



R a i s i n g t h e b a r

Index – Chỉ mục

Module: Java Web Back-end Development

Thảo luận bài cũ

- Hỏi và trao đổi về các khó khăn gặp phải trong bài "Các hàm thông dụng trong SQL"
- Tóm tắt lại các phần đã học từ bài "Các hàm thông dụng trong SQL"

Mục tiêu

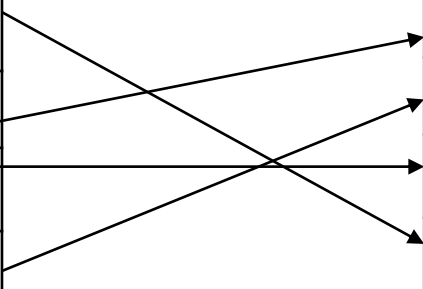
- Trình bày được khái niệm chỉ mục (index)
- Tạo mới, sửa và xóa chỉ mục
- Sử dụng chỉ mục trong truy vấn SQL
- Trình bày được khái niệm khung nhìn (view)
- Tạo mới, sửa và xóa khung nhìn
- Trình bày được khái niệm thủ tục lưu (stored procedure)
- Tạo mới, sửa và xóa thủ tục lưu

Khái niệm chỉ mục - index

- Các bản ghi được lưu trữ vào trong bảng theo đúng thứ tự như khi nhập vào. Do đó có thể dữ liệu này không được sắp xếp.
- Khó khăn: Khi muốn tìm kiếm dữ liệu trên bảng cần phải quét toàn bộ bảng. Điều này làm chậm tốc độ thực thi truy vấn.
- Việc tìm kiếm có thể được trợ giúp nhiều nếu dữ liệu được chứa trong các khối được đánh chỉ mục (index).
- Khi một chỉ mục được tạo ra trên bảng, chỉ mục sẽ tạo nên thứ tự cho các hàng dữ liệu hay các bản ghi trong bảng đó.

Khái niệm chỉ mục – Ví dụ

Emp_No	Emp_No	Emp_Name	Emp_DOB	Emp_DOJ
305	345	James	24-Sep-1968	30-May-1990
345	873	Pamela	27-Jul-1970	19-Nov-1993
693	693	Allan	10-Sep-1970	01-Jul-1992
873	305	Geoff	12-Feb-1973	29-Oct-1996



Chỉ mục – Ưu và nhược điểm

- Ưu điểm:
 - Một chỉ mục cho phép chương trình tìm dữ liệu trong bảng mà không cần duyệt qua toàn bộ bảng.
 - Chỉ mục giúp làm tăng tốc độ thực thi các truy vấn cần nối nhiều bảng hay cần sắp xếp dữ liệu.
 - Một chỉ mục hợp lý có thể cải thiện sự thực thi cơ sở dữ liệu bởi giảm bớt thời gian truy cập.
- Hạn chế:
 - Các bảng có các chỉ mục đòi hỏi nhiều dung lượng bộ nhớ hơn trong CSDL.
 - Các lệnh thao tác dữ liệu đòi hỏi nhiều thời gian xử lý hơn, vì chúng cần cập nhật sự thay đổi tới các chỉ mục.

