

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

Bài 14

Giao dịch (transactions)

Tóm tắt

- Mô tả giao dịch (transaction)
- Thực hiện giao dịch
- Quản lý giao dịch



Mô tả

Một **transaction** (giao dịch) là:

- Một **công việc đơn** (unit of work).
- Là **thành công chỉ khi tất cả** sự **thay đổi** dữ liệu **được thực hiện** trong một transaction được **hoàn tất** (commit) và được **lưu** trong cơ sở dữ liệu vĩnh viễn.

Mô tả

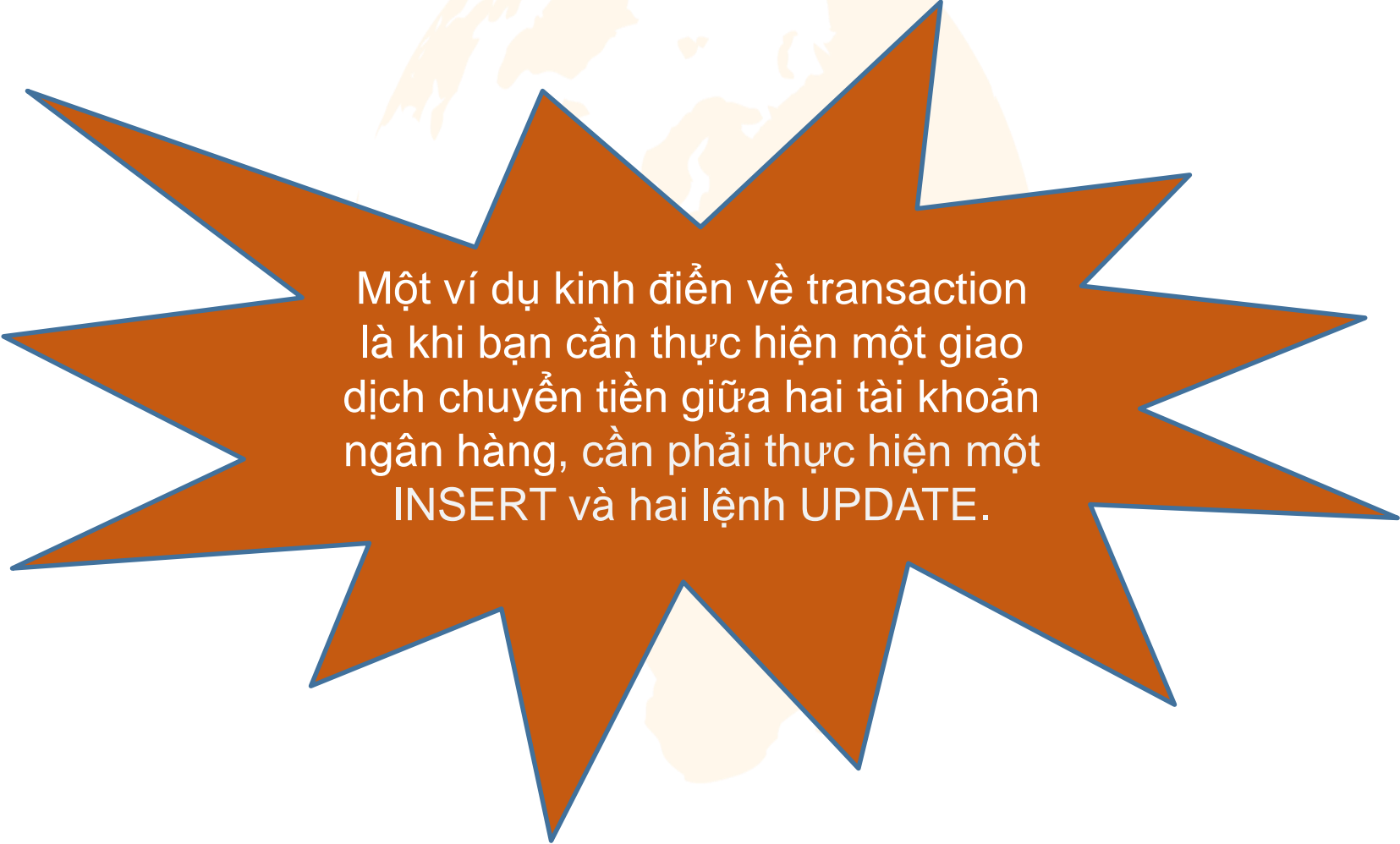
- Nếu **giao dịch bị quay lại** trạng thái ban đầu (rolled) hoặc **hủy bỏ**, thì có nghĩa là transaction (giao dịch) **đã gặp lỗi** và **không xảy ra sự thay đổi nào** với các nội dung của **cơ sở dữ liệu**.
- Một transaction có thể được hoàn tất (**commit**) hoặc quay về trạng thái ban đầu (**roll back**).

