



# Chương 3 Thiết kế cơ sở dữ liệu vật lý

Giáo viên: Nguyễn Mậu Uyên

Bộ môn: Hệ thống thông tin

Biên soạn: Đỗ Thị Mai Hương

# Chỉ mục

- ❖ Sử dụng chỉ mục để tăng tốc độ truy vấn đến dữ liệu trong bảng.
  - Việc truy cập dữ liệu có sử dụng index gọi là '*index access*'
  - Trường hợp ngược lại, gọi là '*table scan*' khi đó các bản ghi sẽ được xử lý tuần tự. Table scan là một trong những động tác có hại nhất cho hiệu suất của SQL Server.
- ❖ Một chỉ mục là một cấu trúc dữ liệu giúp tổ chức các bản ghi dữ liệu trên đĩa để tối ưu các phép toán truy cập. Chúng cung cấp một phương pháp giúp nhanh chóng tìm kiếm dữ liệu dựa trên các giá trị trong các cột.

# Chỉ mục..

## ❖ Ví dụ:

Nếu tạo một Index trên cột khóa chính và sau đó tìm kiếm một dòng dữ liệu dựa trên một trong các giá trị của cột này:

- Đầu tiên SQL Server sẽ tìm giá trị này trong Index.
- Sau đó sử dụng Index để nhanh chóng xác định vị trí của dòng dữ liệu cần tìm.

## ❖ Các loại Index:

- Clustered
- Non-Clustered
- Unique
- Composite index

## ❖ Clustered Index:

- Clustered Index lưu trữ và sắp xếp dữ liệu vật lý trong table hoặc view dựa trên các giá trị khóa của chúng. Các cột khóa này được chỉ định trong định nghĩa **index**.
  - Mỗi table hoặc view chỉ có duy nhất một **Clusterd Index** vì bản thân các dòng dữ liệu được lưu trữ và sắp xếp theo thứ tự vật lý dựa trên các cột trong loại Index này.
- ❖ Khi dữ liệu trong table hoặc view cần được lưu trữ và sắp xếp theo một thứ tự nhất định chính là lúc cần dùng đến Clustered Index.
- ❖ Một bảng có một Clusted Index thì table được gọi là Clustered Table. Nếu không, các dòng dữ liệu của bảng được lưu trong một cấu trúc không được sắp xếp gọi là HEAP

# Chỉ mục..

- **Non-Clustered** có một cấu trúc tách biệt với dòng dữ liệu trong table hoặc view.
- Mỗi một index loại này chứa các giá trị của các cột khóa trong khai báo của index, và mỗi một bản ghi giá trị của khóa trong index này chứa một con trỏ tới dòng dữ liệu tương ứng của nó trong table.
- Mỗi con trỏ từ một dòng của **Non-Clustered index** tới một dòng dữ liệu trong table được gọi là “row locator”.
- Cấu trúc của row locator phụ thuộc vào việc các trang dữ liệu được lưu trong HEAP hay trong một **Clustered Table**.
- Đối với HEAP, row locator là một con trỏ tới dòng dữ liệu, với clustered table, row locator chính là khóa index của **clustered index**.

## ❖ Unique Index:

- Là kiểu Index dùng để đảm bảo tính duy nhất trong các cột được tạo Index. Nếu Index loại này được tạo dựa trên nhiều cột, thì tính duy nhất của giá trị được tính trên tất cả các cột đó, không chỉ riêng rẽ từng cột
- Ví dụ, nếu tạo ra một Index trên các cột FirstName và LastName trong một bảng, thì giá trị của 2 cột này kết hợp với nhau phải là duy nhất, nhưng riêng rẽ từng cột thì giá trị vẫn có thể trùng nhau.

## ❖ Composite index

- Là kiểu Index có nhiều hơn 1 cột.
- Trong SQL Server 2005 và 2008, có thể có đồng thời tối đa 16 cột trong một Index, miễn là kích thước của Index không vượt quá giới hạn 900 byte.
- Cả hai kiểu index cơ sở là Clustered Index và Non Clustered Index cũng có thể đồng thời là kiểu Composite index.



# Chỉ mục..

## ❖ Tạo chỉ mục:

```
CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX  
    index_name ON {table_name | view_name} (column1 [ASC | DESC]  
    [{, column2 [ASC | DESC]} ... ])  
    [INCLUDE (column_name1 [...n])]  
    [WITH [FILLFACTOR = n] [[,] IGNORE_DUP_KEY]  
    [[,] PAD_INDEX] [[,] DROP_EXISTING] [[,] ONLINE = {ON | OFF}]]
```

- ❖ Một index có thể được tạo trên một cột hoặc kết hợp nhiều cột
- ❖ Tham số ONLINE = ON – vẫn cho phép truy cập bảng khi mà chỉ mục đang được tạo.