Hướng dẫn khi tạo chỉ mục

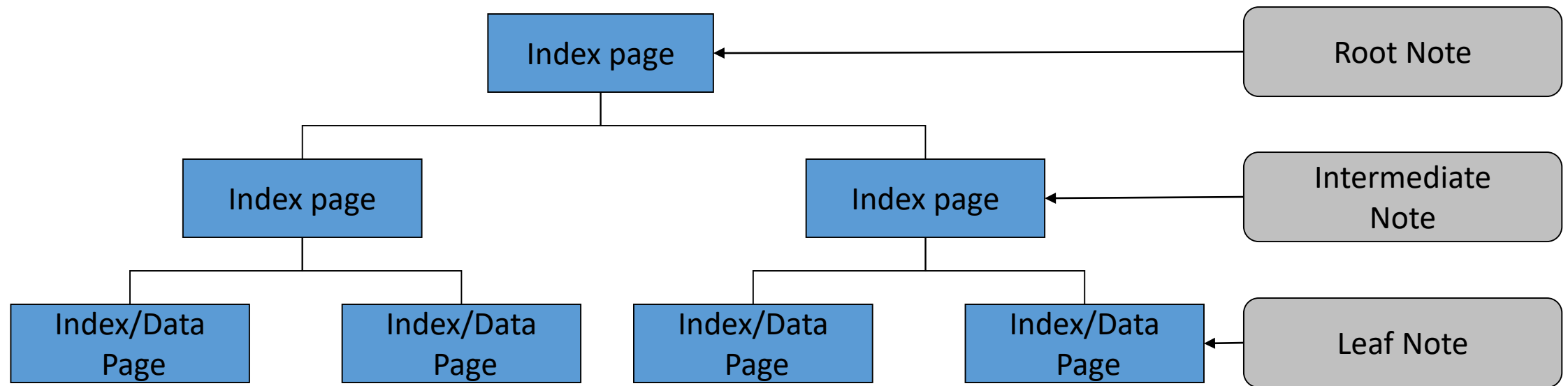
- Cần duy nhất các hàng khi định nghĩa chỉ mục.
- Chỉ mục được tạo và duy trì theo thứ tự sắp xếp tăng hay giảm dần của dữ liệu .
- Một chỉ mục có thể được tạo lập trên một trường hoặc trên nhiều trường của bảng hoặc khung nhìn.
- SQL Server tự động tạo lập các chỉ mục cho các kiểu ràng buộc PRIMARY KEY và UNIQUE.

Hướng dẫn khi tạo chỉ mục

- Nên tạo chỉ mục trên các cột khi:
 - Bảng chứa dữ liệu lớn
 - Cột được sử dụng để tìm kiếm thường xuyên
 - Cột được sử dụng để sắp xếp dữ liệu
 - Dữ liệu trong cột có sự phân biệt cao
- Không nên tạo chỉ mục trên các cột khi:
 - Bảng chỉ chứa một vài hàng
 - Cột thường xuyên có các thao tác thêm sửa xóa
 - Cột có sự trùng lặp dữ liệu cao

