TEST ĐỀ MẪU

Câu 1:

-- Cau 1.1

CREATE PROC sp\_KiemTra

@macd VARCHAR(20), @maxe VARCHAR(20)

AS

BEGIN

-- Nếu đã tồn tại xe và chuyến đi hợp lệ

IF(EXISTS(SELECT \* FROM CHUYENDI cd, XE x WHERE cd.MCD = @macd OR x.MAXE = @maxe))

BEGIN

-- Nếu trong ngày tương ứng không tồn tại chuyến đi nào cho xe đang xét

IF(NOT EXISTS(SELECT \* FROM CHYENDI cd, CD\_XE cdx, Xe x

WHERE x.MAXE = @maxe

AND x.MAXE = cdx.MAXE

AND cdx.MACD = cd.MACD

AND cd.MACD = @macd

AND DATE(cd.NGAYDI) = DATE(GETDATE())

AND MONTH(cd.NGAYDI) = MONTH(GETDATE())

AND YEAR(cd.NGAYDI) = YEAR(GETDATE())

))

BEGIN

INSERT INTO CD\_XE

VALUES(@macd, @maxe)

END

END

-- Nếu không tồn tại xe và chuyến đi hợp lệ

ELSE

BEGIN

PRINT N'Xe và chuyến đi phải hợp lệ!'

END

END

GO

-- Cau 1.2

CREATE TRIGGER tg\_Xe ON dbo.XE

FOR UPDATE

AS

BEGIN

DECLARE @bangSoXeCu VARCHAR(20), @bangSoXeMoi VARCHAR(20), @maXe VARCHAR(20)

SELECT @bangSoXeCu = BANGSO FROM deleted

SELECT @bangSoXeMoi = BANGSO FROM inserted

-- Kiểm tra bảng cũ với bảng mới

IF(@bangSoXeMoi != @bangSoXeCu)

BEGIN

IF(EXISTS(SELECT \* FROM XE x WHERE x.BANGSO = @bangSoXeMoi AND x.MAXE != @maXe))

BEGIN

ROLLBACK TRANSACTION

PRINT N'Biển số xe mới không được trùng với bất kì bảng số xe hiện có nào khác!'

END

END

ELSE

BEGIN

ROLLBACK TRANSACTION

PRINT N'Bảng xe cũ khác bảng xe mới!'

END

END

GO

Câu 2:

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | T2 | T3 | T4 |
| READ(A) |  |  |  |
|  |  | READ(B) |  |
|  |  |  | READ(A) |
|  | READ(C) |  |  |
|  |  | READ(A) |  |
| WRITE(A) |  |  |  |
|  | WRITE(C) |  |  |
|  |  |  | READ(C) |
| READ(C) |  |  |  |
|  |  | WRITE(B) |  |
|  |  |  | WRITE(C) |
| WRITE(C) |  |  |  |

Câu 2a: Các cập xung đột trong S1:

R4(A) -> W1(A)

R2(C) -> W4(C)

R2(C) ->W1(C)

W2(C) -> R1(C)

W2(C) -> W1(C)

W2(C) -> W4(C)

W2(C) -> R4(C)

R3(A) -> W1(A)

R4(C) -> W1(C)

W4(C) ->W1(C)

Câu 2b:

W1(A) = A+7=18

W2(C)=C-2=20

W3(C)=B+A = 6+18 =24

W4(C) = C-A = 20-18 =2

W1(C) = C-7 = 2-7 = -5

Câu 3:

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | T2 | T3 | T4 |
| READ(A) |  |  |  |
|  | READ(B) |  |  |
|  |  |  | READ(D) |
| WRITE(C) |  |  |  |
|  |  | READ(B) |  |
|  |  |  | READ(C) |
|  | WRITE(D) |  |  |
|  |  |  | WRITE(C) |
|  |  | READ(C) |  |
| READ(B) |  |  |  |

T1 T2

T3 T4

S1 là khả tuần tự xung đột, do S1 không có chu trình

Các lịch trình tương đương :

S1: T1-> T3->T4->T2

Câu 4:

T1 T2

T3 T4

S1 có chu trình nên S1 có deadlock

Wait –die

T1,T2 wait

T4 die

T3 chạy bình thường

Wound – wait

T3 die

T4 wait

T1,T2 chạy bình thường