# Chỉ mục..

## ❖ Vô hiệu hóa chỉ mục:

```
alter index index_name on table_name disable
```

## ❖ Xóa chỉ mục:

```
drop index index_name on table_name
```

# Chỉ mục..

❖ Ví dụ về tạo chỉ mục trong CSDL AdventureWorksT

❖ Tạo chỉ mục cho cột EmployeeID trên bảng  
[HumanResources].[Employee]

```
CREATE CLUSTERED INDEX Index_EmpID ON  
[HumanResources].[Employee] (EmployeeID)
```

❖ Tạo chỉ mục gồm nhiều trường

```
CREATE UNIQUE INDEX I_Order_Product_ID ON  
[Production].[WorkOrder] ( [WorkOrderID],[ProductID] ) WITH  
FILLFACTOR = 80
```

❖ Không thể tạo chỉ mục duy nhất trên các trường có giá trị lặp lại hoặc có giá trị NULL

## ❖ Bài thực hành index:

- Tạo bảng Customers(MaKH, TenKH, Diachi, Dienthoai, Fax, Email)
- Chèn hàng triệu bản ghi. Tính thời gian thực hiện câu lệnh tìm kiếm khách hàng theo địa chỉ khi không tạo index và khi có index

# Chỉ mục..

CREATE TABLE Customers(

MaKH varchar (10) NOT NULL ,TenKH nvarchar (30) NULL  
, DiaChi nvarchar (50) NULL ,DienThoai varchar (20) NULL  
, Fax varchar (20) NULL , Email varchar (20) NULL )

- Chèn vào 13000000 bản ghi
- Tạo thủ tục Tinhthoigianthuchien tính thời gian thực hiện câu lệnh tìm kiếm theo địa chỉ.
- Tạo index trên trường địa chỉ.
- Gọi lại thủ tục Tinhthoigianthuchien kiểm tra thời gian thực hiện câu lệnh tìm kiếm
- Tham khảo bài thực hành về index

# Thiết kế an toàn bảo mật dữ liệu

LOGO

## ❖ Cần cân nhắc 3 mục đích:

- Tính bí mật: thông tin không được phép truy cập tùy tiện.
- Tính toàn vẹn: Chỉ những người có thẩm quyền mới được phép cập nhật dữ liệu.
- Tính sẵn có: đối với những người sử dụng có thẩm quyền thì họ luôn được phép truy nhập đến những dữ liệu đã cấp quyền cho họ.

# Thiết kế an toàn bảo mật dữ liệu..

LOGO

- ❖ Xác định các nhóm sử dụng và các quyền hạn
- ❖ Xác định quyền hạn thao tác trên cơ sở dữ liệu
- ❖ Mã hóa dữ liệu

# Quyết định thiết kế cơ sở dữ liệu vật lý

LOGO

- ❖ Quyết định thiết kế chỉ mục.
- ❖ Phân tích nhu cầu sử dụng thông tin
- ❖ Phân tích yếu tố bảo mật



# Quyết định thiết kế cơ sở dữ liệu vật lý..

- ❖ Quyết định thiết kế chỉ mục (index):
  1. Đặt chỉ mục một thuộc tính.
  2. Thuộc tính hoặc những thuộc tính nào được đặt chỉ mục.
  3. Thiết lập chỉ mục nhóm (clustered index).

# Quyết định thiết kế cơ sở dữ liệu vật lý..

LOGO

## ❖ Phân tích nhu cầu sử dụng

1. Bổ sung thuộc tính.
2. Bổ sung bảng tạm.
3. Phi chuẩn hóa

# Quyết định thiết kế cơ sở dữ liệu vật lý..

## ❖ Phi chuẩn hoá (denormalization): :

- Quá trình khi lưu trữ chuyển cơ sở dữ liệu logic ở dạng chuẩn cao (BCNF hoặc 4NF) sang dạng chuẩn thấp hơn, có thể là 1NF, 2NF, hoặc 3NF.
- Sự dư thừa và không nhất quán dữ liệu.
- Nhưng tăng tốc độ thực thi của những truy vấn và giao tác thường xuyên.

# Quyết định thiết kế cơ sở dữ liệu vật lý..

## ❖ Ví dụ:

- Denormalized.
- Một quan hệ ở dạng BCNF có thể được lưu trữ ở nhiều bảng cũng ở dạng BCNF.