Mô tả

Sự cần thiết?

- Có rất nhiều trường hợp người sử dụng **cần phải** thực hiện **những thay đổi** dữ liệu **trong nhiều bảng** của cơ sở dữ liệu.
- Trong vài trường hợp, **dữ liệu sẽ không nhất quán** khi thực thi các **lệnh riêng biệt**.
- Giả sử nếu **câu lệnh đầu tiên** thực hiện một cách **chính xác** nhưng các **câu lệnh khác** thì **thất bại**, lúc này **dữ liệu** sẽ trong trạng thái **sai lệch (incorrect)**.

Dẫn chứng thực tế?



Một ví dụ kinh điển về transaction là khi bạn cần thực hiện một giao dịch chuyển tiền giữa hai tài khoản ngân hàng, cần phải thực hiện một INSERT và hai lệnh UPDATE.

Mô tả

Giả sử bạn có **hai tài khoản A và B** với số tiền tương ứng là **8 tỷ** và **1 tỷ**; nay bạn cần chuyển bớt 2 tỷ từ tài khoản A sang tài khoản B. Sẽ có hai phép cập nhật như sau:

- **Trừ** số tiền hiện có của **tài khoản A** đi 2 tỷ
- **Cộng** thêm số tiền hiện có của **tài khoản B** lên 2 tỷ



Mô tả

Nếu **hai lệnh cập nhật** trên diễn ra **độc lập** (không nằm trong một transaction), và vì một lý do nào đó **lệnh thứ hai bị lỗi**, tài khoản A sẽ còn 6 tỷ và tài khoản B vẫn giữ nguyên 1 tỷ. Điều này không thể chấp nhận được vì **2 tỷ bỗng dưng biến mất!**



Mô tả

Khi thực hiện hai lệnh trên trong một **transaction**, nó sẽ đảm bảo:

- Hoặc cả **hai lệnh** update đều được thực hiện **thành công**. Cả hai tài khoản được cập nhật với số tiền tương ứng.
- Hoặc trong trường hợp giao dịch bị lỗi cả **hai lệnh** đều **không được thực hiện**. Hai tài khoản **giữ nguyên** số tiền **như trước** khi thực hiện transaction.

Định nghĩa Transactions:

Một khối công việc (unit of work) được coi là một giao dịch phải thể hiện (exhibit) bốn thuộc tính **ACID** là:

- Nguyên tử (**atomicity**)
- Nhất quán (**consistency**)
- Cô lập (**isolation**)
- Bền vững (**durability**)

Mô tả

Nguyên tử (atomicity) :
Nếu transaction có nhiều thao tác thì tất cả chúng phải được hoàn tất.

Nhất quán (Consistency): Dãy các thao tác phải nhất quán.

Cô lập (Isolation): Các thao tác phải được thực hiện cô lập khỏi các thao tác khác trên cùng server hoặc trên cùng csdl.

Bền vững (Durability):
Các thao tác được thực hiện trên csdl phải được save và lưu trữ trong csdl vĩnh viễn.

Thực thi transactions: SQL Server hỗ trợ các giao dịch ở nhiều chế độ:

1. Giao dịch hoàn tất tự động (**Autocommit Transactions**): Mỗi câu lệnh đơn được hoàn tất một cách tự động ngay khi nó được hoàn thành.
2. Giao dịch tường minh (**Explicit Transactions**): Mọi transaction bắt đầu tường minh với câu lệnh BEGIN TRANSACTION và kết thúc bằng câu lệnh ROLLBACK hoặc COMMIT.

