



TRANSACTION

(GIAO DỊCH)

Khái niệm giao dịch (TRANSACTION)

Transaction là một gói lệnh dùng đảm bảo các lệnh cập nhật dữ liệu trong gói đều được xử lý một cách trọn vẹn. Nếu có một lệnh nào đó trong **Transaction** bị lỗi thì sẽ tự động bãi bỏ các lệnh trong **Transaction** và phục hồi lại toàn bộ dữ liệu đã bị thay đổi bởi các lệnh trong **Transaction**.

- ❑ CSDL có thể không toàn vẹn trong khi thực hiện giao dịch nhưng phải ở trong trạng thái toàn vẹn trước và sau khi thực hiện.
- ❑ Quản trị giao dịch phải đảm bảo CSDL vẫn được lưu trữ ở trạng thái toàn vẹn nếu có lỗi nào đó xảy ra trong lúc thực hiện giao dịch.
- ❑ SQL Server sử dụng nhật ký giao dịch (transaction log) trong mỗi database để khôi phục lại khi có sự cố.

Ví dụ về giao dịch (TRANSACTION)

- ✓ Ví dụ: Giả sử số dư hiện tại của tài khoản A là 30 triệu, tài khoản B là 10 triệu, tài khoản A chuyển 5 triệu cho tài khoản B. Các thao tác gồm:
 1. Read(A) (Tài khoản A có 30 triệu)
 2. $A = A - 5$ (Trừ tài khoản A 5 triệu (để chuyển cho TK B))
 3. Write(A) (Cập nhật lại số dư tài khoản A (25 triệu))
 4. Read(B) (Tài khoản B có 10 triệu)
 5. $B = B + 5$ (Cộng số dư tài khoản B 5 triệu)
 6. Write(B) (Cập nhật số dư tài khoản B (15 triệu))
- ✓ Nếu tất cả các thao tác trên được thực hiện thành công (số dư còn lại của tài khoản A là 25 triệu và tài khoản B hiện tại là 15 triệu) lúc đó giao dịch đã thực hiện thành công.

Ví dụ về giao dịch (TRANSACTION)

- ✓ Nếu có lỗi xảy ra thì cần phải hủy tất cả các thao tác và phục hồi CSDL trở về trạng thái ban đầu.

1. Read(A) (Tài khoản A có 30 triệu)
2. $A = A - 5$ (Trừ tài khoản A 5 triệu (để chuyển cho TK B))
3. Write(A) (Cập nhật lại số dư tài khoản A (25 triệu))

4. Read(B) (Tài khoản B có 10 triệu)
5. $B = B + 5$ (Cộng số dư tài khoản B 5 triệu)
6. Write(B) (Cập nhật số dư tài khoản B (15 triệu))

Bị lỗi

Một số khái niệm trong giao dịch cần biết

- **BEGIN TRAN[SACTION]** [*transaction_name*]: bắt đầu giao dịch
- **SAVE TRAN[SACTION]** [*savetransaction_name*]: Thiết lập savepoint, đánh dấu một vị trí trong giao dịch (savepoint) để Rollback đến khi có lỗi thay vì Rollback về đầu giao dịch (hủy một phần của giao dịch).
- **ROLLBACK TRAN[SACTION]** [*transaction_name|savetransaction_name*]:
Hủy bỏ giao dịch
- **COMMIT TRAN[SACTION]** [*transaction_name*]: giao dịch thực hiện thành công, chuyển giao kết quả đến database (không có lỗi xảy ra).

