**d**

### Sinh viên thực hiện: NGUYỄN THÀNH PHONG

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR13: *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

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR12: *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

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR14: *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 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR13: *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 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR15: *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 |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR14: *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 |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR16: *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 |  |  |  |