3. Giao dịch ngầm định (**Implicit Transactions**): Một transaction mới được bắt đầu tự động khi transaction trước đó hoàn thành, và mỗi giao dịch được hoàn tất tường minh bằng việc sử dụng lệnh ROLLBACK hoặc COMMIT.
4. Các giao dịch phạm vi trong batch (**Batch-scoped Transactions**): Đây là các giao dịch có liên quan đến Multiple Active Result Sets (MARS).



Kiểm soát giao dịch

Mô tả

Các giao dịch có thể được kiểm soát thông qua các ứng dụng bằng cách xác định điểm bắt đầu và kết thúc của một giao dịch.

Điều này được thực hiện bằng cách sử dụng các hàm API cơ sở dữ liệu hoặc các câu lệnh Transact-SQL.

Khi một giao dịch được bắt đầu trên một kết nối, tất cả các câu lệnh Transact-SQL được thực hiện trên cùng một kết nối và là một phần của kết nối cho đến khi giao dịch kết thúc.

Giao dịch được quản lý ở cấp kết nối.

Bắt đầu và kết thúc phiên giao dịch bằng

Transact-SQL:

- Một trong nhiều cách người dùng có thể bắt đầu và kết thúc các transactions là bằng các **câu lệnh Transact-SQL**.
- Người dùng có thể bắt đầu một transaction trong SQL Server theo các chế độ ngầm định (**implicit**) hoặc tường minh (**explicit**).

Thực hiện

- Chế độ giao dịch tường minh (**Explicit transaction mode**) bắt đầu một giao dịch bằng câu lệnh **BEGIN TRANSACTION**.
- Người dùng có thể kết thúc giao dịch bằng câu lệnh **ROLLBACK** hoặc **COMMIT**.
- Câu lệnh **BEGIN TRANSACTION** đánh dấu điểm bắt đầu của một transaction tường minh hoặc cục bộ.

Thực hiện

Cú pháp:

```
BEGIN { TRAN | TRANSACTION }  
[ { transaction_name | @tran_name_variable }  
[ WITH MARK [ 'description' ] ]  
] [ ; ]
```

Trong đó:

- transaction_name: chỉ ra tên được gán cho transaction specifies the name that is assigned to the transaction. Nó nên tuân theo quy tắc định danh và giới hạn độ dài là 32 kí tự.
- @tran_name_variable: chỉ ra tên biến do người dùng định nghĩa có chứa tên transaction hợp lệ.
- WITH MARK['description']: chỉ ra transaction được đánh dấu trong log. Chuỗi mô tả (description string) định nghĩa điểm đánh dấu.

Thực hiện

Ví dụ: Tạo một Transaction có tên là 'ThemPhongBan' thực hiện công việc thêm một phòng ban mới vào bảng PhongBan của CSDL BKShop

-- Tạo Transaction có tên là ThemPhongBan

```
DECLARE @TranName nvarchar(64)
```

```
SET @TranName = 'ThemPhongBan'
```

```
BEGIN TRAN @TranName
```

```
INSERT INTO BKShop.dbo.PhongBan VALUES (4,  
'Phong Dao Tao')
```

Thực hiện

Câu lệnh **COMMIT TRANSACTION** đánh dấu kết thúc một transaction ngầm định hoặc tường minh thành công.

Cú pháp:

```
COMMIT { TRAN | TRANSACTION } [ transaction_name |  
    @tran_name_variable ] [ ; ]
```

Trong đó:

- transaction_name: chỉ ra tên transaction đã được gán ở câu lệnh BEGIN TRANSACTION trước đó.
- @tran_name_variable: chỉ ra tên biến do người dùng định nghĩa có chứa tên transaction hợp lệ. Biến có thể khai báo kiểu dữ liệu char, varchar, nchar, hoặc nvarchar. Nếu gán cho biến một chuỗi nhiều hơn 32 kí tự, thì chỉ có 32 kí tự được sử dụng còn lại sẽ bị cắt bỏ.

