**d**

BÁO CÁO ĐỒ ÁN THỰC HÀNH

MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

GVHD: cô HLTKNhung – cô TGHồng

**MỤC LỤC**

[THÔNG TIN NHÓM 2](#_Toc487636234)

[BẢNG THIẾT KẾ THỰC THỂ KẾT HỢP 3](#_Toc487636235)

[I. Lược đồ thực thể kết hợp 3](#_Toc487636236)

[II. Lược đồ quan hệ 4](#_Toc487636237)

[III. Ràng buộc dữ liệu bổ sung 4](#_Toc487636238)

[IV. Các điểm đặt biệt trong mô hình thực thể kết hợp 4](#_Toc487636239)

[XÁC ĐỊNH CHỨC NĂNG HỆ THỐNG VÀ THỦ TỤC STORE PROCEDURE. 5](#_Toc487636240)

[I. Các chức năng của hệ thống 5](#_Toc487636241)

[1. Chức năng cho toàn bộ phân hệ 5](#_Toc487636242)

[2. Phân hệ quản trị viên (admin) 5](#_Toc487636243)

[3. Phân hệ …. 5](#_Toc487636244)

[4. Phân hệ ….. 5](#_Toc487636245)

[II. Thủ tục store procedure 6](#_Toc487636246)

[III. Xác định tình huống tranh chấp 6](#_Toc487636247)

[1. Danh sách tình huống tranh chấp đồng thời 6](#_Toc487636248)

[2. Tình huống tranh chấp đồng thời 7](#_Toc487636249)

[- Sinh viên thực hiện: ….. 7](#_Toc487636250)

[3. Giải pháp khắc phục lỗi tranh chấp đồng thời 9](#_Toc487636251)

[- Sinh viên thực hiện: ………… 9](#_Toc487636252)

# **THÔNG TIN NHÓM**

|  |  |  |
| --- | --- | --- |
| **STT** | **MSSV** | **HỌ TÊN** |
| 1 | 1642050 | Nguyễn Thành Phong |
| 2 | 1642067 | Nguyễn Lê Quang Thắng |
| 3 | 1642071 | Trịnh Quốc Thịnh |
| 4 | 1642076 | Huỳnh Duy Thức |
| 5 | 1642084 | Trần Lê Tuấn |

# BẢNG THIẾT KẾ THỰC THỂ KẾT HỢP

# Lược đồ thực thể kết hợp



# Lược đồ quan hệ





# Ràng buộc dữ liệu bổ sung

* Số đáp án của 1 câu hỏi là từ 2 tới 10 đáp án
* Điểm của một bộ đề thi tối đa là 10
* Một câu hỏi phải có tối thiểu 1 đáp án đúng
* Khi chỉnh sửa câu hỏi, nếu chỉnh sửa số câu trả lời thì thuộc tính "Số lượng câu trả lời" phải cập nhật lại.
* Bộ đề thi chỉ chứa câu hỏi của một môn học duy nhất.

# Các điểm đặc biệt trong mô hình thực thể kết hợp

|  |  |  |
| --- | --- | --- |
| **Bảng** | **Thuộc tính** | **Ghi chú** |
| CAUHOI | MucDo | * 0: mức độ dễ * 1: mức độ vừa * 2: mức độ khó |
| CAUTRALOI | LaDapAnDung | * 0: đáp án sai * 1: đáp án đúng |
| NGUOIDUNG | TrangThai | * 0: tài khoản không bị khóa * 1:tài khoản bị khóa |
| NGUOIDUNG | ToanQuyenGV | * 0: chỉ được xem bộ đề thi * 1: được quản lý bộ đề thi |
| BODE\_CAUHOI | Diem | * Là điểm của câu hỏi thuộc đề thi đó, mặc dịnh sẽ là thuộc tính ThangDiem của bảng CAUHOI * Bảng BODE\_CAUHOI được sinh ra từ quan hệ nhiều – nhiều của 2 bảng BODE và CAUHOI |

# XÁC ĐỊNH CHỨC NĂNG HỆ THỐNG VÀ THỦ TỤC STORE PROCEDURE.

# Các chức năng của hệ thống

## Chức năng cho toàn bộ phân hệ

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| ALL1 | Đăng nhập | Đăng nhập vào hệ thống ….. dựa vào tài khoản và mật khẩu. |
|  |  |  |
|  |  |  |

## Phân hệ quản trị viên (admin)

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| AD1 | Tạo tài khoản đăng nhập | Admin sẽ tạo tài khoản đăng nhập sẵn cho …. theo …. và quy tắc dành cho mật khẩu khởi tạo. |
|  |  |  |
|  |  |  |
|  |  |  |

## Phân hệ quản lý giáo viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| QLGV1 | Thêm tài khoản GV | Quản trị sẽ tạo tài khoản cho giáo viên với tên đăng nhập và mật khẩu theo quy định. |
| QLGV2 | Xóa tài khoản GV | Quản trị xóa tài khoản của giáo viên dựa vào mã giáo viên |
| QLGV3 | Thay đổi trạng thái tài khoản GV | Quản trị khóa hoặc mở khóa tài khoản của giáo viên dựa vào mã giáo viên |
| QLGV4 | Sửa tài khoản GV | Quản trị sửa thông tin tài khoản của giáo viên, bao gồm họ tên, tên đăng nhập, mật khẩu dựa vào mã giáo viên |
| QLGV5 | Cấp quyền quản lý cho giáo viên | Quản trị cấp quyền quản lý cho giáo viên dựa vào mã giáo viên |
| QLGV6 | Thay đổi giáo viên quản lý của giáo viên | Quản trị thay đổi giáo viên quản lý (mã giáo viên) của giáo viên |
| QLGV7 | Giáo viên QL thêm môn học mới | Giáo viên quản lý của bộ môn trực thuộc thêm môn học mới trong bô môn |

## Phân hệ giáo viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| GV1 | Tạo bộ đề thi | Giáo viên sau khi đăng nhập thành công sẽ được phép tạo bộ thi |
| GV2 | Cập nhật bộ đề thi | Giáo viên sẽ được phép cập nhật thông tin cho bộ đề thi |
| GV3 | Thêm câu hỏi vào bộ đề thi | Giáo viên sẽ được phép thêm câu hỏi vào bộ đề thi ( Các câu hỏi không được dùng lại cho 2 học kì liên tiếp trong môn học, hoặc cho các hệ khác nhau của môn học này ) |
| GV4 | Cập nhật câu hỏi sang bộ đề thi khác | Giáo viên sẽ được phép cập nhật câu hỏi từ bộ đề thi này sang bộ đề thi khác |
| GV5 | Xóa câu hỏi trong bộ đề thi | Giáo viên sẽ được phép xóa một câu hỏi trong một bộ đề thi |
| GV6 | Tìm kiếm bộ đề thi | Giáo viên sẽ được phép tìm kiếm bộ đề thi theo tên, học kỳ và năm học |
| GV7 | Tìm kiếm tất cả câu hỏi theo bộ đề thi | Giáo viên chọn một bộ đề thi, hệ thống sẽ hiển thị tất cả câu hỏi thuộc bộ đề thi đó |

## Phân hệ quản lý câu hỏi trắc nghiệm

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| CHTN1 | Thêm câu hỏi trắc nghiệm | Giáo viên sau khi đăng nhập thành công và được cấp quyền sẽ được thêm câu hỏi trắc nghiệm |
| CHTN2 | Cập nhật câu hỏi trắc nghiệm | Giáo viên sẽ được phép cập nhật thông tin câu hỏi trắc nghiệm |
| CHTN3 | Xoá câu hỏi trắc nghiệm | Giáo viên sẽ được phép xoá câu hỏi trắc nghiệm |
| CHTN4 | Tìm kiếm câu hỏi trắc nghiệm | Giáo viên sẽ được phép tìm kiếm câu hỏi trắc nghiệm |
| CHTN5 | Xuất danh sách câu hỏi trắc nghiệm | Giáo viên sẽ được phép xem danh cách các câu hỏi trắc nghiệm |
| CHTN6 | Phân loại câu hỏi trắc nghiệm | Giáo viên sẽ được phép phân quyền câu hỏi nghiệm theo các mức dễ , vừa , khó |

