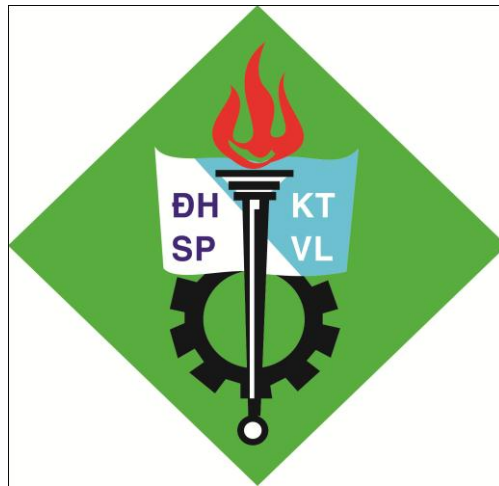


**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG**

**KHOA CÔNG NGHỆ THÔNG TIN**



***TÀI LIỆU GIẢNG DẠY***

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**  
**(DATABASE MANAGEMENT SYSTEM)**

**Mã HP: TH1307**

**Số tín chỉ: 3**

Lưu hành nội bộ

## Mục lục

CHƯƠNG 1: TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MICROSOFT SQL SERVER .....	1
I- Giới thiệu kiến thức tổng quan về hệ QTCSDL:.....	1
II- Giới thiệu hệ quản trị CSDL MS SQL Server: .....	2
III- Mô hình Client/Server: .....	4
1-Tổng quan về cấu trúc Client/Server .....	4
2-Các đặc trưng của mô hình Client/server : .....	5
3-Các tầng cấu trúc: .....	6
4-Các mô hình dữ liệu của hệ thống Client/Server: .....	7
IV- Các thành phần của CSDL trong SQL SERVER: .....	10
1-Tables : .....	10
2-Diagram – Sơ đồ quan hệ: .....	10
3-Views – Khung nhìn hay table ảo: .....	10
4-Stored Procedure – Thủ tục nội:.....	10
5-User Defined Function : .....	10
6-Users : .....	11
7-Roles : .....	11
8-Rules : .....	11
9-Defaults : .....	11
10-User Defined Data Type :.....	11
11-Full Text Catalogs : .....	11
V- Các CSDL hệ thống của SQL SERVER: .....	11
VI- Tính bảo mật trên SQL SERVER : .....	12
1-Mức đăng nhập vào SQL Server : .....	12
2-Quyền thao tác trên SQL Server: .....	12
Bài tập .....	13
CHƯƠNG 2 : TẠO LẬP CSDL TRÊN SQL SERVER .....	14
I- Các loại file lưu trữ csdl : .....	14
II- Tạo cơ sở dữ liệu: .....	14
1-Cú pháp lệnh tạo cơ sở dữ liệu .....	14
2-Xóa cơ sở dữ liệu.....	17
3-Sửa đổi cơ sở dữ liệu .....	17
4-Đổi tên cơ sở dữ liệu: .....	19
5-Xem thông tin cơ sở dữ liệu bằng thủ tục lưu trữ hệ thống: .....	19
III- Tạo và quản lý Table : .....	20
1-Cú pháp lệnh tạo bảng: .....	20
2-Thay đổi cấu trúc bảng – alter table: .....	21
3-Sửa đổi kiểu dữ liệu hoặc kích thước của cột : .....	21

4-Thêm cột .....	22
5-Đổi tên cột : .....	23
6-Xóa cột : .....	23
7-Thêm RBTV cho bảng : .....	23
a. Khai báo ràng buộc trong lúc tạo bảng: .....	23
b. Khai báo ràng buộc trên bảng đã tồn tại.....	24
8-Hủy bỏ RBTV khỏi bảng.....	25
9-Kích hoạt hay tạm ngưng kiểm tra các ràng buộc Foreign key và Check: .....	25
10-Xóa table : .....	26
11-Đổi tên bảng: .....	26
12-Xóa các dòng trên table và giải phóng vùng nhớ:.....	26
Bài tập .....	27
<b>CHƯƠNG 3 : CÁC PHÁT BIỂU CƠ BẢN CỦA TRANSACT-SQL.....</b>	<b>29</b>
I- Giới thiệu ngôn ngữ Transact-SQL :.....	29
II- Kiểu dữ liệu: .....	29
III- Truy xuất dữ liệu : (DQL <i>data query language</i> ) .....	31
1-From clause : .....	31
2-Select clause : .....	31
3-Where clause: .....	36
4-Order by clause:.....	37
5-Tổng hợp dữ liệu: .....	38
6-Toán tử union: .....	38
IV- Data manipulation language (DML) .....	39
1-Chèn mẫu tin mới : .....	39
2-Sửa dữ liệu:.....	39
3-Xóa mẫu tin: .....	41
4-Tạo mới một bảng với các bộ giá trị lấy từ CSDL:.....	41
Bài tập .....	42
<b>Chương IV : TẠO VIEW (BẢNG ẢO).....</b>	<b>50</b>
I- Khái niệm : .....	50
II- Cách tạo và sử dụng khung nhìn : .....	50
1-Tạo View bằng T-SQL:.....	50
2-Các hạn chế khi tạo Views .....	51
3-Mã hóa View: .....	51
4-Sử dụng View để thay đổi dữ liệu:.....	51
5-Thay đổi định nghĩa View:.....	52
6-Xóa View:.....	53
Bài tập .....	53
<b>CHƯƠNG V: T-SQL NÂNG CAO .....</b>	<b>55</b>
I- Khai báo và sử dụng biến .....	55

1-Biến cục bộ .....	55
2-Biến toàn cục .....	56
II- Cấu trúc điều khiển.....	56
1-Lệnh If...else .....	56
2-Lệnh While .....	57
3-Lệnh Case .....	57
III- Thủ tục thường trú (Stored Procedures) .....	59
1-Khái niệm .....	59
2-Khai báo và sử dụng thủ tục .....	59
3-Stored procedure hệ thống.....	63
IV- Kiểu dữ liệu cursor .....	64
1-Khái niệm Cursor .....	64
2-Khai báo và sử dụng Cursor .....	64
3-Biến cursor.....	67
V- Hàm người dùng (User Defined Functions) .....	70
1-Khái niệm hàm người dùng .....	70
2-Khai báo và sử dụng .....	70
3-Các hàm hệ thống .....	73
VI- Triggers và cài đặt ràng buộc dữ liệu .....	74
1-Giới thiệu.....	74
2-Sử dụng Trigger.....	75
Bài tập .....	79
<b>CHƯƠNG 6: BẢO MẬT VÀ AN TOÀN DỮ LIỆU .....</b>	<b>83</b>
I- Bảo mật trong hệ quản trị cơ sở dữ liệu .....	83
1-Khái niệm cơ bản về bảo mật .....	83
2-Lựa chọn bảo mật .....	84
3-Tạo Tài Khoản Đăng Nhập (Login Account): .....	84
4-Thay đổi mật khẩu: .....	85
5-Cấp quyền truy xuất CSDL hiện hành cho Login: .....	85
6-Xóa quyền truy xuất CSDL hiện hành: .....	85
7-Cấp quyền thực thi trên CSDL: .....	85
8-Từ Chối quyền thực thi trên CSDL: .....	87
9-Xóa bỏ quyền thực thi đã cấp hoặc từ chối trên CSDL:.....	87
10-Cấp quyền tạo đối tượng trong CSDL: .....	88
11-Từ chối quyền tạo đối tượng trên csdl: .....	88
12-Xóa bỏ quyền tạo đối tượng đã cấp hoặc từ chối trên CSDL: .....	88
13-Vai Trò Của User Trong SQL Server .....	88
14-Các quyền truy cập trên các đối tượng trong một CSDL:.....	89
15-Vai trò của người sử dụng trong SQL Server và cơ sở dữ liệu.....	89
II- Sao lưu và khôi phục dữ liệu .....	90

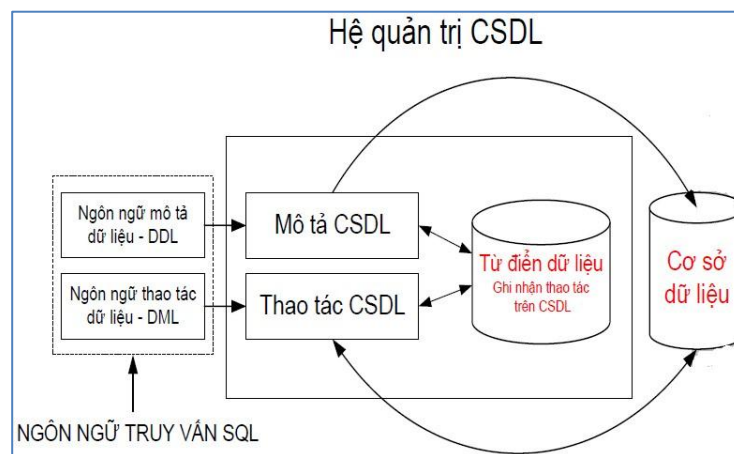
1-Lý do phải sao lưu và khôi phục dữ liệu .....	90
2-Các loại sao lưu dữ liệu .....	91
3-Các mô hình khôi phục dữ liệu .....	91
4-Sao lưu cơ sở dữ liệu (Backup Database) .....	92
5-Khôi phục dữ liệu (Restore Database) .....	93
Bài tập .....	93
TÀI LIỆU THAM KHẢO.....	94
PHỤ LỤC.....	95



## CHƯƠNG 1: TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MICROSOFT SQL SERVER

### I- Giới thiệu kiến thức tổng quan về hệ QTCSDL:

- Thông tin là nguồn tài nguyên quý giá của một tổ chức. Các phần mềm máy tính là những công cụ hiệu quả để xử lý thông tin và hệ quản trị cơ sở dữ liệu là công cụ phổ biến cho phép lưu trữ và rút trích thông tin một cách hiệu quả.
- Cơ sở dữ liệu (CSDL) là một tập hợp dữ liệu được tổ chức và lưu trữ theo một cấu trúc chặt chẽ nhằm phục vụ nhiều đối tượng sử dụng với nhiều mục đích khác nhau.
- Một hệ CSDL (DB system) bao gồm một CSDL (Database) và một hệ quản trị CSDL (DBMS)
- Hệ quản trị CSDL (Database Management System - DBMS) là một công cụ phần mềm tổng quát nhằm hỗ trợ việc lưu trữ, truy xuất và quản trị CSDL. Nó cung cấp cho người dùng và ứng dụng một môi trường thuận tiện và sử dụng hiệu quả tài nguyên dữ liệu. Nói cách khác, Hệ quản trị CSDL là phần mềm chuyên dụng để giải quyết tốt các tình huống mà CSDL đặt ra như: bảo mật, cạnh tranh trong truy xuất.