Thực hiện

Ví dụ: để xác nhận giao dịch ở ví dụ trước cần code

COMMIT WORK -- Đánh dấu sự kết thúc của
Transaction

Hoặc

COMMIT TRAN @TranName -- Giống COMMIT WORK
nhưng lại có thể thêm tên của Transaction

Thực hiện

ROLLBACK TRANSACTION - Giao dịch ngầm định hoặc giao dịch tường minh này bị hủy bỏ hay quay trở lại thời điểm ban đầu của giao dịch, hoặc tới điểm savepoint trong một giao dịch.

```
ROLLBACK { TRAN | TRANSACTION }  
[ transaction_name | @tran_name_variable  
| savepoint_name | @savepoint_variable ] [ ; ]
```

Thực hiện

Trong đó:

- `transaction_name`: chỉ ra tên được gán cho lệnh BEGIN TRANSACTION. Tên được tuân thủ theo qui tắc định danh.
- `@tran_name_variable`: chỉ ra tên của biến người dùng định nghĩa, biến này lưu trữ tên một tên giao dịch hợp lệ. Biến có thể được khai báo các kiểu dữ liệu char, varchar, nchar, hoặc nvarchar.
- `savepoint_name`: chỉ ra tên điểm savepoint_name từ câu lệnh SAVE TRANSACTION. Chỉ dùng savepoint_name khi điều kiện rollback tác động đến một phần giao dịch.
- `@savepoint_variable`: chỉ ra của tên biến savepoint, là biến có chứa tên savepoint hợp lệ. Biến có thể được khai báo với kiểu dữ liệu char, varchar, nchar, hoặc nvarchar.

Thực hiện

Ví dụ: Tạo 2 Transaction, lần lượt tên là 'ThemNhanVienFulltime' và 'ThemNhanVienParttime'. Trong transaction thứ 1 thêm 1 nhân viên và transaction 2 thêm 2 nhân viên. Thực hiện Rollback Transaction thứ nhất và commit transaction thứ 2.

Thực hiện

Code:

```
DECLARE
```

```
@TranName1 nvarchar(64),
```

```
@TranName2 nvarchar(64)
```

```
SET @TranName1 = 'ThemNhanVienFulltime'
```

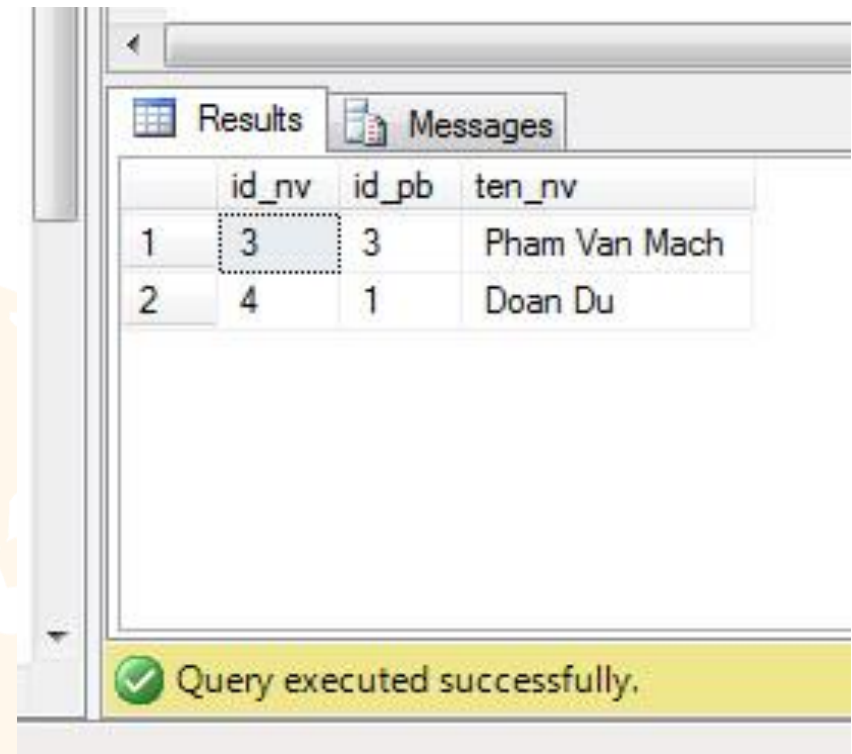
```
SET @TranName2 = 'ThemNhanVienParttime'
```

