

Giao tác (Transaction)

Một giao tác là một đơn vị xử lý nguyên tử gồm nhiều hành động. Khi thực hiện một giao tác hoặc phải thực hiện tất cả các hành động của nó hoặc thì không thực hiện hành động nào hết.

Giao tác trong các loại CSDL quan hệ lớn được sử dụng trong những trường hợp mà các hành động cập nhật dữ liệu trên nhiều bảng khác nhau được thực hiện trong cùng một đơn vị (unit).

29

Giao tác (Transaction)

- Các tính chất
 - Tính nguyên tử (**A**tomicity)
 - Tính nhất quán (**C**onsistency)
 - Tính độc lập (**I**solation)
 - Tính bền vững (**D**urability)

30

Giao tác không tường minh

- Mặc định các lệnh bên trong lô (batch) chứa các câu lệnh không tường minh, điều này có nghĩa là nếu có ít nhất 1 câu lệnh thực hiện không thành công trong lô thì tất cả các lệnh còn lại sẽ không được ghi nhận lại.

31

Giao tác không tường minh (tt)

- Ví dụ: chúng ta cho thực hiện cùng lúc 3 lệnh để cập nhật dữ liệu vào 3 bảng khác nhau trong cùng một lô.

--**Thêm mặt hàng mới**

```
INSERT INTO MAT_HANG(MaMH, TenMH, DVT)
VALUES('D001', 'đèn ngủ', 'cái' )
```

--**Sửa đổi tên nhà cung cấp 'NCC01'**

```
UPDATE NHACC
SET TenNCC = 'Le Khai Hoan'
Where      MaNCC = 'NCC01'
```

--**Xoá đơn đặt hàng 'DH001'**

```
DELETE HOA_DONDH
Where      MaDH = 'DH001'
```

GO

Vì phạm toàn vẹn DL về khóa ngoại. Nên các lệnh trước đó không thực hiện

32

Giao tác tường minh

- Giao tác tường minh trong những trường hợp cập nhật dữ liệu trên nhiều bảng khác nhau và phải đảm bảo các hành động này nằm trong cùng một đơn vị xử lý.

33

[*Giao tác tường minh (1)*

■ Bắt đầu một giao tác

- BEGIN TRAN[SACTION]
[Tên_giao_tác]

■ kết thúc một giao tác

- ROLLBACK TRAN[SACTION]
[Tên_giao_tác]

Hoặc

- COMMIT TRAN[SACTION]
[Tên_giao_tác]

kết thúc giao tác nhưng không ghi nhận lại các hành động cập nhật dữ liệu bên ngoài giao tác.

kết thúc giao tác nhưng đồng ý ghi nhận lại các hành động cập nhật dữ liệu bên trong giao tác.

34

Giao tác tường minh (tt)

- ví dụ 2:
begin tran cap1
 insert into #test values(1,'aaa')
 begin tran cap2
 insert into #test
values(2,'bbb')
 begin tran cap3
 insert into #test
values(3,'ccc')
 commit tran cap3
 go
 commit tran cap2
go
rollback tran cap1

35

Phân vùng trong giao tác

- Ta chia nhỏ các hành động bên trong giao tác thành nhiều phần, tương ứng từng phần nhỏ ta có thể dễ dàng chủ động đồng ý ghi nhận hoặc không ghi nhận lại việc cập nhật dữ liệu.
- Cú pháp
 SAVE TRAN[SACTION] [Tên vùng]
 Các lệnh

36

[*Phân vùng trong giao tác (tt)*]

- Ví dụ:

Begin tran

--Vùng 1

SAVE TRAN Vung_1_2

insert into #test values(1,'aaa')

insert into #test values(2,'bbb')

--Vùng 2

SAVE TRAN Vung_3

insert into #test values(3,'ccc')

RollBack Tran Vung_3

Commit Tran Vung_1_2

chỉ có mẫu
tin thứ 1 và
thứ 2 được
ghi nhận lại

37

TRIGGER

- Khái niệm: **Trigger** là một dạng đặc biệt của thủ tục nội tại. Tuy nhiên khác với thủ tục nội tại:
 - Không có tham số.
 - Không thể gọi trực tiếp bằng lệnh **EXECUTE** như thủ tục nội tại mà thực hiện một cách tự động khi dữ liệu của bảng có liên quan đến trigger bị cập nhật.