Hình 1: Kiến trúc của một hệ quản trị CSDL

### Mục tiêu của hệ quản trị cơ sở dữ liệu:

Hệ quản trị cơ sở dữ liệu phải đảm bảo các mục tiêu sau: dữ liệu sẵn dùng (data availability), tính toàn vẹn dữ liệu (data integrity), an toàn dữ liệu (data security), và độc lập dữ liệu (data independency).

- Dữ liệu sẵn dùng (data availability): dữ liệu được tổ chức sao cho mọi người dùng có thể truy cập dễ dàng theo chức năng và nhiệm vụ của họ.

- Tính toàn vẹn dữ liệu (data integrity): dữ liệu lưu trữ trong cơ sở dữ liệu là đúng đắn, đáng tin cậy.
- An toàn dữ liệu (data security): chỉ những người dùng được phép mới có thể truy cập dữ liệu. Nếu nhiều người dùng truy cập chung một mục dữ liệu cùng lúc thì hệ quản trị cơ sở dữ liệu không cho phép họ thực hiện những thay đổi gây mâu thuẫn dữ liệu.
- Độc lập dữ liệu (data independency): hệ quản trị cơ sở dữ liệu phải cho phép tất cả mọi người dùng được phép lưu trữ, cập nhật và rút trích dữ liệu hiệu quả mà không cần nắm chi tiết về cấu trúc của cơ sở dữ liệu được biểu diễn và cài đặt.

**Quá trình phát triển của hệ quản trị cơ sở dữ liệu:**

- Flat files: 1960s – 1980s
- Hierarchical: 1970s – 1990s
- Network : 1970s – 1990s
- Relational: 1980s – đến nay
- Object-oriented: 1990s – đến nay
- Object-relational: 1990s – đến nay
- Data warehousing: 1980s – đến nay
- Web-enabled: 1990s – đến nay

**II- Giới thiệu hệ quản trị CSDL MS SQL Server:**

SQL viết tắt của Structured Query Language (ngôn ngữ truy vấn có cấu trúc), là công cụ sử dụng để tổ chức, quản lý và truy xuất dữ liệu được lưu trữ trong các CSDL. SQL là một hệ thống ngôn ngữ bao gồm tập các câu lệnh sử dụng để tương tác với CSDL quan hệ.

Ngôn ngữ truy vấn có cấu trúc và các hệ quản trị CSDL quan hệ là một trong những nền tảng kỹ thuật quan trọng trong công nghiệp máy tính. Hiện nay SQL được xem là ngôn ngữ chuẩn trong CSDL. Các hệ quản trị CSDL quan hệ thương mại hiện có như Oracle, SQL Server, Informix, DB2,... đều chọn SQL làm ngôn ngữ cho sản phẩm của mình.

SQL Server là một hệ quản trị CSDL nhiều người dùng kiểu Client/Server. Đây là hệ thống cơ bản dùng lưu trữ dữ liệu cho hầu hết các ứng dụng lớn hiện nay. Mô hình



Client/Server trên SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị CSDL cung cấp cho người dùng các khả năng:

- **Định nghĩa dữ liệu:** SQL cung cấp khả năng định nghĩa các CSDL, các cấu trúc lưu trữ và tổ chức dữ liệu cũng như mối quan hệ giữa các thành phần dữ liệu.
- **Truy xuất và thao tác dữ liệu:** Người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các CSDL.
- **Điều khiển truy cập:** SQL có thể được sử dụng để cấp phát và kiểm soát các thao tác của người sử dụng trên dữ liệu, đảm bảo sự an toàn cho CSDL
- **Đảm bảo toàn vẹn dữ liệu:** SQL định nghĩa các ràng buộc toàn vẹn trong CSDL nhờ đó đảm bảo tính hợp lệ và chính xác của dữ liệu trước các thao tác cập nhật cũng như các lỗi của hệ thống.

SQL Server sử dụng ngôn ngữ lập trình và truy vấn CSDL **Transact-SQL (T-SQL)**, một phiên bản của Structured Query Language. Ngôn ngữ lập trình và truy vấn T-SQL cho phép truy xuất dữ liệu, cập nhật và quản lý hệ thống CSDL quan hệ. Mỗi máy chủ chỉ có một hệ quản trị CSDL SQL Server.

**SQL Server có một số đặc tính sau:**

- Cho phép quản trị một hệ CSDL lớn (lên đến vài tera byte), có tốc độ xử lý dữ liệu nhanh đáp ứng yêu cầu về thời gian.
- Cho phép nhiều người cùng khai thác trong một thời điểm đối với một CSDL và toàn bộ quản trị CSDL (lên đến vài chục ngàn user).
- Có hệ thống phân quyền bảo mật tương thích với hệ thống bảo mật của công nghệ NT (Network Technology), tích hợp với hệ thống bảo mật của Windows NT hoặc sử dụng hệ thống bảo vệ độc lập của SQL Server.
- Hỗ trợ trong việc triển khai CSDL phân tán và phát triển ứng dụng trên Internet
- Cho phép lập trình kết nối với nhiều ngôn ngữ lập trình khác dùng xây dựng các ứng dụng đặc thù (Visual Basic, C, C++, ASP, ASP.NET, XML,...).
- Sử dụng câu lệnh truy vấn dữ liệu Transaction-SQL (Access là SQL, Oracle là PL/SQL).

**Các ấn bản của SQL Server**

---

- Enterprise Manager: Là ấn bản đầy đủ của SQL Server có thể chạy trên 32CPU và 64GB RAM. Có các dịch vụ phân tích dữ liệu Analysis Service.
- Standard: Giống như Enterprise nhưng bị hạn chế một số tính năng cao cấp, có thể chạy trên 2CPU, 4GB RAM.
- Personal: Phiên bản này chủ yếu để chạy trên PC, nên có thể chạy trên các hệ điều hành Windows 9x, Windows XP, Windows 2000, Windows 2003...
- Developer: Là phiên bản tương tự như Enterprise nhưng bị giới hạn bởi số user kết nối đến.
- Desktop Engine: Là phiên bản một engine chỉ chạy trên desktop và không có giao diện người dùng (GUI), kích thước CSDL giới hạn bởi 2GB.
- Win CE: Sử dụng cho các ứng dụng chạy trên Windows CE.
- Trial: Phiên bản dùng thử, bị giới hạn bởi thời gian.
- SQL Client: Là phiên bản dành cho máy khách, khi thực hiện khai thác sẽ thực hiện kết nối đến phiên bản SQL Server, phiên bản này cung cấp giao diện GUI khai thác cho người sử dụng.
- SQL Connectivity only: Là phiên bản sử dụng chỉ cho các ứng dụng để kết nối đến SQL Server, phiên bản này không cung cấp công cụ GUI cho người dùng khai thác SQL Server.

Các phiên bản này được cài đặt phụ thuộc vào bộ cài đặt mà bạn chọn hoặc lựa chọn khai cài đặt (ví dụ phiên bản Enterprise, Standard, Personal,... bạn phải chọn theo bộ cài đặt, phiên bản SQL Client, Connectivity,... do bạn chọn trong các hộp thoại trong quá trình cài đặt).

### III- Mô hình Client/Server:

#### 1- Tổng quan về cấu trúc Client/Server

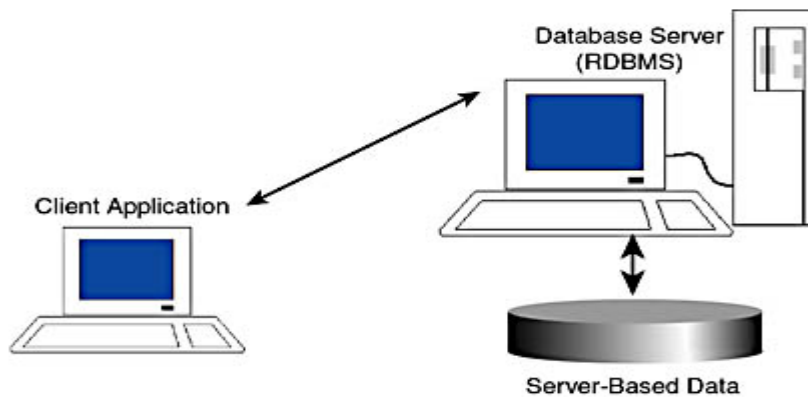
Một CSDL phải đảm bảo được **tính độc lập** giữa dữ liệu và chương trình ứng dụng. Vì vậy, trong mô hình CSDL Client/Server, hệ thống Server lưu trữ CSDL ở trên máy A, thì hệ thống Client chạy các chương trình ứng dụng phải ở trên máy khác.

