**Các dạng scan khi xử lý query**

1. **Hai dạng scan cơ bản Index Scan và Table Scan**
2. ***Index Scan***

Lấy tất cả các hàng trong bảng

### *Clustered Index Scan*

### Quét cả cây clustered index hay có thể được coi như quét một bảng.

### Toàn bộ hoặc phần lớn trong một bảng được quét qua, theo từng hàng để xác định các dữ liệu cần thiết cho truy vấn.

### *Non-clustered Index Scan*

### Lấy tất cả các hàng từ nonclustered index đã được xác định trong cột tham số.

### Nếu trong cột tham số có chứa mệnh đề ORDERED, xử lý truy vấn trả lại các hàng theo thứ tự trong đó Non-clustered index đã được sắp xếp.

### Nếu không có chứa ORDERED thì công cụ lưu trữ tìm kiếm chỉ số theo cách tối ưu mà không cần phân loại đầu ra.

### *Table Scan*

### Một bảng không bao gồm các clustered index.

### Bảng này được lưu trữ như một Heap.

### Kế hoạch quét của nó là đi qua tất cả các trang và lá trong bảng, lấy tất cả các hàng từ bảng đã được xác định trong cột tham số của kế hoạch thực hiện truy vấn.

### Đôi khi tối ưu truy vấn xác định rằng table scan nhanh hơn so với việc dùng chỉ số để lấy một hàng về (thường xảy ra trong các bảng có vài dòng).

### *Dạng truy vấn dữ liệu khác*

### *Index Seek*

### Thường được dùng cho các truy vấn có tính chọn lọc cao.

### Phù hợp với các truy vấn yêu cầu lấy một số ít hàng hoặc chỉ lấy từ 10-15% các hàng trong bảng.

### *Clustered Index Seek*

### Sử dụng Clustered Index để thực hiện việc tìm kiếm các hàng được chỉ định chứa dữ liệu cần thiết nhanh chóng hơn.

### *Non-clustered Index Seek*

### Non-clustered index seek sử dụng một chỉ mục để tìm kiếm các hàng được trả về trực tiếp.

### Sử dụng một Non-clustered index để thực hiện việc điều hành.

1. **Tìm hiểu về Indexes**
2. ***Indexes***

* Cho phép các chương trình trong cơ sở dữ liệu có thể tìm kiếm dữ liệu trong một bảng nhanh chóng mà không cần phải quét toàn bộ bảng.
* Thích hợp với các thao tác tìm kiếm theo kiểu key = value
* Một chỉ mục trong cơ sở dữ liệu là một danh sách các giá trị trong một bảng với mỗi giá trị về vị trí của các hàng được lưu trữ.
* Indexes có thể được tạo trên một cột hay sự kết hợp của các cột trong bảng và được thực hiện dưới dạng B-Tree, cây này được sắp xếp theo từ khóa tìm kiếm và có thể được tìm kiếm tốt với bất kỳ tập con nào của khóa tìm kiếm.

Ví dụ: Một index trên các cột A, B, C có thể được tìm kiếm trên A; trên A, B hay trên A, B, C.

* Index tăng hiệu năng thực hiện truy vấn tuy nhiên chi phí cho không gian lưu trữ có thể cao. Việc insert, update, delete dữ liệu có thể mất nhiều thời gian thực hiện cũng như thời gian xử lý để duy trì các chỉ mục.
* Có 2 loại Index cơ bản là: Clustered index và Non-clustered index. Ngoài ra còn có XML Index, Spatial Index, Fulltext Index.

1. ***Index types***
2. *Clustered Indexes*

* Clustered index tổ chức các bản ghi trong bảng được sắp thứ tự theo trường index (toàn bộ bảng là 1 cây index), với các node lá chứa khóa là các trường được index và đồng thời chứa tất cả các trường còn lại của bảng.
* Clustered index có thể chứa một hoặc nhiều trường
* Khi tạo Primary Key, mặc định Clustered index được tạo kèm với nó
* Khi bảng đã có clustered index thì các index khác (nonclustered) sẽ dùng khóa của trường clustered index làm con trỏ để trỏ về bản ghi tương ứng (nếu bảng không có clustered index thì một giá trị RID nội bộ được dùng).
* Clustered index không phải là duy nhất thì khóa index được gắn thêm một giá trị 4-byte ngẫu nhiên để đảm bảo các node index vẫn là duy nhất. Điều này làm tăng kích thước của clustered index cũng như các index khác.
* Việc tìm kiếm theo trường có clustered index tối ưu hơn so với non-clustered index vì nó bỏ qua được bước bookmark lookup.

1. *Nonclustered Indexes*

* Là một cấu trúc dữ liệu độc lập với bảng (chiếm thêm không gian đĩa).
* Mỗi node lá chứa khóa index và con trỏ trỏ đến trang dữ liệu chứa bản ghi tương ứng.
* Nếu bảng có clustered index, con trỏ này chính là khóa clustered index.
* Nếu bảng không có clustered index, con trỏ này là RowID, một dạng định danh bản ghi kết hợp của fileID + pageID + offset.
* Bảng có thể có nhiều nonclustered index.
* Nonclustered index được lưu trữ tách rời khỏi bảng.
* Trong kế hoạch thực thi thường có thao tác bookmark lookup, đây là thao tác mà hệ thống sau khi tìm kiếm trên cây index nhảy tới bản ghi tương ứng trong bảng để lấy các trường dữ liệu cần.
* Các hàng dữ liệu của bảng bên dưới không được sắp xếp và được lưu trữ theo thứ tự dựa trên các khóa nonclustered.
* Lớp lá của một Nonclustered Index được tạo thành từ các trang chỉ mục thay vì các trang dữ liệu.
* Nonclustered indexes có thể được định nghĩa trên một bảng hoặc xem với một clustered index hay một heap.

1. *So sánh Clustered vs Nonclustered*

* *Clustered index:*
* Tránh phải thực thi bookmark lookup
* Nâng cao độ ổn định cho nonclustered index
* Chỉ được phép tạo một clustered index
* *Non-clustered index:*
* Thực thi bookmark lookup làm giảm hiệu năng
* Cho phép tạo nhiều index trên bảng
* Lưu trữ độc lập với bảng nên tăng khả năng xử lý song song