38

Trigger dùng cho các công việc sau

- Kiểm tra ràng buộc toàn vẹn dữ liệu phức tạp.
- Thực hiện các xử lý thiết kế thi hành tại Server (trong mô hình Client/Server). Các xử lý sẽ tự động thực hiện khi có thao tác INSERT, UPDATE hoặc DELETE xảy ra.
- Trigger dùng thay thế các constraint trong trường hợp ta muốn việc kiểm tra ràng buộc dữ liệu kèm theo các câu thông báo thích hợp theo ý muốn người dùng

39

Ràng buộc dữ liệu toàn vẹn với Trigger

- Để đảm bảo dữ liệu nhất quán và đúng đắn, ta cần kiểm tra thực hiện 3 thao tác: Insert, Update và Delete.
- Có 2 cách kiểm tra
 - Kiểm tra mức giao diện: là công việc lập trình trên các màn hình giao diện
 - Kiểm tra mức CSDL: thực hiện bởi các đối tượng constraint hoặc trigger

40

Ràng buộc dữ liệu toàn vẹn với trigger (tt)

- Đối tượng Constraint kiểm tra
 - Kiểm tra miền giá trị
 - Kiểm tra các ràng buộc giữa các thuộc tính trên cùng một bảng dữ liệu
- Đối tượng Trigger
 - Kiểm tra tính toàn vẹn dữ liệu trên nhiều cột hoặc nhiều dòng của các bảng khác nhau

41

Các dạng ràng buộc toàn vẹn

- RBTV bằng phương pháp mô tả
 - Xác định khoá chính, khoá ngoại, miền giá trị,... và mô tả chúng tại thời điểm tạo Table
 - Thực hiện trước khi cho phép thêm vào Table.
- RBTV theo phương pháp thủ tục
 - Xác định bởi tập các câu lệnh T-SQL. Các lệnh chứa bên trong đối tượng trigger.
 - Được gọi thi hành khi có thao tác Thêm, xoá hoặc sửa dữ liệu trên table tương ứng
 - Thực hiện sau khi dữ liệu được ghi vào Table.

42

BÀI TOÁN

KHACH_HANG(MaKH, TenKhach)

PHIEU_XUAT(MAPX, Ngay_PX, MaKH)

CT_PHIEU_XUAT(MAPX, MaHH, SoLuong, DonGia)

HANG_HOA(MAHH, Ten_HH, DonGiaHienHanH)

- **KHACH_HANG**: một khách hàng có 1 mã duy nhất để phân biệt khách hàng này với khách hàng khác, có 1 tên khách hàng duy nhất
- **HANG_HOA**: một hàng hoá có 1 mã hàng duy nhất dùng để phân biệt với hàng hoá khác, có 1 tên hàng hoá và một đơn vị bán hiện tại duy nhất

43

[BÀI TOÁN (TT)

RB nào bắt buộc phải sử dụng trigger

- **PHIEU_XUAT**: một phiếu xuất có 1 mã duy nhất dùng để phân biệt với phiếu khác, có 1 ngày xuất xác định.
 - Một phiếu xuất liên quan đến 1 khách hàng duy nhất
 - Một phiếu xuất có ít nhất 1 chi tiết xuất
- **CT_PHIEU_XUAT**: một chi tiết phiếu xuất có mã phiếu xuất và mã hàng hoá dùng để xác định khoá của quan hệ. Khoá này dùng để phân biệt với chi tiết xuất khác, có 1 số lượng xác định, 1 đơn giá tương ứng với đơn giá hiện hành của hàng hoá lấy từ thuộc tính DonGiaHienHanH của bảng **HANG_HOA**.

44

Cơ chế hoạt động của Trigger

- 3 biến cố kích hoạt 1 trigger
 - INSERT
 - UPDATE
 - DELETE
- Trigger lưu trữ dữ liệu của mẫu tin vừa thêm vào một table mới có tên là INSERTED.
- Trigger lưu trữ dữ liệu của mẫu tin vừa xóa vào một table có tên là DELETED.
- Trigger lưu trữ dữ liệu của mẫu tin vừa cập nhật là sự phối hợp của 2 table DELETED và INSERTED

45

[Tạo mới trigger]

- Cú pháp:
CREATE TRIGGER Tên_trigger
ON tên_table|tên_view
AFTER | INSTEAD OF
biến_cố_kích_hoạt_trigger
AS
-- Các câu lệnh T-SQL

Có thể thay bằng FOR. After là mặc định, chỉ định nghĩa duy nhất được trên view

46

Ví dụ:

```
CREATE TRIGGER Them_HH  
ON  HANG_HOA  
AFTER  INSERT  
AS
```

```
    Select * From Inserted
```