Trong mô hình này, các thành phần **xử lý ứng dụng** trên hệ thống Client đưa ra yêu cầu cho **phần mềm CSDL** trên máy **Client**, phần mềm này sẽ **kết nối** với **phần mềm CSDL** chạy

trên **Server**. Phần mềm CSDL trên Server sẽ *truy nhập* vào CSDL và gửi trả kết quả cho máy Client.

## 2- Các đặc trưng của mô hình Client/server :

Một ứng dụng kiểu Client/Server bao gồm 2 phần: Một phần chạy trên Server (máy chủ) và phần khác chạy trên các Workstations (máy trạm).



Hình 2: Mô hình Client/Server trên SQL Server

Phần Server: (Máy chủ) chứa các CSDL, cung cấp các chức năng phục vụ cho việc tổ chức và quản lý CSDL, cho phép nhiều người sử dụng cùng truy cập dữ liệu. Điều này không chỉ tiết kiệm mà còn thể hiện tính nhất quán về mặt dữ liệu. Tất cả dữ liệu đều được truy xuất thông qua server, không được truy xuất trực tiếp. Do đó, có độ bảo mật cao, tính năng chịu lỗi, chạy đồng thời, sao lưu dự phòng...

Phần Client (Máy khách): Là các phần mềm chạy trên máy trạm cho phép người sử dụng giao tiếp CSDL trên Server.

Với mỗi Máy chủ chỉ có một hệ thống QTCSDL SQL Server. Nếu muốn có nhiều hệ thống QTCSDL bạn cần có nhiều máy chủ tương ứng.

Hệ thống máy tính Client/Server có 5 mô hình kiến trúc dựa trên cấu hình phân tán về truy nhập dữ liệu, gồm:

- Mô hình CSDL tập trung (Centralized database model)
- Mô hình CSDL theo kiểu file - server (File - server database model)
- Mô hình xử lý từng phần CSDL (Database extract proceSQL Servering model)
- Mô hình CSDL Client/Server (Client/Server database model)
- Mô hình CSDL phân tán (Distributed database model)

### 3- Các tầng cấu trúc:

Theo kiến trúc ANSI-PARC, một CSDL có 3 mức biểu diễn :

#### *a. Mô hình dữ liệu mức thấp (Mức vật lý hay mức trong)*

Đưa ra các khái niệm mô tả chi tiết về cách thức dữ liệu được lưu trữ trong máy tính nên còn được gọi là mức lưu trữ CSDL.

Tại mức này, vấn đề cần giải quyết là dữ liệu gì và được lưu trữ như thế nào, lưu ở đâu (đĩa từ, băng từ, track, sector)? Cần các chỉ mục gì? Việc truy xuất là tuần tự (Sequential Access SQL Server) hay ngẫu nhiên (Random Access SQL Server) đối với từng loại dữ liệu.

Những người hiểu và làm việc với CSDL tại mức này là người quản trị CSDL (Administrator), những người sử dụng (NSD) chuyên môn.

Ví dụ : Mô hình quan hệ, mô hình mạng, mô hình phân cấp.

#### *b. Mô hình dữ liệu mức cao (Mức quan niệm)*

Cung cấp các khái niệm gần gũi với người dùng. Mô hình tự nhiên và giàu ngữ nghĩa. Tại mức này sẽ giải quyết cho câu hỏi CSDL cần phải lưu giữ bao nhiêu loại dữ liệu? Đó là những dữ liệu gì? Mối quan hệ giữa các loại dữ liệu này như thế nào?

Từ thế giới thực các chuyên viên tin học qua quá trình khảo sát và phân tích, cùng với những người sẽ đảm nhận vai trò quản trị CSDL, sẽ xác định được những loại thông tin gì được cho là cần thiết phải đưa vào CSDL, đồng thời mô tả rõ mối liên hệ giữa các thông tin này. Có thể nói cách khác, CSDL mức quan niệm là một sự biểu diễn trừu tượng CSDL mức vật lý; hoặc ngược lại, CSDL vật lý là sự cài đặt cụ thể của CSDL mức quan niệm.

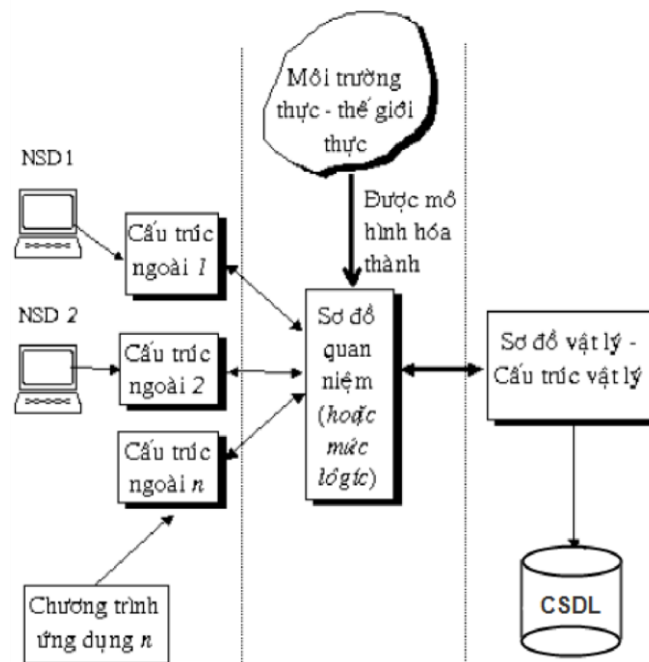
Ví dụ : Mô hình thực thể kết hợp (ERD), mô hình đối tượng.

#### *c. Mô hình dữ liệu mức thực hiện*

Đưa ra các khái niệm người dùng có thể hiểu được nhưng không nằm quá xa với dữ liệu được tổ chức thực sự trên máy. Đây là mức của NSD và các chương trình ứng dụng.

Mỗi người sử dụng hay mỗi chương trình ứng dụng có thể được "nhìn" (View) CSDL theo một góc độ khác nhau. Có thể "nhìn" thấy toàn bộ hay chỉ một phần hoặc chỉ là các thông tin tổng hợp từ CSDL hiện có. Người sử dụng hay chương trình ứng dụng có thể hoàn toàn không được biết về cấu trúc tổ chức lưu trữ thông tin trong CSDL, thậm chí ngay cả tên gọi của các loại dữ liệu hay tên gọi của các thuộc tính. Họ chỉ có thể làm việc trên một

phần CSDL theo cách "nhìn" do người quản trị hay chương trình ứng dụng quy định, gọi là khung nhìn (View).



Hình 3: Kiến trúc tổng quát (ANSI\_PARC của một CSDL)

#### 4- Các mô hình dữ liệu của hệ thống Client/Server:

##### a. Mô hình CSDL tập trung (Centralized database model)

Trong mô hình này, các thành phần xử lý ứng dụng, phần mềm CSDL và bản thân CSDL đều ở trên một bộ xử lý.

Ví dụ: Người dùng máy tính cá nhân có thể chạy các chương trình ứng dụng có sử dụng phần mềm CSDL Oracle để truy nhập tới CSDL nằm trên đĩa cứng của máy tính cá nhân đó. Từ khi các thành phần ứng dụng, phần mềm CSDL và bản thân CSDL cùng nằm trên một máy tính thì ứng dụng đã thích hợp với mô hình tập trung.

Hầu hết công việc xử lý luồng thông tin chính được thực hiện bởi nhiều tổ chức mà vẫn phù hợp với mô hình tập trung. Ví dụ một bộ xử lý mainframe chạy phần mềm CSDL IMS hoặc DB2 của IBM có thể cung cấp cho các trạm làm việc ở các vị trí phân tán sự truy nhập nhanh chóng tới CSDL trung tâm. Tuy nhiên trong rất nhiều hệ thống như vậy, cả 3 thành phần của ứng dụng CSDL đều thực hiện trên cùng một máy mainframe do vậy cấu hình này cũng thích hợp với mô hình tập trung.

##### b. Mô hình CSDL theo kiểu file - server (File - server databasemodel)

Trong mô hình CSDL theo kiểu file - server các thành phần ứng dụng và phần mềm CSDL ở trên một hệ thống máy tính và các file vật lý tạo nên CSDL nằm trên hệ thống máy tính khác. Một cấu hình như vậy thường được dùng trong môi trường cục bộ, trong đó một hoặc nhiều hệ thống máy tính đóng vai trò của server, lưu trữ các file dữ liệu cho hệ thống máy tính khác tham nhập tới. Trong môi trường file - server, phần mềm mạng được thi hành và làm cho các phần mềm ứng dụng cũng như phần mềm CSDL chạy trên hệ thống của người dùng cuối coi các file hoặc CSDL trên file server thực sự như là trên máy tính của người chính họ.

Mô hình file server rất giống với mô hình tập trung. Các file CSDL nằm trên máy khác với các thành phần ứng dụng và phần mềm CSDL; tuy nhiên các thành phần ứng dụng và phần mềm CSDL có thể có cùng thiết kế để vận hành một môi trường tập trung.

