaChoThue** | **Khóa** | **sp\_XemNhaChoThue** | **Khóa** |
| ***Input:*** **Bảng NhaThue**  ***Output***: **Bảng NhaThue đã chỉnh giá** | ***Input***: **Bảng NhaThue**  ***Output***: **Bảng NhaThue** |
| SET TRANSACTION ISOLATION  LEVEL ………………………….. | SET TRANSACTION ISOLATION  LEVEL **UNREPEATEABLE READ** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) ……  IF NOT EXISTS (SELECT \* FROM NhaThue WHERE MaNhaThue = ‘@MaNhaThue’  BEGIN  Select N’Nhà thuê không tồn tại’  END | **R(NhaThue)** |  |  |
| B2: Kiểm tra thông tin (2) ……  IF EXISTS (SELECT \* FROM ChuyenDi WHERE MaNhaThue = @MaNhaThue  BEGIN  Update NhaThue  Set GiaNhaThue = @GiaThue  Where MaNhaThue = @MaNhaThue  END | **W(NhaThue)** |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) ……  IF EXISTS (SELECT \* FROM A WHERE MaNhaThue=@MaNhaThue  BEGIN  Select \* from NhaThue  END | **R(A)** |
|  |  | COMMIT |  |
| IF @error ….  Khi Update một nhà trong bảng NhaThue nếu không có nhà đang thuê thì rollback.  ROLLBACK |  |  |  |