■ Chèn dữ liệu

```
INSERT  
    HANG_HOA(MaHH, TenHH)  
VALUES('TV01', 'Tivi Sony')
```

47

```
CREATE TRIGGER SUA_HH  
ON  HANG_HOA  
AFTER  UPDATE  
AS
```

```
    Select * From Inserted
```

```
    Select * From Deleted
```

■ Cập nhật dữ liệu

```
UPDATE HANG_HOA
```

```
SET      Ten_HH =  
'Man Hinh Sony'
```

```
WHERE      MaHH =  
'TV01'
```

48

```
CREATE TRIGGER Xoa_HH
ON  HANG_HOA
AFTER  DELETE
AS
```

```
    Select * From Inserted
```

```
    Select * From Deleted
```

■ Xóa dữ liệu

```
    DELETE      HANG_HOA
```

```
    WHERE      MaHH
               = 'TV01'
```

49

Các thao tác Trigger phổ biến

Thêm mới mẫu tin

- Kiểm tra ràng buộc dữ liệu
- Khoá ngoại
- Miền giá trị
- Liên bộ trên một quan hệ
- Liên thuộc tính trong cùng một bảng
- Liên thuộc tính của nhiều bảng khác nhau

Ví dụ

- HOADON_DH(MaHD, NgayDH, MaKH)
- PHIEU_XUAT(MaPX, NgayXuat, #MaHD)
- CHITIET_DH(MAHD, MaHH, SoLuong, DonGia)

50

Thêm mới mẫu tin

- Xây dựng Trigger trong bảng PHIEU_XUAT để kiểm tra các ràng buộc toàn vẹn dữ liệu khi người dùng thêm mới thông tin của một phiếu xuất hàng cho một bảng hoá đơn đặt hàng trước đó. Các ràng buộc toàn vẹn dữ liệu bao gồm
 - Khoá ngoại: cần kiểm tra số đặt hàng phải tồn tại trong bảng đơn đặt hàng.
 - Miền giá trị: cần kiểm tra ngày giao hàng phải ở sau ngày đặt hàng.

51

Thêm mới mẫu tin (tt)

```
CREATE TRIGGER      tg_PhieuXuat_Insert
ON PHIEU_XUAT
FOR INSERT
AS
    DECLARE @NgayHD datetime, @ErrMsg
    varchar(200)
-- Kiểm tra số hoá đơn đã có trong bảng DONDH
không?
IF NOT EXISTS(Select *
               From Inserted I, HOADON_DH D
               Where I.MaHD= D.MaHD)
Begin
    Rollback Tran
    Raiserror('Số đơn đặt hàng không tồn tại',
    16,1)
    Return
End
```

52

Thêm mới mẫu tin (tt)

```
--Tính ra ngày đặt hàng
Select @NgayDH=NgayDH
From HoaDon_DH D, Inserted I
Where D.MaHD = I.MaHD
-- Kiểm tra ngày giao hàng phải sau ngày đặt hàng
IF @NgayDH > (Select ngayxuat From Inserted)
Begin
    Set @ErrMsg = 'ngày giao hàng phải ở sau ngày:'
    + Convert(char(10), ngayDH, 103 )
    Raierror(@ErrMsg,16,1)
    Rollback tran
End
```

53

Huỷ bỏ mẫu tin

- Ví dụ: khi xoá một số hoá đơn đặt hàng trong bảng HOADON_DH cần phải kiểm tra các RBTV dữ liệu sau:
 - Kiểm tra xem đơn đặt hàng bị xoá đã được xuất hàng chưa? Nếu đã được xuất rồi thì thông báo không thể xoá đơn đặt hàng được.
 - Ngược lại thì xoá dữ liệu liên quan bên bảng chi tiết đơn đặt hàng (CHITIET_HD)

54

```

CREATE TRIGGER tg_HOADON_Delete
ON HOADON_DH
FORDELETE
AS
    DECLARE @SoPX char(5), @ErrMsg
             char(200), @Delete_Err int
-- Kiểm tra xem đơn hàng đã được xuất chưa
    IF EXISTS(Select MaPX From PHIEU_XUAT
Where MaHD IN(Select MaHD From Deleted))
    Begin
        Select @MaPX = MaPX From PHIEU_XUAT
        Where MaHD In(Select MaHD From Deleted)
        Set @ErrMsg = 'Đơn đặt hàng đã được nhập theo '+'
                    'số xuất hàng '+' @SoPX + char(13) + '.Không thể
                    huỷ được'
        RaiseError(@ErrMsg,16,1)
        Rollback tran
    End