Thực hiện

```
BEGIN TRAN @TranName1
INSERT INTO NhanVien VALUES
(1, 1, 'Kieu Phong'),
(2, 1, 'Mo Dung')
ROLLBACK TRAN @TranName1

BEGIN TRAN @TranName2
INSERT INTO NhanVien VALUES
(4, 1, 'Doan Du')

SELECT @@TRANCOUNT AS 'Số giao dịch'
COMMIT TRAN @TranName2
```



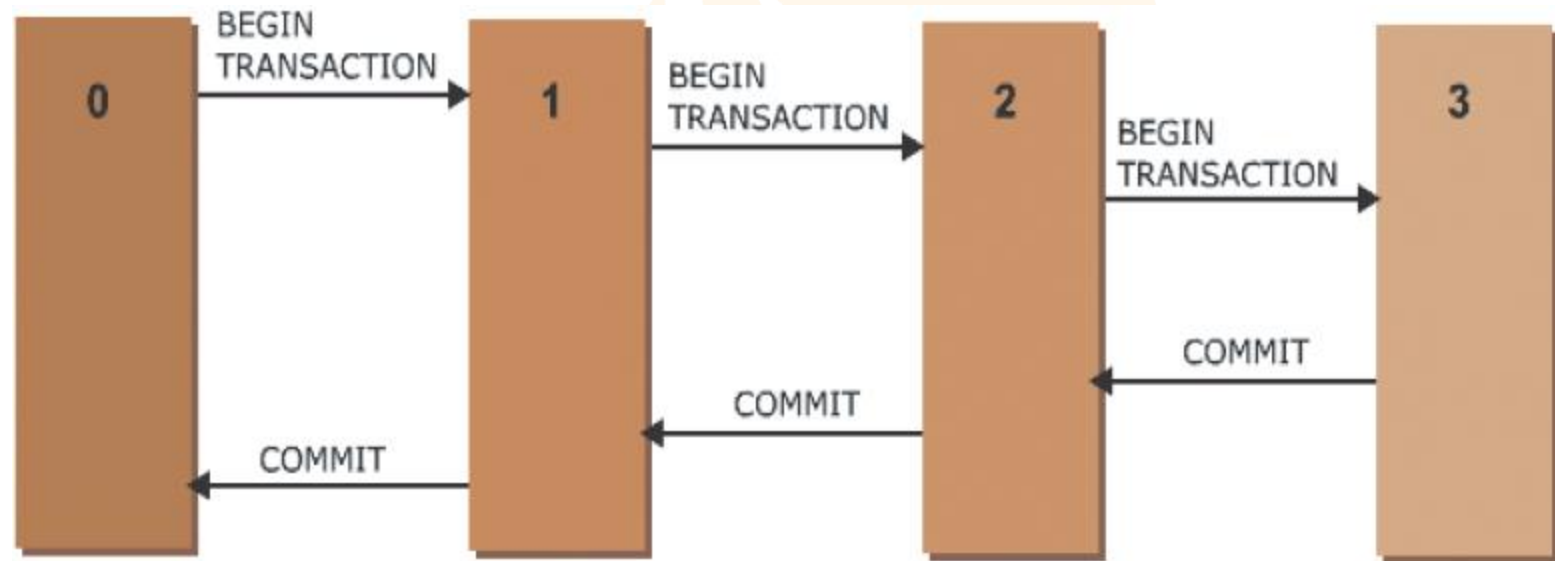
The screenshot shows a SQL Server interface with a 'Results' tab. Below the tab, a table displays the results of a query. The table has three columns: 'id_nv', 'id_pb', and 'ten_nv'. There are two rows of data. The first row has values 1, 3, and 'Pham Van Mach'. The second row has values 2, 4, and 'Doan Du'. A status bar at the bottom of the window indicates 'Query executed successfully.' with a green checkmark icon.

	id_nv	id_pb	ten_nv
1	3	3	Pham Van Mach
2	4	1	Doan Du

Query executed successfully.