# Thủ tục store procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên thủ tục** | **Input** | **Output** | **Nội dung thực hiện** | **Chức năng** |
|  | sp\_TaoTaiKhoan | HoTen, TenDangNhap, MatKhau, TrangThai, ToanQuyenGV,  MaLoai,  MaBM,  MaGVQL | True hoặc False | Kiểm tra sự tồn tại MaLoai, MaBM, MaGVQL, nếu không tồn tại thì thông báo lỗi, ngược lại tiến hành tạo tài khoản mới | QLGV1 |
|  | sp\_XoaTaiKhoan | MaND | True hoặc False | Kiểm tra sự tồn tại của MaND, nếu tồn tại thì thưc hiện xóa tài khoản GV, ngược lại thông báo lỗi | QLGV2 |
|  | sp\_CapNhatTrangThai | MaND, TrangThai | True hoặc False | Kiểm tra sự tồn tại của MaND, nếu tồn tại thì thực hiện cập nhật trạng thái(giá trị = 0: tài khoản không bị khóa, 1: tài khoản đang bị khóa), ngược lại thông báo lỗi | QLGV3 |
|  | sp\_CapNhatThongTin | MaND,  HoTen, TenDangNhap, MatKhau, MaLoai, MaBM | True hoặc False | Kiểm tra sự tồn tại của MaND, MaLoai, MaBM, nếu tồn tại thì thực hiện cập nhật thông tin (HoTen, TenDangNhap, MatKhau, MaLoai, MaBM), ngược lại thông báo lỗi | QLGV4 |
|  | sp\_GVQLCapQuyen | MaGVQL, MaGV | True hoặc False | Kiểm tra MaGVQL có phải là giáo viên quản lý hay không. Kiểm tra MaGV và MaGVQL có cùng thuộc 1 bộ môn hay không. Nếu thoả 2 điều tên thì tiến hành cấp quyền cho MaGV | QLGV5 |
|  | sp\_ThayDoiGVQL | MaND, MaGVQL | True hoặc False | Kiểm tra sự tồn tại của MaND, MaGVQL, nếu không tồn tại thì thông báo lỗi, ngược lại tiến hành cập nhật MaGVQL cho giáo viên này. | QLGV6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên thủ tục** | **Input** | **Output** | **Nội dung thực hiện** | **Chức năng** |
|  | Sp\_ThemCauHoi | Macauhoi,  Monhoc,  Noidung,  Socautraloi,  LoaiMonHoc,  Thangdiemdukien,  Danhsachcautraloi,  MucDo | True or false | Thêm một dòng dữ liệu vào bảng CauHoi , sau đó xác thực hoặc thông báo lỗi | CHTN1 |
|  | sp\_CapNhatCauHoi | Macauhoi,  Monhoc,  Noidung,  Socautraloi,  LoaiMonHoc,  Thangdiemdukien,  Danhsachcautraloi | True or False | Cập nhật câu hỏi trắc nghiệm , sau đó xác thực hoặc thông báo lỗi | CHTN2 |
|  | sp\_XoaCauHoi | Macauhoi | True or False | Kiểm tra xem câu hỏi đó có đang tồn taị hay chưa, sau đó xác thực hoặc thông báo lỗi | CHTN3 |
|  | sp\_TimKiemCauHoi | Keyword | Monhoc,  Noidung,  Socautraloi,  LoaiMonHoc,  Thangdiemdukien,  Danhsachcautraloi,  Mucdo | Nếu tồn tại câu hỏi sẽ xuất ra câu hỏi cần tìm , nếu không tìm thấy sẽ thông báo không tìm thấy câu hỏi cần tìm hoặc thông báo lỗi | CHTN4 |
|  | sp\_XuatDanhSachCauHoi | Không | Danh sách câu hỏi | Xuất danh sách câu hỏi trong kho câu hỏi | CHTN5 |
|  | sp\_PhanLoaiCauhoi | Macauhoi | MucDo | Kiểm tra xem câu hỏi đó có đang tồn taị hay chưa, sau đó xác thực , thay đổi mức độ hoặc thông báo lỗi | CHTN6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên thủ tục** | **Input** | **Output** | **Nội dung thực hiện** | **Chức năng** |
|  | sp\_DangNhap | User và matkhau | True or false | Kiểm tra thông tin về User và matkhau, sau đó xác thực hoặc thông báo lỗi | ALL1 |
|  | sp\_TaoBoDeThi | TenBoDeThi, HocKy và NamHoc | True or False | Thêm một dòng dữ liệu vào bảng BoDeThi , sau đó xác thực hoặc thông báo lỗi | GV1 |
|  | sp\_CapNhatBoDeThi | MABDT, TenBoDeThi, HocKy và NamHoc | True or False | Cập nhật bộ đề thi , sau đó xác thực hoặc thông báo lỗi | GV2 |
|  | sp\_ThemCauHoiVaoBoDeThi | MABDT,MACH | True or False | Kiểm tra bộ đề thi đã có câu hỏi này chưa, sau đó xác thực hoặc thông báo lỗi | GV3 |
|  | sp\_CapNhatCauHoiSangBoDeThiKhac | MACH,MABDTold và MABDTnew | True or False | Kiểm tra xem bộ đề thi mới đã đủ số lượng câu hỏi hay chưa, sau đó xác thực hoặc thông báo lỗi | GV4 |
|  | sp\_XoaCauHoiTrongBodeThi | MACH,MABDT | True or False | Kiểm tra xem câu hỏi đó có đang tồn tại trong bộ đề thi hay chưa, sau đó xác thực hoặc thông báo lỗi | GV5 |
|  | sp\_TimKiemBoDeThi | TenBoDeThi, HocKy và NamHoc | Danh sách bộ đề thi | Tìm kiếm bộ đề thi theo 3 tiêu chí : tên bộ đề thi , học kỳ và năm học | GV6 |
|  | sp\_TimKiemTatCaCauHoiTheoBoDeThi | MABDT | Dach sách câu hỏi | Tìm kiếm danh sách câu hỏi theo mã bộ đề thi | GV7 |
|  | sp\_ThemMonHoc | MaBM, TenMH | True or False | Kiểm tra mã bộ môn xem có tồn tại trong bảng BoMon hay không. Nếu tồn tại, thực hiện thêm | GVQL7 |

# 

# Xác định tình huống tranh chấp

## Danh sách tình huống tranh chấp đồng thời

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Chức năng 1** | **Người dùng** | **Thủ tục tham gia** | **Chức năng 2** | **Người dùng** | **Thủ tục tham gia** | **Lỗi tranh chấp** | **Ghi chú** |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

## Tình huống tranh chấp đồng thời

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty read***  T1 (User = giáo viên A): thực hiện thêm 1 câu trả lời cho câu hỏi C  T2 (User = giáo viên B ): thực hiện xem thông tin ( số lượng câu trả lời ) của câu hỏi C | | | |
| **sp\_ThemCauTraLoiVaoCauHoi\_DEMO** | **Khóa** | **sp\_LayDanhSachCauTraLoiTheoCauHoi\_DEMO** | **Khóa** |
| ***Input:*** *@NOIDUNG, @LADAPANDUNG, @MACH*  ***Output***: @Return | ***Input***: @MACH  ***Output***: @Return |
|  | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : không tồn tại câu hỏi  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE MACH = @MACH )  BEGIN  SET @Return = N'Không tồn tại câu hỏi này'  END | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) : đã tồn tại câu trả lời trong cau hỏi này  IF EXISTS ( SELECT \* FROM CAUTRALOI WHERE MACH = @MACH AND NOIDUNG = @NOIDUNG )  BEGIN  SET @Return = N'Trong câu hỏi đã tồn tại câu trả lời này'  END | **R(CAUTRALOI)** |  |  |
| B3: Thêm thông tin vào bảng CAUTRALOI  INSERT INTO CAUTRALOI(NOIDUNG,LADAPANDUNG,MACH) VALUES(@NOIDUNG,@LADAPANDUNG,@MACH) | **X(CAUTRALOI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : không tồn tại câu hỏi  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE @MACH = MACH )  BEGIN  SET @Return = N'Không tồn tại câu hỏi này'  END | **R(CAUHOI)** |
|  |  | B2: Xem thông tin câu hỏi  SELECT \* FROM CAUHOI WHERE MACH = @MACH | **R(CAUHOI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  ROLLBACK  // tg\_SoLuongCauTraLoiCuaCauHoi > 10  END CATCH |  |  |  |

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR02: *Unrepeatable Read***  T1 (User = giáo viên A): thực hiện lấy danh sách câu hỏi theo nội dung và mức độ  T2 (User = giáo viên B ): thực hiện cập nhật thông tin câu hỏi ( mức độ ) | | | |
| **sp\_LayDanhSachCaiHoiTheoMucDO\_DEMO** | **Khóa** | **sp\_CapNhatCauHoi\_DEMO** | **Khóa** |
| ***Input:*** *@NOIDUNG, @MUCDO*  ***Output***: @Return | ***Input***: @MACH  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS( SELECT \* FROM CAUHOI WHERE @NOIDUNG like '%' + @NOIDUNG + '%' OR @NOIDUNG = '' )  SET @Return = N'Không có câu hỏi chứa nội dung này' | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) :  IF NOT EXISTS( SELECT \* FROM CAUHOI WHERE @NOIDUNG like '%' + @NOIDUNG + '%' AND MUCDO = @MUCDO )  SET @Return = N'Không có câu hỏi có mức độ này' | **R(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B3: Lấy danh sách câu hỏi thỏa điều kiện  SELECT c.\*,m.TENMH FROM CAUHOI c,MonHoc m  WHERE c.MAMH = m.MAMH AND NOIDUNG like '%' + @NOIDUNG + '%' AND c.MUCDO = @MUCDO  SET @Return = ‘’ | **R(CAUHOI)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) :  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE @MACH = MACH )  BEGIN  SET @Return = N'Không tồn tại câu hỏi này'  END | **R(CAUHOI)** |
|  |  | B2: Cập nhật thông tin cho câu hỏi (2) :  UPDATE CAUHOI SET NOIDUNG = @NOIDUNG, THANGDIEM = @THANGDIEM, MUCDO = @MUCDO  WHERE MACH = @MACH  SET @Return = '' | **X(CAUHOI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR03: *Lost Update***  T1 (User = giáo viên A): thực hiện lấy thêm câu trả lời vào câu hỏi  T2 (User = giáo viên B ): thực hiện xóa câu trả lời ( thuộc câu hỏi của T1 ) | | | |
| **sp\_ThemCauTraLoiVaoCauHoi\_DEMO** | **Khóa** | **sp\_XoaCauTraLoi\_DEMO** | **Khóa** |
| ***Input:*** *@NOIDUNG, @LADAPANDUNG, @MACH*  ***Output***: @Return | ***Input***: @MACTL  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE MACH = @MACH )  SET @Return = N'Không tồn tại câu hỏi này' | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) :  IF EXISTS ( SELECT \* FROM CAUTRALOI WHERE MACH = @MACH AND NOIDUNG = @NOIDUNG )  SET @Return = N'Trong câu hỏi đã tồn tại câu trả lời này' | **R(CAUTRALOI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B3: Thêm câu trả lời vào câu hỏi  INSERT INTO CAUTRALOI(NOIDUNG,LADAPANDUNG,MACH) VALUES(@NOIDUNG,@LADAPANDUNG,@MACH) | **X(CAUTRALOI)** |  |  |
| B4: Cập nhật số lượng CTL cho câu hỏi (trigger) | **X(CAUHOI)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) :  IF NOT EXISTS ( SELECT \* FROM CAUTRALOI WHERE MACTL = @MACTL )  SET @Return = hông tồn tại câu trả lời này' | **R(CAUTRALOI)** |
|  |  | B2: Xóa câu trả lời (2) :  DELETE FROM CAUTRALOI WHERE MACTL = @MACTL | **X(CAUTRALOI)** |
|  |  | B3: Cập nhật số lượng câu trả lời cho câu hỏi (3) (trigger) | **X(CAUHOI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR04: *Phantom***  T1 (User = giáo viên A): thực hiện lấy danh sách câu hỏi trong bộ đề thi C  T2 (User = giáo viên B ): thực hiện thêm câu hỏi vào trong bộ đề thi C | | | |
| **sp\_LayDanhSachCauHoiTheoDeThi\_DEMO** | **Khóa** | **sp\_ThemCauHoiVaoBoDeThi\_DEMO** | **Khóa** |
| ***Input:*** *@MABDT*  ***Output***: @Return | ***Input***: @MABDT,@MACH,@Diem,@Return  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) :  DECLARE @slCauHoi INT  SELECT @slCauHoi = COUNT(\*) FROM TAOBODETHI WHERE MABDT = @MABDT  IF ( @slCauHoi < 0 )  SET @Return = N'Bộ đề thi chưa có bất câu hỏi nào' | **R(TAOBODETHI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B2: Lấy danh sách câu hỏi trong bộ đề thi :  SELECT c.\*  FROM TAOBODETHI t , CAUHOI c  WHERE t.MABDT = @MABDT AND t.MACH = c.MACH | **R(TAOBODETHI,CAUHOI)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) :  IF NOT EXISTS ( SELECT \* FROM BODETHI WHERE @MABDT = MABDT )  SET @Return = N'Không tồn tại bộ đề thi này' | **R(CAUHOI)** |
|  |  | B2: Kiểm tra thông tin (2) :  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE @MACH = MACH )  SET @Return = N'Không tồn tại câu hỏi này' | **R(CAUHOI)** |
|  |  | B3: Thêm câu hỏi vào bộ đề thi :  INSERT INTO TAOBODETHI(MABDT,MACH,DIEM) VALUES(@MABDT,@MACH,@DIEM) | **X(TAOBODETHI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  ROLLBACK  END CATCH |  |  |  |