Các tính chất cơ bản của một giao dịch

- ❑ **Tính nguyên tử (Atomicity):** Hoặc tất cả các thao tác trong một giao dịch phải được thực hiện hoặc tất cả các thao tác này phải được huỷ bỏ.
- ❑ **Tính nhất quán (Consistency):** CSDL phải ở trong trạng thái toàn vẹn trước và sau khi thực hiện giao dịch (nhưng có thể không toàn vẹn trong lúc thực hiện giao dịch).
- ❑ **Tính cô lập (Isolation):** Những chỉnh sửa được làm bởi transaction hiện hành phải được cô lập khỏi những chỉnh sửa được làm bởi các transaction hiện hành khác.
- ❑ **Tính bền vững (Durability):** Một khi giao dịch đã thực hiện thành công, những thay đổi mà giao dịch đã thực hiện trên hệ thống là bền vững (ổn định lâu dài trong hệ thống).

Các loại (kiểu) giao dịch

- ❑ **Autocommit transactions (giao dịch tự động - mặc định):** Luôn luôn bắt đầu một giao dịch đối với lệnh thay đổi dữ liệu, tự động **commit** hoặc **rollback** giao dịch khi lệnh thành công hoặc lỗi.
- ❑ **Explicit transactions (giao dịch tường minh):** Khai báo tường minh điểm bắt đầu cũng như **Commit** hoặc **Rollback**. Giao dịch tường minh vượt qua Autocommit transactions.
- ❑ **Implicit transactions (giao dịch ẩn):** Bắt đầu giao dịch tự động nhưng người lập trình phải chỉ ra điểm kết thúc của giao dịch cũng như chấp nhận hay từ chối kết quả.

Giao dịch tự động - Autocommit transactions

❑ Giao dịch tự động:

- Là kiểu quản lý giao dịch mặc định (**Default**) trong SQL Server
- Nếu xảy ra lỗi : **Rollback** (hủy kết quả, trả CSDL về tình trạng ban đầu)
- Tự động **Commit** hoặc **Rollback** một lệnh T-SQL khi hoàn tất
- Nếu không xảy ra lỗi: **Commit** (ghi nhận thay đổi đến CSDL)

Giao dịch tự động - Autocommit transactions

✓ Ví dụ về giao dịch tự động:

```
SQLQuery1.sql - B...O-PC.VD (sa (57))*
Use VD
Go
Create Table #NhanVien
(
    MaNV char(5) not null Primary Key,
    TenNV nvarchar(30) not null
)
Go
Insert Into #NhanVien (MaNV,TenNV) values ('01',N'Đức An')
Insert Into #NhanVien (MaNV,TenNV) values ('02',N'Thảo Hân')
Insert Into #NhanVien (MaNV,TenNV) values ('02',N'Gia Tuệ')
Insert Into #NhanVien (MaNV,TenNV) values ('03',N'Lê Minh')
Go
Select * From #NhanVien
```

	MaNV	TenNV
1	01	Đức An
2	02	Thảo Hân
3	03	Lê Minh

Msg 2627, Level 14, State 1, Line 3
Violation of PRIMARY KEY constraint

Giao dịch tường minh - Explicit transactions

❑ Giao dịch phải được khai báo tường minh (Explicit transaction)

BEGIN TRAN[SACTION] [transaction_name]

Các lệnh trong transaction

[SAVE TRAN[SACTION] [savetransaction_name]]

Các lệnh trong transaction

.....

ROLLBACK TRAN[SACTION] [transaction_name | savetransaction_name]

(Quay lui nếu giao dịch gặp lỗi)

COMMIT TRAN[SACTION] [transaction_name]

(Chuyển giao (commit) kết quả đến Database nếu giao dịch thực hiện thành công)

Giao dịch tường minh - Explicit transactions

❑ Trong đó:

- Để đánh dấu điểm bắt đầu một giao dịch:

`BEGIN TRANS[ACTION] [transaction_name]`

- Đánh dấu một vị trí trong giao dịch (savepoint) để Rollback đến khi có lỗi thay vì Rollback về đầu giao dịch (hủy một phần của giao dịch).

`SAVE TRAN[SACTION] < savetransaction_name >`

- Khi xảy ra lỗi, để hủy bỏ giao dịch.