Thực hiện

Hàm hệ thống **@@TRANCOUNT** trả về một tổng số các câu lệnh **BEGIN TRANSACTION** xuất hiện(occur) trong kết nối hiện tại.



Cú pháp:

@@TRANCOUNT

Thực hiện

Ví dụ:

```
PRINT @@TRANCOUNT
```

```
BEGIN TRAN
```

```
    PRINT @@TRANCOUNT
```

```
    BEGIN TRAN
```

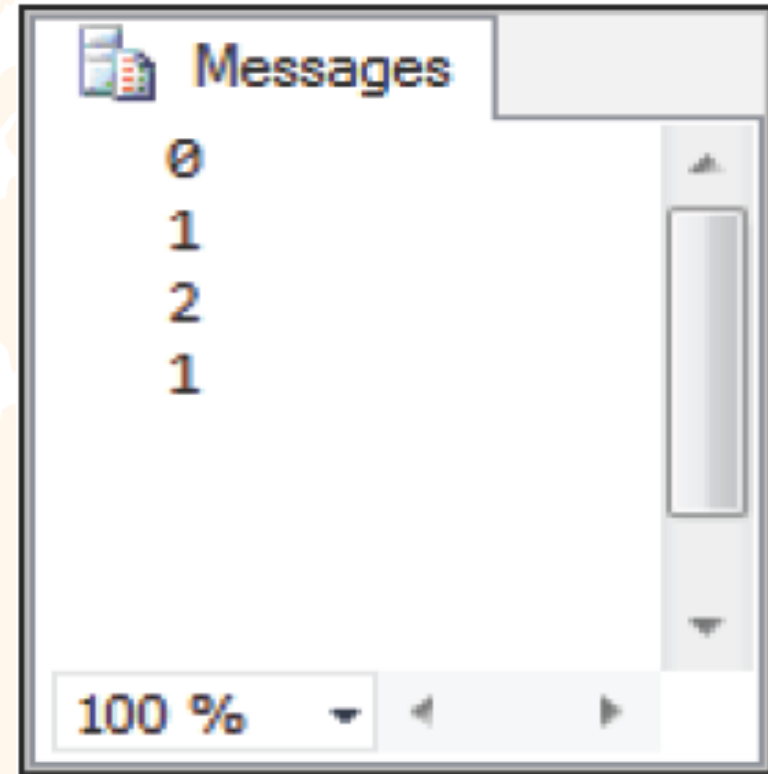
```
        PRINT @@TRANCOUNT
```

```
    COMMIT
```

```
    PRINT @@TRANCOUNT
```

```
COMMIT
```

```
PRINT @@TRANCOUNT
```



Quản lý

- Mặc định **mỗi câu lệnh đơn** được thực thi như là **một transaction** trong SQL Server.
- Nếu **một câu lệnh SQL** đơn được đưa ra, thì **một giao dịch ngầm định** được bắt đầu.
- Khi người dùng sử dụng các câu lệnh **BEGIN TRAN/COMMIT TRAN** tường minh, chúng có thể nhóm chúng lại với nhau như là một giao dịch tường minh.
- SQL Server **thực thi nhiều mức cô lập** giao dịch **đảm bảo** thuộc tính **ACID** của các giao dịch này.

Tóm tắt bài học

- Một **giao dịch** là một dãy các thao tác làm việc như là một **khối duy nhất**.
- Các giao dịch có thể được điều khiển bởi một ứng dụng bằng việc chỉ ra **bắt đầu và kết thúc**.
- **BEGIN TRANSACTION** đánh dấu điểm bắt đầu của một giao dịch tường minh hoặc cục bộ.
- **COMMIT TRANSACTION** đánh dấu kết thúc thành công giao dịch tường minh hoặc giao dịch ngầm định.
- **ROLLBACK** với tùy khóa tùy chọn WORK là để quay lui lại (rolls back) một giao dịch người dùng định nghĩa về thời bắt đầu của giao dịch.
- **@@TRANCOUNT** là hàm hệ thống trả về tổng số câu lệnh BEGIN TRANSACTION xuất hiện trong kết nối hiện hành.

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

Thank for watching!