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR05: *Deadlock***  T1 (User = giáo viên A): Cập nhật câu trả lời sang câu hỏi C  T2 (User = giáo viên B ): Xóa câu trả lời trong câu hỏi C | | | |
| **sp\_CapNhatCauTraLoiSangCauHoiKhac\_DEMO** | **Khóa** | **sp\_XoaCauTraLoiTheoCauHoi\_DEMO** | **Khóa** |
| ***Input:*** *@MACTL, @MACH*  ***Output***: @Return | ***Input***: @MACTL,@MACH  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Cập nhật số lượng CTL của câu hỏi cũ  DECLARE @MACHold BIGINT  SELECT @MACHold = MACH FROM CAUTRALOI WHERE MACTL = @MACTL  UPDATE CAUHOI SET SOCAUTRALOI = SOCAUTRALOI - 1 WHERE MACH = @MACHold | **X(CAUHOI)** |  |  |
| B2: Cập nhật CTL sang câu hỏi khác  UPDATE CAUTRALOI SET MACH = @MACH WHERE MACTL = @MACTL | **X(CAUTRALOI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B3: Cập nhật số lượng CTL của câu hỏi cũ:  UPDATE CAUHOI SET SOCAUTRALOI = SOCAUTRALOI + 1 WHERE MACH = @MACH | **X(CAUHOI)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Cập nhật số lương CTL của câu hỏi  UPDATE CAUHOI SET SOCAUTRALOI = SOCAUTRALOI - 1 WHERE MACH = 1 | **X(CAUHOI)** |
|  |  | WAITFOR DELAY '00:00:10' |  |
|  |  | B3: Xóa câu trả lời :  DELETE FROM CAUTRALOI WHERE MACTL = @MACTL | **X(CAUTRALOI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  IF ( ERROR\_NUMBER() = 1205 )  BEGIN  SET @Return = ERROR\_MESSAGE()  END  ROLLBACK TRAN  ROLLBACK  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR06: *Dirty read***  T1 (User = Quản trị A): thực hiện cập nhật thông tin (khóa tài khoản) của User B  T2 (User = Giáo viên B): thực hiện đăng nhập | | | |
| **sp\_CapNhatThongTin** | **Khóa** | **sp\_DangNhap** | **Khóa** |
| ***Input:***  *@MaND*  *@HoTen*  *@TenDangNhap*  *@ToanQuyenGV*  *@TrangThai,*  *@MaLoai*  *@MaBM*  *@MaGVQ*  ***Output***: @Return | ***Input***:  @tenDangNhap  @matKhau  @maNguoiDung  ***Output***:  @return bit out,  @islocked bit out |
|  | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại giáo viên  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  SET @Return = 1  END | **R(NGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA lOAI NGUOI DUNG KHONG TON TAI'  SET @Return = 2  END | **R(LOAINGUOIDUNG)** |  |  |
| B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 3  END | **R(BOMON)** |  |  |
| B4: Cập nhật thông tin bảng NGUOIDUNG  IF @Return = -1  BEGIN  UPDATE [NGUOIDUNG] SET TENDANGNHAP = @TenDangNhap,HOTEN = @HoTen, TRANGTHAI = @TrangThai, TOANQUYENGV = @ToanQuyenGV, MALOAI = @MaLoai, MABM = @MaBM, MAGVQL = @MaGVQL WHERE [MAND] = @MaND  SET @Return = 0;  END | **X(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra trạng thái của tài khoản  SET @islocked = (SELECT TRANGTHAI FROM NGUOIDUNG WHERE TENDANGNHAP = @tenDangNhap)  IF @islocked = 0  BEGIN  SET @return = 0  END | **R(NGUOIDUNG)** |
|  |  | B2: Kiểm tra sự hợp lệ của username và password  IF EXISTS ( SELECT MAND FROM NGUOIDUNG WHERE TENDANGNHAP = @tenDangNhap AND MATKHAU = @matKhau)  BEGIN  SET @maNguoiDung = ( SELECT MAND FROM NGUOIDUNG WHERE TENDANGNHAP = @tenDangNhap AND MATKHAU = @matKhau ) SET @return = 1  END | **R(NGUOIDUNG)** |
|  |  | COMMIT |  |
| ROLLBACK |  |  |  |
| BEGIN CATCH  PRINT ERROR\_MESSAGE()  SET @Return = 4  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR07: *Unrepeatable Read***  T1 (User = Quản trị A): thực hiện tạo tài khoản mới cho giáo viên C với bộ môn D  T2 (User = Quản trị B ): thực hiện xóa bộ môn D | | | |
| **sp\_TaoTaiKhoan** | **Khóa** | **sp\_XoaBoMon** | **Khóa** |
| ***Input:***  *@HoTen*  *@TenDangNhap*  *@MatKhau*  *@TrangThai*  *@ToanQuyenGV*  *@MaLoai*  *@MaBM*  *@MaGVQL* | ***Input***: @MaBM |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA LOAI NGUOI DUNG KHONG TON TAI'  ROLLBACK TRAN  RETURN 1  END | **R(LOAINGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  ROLLBACK TRAN  RETURN 2  END | **R(BOMON)** |  |  |
| B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của giáo viên quản lý  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaGVQL)  BEGIN  PRINT N'MA GIAO VIEN QUAN LY KHONG TON TAI'  ROLLBACK TRAN  RETURN 3  END | **R(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 1  END | **R(BOMON)** |
|  |  | B2: Xóa bộ môn :  DELETE BOMON WHERE [MABM] = @MaBM | **X(BOMON)** |
|  |  | COMMIT |  |
| B3: Thêm tài khoản mới  INSERT INTO [NGUOIDUNG] ([HOTEN], [TENDANGNHAP], [MATKHAU],[TRANGTHAI], [TOANQUYENGV], [MALOAI], [MABM], [MAGVQL])  VALUES (@HoTen, @TenDangNhap, @MatKhau, @TrangThai, @ToanQuyenGV, @MaLoai, @MaBM, @MaGVQL) | **R(NGUOIDUNG)** |  |  |
|  |  |  |  |
| BEGIN CATCH  PRINT N'LOI HE THONG'  ROLLBACK TRAN  RETURN 1  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR08: *Lost Update***  T1 (User = Quản trị A): thực hiện cập nhật thông tin cho giáo viên C  T2 (User = Quản trị B): thực hiện cập nhật giáo viên quản lý cho giáo viên C | | | |
| **sp\_CapNhatThongTin** | **Khóa** | **sp\_ThayDoiGVQL** | **Khóa** |
| ***Input:***  *@MaND*  *@HoTen*  *@TenDangNhap*  *@ToanQuyenGV*  *@TrangThai,*  *@MaLoai*  *@MaBM*  *@MaGVQL*  ***Output***: @Return | ***Input***:  @MaND  @MaGVQL  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại giáo viên  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  SET @Return = 1  END | **R(NGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA lOAI NGUOI DUNG KHONG TON TAI'  SET @Return = 2  END | **R(LOAINGUOIDUNG)** |  |  |
| B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 3  END | **R(BOMON)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của người dùng  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  SET @Return = 1  END | **R(NGUOIDUNG)** |
|  |  | B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại của giáo viên quản lý  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaGVQL)  BEGIN  PRINT N'MA GVQL KHONG TON TAI'  SET @Return = 2  END | **R(NGUOIDUNG)** |
|  |  | B3: Cập nhật giáo viên quản lý cho giáo viên  UPDATE [NGUOIDUNG] SET MAGVQL = @MaGVQL WHERE [MAND] = @MaND | **X(NGUOIDUNG)** |
|  |  | COMMIT |  |
| B4: Cập nhật thông tin bảng NGUOIDUNG  IF @Return = -1  BEGIN  UPDATE [NGUOIDUNG] SET TENDANGNHAP = @TenDangNhap,HOTEN = @HoTen, TRANGTHAI = @TrangThai, TOANQUYENGV = @ToanQuyenGV, MALOAI = @MaLoai, MABM = @MaBM, MAGVQL = @MaGVQL WHERE [MAND] = @MaND  SET @Return = 0;  END | **X(NGUOIDUNG)** |  |  |
| BEGIN CATCH  PRINT ERROR\_MESSAGE()  SET @Return = 4  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR09: *Phantom***  T1 (User = Quản trị A): thực hiện xóa bộ môn C  T2 (User = Quản trị B): thực hiện tạo tài khoản với bộ môn C | | | |
| **sp\_XoaBoMon** | **Khóa** | **sp\_TaoTaiKhoan** | **Khóa** |
| ***Input:*** *@MaBM*  ***Output***: @Return | ***Input:***  *@HoTen*  *@TenDangNhap*  *@MatKhau*  *@TrangThai*  *@ToanQuyenGV*  *@MaLoai*  *@MaBM*  *@MaGVQL* |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 1  END | **R(BOMON)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra các bộ môn đang được sử dụng  IF EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MABM] = @MaBM)  BEGIN  PRINT N'BO MON DA DUOC SU DUNG. KHONG THE XOA DUOC'  SET @Return = 2  END |  |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA LOAI NGUOI DUNG KHONG TON TAI'  ROLLBACK TRAN  RETURN 1  END | **R(LOAINGUOIDUNG)** |
|  |  | B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  ROLLBACK TRAN  RETURN 2  END | **R(BOMON)** |
|  |  | B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của giáo viên quản lý  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaGVQL)  BEGIN  PRINT N'MA GIAO VIEN QUAN LY KHONG TON TAI'  ROLLBACK TRAN  RETURN 3  END | **R(NGUOIDUNG)** |
|  |  | B3: Thêm tài khoản mới  INSERT INTO [NGUOIDUNG] ([HOTEN], [TENDANGNHAP], [MATKHAU],[TRANGTHAI], [TOANQUYENGV], [MALOAI], [MABM], [MAGVQL])  VALUES (@HoTen, @TenDangNhap, @MatKhau, @TrangThai, @ToanQuyenGV, @MaLoai, @MaBM, @MaGVQL) | **R(NGUOIDUNG)** |
|  |  | COMMIT |  |
| B2: Xóa bộ môn :  DELETE BOMON WHERE [MABM] = @MaBM | **X(BOMON)** |  |  |
| BEGIN CATCH  PRINT N'LOI HE THONG'  SET @Return = 1  ROLLBACK TRAN  RETURN 1  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR10: *Deadlock***  T1 (User = Quản trị A): Cập nhật thông tin người dùng C  T2 (User = Quản trị B ): Xóa tài khoản người dùng C | | | |
| **sp\_CapNhatThongTin** | **Khóa** | **sp\_XoaTaiKhoan** | **Khóa** |
| ***Input:***  *@MaND*  *@HoTen*  *@TenDangNhap*  *@ToanQuyenGV*  *@TrangThai,*  *@MaLoai*  *@MaBM*  *@MaGVQL*  ***Output***: @Return | ***Input***:  *@MaND* |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại giáo viên  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  SET @Return = 1  END | **R(NGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA lOAI NGUOI DUNG KHONG TON TAI'  SET @Return = 2  END | **R(LOAINGUOIDUNG)** |  |  |
| B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 3  END | **R(BOMON)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1): kiểm tra sự tồn tại của người dùng  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  ROLLBACK TRAN  RETURN 1  END | **R(NGUOIDUNG)** |
|  |  | B2: Xóa tài khảon  DELETE [NGUOIDUNG] WHERE [MAND] = @MaND | **X(NGUOIDUNG)** |
|  |  | COMMIT |  |
| BEGIN CATCH  PRINT ERROR\_MESSAGE()  ROLLBACK TRAN  RETURN 1  END CATCH |  |  |  |