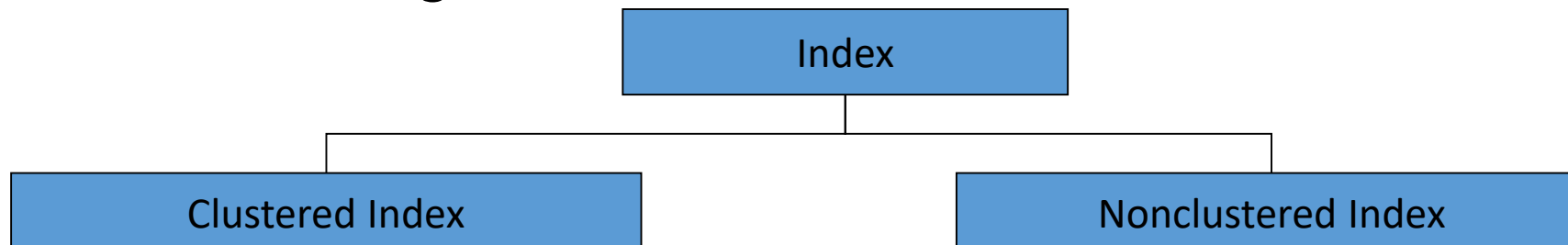
Cấu trúc chỉ mục

- Cây nhị phân chỉ mục



Phân loại chỉ mục

- Chỉ mục phân cụm - Clustered Indexes: xác định thứ tự lưu trữ thực sự của dữ liệu trong bảng. Khi tạo chỉ mục phân cụm trong bảng, dữ liệu trong bảng sẽ được sắp xếp lại về mặt vật lý theo một thứ tự liên tiếp.
- Chỉ mục không phân cụm - Nonclustered Indexes: xác định thứ tự lưu trữ dữ liệu logic của bảng. Chỉ mục không phân cụm sẽ chứa các giá trị khóa và các bộ định vị hàng để chỉ ra vị trí lưu trữ của dữ liệu trong bảng.
- Nên tạo chỉ mục Clustered trước khi tạo chỉ mục Nonclustered để chỉ mục Nonclustered không bị tạo lại sau khi chỉ mục Clustered được tạo.



Phân loại chỉ mục – so sánh

Clustered Indexes	Nonclustered Indexes
Được sử dụng cho các truy vấn trả về số lượng lớn tập kết quả	Được sử dụng cho truy vấn trả về số lượng nhỏ tập kết quả
Chỉ một chỉ mục clustered index có thể được tạo trên một bảng	Nhiều chỉ mục nonclustered index có thể được tạo trên bảng, tối đa là 249
Dữ liệu được sắp xếp vật lý dựa theo khóa clustered	Dữ liệu không được sắp xếp dựa theo khóa nonclustered
Nút lá của chỉ mục clustered chứa trang dữ liệu	Nút lá của chỉ mục nonclustered chứa trang dữ liệu

Cú pháp tạo chỉ mục

- Sử dụng câu lệnh "CREATE INDEX".
- Cú pháp:

```
CREATE[UNIQUE][CLUSTERED|NONCLUSTERED]INDEX index_name  
ON <table or view> ( column_name [ ASC | DESC ] [ ,...n ] )  
[WITH <backward_compatible_index_option> [ ,...n ] ]
```

- Ví dụ:

```
CREATE INDEX IX_Country ON Customer(Country);
```

Chỉ mục duy nhất – Unique Index

- Các chỉ mục duy nhất không cho phép các giá trị lặp lại trong cột hoặc tổ hợp cột đánh chỉ mục.
- Dữ liệu lặp lại phải được cập nhật hoặc xóa trước khi chỉ mục duy nhất được tạo lập.
- Một giá trị rỗng trong một cột khoá phản ánh như một giá trị duy nhất.
- Cú pháp:

```
CREATE UNIQUE [CLUSTERED|NONCLUSTERED]INDEX index_name  
ON <table or view> ( column_name [ ASC | DESC ] [ ,...n ] )  
[WITH <backward_compatible_index_option> [ ,...n ] ]
```

Chỉ mục phức hợp – Composite Index

- Là chỉ mục được tạo lập trên hai hoặc nhiều cột.
- Tạo lập một chỉ số phức hợp khi hai hoặc nhiều cột thường xuyên được tìm kiếm như một nhóm.
- Xác định danh sách các cột hợp thành giá trị duy nhất để tạo lập chỉ mục phức hợp.
- Thứ tự và số lượng các cột trong một chỉ mục tổ hợp ảnh hưởng đến sự thực thi của truy vấn.
- Cú pháp:

```
CREATE UNIQUE [CLUSTERED|NONCLUSTERED]INDEX index_name
```

```
ON <table or view> ( column1, column2[ ,...n ] )
```

```
[WITH <backward_compatible_index_option> [ ,...n ] ]
```

Tùy chọn thuộc tính chỉ mục

- Cú pháp:

```
CREATE UNIQUE [CLUSTERED|NONCLUSTERED]INDEX index_name  
ON <table or view> ( column1, column2[ ,...n ] )  
[WITH <backward_compatible_index_option> [ ,...n ] ]
```

- <backward_compatible_index_option> ::=

```
{  
  PAD_INDEX  
  | FILLFACTOR = number  
  | SORT_IN_TEMPDB  
  | IGNORE_DUP_KEY  
  | STATISTICS_NORECOMPUTE  
  | DROP_EXISTING  
}
```