***c. Mô hình xử lý từng phần CSDL (Database extract proceSQL Servering model)***

Một CSDL ở xa có thể được truy nhập bởi phần mềm CSDL, được gọi là xử lý dữ liệu từng phần. Với mô hình này, người sử dụng có thể tại một máy tính cá nhân kết nối với hệ thống máy tính ở xa nơi có dữ liệu mong muốn. Người sử dụng có thể tác động trực tiếp đến phần mềm chạy trên máy ở xa và tạo yêu cầu để lấy dữ liệu từ CSDL đó. Người sử dụng cũng có thể chuyển dữ liệu từ máy tính ở xa về chính máy tính của mình và vào đĩa cứng và có thể thực hiện việc sao chép bằng phần mềm CSDL trên máy cá nhân.

Với cách tiếp cận này, người sử dụng phải biết chắc chắn là dữ liệu nằm ở đâu và làm như thế nào để truy nhập và lấy dữ liệu từ một máy tính ở xa. Phần mềm ứng dụng đi kèm cần phải có trên cả hai hệ thống máy tính để kiểm soát sự truy nhập dữ liệu và chuyển dữ liệu giữa hai hệ thống. Tuy nhiên, phần mềm CSDL chạy trên hai máy không cần biết rằng việc xử lý CSDL từ xa đang diễn ra vì người sử dụng tác động tới chúng một cách độc lập.

***d. Mô hình CSDL Client/Server (Client/Server database model)***

Mô hình CSDL Client/Server gần giống như mô hình file - server, tuy nhiên mô hình Client/Server có rất nhiều thuận lợi hơn mô hình file - server.

Xét ví dụ sau đây: Một người dùng cuối muốn tạo ra một văn tin để lấy dữ liệu tổng số, yêu cầu đòi hỏi lấy dữ liệu từ 1000 bản ghi.

Với cách tiếp cận File-Server nội dung của tất cả 1000 bản ghi phải đưa lên mạng, vì phần mềm CSDL chạy trên máy của người sử dụng phải truy nhập từng bản ghi để thỏa mãn yêu cầu của người sử dụng.

Với cách tiếp cận CSDL Client/Server, chỉ có lời văn tin khởi động ban đầu và kết quả cuối cùng cần đưa lên mạng. Phần mềm CSDL chạy trên máy lưu giữ CSDL sẽ truy nhập các bản ghi cần thiết, xử lý chúng và gọi các thủ tục cần thiết để đưa ra kết quả cuối cùng.

#### *Front-end software*

Trong mô hình CSDL Client/Server, thường nói đến các phần mềm front-end software và back-end software. Front-end software được chạy trên một máy tính cá nhân hoặc một workstation và đáp ứng các yêu cầu đơn lẻ riêng biệt, phần mềm này đóng vai trò của Client trong ứng dụng CSDL Client/Server và thực hiện các chức năng hướng tới nhu cầu của người dùng cuối cùng, phần mềm Front-end software thường được chia thành các loại sau:

- End user database software: Được thực hiện bởi người sử dụng cuối trên chính hệ thống của họ để truy nhập các CSDL cục bộ nhỏ cũng như kết nối với các CSDL lớn hơn trên CSDL Server.
- Simple query and reporting software: Được thiết kế để cung cấp các công cụ để dùng hơn trong việc lấy dữ liệu từ CSDL và tạo các báo cáo đơn giản từ dữ liệu đã có.
- Data analysis software: Cung cấp các hàm về tìm kiếm, khôi phục, chúng có thể cung cấp các phân tích phức tạp cho người dùng.
- Application development tools: Cung cấp các khả năng về ngôn ngữ mà các nhân viên hệ thống thông tin chuyên nghiệp sử dụng để xây dựng các ứng dụng CSDL.
- Database administration Tools: Các công cụ này cho phép người quản trị CSDL sử dụng máy tính cá nhân hoặc trạm làm việc để thực hiện việc quản trị CSDL như định nghĩa các CSDL, thực hiện lưu trữ hay phục hồi.

#### *Back-end software*

Phần mềm này bao gồm phần mềm CSDL Client/Server và phần mềm mạng chạy trên máy đóng vai trò là Server CSDL.

#### ***e. Mô hình CSDL phân tán (Distributed database model)***

Cả hai mô hình File - Server và Client/Server đều giả định là dữ liệu nằm trên một bộ xử lý và chương trình ứng dụng truy nhập dữ liệu nằm trên một bộ xử lý khác, còn mô hình CSDL phân tán lại giả định bản thân CSDL có ở trên nhiều máy khác nhau.

#### **IV- Các thành phần của CSDL trong SQL SERVER:**

Mỗi CSDL có các đối tượng sau:

##### **1- Tables :**

Table là đối tượng chính của CSDL dùng lưu trữ dữ liệu cần quản lý. Mỗi table có 1 hay nhiều Field. Mỗi Field ứng với một loại dữ liệu cần lưu trữ.

Table còn có các thành phần liên quan như :

##### **a) Constraint – Ràng buộc:**

Constraint là các chỉ định ràng buộc dữ liệu trong bảng hoặc các bảng khác nhau theo một quy tắc nào đó.

##### **b) Triggers – Bẫy Lỗi:**

Trigger thường chứa các mã lệnh kiểm tra dữ liệu, có tính năng tự động thực hiện khi có hành động nào đó xảy ra đối với dữ liệu trong Table như Insert, Update, Delete.

##### **c) Indexs – Chỉ mục :**

Hỗ trợ cho việc sắp xếp và tìm kiếm nhanh thông tin trên table.

##### **2- Diagram – Sơ đồ quan hệ:**

Thể hiện mối quan hệ dữ liệu giữa các table.

##### **3- Views – Khung nhìn hay table ảo:**

Là đối tượng dùng hiển thị dữ liệu được rút trích, tính toán từ các Table theo nhu cầu của người dùng.

##### **4- Stored Procedure – Thủ tục nội:**

Chứa các lệnh T-SQL dùng thực hiện một số tác vụ nào đó. Stored Proc có thể nhận và truyền tham số. Stored Proc được biên dịch trước, do đó thời gian thực hiện nhanh khi được gọi. Có nhiều Stored Proc hệ thống được định nghĩa với tiền tố “sp\_” có nhiệm vụ thu thập thông tin từ các bảng hệ thống và rất có ích cho việc quản trị.

##### **5- User Defined Function :**



Hàm do người dùng định nghĩa

#### 6- Users :

Chứa danh sách User sử dụng CSDL. Người quản trị hệ thống cao nhất có User Name là **dbo**, tên đăng nhập (Login Name) hệ thống mặc định là **sa**. Tài khoản **sa** luôn tồn tại và không thể bỏ đi. Để thay đổi mật khẩu của **sa**, cách nhanh nhất là:

Mở trình Query Analyzer

Thực hiện thủ tục hệ thống : EXEC **SP\_PASSWORD** NULL, <NewPass>

#### 7- Roles :

Các qui định vai trò và chức năng của User trong hệ thống SQL Server

#### 8- Rules :

Các qui tắc ràng buộc dữ liệu được lưu trữ trên Table

#### 9- Defaults :

Các khai báo giá trị mặc định

#### 10- User Defined Data Type :

Kiểu dữ liệu do người dùng tự định nghĩa

#### 11- Full Text Catalogs :

Tập phân loại dữ liệu Text.

### V- Các CSDL hệ thống của SQL SERVER:

Sau khi cài đặt, SQL Server có 4 CSDL hệ thống:

- 1- **Master:** là CSDL kiểm soát tất cả các hoạt động trên SQL Server, chứa thông tin về hệ thống SQL Server : Các tài khoản đăng nhập, cấu hình hệ thống, thông tin về các CSDL đã tạo, các thủ tục hệ thống thực hiện các tác vụ quản trị hệ thống, các thủ tục của người dùng tạo thêm...

Ví dụ khi User tạo CSDL mới, thêm hay xóa Stored Procedure, các thông tin này đều được lưu trữ trong CSDL master.

*Chú ý: Cần Backup CSDL Master mỗi khi bạn sửa đổi hệ thống.*

- 2- **Model :** Chứa các template dùng làm mẫu để tạo CSDL mới. Khi bạn tạo CSDL thì SQL Server lấy tất cả các mẫu (bao gồm Tables, Views,...) từ CSDL này.

- 3- **MSDB:** dùng hỗ trợ dịch vụ SQL Server Agent, bao gồm sắp xếp thông tin về các công việc theo lịch biểu, các cảnh báo lỗi, các sự kiện, nhân bản. Lịch sử về các hoạt động Backup đều được lưu trong CSDL này.

Ví dụ: Khi tạo một lịch trình cho việc backup dữ liệu hay lịch trình để thực hiện Stored Procedure, tất cả các tác vụ này đều lưu trong CSDL này.

Nếu xóa CSDL này, phải cài đặt lại nó khi cần dùng hoặc khi hệ thống yêu cầu.

- 4- **Tempdb:** là nơi lưu trữ các thông tin tạm thời của các hoạt động trên SQL, ví dụ như các table tạm phục vụ cho việc sắp xếp dữ liệu. CSDL tempdb tự khởi tạo lại mỗi khi SQL Server được khởi động lại.

## **VI- Tính bảo mật trên SQL SERVER :**

SQL Server kiểm tra User ở 2 mức :

- Mức đăng nhập vào SQL Server
- Mức sử dụng các đối tượng trên SQL Server.

### **1- Mức đăng nhập vào SQL Server :**

Để kết nối với SS, người sử dụng phải có một tài khoản đăng nhập (Login Account) được cung cấp bởi người quản trị hệ thống.