### Sinh viên thực hiện: TRỊNH QUỐC THỊNH

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR11: *Dirty read***  T1 (User = Giáo Viên A): Thực hiện xoá câu hỏi  T2 (User = Giáo Viên B): Thực hiện lấy danh sách các câu hỏi | | | |
| **sp\_XoaCauHoi** | **Khóa** | **sp\_Laydanhsachcauhoi** | **Khóa** |
| ***Input:***  *@Macauhoi*  ***Output***: @Return | ***Input***:  *@Macauhoi*  ***Output***:  @return nvarchar(100) out |
|  | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |  |  |
| B2: Xoá Câu hỏi bảng CAUHOI  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **X(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |
|  |  | B2: Lấy danh sách các câu hỏi  SELECT \* FROM CAUHOI WHERE MACH = @MaCauHoi | **R(CAUHOI)** |
|  |  | COMMIT |  |
| ROLLBACK |  |  |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRỊNH QUỐC THỊNH

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR12: *Unrepeatable Read***  T1 (User = Giáo Viên A): thực hiện xem số lượng câu trả lời của 1 câu hỏi  T2 (User = Giáo Viên B ): thực hiện cập nhật số lượng câu trả lời của câu hỏi đó | | | |
| **sp\_Soluongcautraloicuacauhoi** | **Khóa** | **sp\_CapNhatSoLuongCauHoi** | **Khóa** |
| ***Input:***  *@Macauhoi*  ***Output***: @Return | ***Input***:  *@Macauhoi*  *@Socautraloi*  *@Thangdiemdukien*  *@Mucdo*  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |  |  |
| B2: Xem số lượng câu trả lời của 1 câu hỏi  SELECT SOCAUTRALOI FROM CAUHOI c WHERE c.MACH = @Macauhoi | **R(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |
|  |  | B2: Cập nhật số câu hỏi của số lương câu hỏi mà giáo viên A vừa xem :  UPDATE [CAUHOI] SET SOCAUTRALOI = @Socautraloi, THANGDIEM = @Thangdiemdukien , MUCDO = @Mucdo WHERE [MACH] = @Macauhoi | **X(CAUHOI)** |
|  |  | COMMIT |  |
| B3: Xem lại số lượng câu trả lời của câu hỏi trước đó đã xem  SELECT SOCAUTRALOI FROM CAUHOI c WHERE c.MACH = @Macauhoi | **R(CAUHOI)** |  |  |
|  |  |  |  |
| BEGIN CATCH  SET @Return = N'LOI HE THONG'  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRỊNH QUỐC THỊNH

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR13: *Lost Update***  T1 (User = Giáo Viên A): thực hiện cập nhật thông tin của 1 câu hỏi  T2 (User = Giáo Viên B): thực hiện cập nhật môn hoc của 1 câu hỏi | | | |
| **sp\_CapNhatCauHoi** | **Khóa** | **sp\_ThayDoiMonHoc** | **Khóa** |
| ***Input***:  *@Macauhoi*  *@Monhoc*  *@Noidung*  *@Socautraloi*  *@Thangdiemdukien*  *@Mucdo*  ***Output***: @Return | ***Input***:  *@Macauhoi*  *@Monhoc*  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại môn học  ELSE IF NOT EXISTS(SELECT \* FROM MONHOC WHERE [MAMH] = @Monhoc)  BEGIN  SET @Return = N'MA MON HOC KHONG TON TAI'  END | **R(MONHOC)** |  |  |
| B3: Cập nhật câu hỏi  UPDATE [CAUHOI] SET MAMH = @Monhoc,NOIDUNG = @Noidung, SOCAUTRALOI = @Socautraloi, THANGDIEM = @Thangdiemdukien , MUCDO = @Mucdo WHERE [MACH] = @Macauhoi | **X(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |
|  |  | B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại môn học  ELSE IF NOT EXISTS(SELECT \* FROM MONHOC WHERE [MAMH] = @Monhoc)  BEGIN  SET @Return = N'MA MON HOC KHONG TON TAI'  END | **R(MONHOC)** |
|  |  | B3: Cập nhật môn học của câu hỏi mà giáo viên A đang cập nhật  UPDATE [NGUOIDUNG] SET MAGVQL = @MaGVQL WHERE [MAND] = @MaND | **X(MONHOC)** |
|  |  | COMMIT |  |
| B4: Xem lại kết quả cập nhật của mình  Select \* from CAUHOI WHERE [MACH] = @Macauhoi | **R(CAUHOI)** |  |  |
| BEGIN CATCH  SET @Return = N'LOI HE THONG'  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRỊNH QUỐC THỊNH

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR14: *Phantom***  T1 (User = Giáo Viên A): thực hiện lấy số lượng câu hỏi của môn học A  T2 (User = Giáo Viên B): thực hiện thêm câu hỏi vào môn học A | | | |
| **sp\_Soluongcauhoicuabomon** | **Khóa** | **sp\_Themcauhoi** | **Khóa** |
| ***Input***:  *@Mamonhoc*  ***Output***: @Return | ***Input***:  *@Monhoc*  *@Noidung*  *@Socautraloi*  *@Thangdiemdukien*  *@Mucdo*  ***Output***: @Return |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra môn học đó có câu hỏi nào chưa  DECLARE @slch INT  SELECT @slch = COUNT(\*) FROM CAUHOI c WHERE c.MAMH = @Mamonhoc  IF ( @slch < 1 )  SET @Return = N'Mon hoc nay chua co cau hoi nao' | **R(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Thêm câu hỏi mới  INSERT INTO CAUHOI(MAMH,NOIDUNG,SOCAUTRALOI,THANGDIEM,MUCDO) VALUES(@Monhoc,@Noidung,@Socautraloi,@Thangdiemdukien,@Mucdo) | **X(CAUHOI)** |
|  |  | COMMIT |  |
| B2: Lấy số lượng câu hỏi của môn học A  SELECT COUNT(\*) FROM CAUHOI c WHERE c.MAMH = @Mamonhoc | **X(BOMON)** |  |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRỊNH QUỐC THỊNH

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR15: *Deadlock***  T1 (User = Giáo Viên A): Cập nhật câu hỏi A  T2 (User = Giáo Viên B ): Xóa câu hỏi A | | | |
| **sp\_CapNhatCauHoi** | **Khóa** | **sp\_XoaCauHoi** | **Khóa** |
| ***Input***:  *@Macauhoi*  *@Monhoc*  *@Noidung*  *@Socautraloi*  *@Thangdiemdukien*  *@Mucdo*  ***Output***: @Return | ***Input***:  *@Macauhoi* |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại môn học  ELSE IF NOT EXISTS(SELECT \* FROM MONHOC WHERE [MAMH] = @Monhoc)  BEGIN  SET @Return = N'MA MON HOC KHONG TON TAI'  END | **R(MONHOC)** |  |  |
| B3: Cập nhật câu hỏi  UPDATE [CAUHOI] SET MAMH = @Monhoc,NOIDUNG = @Noidung, SOCAUTRALOI = @Socautraloi, THANGDIEM = @Thangdiemdukien , MUCDO = @Mucdo WHERE [MACH] = @Macauhoi | **X(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |
|  |  | B2: Xóa câu hỏi  DELETE CauHoi WHERE MACH = @Macauhoi | **X(CAUHOI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = N'LOI HE THONG'  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: HUỲNH DUY THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR16: *Dirty read***  T1 Giáo viên quản lý cập nhật BỎ quyền thay đổi câu trắc nghiệm của giáo viên  T2 Lấy danh sách giáo viên có quyền thay đổi câu trắc nghiệm | | | |
| **SP\_GVQLCAPQUYEN** | **Khóa** | **SP\_LOADDSGV** | **Khóa** |
| ***Input:*** *@MAGVQL, @MAGV*  ***Output***: @Return | ***Input***:  ***Output***: |
|  | SET TRANSACTION ISOLATION  LEVEL **READ UNCOMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS (SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGVQL AND ND.MALOAI = 2)  RETURN 0 | **R(NGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) :  IF NOT EXISTS (SELECT 1 FROM NGUOIDUNG ND1, NGUOIDUNG ND2  WHERE ND1.MAND = @MAGV AND ND2.MAND = @MAGVQL AND ND1.MABM = ND2.MABM)  RETURN 0 | **R(NGUOIDUNG)** |  |  |
| B3: Update BỎ quyền thao tác câu trắc nghiệm của giáo viên  UPDATE NGUOIDUNG  SET TOANQUYENGV = 0  WHERE MAND = @MAGV | **X(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | BEGIN TRY  SELECT ND.MAND FROM NGUOIDUNG ND  WHERE ND.TOANQUYENGV = 1  END TRY | **R(NGUOIDUNG)** |
|  |  | BEGIN CATCH  PRINT 'LOI LOAD GV'  ROLLBACK TRAN  END CATCH |  |
|  |  | END TRAN |  |
| ROLLBACK  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| END TRAN |  |  |  |