```

55

```

Else
Begin
-- Xoá tự động chi tiết các đơn đặt hàng liên quan
Delete CHITIET_DH
Where MaHD In(Select MaHD From DELETED)
Set @Delete_Err = @@ERROR
IF @Delete_Err <> 0
Begin
    Set @ErrMsg = 'Lỗi vi phạm xóa trên bảng chi tiết đặt
    hàng'
    RaiseError(@ErrMsg, 16, 1)
    Rollback Tran
End
End

```

56

Sửa đổi mẫu tin

- Kiểm tra ràng buộc dữ liệu
 - Kiểm tra ràng buộc dữ liệu
 - Khoá ngoại
 - Miền giá trị
 - Liên bộ trên một quan hệ
 - Liên thuộc tính trong cùng một bảng
 - Liên thuộc tính của nhiều bảng khác nhau

57

- Hàm Update
 - Ý nghĩa
 - kiểm tra dữ liệu của cột bên trong bảng có bị thay đổi trong các trigger sửa đổi dữ liệu
 - Cú pháp
 - UPDATE (tên_cột) (biểu thức luận lý)
 - Tên_cột: tên cột mà chúng ta muốn kiểm tra xem dữ liệu tại đó có bị sửa đổi trong trigger không.
 - Biểu thức luận lý: trả về True khi giá trị dữ liệu của cột đã bị sửa đổi, ngược lại trả về False khi giá trị dữ liệu của cột không bị sửa đổi

58

- Ví dụ: sửa đổi thông tin của một số đặt hàng bên trong bảng HOADON_DH cần phải kiểm tra các ràng buộc toàn vẹn dữ liệu sau:
 - Không cho phép sửa đổi dữ liệu tại cột MaDH hoặc MaKH vì khi đó dữ liệu sẽ ảnh hưởng đến nhiều bảng.
 - Sửa đổi giá trị cột ngày đặt hàng thì phải đảm bảo luôn luôn trước ngày giao hàng đầu tiên của số đặt hàng đó (nếu đơn đặt hàng đã có giao hàng).

59

```
CREATE TRIGGER
    tg_HOADON_DH_Update ON DONDH
FOR UPDATE
AS    Declare @MinNgayXH date, @ErrMsg
      varchar(200)
-- Khi sửa đổi các cột MaDH hoặc MaKH
IF Update(MaDH) OR Update(MaKH)
Begin
    Rollback Tran
    Set @ErrMsg = 'Không thể thay đổi số
đặt hàng hoặc
mã khách hàng'
    RaiseError(@ErrMsg, 16, 1)
    Return
End
```

60


```

-- Khi sửa đổi ngày đặt hàng
IF Update(NgayDH)
Begin
-- Kiểm tra đơn đặt hàng đã được xuất chưa
IF EXISTS (Select MaPX From
PHIEU_XUAT PX, deleted d
where px.mapx=d.mapx)

Begin
-- Tính ra ngày xuất hàng đầu tiên
Select @MinNgayXH = Min(NgayXuat)
From PHIEU_XUAT PX, DELETED D
Where PX.MaDH = D.MaDH

```

61

```

--kiểm tra giá trị ngày đặt hàng sau khi sửa đổi
--phải luôn trước ngày giao hàng đầu tiên
IF @MinNgayXH < (Select NgayDH From
Inserted)
Begin
Rollback tran
Set @ErrMsg = 'Ngày đặt hàng phải ở
trước ngày:' +
Convert(char(10),@MinNgayXH, 103)
RaiseError(@ErrMsg, 16, 1)
End
End
End

```

62

Trigger cập nhật giá trị tự động

- Ví dụ: dữ liệu của bảng TONKHO sẽ được tính tự động từ dữ liệu của các bảng liên quan đến việc nhập hàng và việc xuất hàng, cụ thể sẽ là các bảng: PNHAP, CTPNHAP, PXUAT, và CTPXUAT
- Khi thêm mới các thông tin của chi tiết một phiếu nhập hàng vào bảng CTNHAP, chúng ta cần kiểm tra các RBTV dữ liệu
 - Kiểm tra số phiếu nhập phải có trong bảng PNHAP.
 - Kiểm tra mã vật tư phải có trong danh sách chi tiết, danh sách các mã vật tư phải có trong chi tiết đơn đặt hàng trước đó.
 - Kiểm tra tổng số lượng nhập hàng vẫn còn ít hơn số lượng đặt hàng của vật của vật tư đó.
- Nếu tất cả các RBTV dữ liệu ở trên đều hợp lệ thì tăng giá trị của cột tổng số lượng nhập trong bảng TONKHO và cột tổng trị giá trong bảng PNHAP.