Khi SQL Server chạy trên Windows NT, Người Quản trị hệ thống có thể chỉ định nó chạy ở 1 trong 2 chế độ xác nhận :

#### **a- Chế độ xác nhận Windows NT (Windows NT Authentication Mode):**

Khi đó User chỉ cần sử dụng tài khoản đăng nhập của Windows NT truy cập vào mạng là có thể kết nối tới SS và các tài nguyên khác trên mạng. Người quản trị hệ thống phải gán quyền truy xuất SQL Server cho mỗi tài khoản người dùng hoặc tài khoản nhóm người dùng trên mạng. SQL Server cài đặt trên Windows 9x không hỗ trợ chế độ này.

#### **b- Chế độ hỗn hợp (Mixed Mode):**

Một số User có thể kết nối với SQL Server với xác nhận Windows NT. Một số User khác chỉ sử dụng xác nhận của SQL Server (SQL Server Authentication) dựa trên Login Name và Password do người quản trị CSDL cấp.

### **2- Quyền thao tác trên SQL Server:**

Tùy theo yêu cầu, mỗi người dùng có thể được gán hoặc không gán các quyền như :

- Quyền sử dụng các ứng dụng CSDL trong SQL Server
- Quyền tạo và sửa đổi cấu trúc các đối tượng trong SQL Server
- Quyền truy cập và xử lý dữ liệu.

### **Bài tập**

Câu 1: Phân biệt các khái niệm CSDL, hệ quản trị CSDL, hệ CSDL, SQL, T-SQL?

Câu 2: Cho ví dụ về một số hệ quản trị CSDL theo mô hình Client/Server ?

Câu 3: Hãy trình bày các đặc trưng của mô hình Client/Server?

Câu 4: Hãy so sánh mô hình dữ liệu tập trung và mô hình dữ liệu phân tán ?

Câu 5: Cho ví dụ về mô hình dữ liệu tập trung và mô hình dữ liệu phân tán hiện nay?

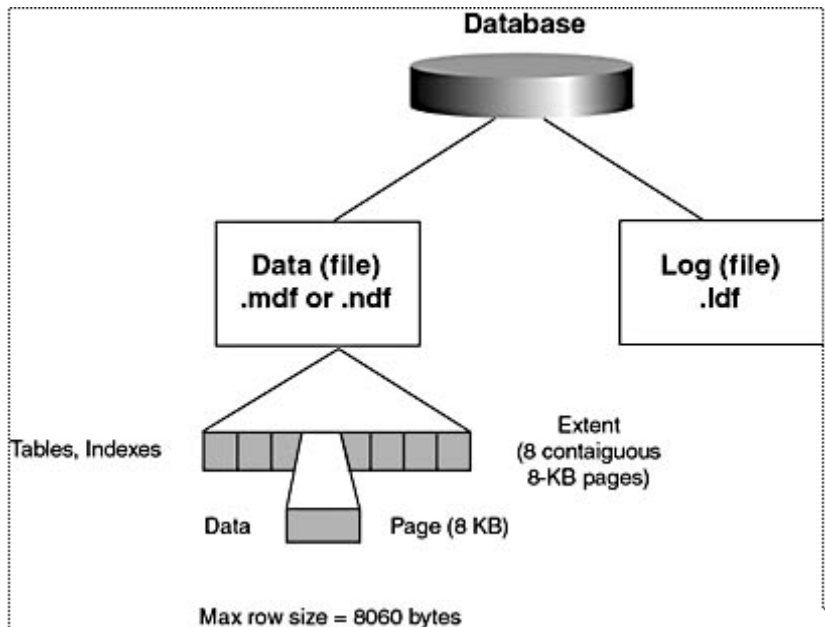
## CHƯƠNG 2 : TẠO LẬP CSDL TRÊN SQL SERVER

### I- Các loại file lưu trữ csdl :

Khi tạo một CSDL, SQL Server sẽ tạo những file lưu trữ. Có 2 loại file như sau:

✚ File dữ liệu (Data File) bao gồm:

- File chính (Primary data file): Mỗi CSDL chỉ có 1 file dữ liệu chính có phần mở rộng là MDF.
- Các file dữ liệu phụ



(Secondary data files): Các file này chứa các dữ liệu và đối tượng không nằm vừa trong Primary file. Một số CSDL có thể lớn đến nỗi phải cần nhiều file dữ liệu phụ hay cần sử dụng các file phụ trên các ổ đĩa riêng để phân dữ liệu qua nhiều đĩa. Các file dữ liệu phụ tiếp theo nên có phần mở rộng là NDF.

✚ Các file nhật ký (Log files): lưu trữ nhật ký giao tác (LDF) (Transaction log) thực hiện trên CSDL, nhằm mục đích phục hồi CSDL khi có sự cố.

Khi sử dụng nhiều file dữ liệu, SQL Server tự động trải dữ liệu qua tất cả các file dữ liệu. Điều này làm giảm tranh chấp và các điểm nóng (hotspot) trong dữ liệu.

Tuy nhiên, đối với file nhật ký, SQL không trải thông tin trên các file nhật ký. Khi 1 file nhật ký đầy, thông tin sẽ được ghi tiếp vào file khác.

### II- Tạo cơ sở dữ liệu:

#### 1- Cú pháp lệnh tạo cơ sở dữ liệu

**CREATE DATABASE** *Database\_name*

[ **ON [PRIMARY]** ( <Thông tin File Dữ Liệu> ),... ]

[ **LOG ON** ( <Thông tin File Log> ),... ]

Trong đó :

<Thông tin File> = ( **FILENAME** = 'D:\Path\FileName'

[, **NAME** = *LogicalName* ]

[, **SIZE** = <*Size MB or KB*>]

[, **MAXSIZE** = <*MaxSize* > ]

[, **FILEGROWTH** = <*No of KyloByte or Percentage*>] )

### Arguments

- **Database\_name** : Tên Database phải được phân biệt trên cùng server (tối đa 128 ký tự)
- **ON** : Khai báo các file chứa cơ sở dữ liệu
- **PRIMARY**: Dùng chỉ định file chính của CSDL. Nếu không chỉ định Primary, file đầu tiên được liệt kê trong phát biểu Create Database trở thành primary file.
- **NAME = 'LogicalName'** : Tên luận lý của file lưu trữ CSDL. Tên này được sử dụng trong các phát biểu của T-SQL, cần phân biệt.
- **FILENAME = 'FileName'** : Tên lưu trên đĩa. Bao gồm cả ổ đĩa, thư mục
- **SIZE = <size MB or KB>** : Kích thước File theo đơn vị MB (mặc định) hoặc KB. Thấp nhất 512 KB, mặc định 1 MB.
- **MAXSIZE = max\_size** : Chỉ định kích thước tối đa mà file có thể tăng. Nếu không chỉ định, kích thước file sẽ tăng cho đến khi đĩa đầy.
- **FILEGROWTH** : Khai báo số gia khi tăng kích thước File, không được lớn hơn MaxSize. Mặc định là 10% và giá trị nhỏ nhất là 64 KB.
- **LOG ON** : Khai báo các file dùng lưu trữ nhật ký thao tác trên database. Nếu không chỉ định LOG ON, SQL tự tạo một file nhật ký có size bằng 25 percent của tổng kích thước của tất cả các data files trên database.

Mặc định, các tập tin dữ liệu và log được lưu trong thư mục MSSQL\ Data của thư mục cài đặt SQL Server.

**Ví dụ 1:** Tạo CSDL BanHang, có kích thước 20MB – trong đó, 15MB dành cho file dữ liệu và 5MB dành cho file nhật ký.

```
CREATE DATABASE BanHang
```

```
ON ( NAME = 'Sales_dat',
```

```

FILENAME = 'D:\data\saledat.mdf',

SIZE = 15MB, MAXSIZE = 50MB,

FILEGROWTH = 20% )

LOG ON ( NAME = 'Sales_log',

FILENAME = 'D:\data\salelog.ldf',

SIZE = 5MB, MAXSIZE = 20MB,

FILEGROWTH = 1MB )

```

**Ví dụ 2:** Tạo CSDL lưu ở nhiều file. Theo Microsoft, File dữ liệu đầu tiên có phần mở rộng là MDF, các file dữ liệu còn lại có phần mở rộng là .NDF. Các file nhật ký có phần mở rộng là LDF.

<pre> CREATE DATABASE BanHang ON ( NAME = 'Sales_dat1',     FILENAME = 'D:\data\sale.mdf',     SIZE = 10,     MAXSIZE = 50,     FILEGROWTH = 5 ), ( NAME = 'Sales_dat2',     FILENAME = 'D:\data\sale1.ndf',     SIZE = 10, MAXSIZE = 50,     FILEGROWTH = 5 ) LOG ON ( NAME = 'Sales_log1',     FILENAME = 'D:\data\sale.ldf',     SIZE = 5, MAXSIZE = 20,     FILEGROWTH = 5 ), ( NAME = 'Sales_log2',     FILENAME = 'D:\data\sale1.ldf',     SIZE = 5, MAXSIZE = 20,     FILEGROWTH = 5 ) </pre> <p><i>Do không dùng từ khóa PRIMARY nên mặc định file đầu tiên (Sale_dat1) là file chính.</i></p>	<pre> CREATE DATABASE BanHang ON ( NAME = 'Sales_dat1',     FILENAME = 'D:\data\sale.mdf',     SIZE = 10,     MAXSIZE = 50,     FILEGROWTH = 5 ), <b>PRIMARY</b> ( NAME = 'Sales_dat2',     FILENAME = 'D:\data\sale1.ndf',     SIZE = 10, MAXSIZE = 50,     FILEGROWTH = 5 ) LOG ON ( NAME = 'Sales_log1',     FILENAME = 'D:\data\sale.ldf',     SIZE = 5, MAXSIZE = 20,     FILEGROWTH = 5 ), ( NAME = 'Sales_log2',     FILENAME = 'D:\data\sale1.ldf',     SIZE = 5, MAXSIZE = 20,     FILEGROWTH = 5 ) </pre> <p><i>Do dùng từ khóa PRIMARY nên file (Sale_dat2) là file chính.</i></p>
--	---

Chú thích:

- CSDL mới được tạo là bản sao của CSDL Model, nên mọi thứ trong CSDL Model sẽ có trong CSDL mới. Mặc định, các thành viên có vai trò *sysadmin* – System Administrator và *dbcreator* – Database Creators mới có quyền tạo Database mới.