➤ EMPLOYEE(Ssn,Name,Phone,Grade,Salary)

→ EMP1(Ssn,Name,Phone)

EMP2(Ssn,Grade,Salary)

# Quyết định thiết kế cơ sở dữ liệu vật lý..

LOGO

## ❖ Phân tích yếu tố bảo mật

1. Bổ sung thêm các bảng người dùng.
2. Bổ sung thêm bảng nhóm người dùng.

# Database tuning

- ❖ Khi cơ sở dữ liệu được triển khai và vận hành, có thể xuất hiện một số vấn đề chưa được tính toán trong quá trình thiết kế cơ sở dữ liệu vật lý.
- Cần xem xét lại thiết kế cơ sở dữ liệu vật lý.
- ❖ **Mục tiêu của tuning:**
  - Tăng tốc độ thực thi của ứng dụng.
  - Giảm thời gian đáp ứng của truy vấn và giao tác.
  - Cải thiện thông năng của giao tác.

# Một số điều chỉnh

A decorative header image featuring a close-up of a fountain pen nib and a small logo with the word "LOGO" in green capital letters.

- ❖ Điều chỉnh chỉ mục (tuning indexes).
- ❖ Điều chỉnh thiết kế cơ sở dữ liệu (tuning the database design).
- ❖ Điều chỉnh truy vấn (tuning queries).



# Điều chỉnh chỉ mục



## ❖ Chỉ mục nào?

- Truy vấn chạy quá lâu vì ít chỉ mục.
- Những chỉ mục có thể không sử dụng.
- Chỉ mục trải qua quá nhiều update vì chỉ mục được đặt trên thuộc tính thay đổi thường xuyên.

## ❖ Những lựa chọn điều chỉnh chỉ mục:

- Bỏ hoặc thiết lập chỉ mục mới.
- Đổi non-clustered index thành clustered index hoặc ngược lại.
- Tổ chức lại chỉ mục.

# Điều chỉnh thiết kế cơ sở dữ liệu

LOGO

❖ Nếu thiết kế cơ sở dữ liệu không đạt được mục tiêu mong đợi

→ Xem lại thiết kế cơ sở dữ liệu.

# Tổng kết

A decorative header image featuring a close-up of a fountain pen nib and a small logo with the word "LOGO" in green capital letters.

- ❖ Quá trình thiết kế cơ sở dữ liệu vật lý
- ❖ Trong quá trình ứng dụng cơ sở dữ liệu, có thể nảy sinh vấn đề.
  - Điều chỉnh cơ sở dữ liệu (database tuning).

# Bài toán bán sách online



## ❖ Lựa chọn những chỉ mục nào được tạo:

- Những quan hệ nào được chỉ mục và trường nào hoặc kết hợp một số trường để tạo ra chỉ mục khoá tìm kiếm.
- Với mỗi chỉ mục, xác định xem nó nên được phân cụm hay không phân cụm?

## ❖ Điều chỉnh lược đồ logic

- Phân tích nhu cầu sử dụng thông tin
- Phân tích nhu cầu bảo mật thông tin.

## ❖ Mô hình quan hệ

Nhaxuatban(MaNXB, TenNXB, Diachi, Dienthoai)

Tacgia(MaTG, Hoten, Diachi, Dienthoai)

Chude(MaCD, TenCD)

Sach(Masach, Tensach, soluong, dongia, MaNXB, MaCD)

Tacgia\_Sach(MaTG, Masach, sotrang, ghichu)

Khachhang(MaKH, Hoten, Diachi, Dienthoai)

Donhang(MaDH, MaKH, NgayDH)

Chitietdonhang(MaDH, MaSach, Soluong)

## ❖ Phân tích nhu cầu sử dụng thông tin:

- Mô hình dữ liệu có lưu trữ được giá tại thời điểm bán cho mỗi đơn hàng?
- Việc tính thành tiền và tổng tiền cho các mặt hàng có trong đơn hàng có lặp đi lặp lại gây mất thời gian?
- Có phân biệt được đơn hàng nào là đang đặt, đơn hàng nào là đã đặt nhưng chưa giao, đơn hàng nào là đã giao hàng?

## ❖ Phân tích nhu cầu bảo mật thông tin:

- Đối với người quản lý: Có thể chia ra thành mấy đối tượng? Đối tượng quản trị hệ thống, đối tượng bán hàng, đối tượng giao hàng, đối tượng quản lý đặt hàng?
- Đối với người mua: có cấp quyền truy cập vào hệ thống bán hàng?



# Bài toán bán sách online

LOGO

## ❖ Phân tích yêu cầu phát sinh

- Nhu cầu giảm giá trên đơn hàng mua?
- Nhu cầu giảm giá cho khách Vip?
- Nhu cầu giảm giá theo dịp đặc biệt?
- Nhu cầu giảm giá trên một số sách?
- Nhu cầu giảm giá trên tất cả các sách?

# Bài tập

LOGO

- ❖ **Bài 1: Phân tích dung lượng dữ liệu mô hình khai thác của “Hệ thống quản lý bán sách online”**
- ❖ **Bài 2: Phân tích dung lượng dữ liệu mô hình khai thác của “Hệ thống quản lý đào tạo LQDUNI”**