### Sinh viên thực hiện: HUỲNH DUY THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR17: *Unrepeatable read***  T1 Hiển thị danh sách giáo viên của giáo viên quản lý A nào đó  T2 Cập nhật giáo viên quản lý A thành B của giáo viên K | | | |
| **SP\_LOAD\_DSGV\_CUNG\_GVQL** | **Khóa** | **SP\_UPDATEGVQL** | **Khóa** |
| ***Input:***  *@MAGVQL*  ***Output***: @Return | ***Input***: @MAGV, @MAGVQL  ***Output***: |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGVQL AND  ND.MALOAI = 2) -- 2: GVQL  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |  |  |
| B2: Hiển thị số lượng giáo viên (2) :  DECLARE @COUNT INT = (SELECT COUNT(\*) FROM  NGUOIDUNG WHERE MAGVQL = @MAGVQL)  PRINT 'CO ' + CAST(@COUNT AS VARCHAR(10)) + ' GV THUOC CUNG GVQL LA ' + CAST(@MAGVQL AS VARCHAR(10)) + ' :' | **R(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1):  IF NOT EXISTS (SELECT \* FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGVQL AND  ND.MALOAI = 2) -- 2: GVQL  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |
|  |  | B3: Update mã giáo viên quản lý (3):  UPDATE NGUOIDUNG  SET MAGVQL = @MAGVQL  WHERE MAND = @MAGV | **X(NGUOIDUNG)** |
|  |  | BEGIN CATCH  PRINT 'LOI LOAD GV'  ROLLBACK TRAN  END CATCH |  |
|  |  | COMMIT TRAN |  |
| B3: In danh sách giáo viên (3): |  |  |  |
| SELECT \* FROM NGUOIDUNG WHERE MAGVQL = @MAGVQL |  |  |  |
| BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| COMMIT TRAN |  |  |  |

### Sinh viên thực hiện: HUỲNH DUY THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR18: Lost update**  T1 Update quyền giáo viên  T2 Update quyền giáo viên | | | |
| **sp\_UpdateToanQuyenGV1** | **Khóa** | **sp\_UpdateToanQuyenGV2** | **Khóa** |
| ***Input:***  *@MaGV, @MaGVQL*  ***Output***: | ***Input***: @MAGV, @MAGVQL  ***Output***: |
|  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGV AND  ND.MAGVQL = @MaGVQL)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGV AND  ND.MAGVQL = @MaGVQL)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |
|  |  | B2: Update toàn quyền giáo viên (2):  DECLARE @VALUE BIT = (SELECT ND.TOANQUYENGV FROM NGUOIDUNG ND WHERE ND.MAND = @MaGV)  UPDATE NGUOIDUNG  SET TOANQUYENGV = 1 - @VALUE  WHERE MAND = @MaGV | **X(NGUOIDUNG)** |
|  |  | BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |
|  |  | COMMIT TRAN |  |
| B2: Update toàn quyền giáo viên (2):  DECLARE @VALUE BIT = (SELECT ND.TOANQUYENGV FROM NGUOIDUNG ND WHERE ND.MAND = @MaGV)  UPDATE NGUOIDUNG  SET TOANQUYENGV = 1 - @VALUE  WHERE MAND = @MaGV |  |  |  |
| BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| COMMIT TRAN |  |  |  |

### Sinh viên thực hiện: HUỲNH DUY THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR19: Deadlock**  T1 Update quyền giáo viên  T2 Update quyền giáo viên | | | |
| **sp\_UpdateToanQuyenGV1** | **Khóa** | **sp\_UpdateToanQuyenGV2** | **Khóa** |
| ***Input:***  *@MaGV, @MaGVQL*  ***Output***: | ***Input***: @MAGV, @MAGVQL  ***Output***: |
| SET TRAN ISOLATION LEVEL REPEATABLE READ | SET TRAN ISOLATION LEVEL REPEATABLE READ |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGV AND  ND.MAGVQL = @MaGVQL)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGV AND  ND.MAGVQL = @MaGVQL)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |
|  |  | B2: Update toàn quyền giáo viên (2):  DECLARE @VALUE BIT = (SELECT ND.TOANQUYENGV FROM NGUOIDUNG ND WHERE ND.MAND = @MaGV)  UPDATE NGUOIDUNG  SET TOANQUYENGV = 1 - @VALUE  WHERE MAND = @MaGV | **X(NGUOIDUNG)** |
|  |  | BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |
|  |  | COMMIT TRAN |  |
| B2: Update toàn quyền giáo viên (2):  DECLARE @VALUE BIT = (SELECT ND.TOANQUYENGV FROM NGUOIDUNG ND WHERE ND.MAND = @MaGV)  UPDATE NGUOIDUNG  SET TOANQUYENGV = 1 - @VALUE  WHERE MAND = @MaGV |  |  |  |
| BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| COMMIT TRAN |  |  |  |

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR21: *Dirty read***  T1 (User = giáo viên A): Thực hiện đổi thông tin Môn học có mã là “x” và đổi Tên môn học thành “X”  T2 (User = giáo viên B ): Thực hiện lấy Môn học có mã là “x” và Tên môn học là “X” | | | |
| **sp\_CapNhatMonHoc** | **Khóa** | **sp\_LayMonHoc** | **Khóa** |
| ***Input:*** *@MAMH, @TENMH*  ***Output***: *@KETQUA* | ***Input:*** *@MAMH, @TENMH*  ***Output***: *@MAMH, @TENMH, @MABM* |
| BEGIN TRAN |  | SET TRAN ISOLATION LEVEL READ UNCOMMITTED |  |
| BEGIN TRY  UPDATE MONHOC  SET @TENMH = @TENMH  WHERE MAMH = @MAMH | **X(MONHOC)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | BEGIN TRY  SELECT \* FROM MONHOC WHERE MAMH = @MAMH AND TENMH = @TENMH  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH | **R(MONHOC)**  //Không cần xin khoá |
|  |  | COMMIT |  |
|  |  | END TRAN |  |
| ROLLBACK TRAN  RETURN 0  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| END TRAN |  |  |  |

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR22: *Unrepeatable Read***  T1 (User = giáo viên A): thực hiện lấy thông tin Môn học có tên là “a” và có Mã bộ môn là “b”  T2 (User = giáo viên B ): thực hiện cập nhật thông tin của Môn học có tên là “a” và có Mã bộ môn là “b” đổi thành Mã bộ môn “c” | | | |
| **sp\_LayMonHoc** | **Khóa** | **sp\_CapNhatMonHoc** | **Khóa** |
| ***Input:*** *@MAMH, @MABM*  ***Output****: @Return* | ***Input:*** *@MAMH, @MABM*  ***O*utput**: @Return |
| BEGIN TRAN |  |  |  |
| B1: In ra số lượng Môn học  BEGIN TRY  DECLARE @COUNT INT = (SELECT COUNT(\*) FROM MONHOC WHERE MAMH = @MAMH AND MABM = @MABM)  PRINT 'CO ' + CAST(@COUNT AS VARCHAR(10)) + ' KET QUA.' | **R(MONHOC)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B3: Lấy danh sách các môn học thỏa điều kiện  SELECT \* FROM MONHOC WHERE MAMH = @MAMH AND MABM = @MABM  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH | **R(MONHOC)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin Mã bộ môn có tồn tại  BEGIN TRY  IF NOT EXISTS (SELECT \*  FROM BOMON  WHERE MABM = @MABM)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(MONHOC)**  //Không cần xin khoá |
|  |  | B2: Cập nhật Môn học  UPDATE MONHOC  SET MABM = @MABM  WHERE MAMH = @MAMH  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH  RETURN 1 | **X(MONHOC)**  //Không cần xin khoá |
|  |  | COMMIT |  |

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR23: *Phantom***  T1 (User = giáo viên A): lấy danh sách môn học có Mã bộ môn là X (Trả về số lượng và Kết quả)  T2 (User = giáo viên B ): thực hiện thêm môn học có Mã bộ môn là X | | | |
| **sp\_LayMonHoc** | **Khóa** | **sp\_ThemMonHoc** | **Khóa** |
| ***Input:*** *@MABM*  ***Output****: @Return* | ***Input****: @MaBM, @TenMH*  ***Output****: @Return* |
| SET TRAN ISOLATION LEVEL REPEATABLE READ | SET TRAN ISOLATION LEVEL READ COMMITTED |
| BEGIN TRAN |  |  |  |
| BEGIN TRY  --Đếm xem có bao nhiêu phần tử  DECLARE @COUNT INT = (SELECT COUNT(\*) FROM MONHOC WHERE MABM = @MABM)  PRINT 'CO ' + CAST(@COUNT AS VARCHAR(10)) + ' KET QUA.' | **R(MONHOC)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | --Thực hiện thêm môn học mới  BEGIN TRY  INSERT INTO MONHOC(TENMH, MABM) VALUES (@TENMH, @MABM)  END TRY  BEGIN CATCH  ROLLBACK TRAN  END CATCH | **X(MONHOC)** |
|  |  | COMMIT |  |
| --Lấy danh sách các phần tử  SELECT \* FROM MONHOC WHERE MABM = @MABM  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH | **R(MONHOC)** |  |  |
| COMMIT |  |  |  |

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR24: *Deadlock***  T1 (User = Quản trị A): Cập nhật môn học A  T2 (User = Quản trị B ): Xóa môn học A | | | |
| **sp\_CapNhatMonHoc** | **Khóa** | **sp\_XoaMonHoc** | **Khóa** |
| ***Input:***  *@MaMH*  *@TenMH* | ***Input***:  *@MaMH* |
| SET TRAN ISOLATION LEVEL SERIALIZABLE | SET TRAN ISOLATION LEVEL SERIALIZABLE |
| BEGIN TRAN |  |  |  |
| --Kiểm tra tồn tại môn học  BEGIN TRY  IF NOT EXISTS(SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học không tồn tại'  ROLLBACK TRAN  END | **R(MONHOC)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | --Kiểm tra tồn tại môn học  BEGIN TRY  IF NOT EXISTS(SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học không tồn tại'  ROLLBACK TRAN  END | **R(MONHOC)** |
|  |  | --Thực hiện xóa môn học  DELETE FROM MONHOC WHERE MAMH = @MAMH  END TRY  BEGIN CATCH  PRINT ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH | **X(MONHOC)** |
|  |  | COMMIT |  |
| UPDATE MONHOC  SET TENMH = @TENMH  WHERE MAMH = @MAMH  END TRY  BEGIN CATCH  PRINT ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH | **X(MONHOC)** |  |  |
| COMMIT |  |  |  |