63

HÀM DO NGƯỜI DÙNG ĐỊNH NGHĨA (USER DEFINED FUNCTIONS)

- **Khái quát:** UDFs là một chương trình con đảm trách một xử lý nào đó với đặc tính là sẽ nhận các tham số đầu vào và trả về một giá trị kết quả xử lý tính toán được dựa trên các tham số đầu vào đã nhận.
- UDFs phân thành 2 nhóm:
 - Hàm xác định (deterministic)
 - Hàm không xác định (non-deterministic).

64

HÀM DO NGƯỜI DÙNG ĐỊNH NGHĨA (tt)

- Hàm xác định luôn trả về cùng giá trị nếu giá trị các tham số được truyền vào là như nhau.
- Hàm không xác định có thể cho ra kết quả khác biệt tại mỗi thời điểm chúng được gọi

65

HÀM DO NGƯỜI DÙNG ĐỊNH NGHĨA (tt)

UDFs là sự kết hợp của 2 đối tượng View và Store Procedure

Khắc phục một số hạn chế của View và Store Procedure

- Store Procedure không thể là một phần của câu lệnh SELECT nhưng UDFs thì có
- View không hỗ trợ đệ quy trong khi UDFs thì có thể làm được điều này.

66

HÀM DO NGƯỜI DÙNG ĐỊNH NGHĨA (tt)

- Tạo UDF
`CREATE FUNCTION`
 [Tên_Function](tham số)
`RETURNS`
 kiểu_dữ_liệu_trả_về
`AS`
`BEGIN`
 --Các lệnh
 Return
`END`

67

HÀM DO NGƯỜI DÙNG ĐỊNH NGHĨA (tt)

- Returns: thiết lập kiểu dữ liệu trả về của UDFs. Có 2 cách thiết lập chính
 - Trả về giá trị kiểu vô hướng: một chuỗi, một giá trị logic hoặc một kiểu số.
 - Trả về một bảng: có thể trả về hai loại bảng
 - Inline table: khắc phục được nhược điểm không có tham số của VIEW. Có nghĩa rằng UDFs loại inline table giống như một VIEW có tham số.
 - Multistatement table: UDFs loại này giống với Store Procedure. Loại này luôn trả về 1 biến table. Thực hiện các câu SELECT phức tạp, cho phép thực hiện các câu lệnh logic khác như UPDATE, INSERT INTO...

68

[Ví dụ]

- Ví dụ 1: Tạo View như sau

CREATE VIEW DS_SinhVien

RETURNS chứa duy nhất từ khoá table

AS

Select * From SINHVIEN Where MaLop = 'C5CT09'

CREATE FUNCTION DS_SinhVien(@MaLop char(6))

RETURNS Table AS

Return (Select * From SinhVien Where MaLop=@MaLop)

Chứa một câu SELECT đơn giản nằm trong cặp dấu ngoặc đơn.

69

Ví dụ (tt)

```
CREATE FUNCTION F_DSHangHoa(@LoaiHang
varchar(50), @PhanTram numeric )
RETURNS @DSHangHoa Table
( MaMH char(10),
  TenMH varchar(50),
  DonGiaKhuyenMai numeric)
AS
Begin
  INSERT INTO @DSHangHoa(MaMH, TenMH,
DonGiaKhuyenMai)
  Select MaMH, TenMH, DonGiaHienHanh
  From HANGHOA
  Where loaihang = @LoaiHang

  Update @DSHangHoa
  Set DonGiaKhuyenMai= DonGiaKhuyenMai-
(DonGiaKhuyenMai
* @Phantram)/100
  Return
End
```

70

```

CREATE FUNCTION
    DonGiaHienHanh(@MaMH
        varchar(50))
RETURNS numeric
AS
BEGIN
    Return(Select DonGiaHienHanh
        From  HANGHOA
        WHERE      MaMH =
            @MaMH)
END

```

71

```

CREATE FUNCTION Test_function(@b int, @c int)
RETURNS int
as
BEGIN
    declare @kq int
    if @b>2
        set @kq=@b+@c
    else
        set @kq=@b+@c+1
    return @kq
END

```

72