`ROLLBACK TRAN[SACTION] [transaction_name | savetransaction_name]`

- Khi giao dịch thực hiện thành công (không có lỗi xảy ra), để chuyển giao kết quả đến database

`COMMIT TRANS[ACTION] [transaction_name]`

Giao dịch tường minh - Explicit transactions

✓ Ví dụ về giao dịch tường minh:

BEGIN TRANSACTION TranStart

INSERT INTO Sinhvien (MaNV, TenNV) VALUES ('01', N'Mỹ Dung')

→ **SAVE TRAN** FirstPoint

INSERT INTO Sinhvien (MaNV, TenNV) VALUES ('02', N'Diễm Hương')

INSERT INTO Sinhvien (MaNV, TenNV) VALUES ('03', N'Minh Tài')

→ **ROLLBACK TRAN** FirstPoint

INSERT INTO Sinhvien (MaNV, TenNV) VALUES ('04', N'Nam Dương')

→ **SAVE TRAN** SecondPoint

INSERT INTO Sinhvien (MaNV, TenNV) VALUES ('05', N'Quốc Hùng')

→ **ROLLBACK TRAN** SecondPoint

INSERT INTO Sinhvien (MaNV, TenNV) VALUES ('06', N'Mỹ Hậu')

COMMIT TRAN TranStart

Giao dịch ẩn - Implicit transactions

- ❑ Bất kỳ Transaction nào mà bắt đầu bằng lệnh **Begin Transaction** và kết thúc bằng lệnh **Commit Transaction** hoặc **Rollback Transaction** đều là **Explicit Transaction** (giao dịch tường minh).

- ❑ **Giao dịch ẩn:**

Bắt đầu một giao dịch ẩn như giao dịch tự động nhưng kết thúc giao dịch ẩn phải có lệnh **Commit** hay **Rollback**.

- ❑ Khởi động/ tắt chế độ giao dịch ẩn:

SET IMPLICIT_TRANSACTIONS ON | OFF

(Mặc định là **OFF**)

- ❑ Khi đặt **Implicit_Transactions** là **ON**, thì ở cuối các **Tracsaction** cần phải có lệnh **Commit** hoặc **RollBack**. Nếu không thì các lệnh trong Transaction và tất cả dữ liệu đã thay đổi **sẽ bị bỏ qua** khi người dùng kết thúc kết nối.

Giao dịch ẩn - Implicit transactions

- ❑ **Lưu ý:** Khi `Implicit_Transactions` là `On` nếu không ở trong một Transaction nào, các phát biểu sau đây sẽ tự động khởi đầu một Transaction mới.

ALTER TABLE

FETCH

REVOKE

CREATE

GRANT

SELECT

DELETE

INSERT

TRUNCATE TABLE

DROP

OPEN

UPDATE

Xử lý lỗi (Biến hệ thống @@error)

- ❑ Biến @@error chứa mã lỗi cho phát biểu vừa thực hiện (phát biểu T-SQL cuối cùng). Nó tự động xóa và đặt lại mã lỗi với mỗi phát biểu được thực hiện.
 - Nếu không có lỗi biến hệ thống @@error có giá trị bằng 0
 - Nếu có lỗi biến hệ thống @@error có giá trị khác 0

- ❑ Thường được sử dụng:

Phát biểu T-SQL (Thêm, sửa, xóa)

IF @@error != 0 (Khi có lỗi xảy ra)

Lệnh xử lý lỗi

Xử lý lỗi (Biến hệ thống @@error)

✓ Ví dụ:

```
BEGIN TRAN
```

```
    UPDATE HANGHOA SET SLTon += 50
```

```
    IF (@@ERROR != 0 )
```

```
        BEGIN
```

```
            RAISERROR (N'Lỗi khi cập nhật SLT',16,1)
```

```
            ROLLBACK TRAN
```

```
        END
```

```
    INSERT INTO CTPN (SOPN,MAMH, SLNHAP) VALUES ('01','X',50)
```

```
    IF (@@ERROR != 0 )
```

```
        BEGIN
```

```
            RAISERROR (N' Lỗi khi thêm CTPN',16,1)
```

```
            ROLLBACK TRAN
```

```
        END
```

```
COMMIT TRAN
```