## Giải pháp khắc phục lỗi tranh chấp đồng thời

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Dirty read***  T1 (User = giáo viên A): thực hiện thêm 1 câu trả lời cho câu hỏi C  T2 (User = giáo viên B ): thực hiện xem danh sách câu trả lời cho câu hỏi C  Cách khác phục : sử dụng múc cô lập Read Commited Phát S khi đọc  //Cụ thể khắc phục ở chỗ nào???? | | | |
| **sp\_Them1ChuyenXe** | **Khóa** | **sp\_ThongKeChuyenXe-CoDieuKien** | **Khóa** |
| ***Input:*** *@NOIDUNG, @LADAPANDUNG, @MACH*  ***Output***: @Return | ***Input***: @MACH  ***Output***: @Return |
|  | SET TRANSACTION ISOLATION  LEVEL **READ COMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : không tồn tại câu hỏi  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE MACH = @MACH )  BEGIN  SET @Return = N'Không tồn tại câu hỏi này'  END | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) : đã tồn tại câu trả lời trong cau hỏi này  IF EXISTS ( SELECT \* FROM CAUTRALOI WHERE MACH = @MACH AND NOIDUNG = @NOIDUNG )  BEGIN  SET @Return = N'Trong câu hỏi đã tồn tại câu trả lời này'  END | **R(CAUTRALOI)** |  |  |
| B3: Thêm thông tin vào bảng CAUTRALOI  INSERT INTO CAUTRALOI(NOIDUNG,LADAPANDUNG,MACH) VALUES(@NOIDUNG,@LADAPANDUNG,@MACH) | **X(CAUTRALOI)** |  |  |
| WAITFOR DELAY '00:00:20' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : không tồn tại câu hỏi  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE @MACH = MACH )  BEGIN  SET @Return = N'Không tồn tại câu hỏi này'  END | **R(CAUHOI)**  //Không cần xin khoá |
|  |  | B2: Xem thông tin câu hỏi  SELECT \* FROM CAUHOI WHERE MACH = @MACH | **R(CAUHOI)**  //Không cần xin khoá |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  ROLLBACK  // tg\_SoLuongCauTraLoiCuaCauHoi > 10  END CATCH |  |  |  |

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR02: *Unrepeatable Read***  T1 (User = giáo viên A): thực hiện lấy danh sách câu hỏi theo nội dung và mức độ  T2 (User = giáo viên B ): thực hiện cập nhật thông tin câu hỏi ( mức độ )  Cách khắc phục : sử dụng múc cô lập Repeatable Read Phát S khi đọc và giữ S đến khi transaction kết thúc  Cụ thể khắc phục ở chỗ : Do T1 phát và giữ S cho đến khi commit nên T1 luôn kết thúc trước T2. Vì vậy T1 sẽ ko bị lỗi | | | |
| **sp\_LayDanhSachCaiHoiTheoMucDO\_DEMO** | **Khóa** | **sp\_CapNhatCauHoi\_DEMO** | **Khóa** |
| ***Input:*** *@NOIDUNG, @MUCDO*  ***Output***: @Return | ***Input***: @MACH  ***Output***: @Return |
| SET TRAN ISOLATION LEVEL REPEATABLE READ |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS( SELECT \* FROM CAUHOI WHERE @NOIDUNG like '%' + @NOIDUNG + '%' OR @NOIDUNG = '' )  SET @Return = N'Không có câu hỏi chứa nội dung này' | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) :  IF NOT EXISTS( SELECT \* FROM CAUHOI WHERE @NOIDUNG like '%' + @NOIDUNG + '%' AND MUCDO = @MUCDO )  SET @Return = N'Không có câu hỏi có mức độ này' | **R(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B3: Lấy danh sách câu hỏi thỏa điều kiện  SELECT c.\*,m.TENMH FROM CAUHOI c,MonHoc m  WHERE c.MAMH = m.MAMH AND NOIDUNG like '%' + @NOIDUNG + '%' AND c.MUCDO = @MUCDO  SET @Return = ‘’ | **R(CAUHOI)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) :  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE @MACH = MACH )  BEGIN  SET @Return = N'Không tồn tại câu hỏi này'  END | **R(CAUHOI)**  //Không cần xin khoá |
|  |  | B2: Cập nhật thông tin cho câu hỏi (2) :  UPDATE CAUHOI SET NOIDUNG = @NOIDUNG, THANGDIEM = @THANGDIEM, MUCDO = @MUCDO  WHERE MACH = @MACH  SET @Return = '' | **X(CAUHOI)**  //Không cần xin khoá |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR03: *Lost Update***  T1 (User = giáo viên A): thực hiện lấy thêm câu trả lời vào câu hỏi  T2 (User = giáo viên B ): thực hiện xóa câu trả lời ( thuộc câu hỏi của T1 )  Cách khắc phục : sử dụng múc cô lập Repeatable Read Phát S khi đọc và giữ S đến khi transaction kết thúc  Cụ thể khắc phục ở chỗ : mức cô lập chưa giải quyết được trường hợp này. T2 là nạn nhân sẽ được HQTCSDL hủy | | | |
| **sp\_ThemCauTraLoiVaoCauHoi\_DEMO** | **Khóa** | **sp\_XoaCauTraLoi\_DEMO** | **Khóa** |
| ***Input:*** *@NOIDUNG, @LADAPANDUNG, @MACH*  ***Output***: @Return | ***Input***: @MACTL  ***Output***: @Return |
| SET TRAN ISOLATION LEVEL REPEATABLE READ | SET TRAN ISOLATION LEVEL REPEATABLE READ |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE MACH = @MACH )  SET @Return = N'Không tồn tại câu hỏi này' | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) :  IF EXISTS ( SELECT \* FROM CAUTRALOI WHERE MACH = @MACH AND NOIDUNG = @NOIDUNG )  SET @Return = N'Trong câu hỏi đã tồn tại câu trả lời này' | **R(CAUTRALOI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B3: Thêm câu trả lời vào câu hỏi  INSERT INTO CAUTRALOI(NOIDUNG,LADAPANDUNG,MACH) VALUES(@NOIDUNG,@LADAPANDUNG,@MACH) | **X(CAUTRALOI)** |  |  |
| B4: Cập nhật số lượng CTL cho câu hỏi (trigger) | **X(CAUHOI)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) :  IF NOT EXISTS ( SELECT \* FROM CAUTRALOI WHERE MACTL = @MACTL )  SET @Return = hông tồn tại câu trả lời này' | **R(CAUTRALOI)** |
|  |  | B2: Xóa câu trả lời (2) :  DELETE FROM CAUTRALOI WHERE MACTL = @MACTL | **X(CAUTRALOI)** |
|  |  | B3: Cập nhật số lượng câu trả lời cho câu hỏi (3) (trigger) | **X(CAUHOI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: TRẦN LÊ TUẤN