## 2- Xóa cơ sở dữ liệu

**Cú pháp:**

**DROP DATABASE** <DatabaseName> [, ...]

Ví dụ:      **DROP DATABASE** mydb1, mydb2

Chú ý:

- CSDL Master có Table **SYSDATABASES** chứa thông tin như Name, ID,... của các database trên Server

**If Exists(Select 'True'**

**From master.SysDatabases**

**Where Name = 'Thuvien')**

**Drop Database ThuVien**

- Bạn phải có vai trò db\_owner trên CSDL.
- Không ai đang làm việc với CSDL

## 3- Sửa đổi cơ sở dữ liệu

Để thêm hay xóa file và nhóm file hoặc thay đổi các thuộc tính của file và nhóm file, như thay đổi tên và dung lượng của file sử dụng

**Cú pháp:**

**ALTER DATABASE** database\_name

**ADD FILE** <Thông tin File Dữ Liệu> [...n]

| **ADD LOG FILE** <Thông tin file Log > [...n]

| **REMOVE FILE** <Tên logic>

| **MODIFY FILE** <Thông tin file>

**Các ví dụ:**

1. Thêm 1 file chứa dữ liệu cho Database Test có tên là Testdat2

```
CREATE DATABASE Test
```

```
ON (NAME = 'Testdat1',
```

```
    FILENAME = 'D:\data\Testdat1.ndf',
```

```
    SIZE = 5MB, MAXSIZE = 100MB,
```

```
    FILEGROWTH = 5MB
```

```
)
```

```
ALTER DATABASE Test
```

```
    ADD FILE ( NAME = 'Testdat2',
```

```
        FILENAME = 'D:\data\Testdat2.ndf',
```

```
        SIZE = 5MB, MAXSIZE = 100MB,
```

```
        FILEGROWTH = 5MB
```

```
)
```

2. Thêm 2 file Log kích thước 5-MB cho Database Test có tên là Testlog2 và Testlog3

```
ALTER DATABASE Test
```

```
    ADD LOG FILE
```

```
        ( NAME = 'Testlog2',
```

```
            FILENAME = 'D:\data\Testlog2.ldf',
```

```
            SIZE = 5MB, MAXSIZE = 100MB,
```

```
            FILEGROWTH = 5MB),
```

```
        ( NAME = 'Testlog3',
```

```
            FILENAME = 'D:\data\Testlog3.ldf',
```

```
            SIZE = 5MB, MAXSIZE = 100MB, FILEGROWTH = 5MB)
```

3. Xóa file của Database Test có tên là Testdat2

```
ALTER DATABASE Test REMOVE FILE Testdat2
```



4. Sửa file : Tăng kích thước file cho Testdat1 trong Database Test lên là 20M

ALTER DATABASE Test

MODIFY FILE (NAME = 'Testdat1',SIZE = 20MB)

#### 4- Đổi tên cơ sở dữ liệu:

Cú pháp:

**EXEC SP\_RENAMEDB** 'OldName', 'NewName'

Chú ý:

- ☒ Bạn phải là có quyền Admin
- ☒ CSDL phải ở chế độ SingleUser
- ☒ Tên các file và nhóm file không bị ảnh hưởng bởi sự thay đổi.

#### 5- Xem thông tin cơ sở dữ liệu bằng thủ tục lưu trữ hệ thống:

Thông tin của các Database được lưu trữ trong table hệ thống SYSDATABASES của CSDL Master. Bạn có thể liệt kê bằng lệnh:

Select \* From Master.dbo.SysDatabases

Ngoài ra, bạn có thể dùng các system stored procedures để hiển thị thông tin about databases and database options:

System stored procedure	Description
SP_HELPDB	Hiển thị thông tin của tất cả Databases trên Server gồm: name, size, owner, ID, creation date, and status information.
SP_HELPDB <i>database_name</i>	Hiển thị thông tin của database được chỉ định, gồm : name, size, owner, ID, creation date, and status information. Ngoài ra còn cho biết chi tiết các file dữ liệu và log file.
SP_SPACEUSED [ <i>objname</i> ]	Kích thước của current database hoặc table trong current database.

Ví dụ 1: Xem thông tin của CSDL TheThao

**EXEC SP\_HELPDB** TheThao

Ví dụ 2: Xem kích thước CSDL TheThao.

USE TheThao

## EXEC SP\_SPACEUSED

Ví dụ 3: Xem kích thước table tblCLB

USE TheThao

EXEC SP\_SPACEUSED tblCLB

### III- Tạo và quản lý Table :

Khi tạo table cần quan tâm đến các yếu tố trên các Field được tạo như :

- Key : Field đó là khóa hay không
- ID : Field đó có thuộc tính Identity hay không
- Column Name : Tên Field
- Data Type : Kiểu dữ liệu của Field
- Size : Kích thước lưu trữ
- Allow Null : Cho phép Null
- Default : Giá trị mặc định
- Identity : Field số có giá trị tăng tự động, với giá trị ban đầu là bao nhiêu (Identity Seed) và số gia(Identity Increment) là bao nhiêu.
- Tên Table bạn nên đặt với tiếp đầu ngữ là “tbl”.

Lưu ý:

- Luôn tạo khóa chính cho một bảng.
- Ràng buộc khóa ngoại nên được tạo sau khi đã tạo xong tất cả các bảng liên quan.

#### 1- Cú pháp lệnh tạo bảng:

```
Create table Table_name
(
    { Column_name      Data_type [null | not null]
                                     [default default_value ]
                                     [identity [( seed, increment)] ]
    } [...n]
    [, constraint constraint_name primary key ( Column_name [...n] ) ]
)
```

Ví dụ: Tạo bảng học sinh có khóa chính là (STT, Lop)

```
Create table HOCSINH
(
    STT          int not null,
    Lop          char(5) not null default '11A1' ,
    HoTen        nvarchar(30) not null,
    NgaySinh     datetime not null,
    DiaChi       nvarchar(100),
    constraint   pk_HS primary key (STT, Lop)
)
```

## 2- Thay đổi cấu trúc bảng – alter table:

Các trường hợp thay đổi cấu trúc bảng, đó là: Bổ sung thêm cột mới (*ADD Column*); Xóa cột; Sửa đổi định nghĩa của cột (*MODIFY Column*) và lệnh hủy bỏ RBTV trên cột hay trên cả bảng.

### Cú pháp chung:

**ALTER TABLE** table\_name

**ALTER COLUMN** column\_name {data\_type [(p[, s])] [NULL|NOT NULL]}

| **ADD** {[<column\_definition>] | ColName AS Expression } ,...

| **DROP COLUMN** column [,...n]

| [**WITH CHECK** | **WITH NOCHECK**] **ADD** {<table\_constraint> } [,...n]

| **DROP** [CONSTRAINT] constraint\_name

| {**CHECK** | **NOCHECK**} **CONSTRAINT** {ALL | constraint\_name[,...n]}

| {**ENABLE** | **DISABLE**} **TRIGGER** {ALL | trigger\_name[,...n]}

## 3- Sửa đổi kiểu dữ liệu hoặc kích thước của cột :

**ALTER TABLE** table

**ALTER COLUMN** ColName DataType [(p [, s])] [NULL | NOT NULL ]

Ví dụ: Sửa lại kích thước cột tên nhân viên (**Name**) thành 25 ký tự:

```
ALTER TABLE NhanVien ALTER COLUMN HoTenNV CHAR (25)
```

Chú ý: Không thể sửa cột :

- ☑ Có kiểu **text**, **image**, **ntext**, or **timestamp**
- ☑ Cột tính toán hoặc đã dùng trong cột tính toán
- ☑ Đã sử dụng trong các ràng buộc ngoại trừ chỉ thay đổi kích thước.
- ☑ Không được phép sửa đổi kích thước của cột cho nhỏ lại, và cũng không được phép thay đổi kiểu dữ liệu của cột, trừ trường hợp cột đó chưa có dữ liệu gì.
- ☑ Không thể sửa đổi cột hiện chứa giá trị NULL từ thuộc tính NULL thành NOT NULL.