Xử lý lỗi (TRY ... CATCH)

✓ Cú pháp:

BEGIN TRY

Các lệnh có khả năng phát sinh lỗi

END TRY

BEGIN CATCH

Lệnh xử lý nếu lỗi phát sinh

END CATCH

Xử lý lỗi (TRY ... CATCH)

✓ Ví dụ:

BEGIN TRY

BEGIN TRAN

UPDATE HANGHOA SET SLTon += 50

INSERT INTO CTPN (SOPN, MAMH, SLNHAP)

VALUES ('01','X',50)

COMMIT TRAN

END TRY

BEGIN CATCH

RAISERROR(N'Có lỗi! Không thể thêm CTPN',16,1)

ROLLBACK TRAN

END CATCH

Lệnh SET XACT_ABORT ON|OFF

SET XACT_ABORT ON|OFF

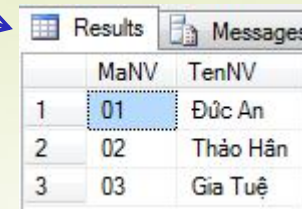
- ❑ **OFF: (mặc định)** Chỉ hủy bỏ lệnh gây lỗi, tiếp tục thực hiện lệnh tiếp theo (thường dùng khi lỗi không nghiêm trọng)
- ❑ **ON:** Hủy bỏ tất cả lệnh trong transaction nếu một câu lệnh Transact-SQL nào đó phát sinh lỗi run-time (thường dùng khi lỗi là nghiêm trọng)

Lệnh SET XACT_ABORT ON|OFF

✓ Ví dụ: Chỉ bỏ qua dòng lệnh bị lỗi

```
SQLQuery1.sql - B...O-PC.VD (sa (53))*
Use VD
Go
Create Table #NhanVien
(
    MaNV char(5) not null Primary Key,
    TenNV nvarchar(30) not null
)
Go
Set xact_abort off
Begin Tran
    Insert Into #NhanVien (MaNV,TenNV) values ('01',N'Đức An')
    Insert Into #NhanVien (MaNV,TenNV) values ('02',N'Thảo Hân')
    Insert Into #NhanVien (MaNV,TenNV) values ('03',N'Gia Tuệ')
    Insert Into #NhanVien (MaNV,TenNV) values ('03',N'Lê Minh')
Commit
Go
Select * From #NhanVien
```

Violation of
PRIMARY KEY
constraint



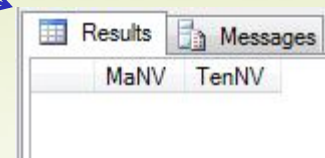
	MaNV	TenNV
1	01	Đức An
2	02	Thảo Hân
3	03	Gia Tuệ

Lệnh SET XACT_ABORT ON|OFF

✓ Ví dụ: khi có lỗi, hủy bỏ tất cả các lệnh trong giao dịch

```
SQLQuery1.sql - B...O-PC.VD (sa (53))*
Use VD
Go
Create Table #NhanVien
(
    MaNV char(5) not null Primary Key,
    TenNV nvarchar(30) not null
)
Go
Set xact_abort on
Begin Tran
    Insert Into #NhanVien (MaNV,TenNV) values ('01',N'Đức An')
    Insert Into #NhanVien (MaNV,TenNV) values ('02',N'Thảo Hân')
    Insert Into #NhanVien (MaNV,TenNV) values ('03',N'Gia Tuệ')
    Insert Into #NhanVien (MaNV,TenNV) values ('03',N'Lê Minh')
Commit
Go
Select * From #NhanVien
```

Violation of
PRIMARY KEY
constraint



MaNV	TenNV
------	-------