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR04: *Phantom***  T1 (User = giáo viên A): thực hiện lấy danh sách câu hỏi trong bộ đề thi C  T2 (User = giáo viên B ): thực hiện thêm câu hỏi vào trong bộ đề thi C  Cách khắc phục : sử dụng múc cô lập Serialization cho T1 để phát S khi đọc và giữ S đến khi transaction kết thúc, có ngăn chặn lệnh Insert  Cụ thể khắc phục ở chỗ : Do khóa S đươc phát giữ đến hết T1 và ngăn chặn Insert nên T1 luôn kết thúc trước T2. | | | |
| **sp\_LayDanhSachCauHoiTheoDeThi\_DEMO** | **Khóa** | **sp\_ThemCauHoiVaoBoDeThi\_DEMO** | **Khóa** |
| ***Input:*** *@MABDT*  ***Output***: @Return | ***Input***: @MABDT,@MACH,@Diem,@Return  ***Output***: @Return |
| SET TRANSACTION ISOLATION LEVEL SERIALIZABLE |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) :  DECLARE @slCauHoi INT  SELECT @slCauHoi = COUNT(\*) FROM TAOBODETHI WHERE MABDT = @MABDT  IF ( @slCauHoi < 0 )  SET @Return = N'Bộ đề thi chưa có bất câu hỏi nào' | **R(TAOBODETHI)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B2: Lấy danh sách câu hỏi trong bộ đề thi :  SELECT c.\*  FROM TAOBODETHI t , CAUHOI c  WHERE t.MABDT = @MABDT AND t.MACH = c.MACH | **R(TAOBODETHI,CAUHOI)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) :  IF NOT EXISTS ( SELECT \* FROM BODETHI WHERE @MABDT = MABDT )  SET @Return = N'Không tồn tại bộ đề thi này' | **R(CAUHOI)** |
|  |  | B2: Kiểm tra thông tin (2) :  IF NOT EXISTS ( SELECT \* FROM CAUHOI WHERE @MACH = MACH )  SET @Return = N'Không tồn tại câu hỏi này' | **R(CAUHOI)** |
|  |  | B3: Thêm câu hỏi vào bộ đề thi :  INSERT INTO TAOBODETHI(MABDT,MACH,DIEM) VALUES(@MABDT,@MACH,@DIEM) | **X(TAOBODETHI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = ERROR\_MESSAGE()  ROLLBACK TRAN  ROLLBACK  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR06: *Dirty read***  T1 (User = Quản trị A): thực hiện cập nhật thông tin (khóa tài khoản) của User B  T2 (User = Giáo viên B): thực hiện đăng nhập  Cách khắc phục : sử dụng mức cô lập Read Commited phát khóa S khi đọc cho sp\_DangNhap | | | |
| **sp\_CapNhatThongTin** | **Khóa** | **sp\_DangNhap** | **Khóa** |
| ***Input:***  *@MaND*  *@HoTen*  *@TenDangNhap*  *@ToanQuyenGV*  *@TrangThai,*  *@MaLoai*  *@MaBM*  *@MaGVQ*  ***Output***: @Return | ***Input***:  @tenDangNhap  @matKhau  @maNguoiDung  ***Output***:  @return bit out,  @islocked bit out |
|  | SET TRANSACTION ISOLATION  LEVEL **READ COMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại giáo viên  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  SET @Return = 1  END | **R(NGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA lOAI NGUOI DUNG KHONG TON TAI'  SET @Return = 2  END | **R(LOAINGUOIDUNG)** |  |  |
| B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 3  END | **R(BOMON)** |  |  |
| B4: Cập nhật thông tin bảng NGUOIDUNG  IF @Return = -1  BEGIN  UPDATE [NGUOIDUNG] SET TENDANGNHAP = @TenDangNhap,HOTEN = @HoTen, TRANGTHAI = @TrangThai, TOANQUYENGV = @ToanQuyenGV, MALOAI = @MaLoai, MABM = @MaBM, MAGVQL = @MaGVQL WHERE [MAND] = @MaND  SET @Return = 0;  END | **X(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra trạng thái của tài khoản  SET @islocked = (SELECT TRANGTHAI FROM NGUOIDUNG WHERE TENDANGNHAP = @tenDangNhap)  IF @islocked = 0  BEGIN  SET @return = 0  END | **R(NGUOIDUNG)** |
|  |  | B2: Kiểm tra sự hợp lệ của username và password  IF EXISTS ( SELECT MAND FROM NGUOIDUNG WHERE TENDANGNHAP = @tenDangNhap AND MATKHAU = @matKhau)  BEGIN  SET @maNguoiDung = ( SELECT MAND FROM NGUOIDUNG WHERE TENDANGNHAP = @tenDangNhap AND MATKHAU = @matKhau ) SET @return = 1  END | **R(NGUOIDUNG)** |
|  |  | COMMIT |  |
| ROLLBACK |  |  |  |
| BEGIN CATCH  PRINT ERROR\_MESSAGE()  SET @Return = 4  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR07: *Unrepeatable Read***  T1 (User = Quản trị A): thực hiện tạo tài khoản mới cho giáo viên C với bộ môn D  T2 (User = Quản trị B ): thực hiện xóa bộ môn D  Cách khắc phục : sử dụng mức cô lập Repeatable Read CHO sp\_TaoTaiKhoan phát S khi đọc và giữ S đến khi giao tác kết thúc  Do T1 phát và giữ S cho đến khi commit nên T1 luôn kết thúc trước T2. Vì vậy T1 sẽ ko bị xung đột với T2 | | | |
| **sp\_TaoTaiKhoan** | **Khóa** | **sp\_XoaBoMon** | **Khóa** |
| ***Input:***  *@HoTen*  *@TenDangNhap*  *@MatKhau*  *@TrangThai*  *@ToanQuyenGV*  *@MaLoai*  *@MaBM*  *@MaGVQL* | ***Input***: @MaBM |
| SET TRANSACTION ISOLATION  LEVEL **REPEATABLE READ** |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA LOAI NGUOI DUNG KHONG TON TAI'  ROLLBACK TRAN  RETURN 1  END | **R(LOAINGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  ROLLBACK TRAN  RETURN 2  END | **R(BOMON)** |  |  |
| B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của giáo viên quản lý  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaGVQL)  BEGIN  PRINT N'MA GIAO VIEN QUAN LY KHONG TON TAI'  ROLLBACK TRAN  RETURN 3  END | **R(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 1  END | **R(BOMON)** |
|  |  | B2: Xóa bộ môn :  DELETE BOMON WHERE [MABM] = @MaBM | **X(BOMON)** |
|  |  | COMMIT |  |
| B3: Thêm tài khoản mới  INSERT INTO [NGUOIDUNG] ([HOTEN], [TENDANGNHAP], [MATKHAU],[TRANGTHAI], [TOANQUYENGV], [MALOAI], [MABM], [MAGVQL])  VALUES (@HoTen, @TenDangNhap, @MatKhau, @TrangThai, @ToanQuyenGV, @MaLoai, @MaBM, @MaGVQL) | **R(NGUOIDUNG)** |  |  |
|  |  |  |  |
| BEGIN CATCH  PRINT N'LOI HE THONG'  ROLLBACK TRAN  RETURN 1  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR08: *Lost Update***  T1 (User = Quản trị A): thực hiện cập nhật thông tin cho giáo viên C  T2 (User = Quản trị B): thực hiện cập nhật giáo viên quản lý cho giáo viên C  Cách khắc phục : sử dụng mức cô lập Repeatable Read cho sp\_CapNhatThongTin phát S khi đọc và giữ S đến khi giao tác kết thúc.  Do T1 phát và giữ S cho đến khi commit nên T1 luôn kết thúc trước T2. Vì vậy T1 sẽ ko bị xung đột với T2 | | | |
| **sp\_CapNhatThongTin** | **Khóa** | **sp\_ThayDoiGVQL** | **Khóa** |
| ***Input:***  *@MaND*  *@HoTen*  *@TenDangNhap*  *@ToanQuyenGV*  *@TrangThai,*  *@MaLoai*  *@MaBM*  *@MaGVQL*  ***Output***: @Return | ***Input***:  @MaND  @MaGVQL  ***Output***: @Return |
| SET TRANSACTION ISOLATION  LEVEL **REPEATABLE READ** |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại giáo viên  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  SET @Return = 1  END | **R(NGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA lOAI NGUOI DUNG KHONG TON TAI'  SET @Return = 2  END | **R(LOAINGUOIDUNG)** |  |  |
| B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 3  END | **R(BOMON)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của người dùng  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaND)  BEGIN  PRINT N'MA NGUOI DUNG KHONG TON TAI'  SET @Return = 1  END | **R(NGUOIDUNG)** |
|  |  | B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại của giáo viên quản lý  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaGVQL)  BEGIN  PRINT N'MA GVQL KHONG TON TAI'  SET @Return = 2  END | **R(NGUOIDUNG)** |
|  |  | B3: Cập nhật giáo viên quản lý cho giáo viên  UPDATE [NGUOIDUNG] SET MAGVQL = @MaGVQL WHERE [MAND] = @MaND | **X(NGUOIDUNG)** |
|  |  | COMMIT |  |
| B4: Cập nhật thông tin bảng NGUOIDUNG  IF @Return = -1  BEGIN  UPDATE [NGUOIDUNG] SET TENDANGNHAP = @TenDangNhap,HOTEN = @HoTen, TRANGTHAI = @TrangThai, TOANQUYENGV = @ToanQuyenGV, MALOAI = @MaLoai, MABM = @MaBM, MAGVQL = @MaGVQL WHERE [MAND] = @MaND  SET @Return = 0;  END | **X(NGUOIDUNG)** |  |  |
| BEGIN CATCH  PRINT ERROR\_MESSAGE()  SET @Return = 4  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: NGUYỄN LÊ QUANG THẮNG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR09: *Phantom***  T1 (User = Quản trị A): thực hiện xóa bộ môn C  T2 (User = Quản trị B): thực hiện tạo tài khoản với bộ môn C  Cách khắc phục : sử dụng mức cô lập Serialization cho T1 để phát S khi đọc và giữ S đến khi giao tác kết thúc, có ngăn chặn lệnh Insert  Do khóa S đươc phát giữ đến hết T1 và ngăn chặn Insert. Do đó T2 phải chờ T1 kết thúc rồi mới thực hiện lệnh Insert. | | | |
| **sp\_XoaBoMon** | **Khóa** | **sp\_TaoTaiKhoan** | **Khóa** |
| ***Input:*** *@MaBM*  ***Output***: @Return | ***Input:***  *@HoTen*  *@TenDangNhap*  *@MatKhau*  *@TrangThai*  *@ToanQuyenGV*  *@MaLoai*  *@MaBM*  *@MaGVQL* |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  SET @Return = 1  END | **R(BOMON)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra các bộ môn đang được sử dụng  IF EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MABM] = @MaBM)  BEGIN  PRINT N'BO MON DA DUOC SU DUNG. KHONG THE XOA DUOC'  SET @Return = 2  END |  |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của loại người dùng  IF NOT EXISTS(SELECT \* FROM LOAINGUOIDUNG WHERE [MALOAI] = @MaLoai)  BEGIN  PRINT N'MA LOAI NGUOI DUNG KHONG TON TAI'  ROLLBACK TRAN  RETURN 1  END | **R(LOAINGUOIDUNG)** |
|  |  | B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại của bộ môn  IF NOT EXISTS(SELECT \* FROM BOMON WHERE [MABM] = @MaBM)  BEGIN  PRINT N'MA BO MON KHONG TON TAI'  ROLLBACK TRAN  RETURN 2  END | **R(BOMON)** |
|  |  | B3: Kiểm tra thông tin (3) : kiểm tra sự tồn tại của giáo viên quản lý  IF NOT EXISTS(SELECT \* FROM NGUOIDUNG WHERE [MAND] = @MaGVQL)  BEGIN  PRINT N'MA GIAO VIEN QUAN LY KHONG TON TAI'  ROLLBACK TRAN  RETURN 3  END | **R(NGUOIDUNG)** |
|  |  | B3: Thêm tài khoản mới  INSERT INTO [NGUOIDUNG] ([HOTEN], [TENDANGNHAP], [MATKHAU],[TRANGTHAI], [TOANQUYENGV], [MALOAI], [MABM], [MAGVQL])  VALUES (@HoTen, @TenDangNhap, @MatKhau, @TrangThai, @ToanQuyenGV, @MaLoai, @MaBM, @MaGVQL) | **R(NGUOIDUNG)** |
|  |  | COMMIT |  |
| B2: Xóa bộ môn :  DELETE BOMON WHERE [MABM] = @MaBM | **X(BOMON)** |  |  |
| BEGIN CATCH  PRINT N'LOI HE THONG'  SET @Return = 1  ROLLBACK TRAN  RETURN 1  END CATCH |  |  |  |

### Sinh viên thực hiện: TRỊNH QUỐC THỊNH

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR15: *Deadlock***  T1 (User = Giáo Viên A): Cập nhật câu hỏi A  T2 (User = Giáo Viên B ): Xóa câu hỏi A | | | |
| **sp\_CapNhatCauHoi** | **Khóa** | **sp\_XoaCauHoi** | **Khóa** |
| ***Input***:  *@Macauhoi*  *@Monhoc*  *@Noidung*  *@Socautraloi*  *@Thangdiemdukien*  *@Mucdo*  ***Output***: @Return | ***Input***:  *@Macauhoi* |
| SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** | SET TRANSACTION ISOLATION  LEVEL **SERIALIZABLE** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |  |  |
| B2: Kiểm tra thông tin (2) : kiểm tra sự tồn tại môn học  ELSE IF NOT EXISTS(SELECT \* FROM MONHOC WHERE [MAMH] = @Monhoc)  BEGIN  SET @Return = N'MA MON HOC KHONG TON TAI'  END | **R(MONHOC)** |  |  |
| B3: Cập nhật câu hỏi  UPDATE [CAUHOI] SET MAMH = @Monhoc,NOIDUNG = @Noidung, SOCAUTRALOI = @Socautraloi, THANGDIEM = @Thangdiemdukien , MUCDO = @Mucdo WHERE [MACH] = @Macauhoi | **X(CAUHOI)** |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1) : kiểm tra sự tồn tại của câu hỏi  IF NOT EXISTS(SELECT \* FROM CAUHOI WHERE [MACH] = @Macauhoi)  BEGIN  SET @Return = N'MA CAU HOI KHONG TON TAI'  END  BEGIN  DELETE CauHoi WHERE MACH = @Macauhoi  SET @Return = '';  END | **R(CAUHOI)** |
|  |  | B2: Xóa câu hỏi  DELETE CauHoi WHERE MACH = @Macauhoi | **X(CAUHOI)** |
|  |  | COMMIT |  |
| BEGIN CATCH  SET @Return = N'LOI HE THONG'  ROLLBACK TRAN  END CATCH |  |  |  |