#### 4- Thêm cột

**ALTER TABLE** table\_name **ADD** <Định nghĩa cột> [...]

Ví dụ: Thêm vài cột có ràng buộc:

**CREATE TABLE** ViDu ( CotA INT CONSTRAINT CotA\_un UNIQUE)

**ALTER TABLE** ViDu **ADD**

*/\* Thêm cột khóa chính \*/*

CotB INT IDENTITY CONSTRAINT CotB\_pk PRIMARY KEY,

*/\* Thêm cột tham chiếu với cột khác trên cùng table\*/*

CotC INT NULL

CONSTRAINT CotC\_fk REFERENCES ViDu(CotA),

*/\* Thêm cột với ràng buộc dạng thức của dữ liệu \*/*

CotD VARCHAR(16) NULL

CONSTRAINT CotD\_chk

CHECK (CotD IS NULL OR CotD LIKE "[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]" OR CotD LIKE "([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]"),

*/\* Thêm cột với giá trị mặc định \*/*

CotE DECIMAL(3,3) CONSTRAINT CotE\_defa DEFAULT .081

**EXEC SP\_HELP** ViDu

Chú ý: Trong trường hợp Table đã có dữ liệu, nếu bạn cần:

- **Thêm cột NOT NULL:** Phải thực hiện qua 3 bước: (i) thêm cột với thuộc tính NULL, (ii) điền đầy đủ các giá trị cho cột, (iii) đổi lại thuộc tính của cột thành NOT NULL.
- **Thêm cột mới và điền giá trị Default cho các dòng đang tồn tại trong bảng:**

Dùng DEFAULT với thuộc tính **WITH VALUES** để cung cấp giá trị cho mỗi dòng đang tồn tại trong bảng.

```
ALTER TABLE MyTable ADD AddDate smalldatetime NOT NULL  
CONSTRAINT AddDateDflt DEFAULT GetDate() WITH VALUES
```

#### 5- Đổi tên cột :

```
SP_RENAME 'TableName.OldColName', 'NewColName', 'COLUMN'
```

Ví dụ: SP\_RENAME 'customers.[contact title]', 'title', 'COLUMN'

#### 6- Xóa cột :

```
ALTER TABLE <tablename> DROP COLUMN ColName [...n]
```

Ví dụ : CREATE TABLE ViDuXoaCot ( CotA INT, CotB VARCHAR(20) NULL)

```
ALTER TABLE ViDuXoaCot DROP COLUMN CotB
```

```
EXEC sp_help ViDuXoaCot
```

Chú ý: Không thể xóa những cột:

- Đang dùng trong một Index.
- Có ràng buộc CHECK, FOREIGN KEY, UNIQUE, or PRIMARY KEY.
- Có chỉ định giá trị DEFAULT.
- Có chỉ định rule.

#### 7- Thêm RBTV cho bảng :

##### a. Khai báo ràng buộc trong lúc tạo bảng:

Cú pháp:

```
Create table Table_name  
( ...  
[, constraint Constraint_name
```

```
{ primary key (Column_name [...n])  
| unique ( Column_name [...n])  
| check ( logical_expression ) }  
[ [...n]  
)
```

Ví dụ

Create table SinhVien

```
(  
    MaSV      char(10) not null,  
    HoTen     nvarchar(30) not null,  
    Nam       tinyint,  
    CMND      char(10),  
    Khoa      char(5),  
    constraint pk_SV primary key (MaSV), constraint u_CMND unique (CMND),  
    constraint chk_Nam check (Nam > 0 and Nam <= 4)  
)
```

**b. Khai báo ràng buộc trên bảng đã tồn tại**

**ALTER TABLE** TableName

**[WITH CHECK | WITH NOCHECK] ADD { <TableConstraint> }[,...n]**

Trong đó

**<TableConstraint>::= [CONSTRAINT *constraint\_name* ]**

**PRIMARY KEY( *Col1* [ ,...*n* ] )**

**UNIQUE ( *Col1* [ ,...*n* ] )**

**FOREIGN KEY ( *Col1* [ ,...*n* ] ) REFERENCES *ref\_table* ( *ref\_col* [ ,...*n* ] )**

**[ ON DELETE { CASCADE | NO ACTION } ]**

**[ ON UPDATE { CASCADE | NO ACTION } ]**

**DEFAULT <Exp> FOR *Col* [ WITH VALUES ]**

**CHECK( *BT Điều kiện* )**

Ví dụ: ALTER TABLE SanPham

ADD CONSTRAINT SoTon\_defa **DEFAULT 0 FOR** SoTon

- **WITH NOCHECK:** Không kiểm tra ràng buộc đối với những dữ liệu đang có trên cột

ALTER TABLE ViDu WITH NOCHECK

ADD CONSTRAINT CotA\_chk CHECK (CotA > 1)

ALTER TABLE CTDH WITH NOCHECK

ADD CONSTRAINT SoLuong\_chk CHECK (SoLuong >=0)

**Chú ý:** Để sửa đổi RBTV, trước hết cần loại bỏ (DROP) chúng rồi sau đó bổ sung RBTV mới.

Ví dụ

*/\* giả sử đã tồn tại bảng KHOA( MaKhoa, ...) \*/*

Alter table SINHVIEN with check add constraint u\_CMND unique (CMND),  
constraint chk\_Nam check (Nam in (1, 2, 3, 4)),  
constraint fk\_SV\_maKhoa foreign key (Khoa),references KHOA(MaKhoa)

## 8- Hủy bỏ RBTV khỏi bảng

ALTER TABLE table **DROP CONSTRAINT** <Tên RBTV>

Ví dụ:

Alter table SINHVIEN  
  
drop constraint u\_CMND, chk\_Nam

## 9- Kích hoạt hay tạm ngưng kiểm tra các ràng buộc Foreign key và Check:

ALTER TABLE table

**{CHECK | NOCHECK} CONSTRAINT {ALL| ConstraintName[,...n]}**

Ví dụ 1 :-- *Disable the constraint and try again.*

ALTER TABLE ViDu NOCHECK CONSTRAINT CotA\_chk

-- *Reenable the constraint and try another insert, will fail.*

ALTER TABLE ViDu CHECK CONSTRAINT CotA\_chk

Ví dụ 2:

alter table SINHVIEN nocheck constraint u\_CMND, chk\_Nam

## 10- Xóa table :

Cú pháp:: **DROP TABLE** <tên bảng> [, ...]

### Ví dụ:

Xóa table trong database hiện hành: DROP TABLE titles1

Xóa table trong database khác : DROP TABLE pubs.dbo.authors2

Chú ý: Không thể xóa Table cha được tham chiếu bởi Table khác.

## 11- Đổi tên bảng:

Cú pháp: EXEC **SP\_RENAME** 'OldName', 'NewName'

Ví dụ: Đổi tên table **customers** thành **custs**.

EXEC sp\_rename 'customers', 'custs'

## 12- Xóa các dòng trên table và giải phóng vùng nhớ:

Cú pháp: **TRUNCATE TABLE** <TableName>

Ngữ nghĩa: Khác với DELETE FROM <tên bảng> được sử dụng để xóa bản ghi khỏi bảng nhưng vùng nhớ trên bộ nhớ thứ cấp (đĩa từ hoặc vật mang tin từ tính khác) vẫn không được giải phóng để dùng lại. Lệnh TRUNCATE sau khi đã xóa bỏ các bản ghi khỏi bảng thì vùng nhớ của các bản ghi này sẽ được thu hồi và cho phép các bảng khác sử dụng.

Ghi chú: Chỉ có người tạo ra bảng hoặc những người quản trị CSDL mới có quyền hạn TRUNCATE bảng.



## Bài tập

*Cho các CSDL như sau:*

**Bài 1:** Tạo CSDL QLTV gồm các table sau:

**NhomSach**(MaNhom char(5), TenNhom nvarchar(25))

**NhanVien**(MaNV char(5), HoLot nvarchar(25), TenNV nvarchar(10), Phai varchar(3), NgaySinh smalldatetime, DiaChi nvarchar(40))

**DanhMucSach**(MaSach char(5), TenSach nvarchar(40), TacGia nvarchar(20), MaNhom char(5), DonGia Numeric(5), SLTon numeric(5))