Hệ số điền đầy – fillfactor

- Sử dụng kèm với PAD_INDEX.
- Khi các bản ghi được thêm vào bảng hoặc các giá trị trong các cột chỉ mục bị thay đổi.
- SQL Server sẽ dịch chuyển khoảng một nửa số hàng sang trang mới để có chỗ trống cho hàng mới thêm vào.
- Hệ số điền đầy nhằm dành riêng tỉ lệ phần trăm khoảng trống của chỉ mục trên mỗi trang để lưu trữ dữ liệu mở rộng trong tương lai.
- Giá trị thay đổi từ 0 đến 100. Nếu dữ liệu không thay đổi, chỉ định một giá trị 100 sao cho các trang sẽ được điền đầy và tối thiểu dung lượng trống.

Sửa chỉ mục

- Sử dụng câu lệnh "ALTER INDEX" để sửa các tùy chọn của chỉ mục
- **Cú pháp:**

```
ALTER INDEX index_name      ON <table or view> REBUILD
```

```
[WITH < rebuild_index_option > [ ,...n ] ]
```

- **Ví dụ:**

```
CREATE UNIQUE INDEX IX_Country ON Customer(Country) WITH (IGNORE_DUP_KEY=ON );  
ALTER INDEX IX_Country ON Customer WITH (IGNORE_DUP_KEY=OFF) ;
```

Xem thông tin chỉ mục

- `sp_helpindex` là một thủ tục lưu trữ hệ thống đưa ra thông báo thông tin về các chỉ mục trên bảng.
- Cú pháp:
`sp_helpindex <Table_name>`

Sử dụng chỉ mục

- SQL Server sử dụng kỹ thuật tối ưu truy vấn để chọn cách tốt nhất để thực thi các truy vấn, bao gồm các chỉ mục để sử dụng
- Kỹ thuật tối ưu cố gắng lựa chọn phương thức truy cập sao cho giá ước lượng thấp nhất trong các mục của trang I/O

- **Cú pháp:**

(INDEX=index_name)

- **Ví dụ:**

```
SELECT * FROM sales (INDEX =nclindx_ordnum)  
WHERE ord_num = 'P3087a'
```

Xóa chỉ mục

- Các chỉ mục không còn cần thiết có thể bị xóa khỏi CSDL để giải phóng bộ nhớ bằng cách sử dụng lệnh DROP INDEX.

- Cú pháp:

```
DROP INDEX table_name.index_name
```

- Ví dụ:

```
DROP INDEX sales.NCLINDX_ordnum
```

- Lệnh DROP INDEX không được áp dụng trên các chỉ số được tạo lập bởi các ràng buộc PRIMARY KEY hoặc UNIQUE.
- DROP INDEX cũng không thể được sử dụng để xóa chỉ mục trên một bảng hệ thống.

Cột được tính toán - Computed Columns

- Cú pháp:

```
CREATE TABLE <table_name> ([<column_name> as  
<computed_column_expression>]);
```

- Ví dụ:

```
CREATE TABLE Calc(Length int, Breadth int, Area as Length*Breadth);
```

- Lưu ý: Một cột được tính toán không thể tạo thành một phần của PRIMARY KEY, UNIQUE KEY hay ràng buộc CHECK.
- Có thể tạo chỉ mục trên cột được tính toán như các cột khác.
- Ví dụ: CREATE INDEX IX_Area ON Calc(Area);

Tóm tắt bài học

- Trình bày được khái niệm chỉ mục (index)
- Tạo mới, sửa và xóa chỉ mục
- Sử dụng chỉ mục trong truy vấn SQL
- Trình bày được khái niệm khung nhìn (view)
- Tạo mới, sửa và xóa khung nhìn
- Trình bày được khái niệm thủ tục lưu (stored procedure)
- Tạo mới, sửa và xóa thủ tục lưu

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: Database and CRUD



R a i s i n g t h e b a r