### Sinh viên thực hiện: HUỲNH DUY THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR16: *Dirty read***  T1 Giáo viên quản lý cập nhật BỎ quyền thay đổi câu trắc nghiệm của giáo viên  T2 Lấy danh sách giáo viên có quyền thay đổi câu trắc nghiệm  Cách khắc phục : sử dụng múc cô lập Read committed khi đọc ở T2 | | | |
| **SP\_GVQLCAPQUYEN** | **Khóa** | **SP\_LOADDSGV** | **Khóa** |
| ***Input:***  *@MAGVQL, @MAGV*  ***Output***: @Return | ***Input***:  ***Output***: |
|  | SET TRANSACTION ISOLATION  LEVEL **READ COMMITTED** |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS (SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGVQL AND ND.MALOAI = 2)  RETURN 0 | **R(NGUOIDUNG)** |  |  |
| B2: Kiểm tra thông tin (2) :  IF NOT EXISTS (SELECT 1 FROM NGUOIDUNG ND1, NGUOIDUNG ND2  WHERE ND1.MAND = @MAGV AND ND2.MAND = @MAGVQL AND ND1.MABM = ND2.MABM)  RETURN 0 | **R(NGUOIDUNG)** |  |  |
| B3: Update BỎ quyền thao tác câu trắc nghiệm của giáo viên  UPDATE NGUOIDUNG  SET TOANQUYENGV = 0  WHERE MAND = @MAGV | **X(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | BEGIN TRY  SELECT ND.MAND FROM NGUOIDUNG ND  WHERE ND.TOANQUYENGV = 1  END TRY | **R(NGUOIDUNG)** |
|  |  | BEGIN CATCH  PRINT 'LOI LOAD GV'  ROLLBACK TRAN  END CATCH |  |
|  |  | END TRAN |  |
| ROLLBACK  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| END TRAN |  |  |  |

### Sinh viên thực hiện: HUỲNH DUY THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR17: *Unrepeatable read***  T1 Hiển thị danh sách giáo viên của giáo viên quản lý A nào đó  T2 Cập nhật giáo viên quản lý A thành B của giáo viên K  Cách khắc phục : sử dụng múc cô lập Repeatable read khi đọc ở T1 | | | |
| **SP\_LOAD\_DSGV\_CUNG\_GVQL** | **Khóa** | **SP\_UPDATEGVQL** | **Khóa** |
| ***Input:***  *@MAGVQL*  ***Output***: @Return | ***Input***: @MAGV, @MAGVQL  ***Output***: |
| SET TRANSACTION ISOLATION  LEVEL **REPEATABLE READ** |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGVQL AND  ND.MALOAI = 2) -- 2: GVQL  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |  |  |
| B2: Hiển thị số lượng giáo viên (2) :  DECLARE @COUNT INT = (SELECT COUNT(\*) FROM  NGUOIDUNG WHERE MAGVQL = @MAGVQL)  PRINT 'CO ' + CAST(@COUNT AS VARCHAR(10)) + ' GV THUOC CUNG GVQL LA ' + CAST(@MAGVQL AS VARCHAR(10)) + ' :' | **R(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1):  IF NOT EXISTS (SELECT \* FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGVQL AND  ND.MALOAI = 2) -- 2: GVQL  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |
|  |  | B3: Update mã giáo viên quản lý (3):  UPDATE NGUOIDUNG  SET MAGVQL = @MAGVQL  WHERE MAND = @MAGV | **X(NGUOIDUNG)** |
|  |  | BEGIN CATCH  PRINT 'LOI LOAD GV'  ROLLBACK TRAN  END CATCH |  |
|  |  | COMMIT TRAN |  |
| B3: In danh sách giáo viên (3): |  |  |  |
| SELECT \* FROM NGUOIDUNG WHERE MAGVQL = @MAGVQL |  |  |  |
| BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| COMMIT TRAN |  |  |  |

### Sinh viên thực hiện: HUỲNH DUY THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR18: Lost update**  T1 Update quyền giáo viên  T2 Update quyền giáo viên  Cách khắc phục: đặt mức cô lập REPEATABLE READ ở T1, nếu đặt cả T2 sẽ xảy ra Deadlock | | | |
| **sp\_UpdateToanQuyenGV1** | **Khóa** | **sp\_UpdateToanQuyenGV2** | **Khóa** |
| ***Input:***  *@MaGV, @MaGVQL*  ***Output***: | ***Input***: @MAGV, @MAGVQL  ***Output***: |
| SET TRAN ISOLATION LEVEL REPEATABLE READ |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGV AND  ND.MAGVQL = @MaGVQL)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin (1)  IF NOT EXISTS(SELECT \*  FROM NGUOIDUNG ND  WHERE ND.MAND = @MAGV AND  ND.MAGVQL = @MaGVQL)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(NGUOIDUNG)** |
|  |  | B2: Update toàn quyền giáo viên (2):  DECLARE @VALUE BIT = (SELECT ND.TOANQUYENGV FROM NGUOIDUNG ND WHERE ND.MAND = @MaGV)  UPDATE NGUOIDUNG  SET TOANQUYENGV = 1 - @VALUE  WHERE MAND = @MaGV | **X(NGUOIDUNG)** |
|  |  | BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |
|  |  | COMMIT TRAN |  |
| B2: Update toàn quyền giáo viên (2):  DECLARE @VALUE BIT = (SELECT ND.TOANQUYENGV FROM NGUOIDUNG ND WHERE ND.MAND = @MaGV)  UPDATE NGUOIDUNG  SET TOANQUYENGV = 1 - @VALUE  WHERE MAND = @MaGV |  |  |  |
| BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| COMMIT TRAN |  |  |  |

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR21: *Dirty read***  T1 (User = giáo viên A): Thực hiện đổi thông tin Môn học có mã là “x” và đổi Tên môn học thành “X”  T2 (User = giáo viên B ): Thực hiện lấy Môn học có mã là “x” và Tên môn học là “X”  Cách khắc phục : sử dụng mức cô lập Read Commited Phát S khi đọc | | | |
| **sp\_CapNhatMonHoc\_Fix** | **Khóa** | **sp\_LayMonHoc\_Fix** | **Khóa** |
| ***Input:*** *@MAMH, @TENMH*  ***Output***: *@KETQUA* | ***Input:*** *@MAMH, @TENMH*  ***Output***: *@MAMH, @TENMH, @MABM* |
| BEGIN TRAN |  | SET TRAN ISOLATION LEVEL READ COMMITTED |  |
| BEGIN TRY  UPDATE MONHOC  SET @TENMH = @TENMH  WHERE MAMH = @MAMH | **X(MONHOC)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | BEGIN TRY  SELECT \* FROM MONHOC WHERE MAMH = @MAMH AND TENMH = @TENMH  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH | **R(MONHOC)**  //Không cần xin khoá |
|  |  | COMMIT |  |
|  |  | END TRAN |  |
| ROLLBACK TRAN  RETURN 0  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| END TRAN |  |  |  |

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR22: *Unrepeatable Read***  T1 (User = giáo viên A): thực hiện lấy thông tin Môn học có tên là “a” và có Mã bộ môn là “b”  T2 (User = giáo viên B ): thực hiện cập nhật thông tin của Môn học có tên là “a” và có Mã bộ môn là “b” đổi thành Mã bộ môn “c”  Cách khắc phục : sử dụng mức cô lập Repeatable Read Phát S khi đọc và giữ S đến khi transaction kết thúc  Cụ thể khắc phục ở chỗ : Do T1 phát và giữ S cho đến khi commit nên T1 luôn kết thúc trước T2. Vì vậy T1 sẽ ko bị lỗi | | | |
| **sp\_LayMonHoc\_Fix** | **Khóa** | **sp\_CapNhaMonHoc\_Fix** | **Khóa** |
| ***Input:*** *@MAMH, @MABM*  ***Output****: @Return* | ***Input:*** *@MAMH, @MABM*  ***Output****: @Return* |
| SET TRAN ISOLATION LEVEL REPEATABLE READ |  |  |  |
| BEGIN TRAN |  |  |  |
| B1: In ra số lượng Môn học  BEGIN TRY  DECLARE @COUNT INT = (SELECT COUNT(\*) FROM MONHOC WHERE MAMH = @MAMH AND MABM = @MABM)  PRINT 'CO ' + CAST(@COUNT AS VARCHAR(10)) + ' KET QUA.' | **R(MONHOC)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
| B3: Lấy danh sách các môn học thỏa điều kiện  SELECT \* FROM MONHOC WHERE MAMH = @MAMH AND MABM = @MABM  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH | **R(MONHOC)** |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin Mã bộ môn có tồn tại  BEGIN TRY  IF NOT EXISTS (SELECT \*  FROM BOMON  WHERE MABM = @MABM)  BEGIN  ROLLBACK TRAN  RETURN 0  END | **R(MONHOC)**  //Không cần xin khoá |
|  |  | B2: Cập nhật Môn học  UPDATE MONHOC  SET MABM = @MABM  WHERE MAMH = @MAMH  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH | **X(MONHOC)**  //Không cần xin khoá |
|  |  | COMMIT |  |

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR23: *Phantom***  T1 (User = giáo viên A): lấy danh sách môn học có Mã bộ môn là X (Trả về số lượng và Kết quả)  T2 (User = giáo viên B ): thực hiện thêm môn học có Mã bộ môn là X  Nguyên nhân trong quá trình thực hiện, T1 khôn thể ngăn chặn được lệnh Insert đến từ T2  Cách khắc phục : sử dụng múc cô lập Serialization cho T1 để phát S khi đọc và giữ S đến khi transaction kết thúc. Như vậy T1 sẽ kết thúc trước T2 | | | |
| **sp\_LayMonHoc\_Fix** | **Khóa** | **sp\_ThemMonHoc\_Fix** | **Khóa** |
| ***Input:*** *@MABM*  ***Output****: @Return* | ***Input****: @MaBM, @TenMH*  ***Output****: @Return* |
| SET TRAN ISOLATION LEVEL SERIALIZABLE | SET TRAN ISOLATION LEVEL READ COMMITTED |
| BEGIN TRAN |  |  |  |
| BEGIN TRY  --Đếm xem có bao nhiêu phần tử  DECLARE @COUNT INT = (SELECT COUNT(\*) FROM MONHOC WHERE MABM = @MABM)  PRINT 'CO ' + CAST(@COUNT AS VARCHAR(10)) + ' KET QUA.' | **R(MONHOC)** |  |  |
| WAITFOR DELAY '00:00:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | --Thực hiện thêm môn học mới  BEGIN TRY  INSERT INTO MONHOC(TENMH, MABM) VALUES (@TENMH, @MABM)  END TRY  BEGIN CATCH  ROLLBACK TRAN  END CATCH | **X(MONHOC)** |
|  |  | COMMIT |  |
| --Lấy danh sách các phần tử  SELECT \* FROM MONHOC WHERE MABM = @MABM  END TRY  BEGIN CATCH  ROLLBACK TRAN  RETURN 0  END CATCH |  |  |  |
| COMMIT |  |  |  |

**KẾT THÚC BÁO CÁO**