**HoaDon**(MaHD char(5), NgayBan SmallDatetime, MaNV char(5))

**ChiTietHoaDon**(MaHD char(5), MaSach char(5), SoLuong numeric(5))

**Bài 2:** Tạo CSDL QLGD gồm các table sau:

**Khoa**( MaK , TenK)

**ChucVu**( MaCV, TenCV, HeSoPC)

**QueQuan**( MaQQ, TenQQ)

**Mon**( MaMon, TenMon, SoTiet, HinhThucThi)

**GiaoVien**( MaGV, HoTen, Phai, NS, HSL, MaK, MaQQ, MaCV)

**GiangDay**( MaGV, MaMon)

**Bài 3:** Tạo CSDL QLBH gồm các table sau:

**Loai**( MaLoai, TenLoai)

**MatHang**( MaMH, TenMH, Dvt, MaLoai)

**NhanVien**( MaNV, HoTen, NgayBDLV)


**KhachHang**( MaKH, HoTen, SoDT, DiaChi)

**HoaDonCT**( MaHD, MaNV, MaKH, NgayHD , ThanhTien)

**HoaDon**( MaHD, MaMH, Dongia, Soluong)

**Bài 4:** Tạo CSDL QLĐT

**Khoa**( Mak, TenK)

**MonHoc**( MaM, TenMH, Sotiet) 

**CTDaoTao**(MaK, MaM, KhoaHoc)

**SinhVien**(MaSV, HoTen, MaK, KhoaHoc) 

**DangKy**(MaSV, MaM, DiemThi)

**Bài 5:** Tạo CSDL QLKH

**KhoaHoc**( MaKH, TenKH, BatDau, KetThuc)

**HocVien**( MaHV, Ho, Ten, NS, NgheNghiep)

**Giaovien**( MaGV, HoTen, NS, DiaChi)

**LopHoc**( MaLop, TenLop, MaKH, MaGV, SiSo, PhongHoc, LopTruong)

**BienLai**( MaKH, MaHV, MaLop, SoBL, Diem, KetQua, XepLoai, TienNop)

**Bài 6:** Tạo CSDL QLSV

**Lop**( MaLop, TenLop, SiSo)

**SinhVien**( MaSV, HoTen, NamSinh, MaLop)

**Mon**( MaMon, TenMon, Sdvht)

**Giaovien**( MaGV, HoTen, Phai, NamSinh, MaMon)

**Ketqua**( MaSV, MaMon, LanThi, Diem)

**Bài 7:** Tạo CSDL QLNV

**PhongBan**( MaPB, TenPB, NoiDat\_VP)

**NhanVien**( MaNV, HoTen, MaNVQL, NgayBDLV, LuongThang, PCThang, MaPB)

**Yêu cầu:**

1. Hãy tạo các CSDL bài tập 1,2,3,4 sử dụng câu lệnh SQL với gồm 3 file lưu trữ (file chính, file phụ và file log) với các thông số tự cho (xem ví dụ trong phần lý thuyết chương 2). Bài tập 5,6,7 sử dụng Enterprise Manager (Đồ họa).
2. Hãy tạo các table và các ràng buộc cho Bài tập 1,2,3,4 sử dụng câu lệnh SQL, Bài tập 5,6,7 sử dụng Enterprise Manager (Đồ họa).
3. Sử dụng Backup thực hiện công việc sao lưu dữ liệu cho các bài tập vừa tạo ở trên.

## CHƯƠNG 3 : CÁC PHÁT BIỂU CƠ BẢN CỦA TRANSACT-SQL

### I- Giới thiệu ngôn ngữ Transact-SQL :

T-SQL là ngôn ngữ thủ tục thể hệ thứ 3. Không giống như những ngôn ngữ lập trình khác, không thể dùng nó để tạo ra các chương trình ứng dụng độc lập. Các phát biểu của nó chỉ được thực hiện trong môi trường SQL Server với mục đích truy vấn và hiệu chỉnh dữ liệu trong CSDL quan hệ.

T-SQL có các phát biểu được phân loại như sau :

- **Data Control Language (DCL):** Chứa các lệnh điều khiển, phân quyền truy xuất dữ liệu.
- **Data Definition Language (DDL):** Dùng tạo, sửa xóa các đối tượng trong CSDL – như Database, Table, Index, Default, Procedure, Function, Schema, View, và Trigger
- **Data Manipulation Language (DML):** Chứa các lệnh thêm, sửa, xóa dữ liệu
- **Data Query Language (DQL) :** Chỉ chứa 1 phát biểu SELECT dùng truy vấn dữ liệu
- Các thành phần khác của ngôn ngữ như kiểu dữ liệu, biến, toán tử, hàm, các cấu trúc điều khiển và chú thích.

### II- Kiểu dữ liệu:

Các kiểu dữ liệu trong SQL gồm có các loại sau:

	Thực hàng
<p><b>Số Chính Xác - Exact Numerics</b></p> <p><b><u>Số nguyên:</u></b></p> <p><b>Bigint</b> (8 bytes) giá trị từ <math>-2^{63}</math> đến <math>2^{63}-1</math></p> <p><b>Int</b> : (4 bytes) giá trị từ <math>-2^{31}</math> đến <math>2^{31} - 1</math>.</p> <p><b>SmallInt</b> : (2 bytes) giá trị từ <math>2^{15}</math> đến <math>2^{15} - 1</math>.</p> <p><b>Tinyint</b> : (1 byte) giá trị từ 0 đến 255.</p> <p><b><u>Luận lý</u></b></p> <p><b>Bit</b> : có giá trị 0, 1 hoặc NULL.</p> <p><b><u>Số thực</u></b></p> <p><b>Decimal(n, d)</b> : <math>-10^{38} + 1</math> đến <math>10^{38} - 1</math>. <math>n \leq 38</math>; d mặc định bằng 0</p> <p><b>Numeric(n, d)</b> : Giống như decimal.</p> <p><b><u>Tiền tệ :</u></b></p>	

<p><b>Money</b> : <math>-2^{63}</math> đến <math>2^{63} - 1</math>, độ chính xác đến 2 chữ số sau dấu thập phân..</p> <p><b>SmallMoney</b> : -214,748.3648 đến +214,748.3647, với độ chính xác đến 2 chữ số sau dấu thập phân .</p>	
<p><b>Số gần đúng - Approximate Numerics</b></p> <p><b>Float[(n)]</b> : <math>-1.79E + 308</math> đến <math>1.79E + 308</math>. <b>n</b> có giá trị từ 1 – 53, là số bit dùng lưu phần định trị</p> <p><b>Real</b> : <math>-3.40E + 38</math> đến <math>3.40E + 38</math>.</p>	
<p><b>Ngày giờ</b></p> <p><b>Datetime</b> : 1-1- 1753 đến 31-12- 9999, độ chính xác 3/100 giây hay 3.33 milliseconds.</p> <p><b>Smalldatetime</b> : 1-1- 1900 đến 6-6- 2079, với độ chính xác là 1 phút.</p>	Bao trong dấu nhảy đơn.
<p><b>Chuỗi ký tự (không theo Unicode) - Character Strings</b></p> <p><b>Char(n)</b> : độ dài cố định, tối đa là 8000 ký tự.</p> <p><b>Varchar(n)</b>: độ dài không cố định, tối đa là 8000 ký tự.</p> <p><b>Text</b> : độ dài không cố định, tối đa là <math>2^{31} - 1</math> ký tự.</p>	Bao trong dấu nhảy đơn
<p><b>Chuỗi ký tự Unicode - Unicode Character Strings</b></p> <p><b>nChar(n)</b> : độ dài cố định, tối đa là 4000 ký tự.</p> <p><b>nVarchar(n)</b> : độ dài không cố định, tối đa là 4000 ký tự.</p> <p><b>nText</b> : độ dài không cố định, tối đa là <math>2^{30} - 1</math> ký tự.</p>	Bao trong dấu nhảy đơn và phải bắt đầu bằng chữ N: N'sssss'
<p><b>Số nhị phân - Binary Strings</b></p> <p><b>Binary(n)</b> : độ dài cố định (tối đa 8000 bytes).</p> <p><b>Varbinary(n)</b> : độ dài thay đổi (tối đa 8000 bytes).</p> <p><b>Image</b> : độ dài thay đổi (tối đa <math>2^{31}-1</math> bytes).</p>	0Xnnnn
<p><b>Other Data Types</b></p> <p><b>Cursor</b> : kiểu con trỏ</p> <p><b>Sql_Variant</b> : Nhận giá trị của nhiều kiểu dữ liệu khác nhau trong SQL Server ngoại trừ các kiểu text, ntext, timestamp, và sql_variant.</p> <p><b>Table</b> : dùng lưu trữ các tập dữ liệu cho lần xử lý sau.</p> <p><b>Timestamp</b> : kiểu số (binary(8) hay varbinary(8)). Cột khai báo kiểu này sẽ được tự động cập nhật với giá trị phân biệt mỗi khi thêm một mẫu tin mới..</p> <p><b>UniqueIdentifier</b> : A globally unique identifier (GUID).</p>	

**Chú ý:** Kiểu Text, nText, và Image không dùng cho biến cục bộ.

**Trực hằng (Literals):** bao gồm hằng số (*Number - Ví dụ. 1234.56 1234.56*), hằng văn bản (*Text*) và ngày giờ trong cặp dấu nháy đơn (*Ví dụ. 'Nguyễn Hồng Anh'* ) và hằng logic (*True* hay *False*)

### III- Truy xuất dữ liệu : (DQL *data query language*)

SELECT <danhsách các cột>

[INTO *new\_table* ]

[FROM <bảng nguồn>]

[WHERE <điều kiện chọn dòng> ]

[GROUP BY <danhsách cột khóa phân nhóm>]

[HAVING <điều kiện chọn nhóm> ]

[ORDER BY <danhsách cột khóa sắp xếp> [ASC | DESC ]]

Cú pháp đầy đủ của phát biểu Select khá phức tạp. Chúng ta sẽ lần lượt làm rõ từng phần của phát biểu này.

#### 1- From clause :

Chỉ định nguồn dữ liệu. Để truy vấn thông tin từ nhiều bảng, sử dụng phép kết giữa các bảng trong mệnh đề FROM: (theo ANSI)

<table\_source> <join\_type> <table\_source> **ON** <search\_condition>

<join\_type>: Gồm các phép kết :

INNER JOIN, LEFT [OUTER] JOIN; RIGHT [OUTER] JOIN;

FULL [OUTER] JOIN : kết hợp Left Join và Right Join

CROSS JOIN : không có mệnh đề ON và là phép tích Cartesian.

<search\_condition>: chỉ định điều kiện liên kết giữa 2 bảng.

#### 2- Select clause :

SELECT [DISTINCT] [TOP n [PERCENT] [ WITH TIES] ] <column\_list>

[FROM <table\_list>]

- **DISTINCT** : Chỉ hiện những dòng có dữ liệu phân biệt. Mặc định các dòng trùng dữ liệu đều được xuất hiện trong kết quả.



**Tên biến** (*Variable*)

**c- Các hàm toán học:**

ABS( <i>x</i> )	: Trị tuyệt đối của <i>x</i>
SQRT( <i>x</i> )	: Căn bậc hai của <i>x</i>
SQUARE( <i>x</i> )	: $x^2$
POWER( <i>y</i> , <i>x</i> )	: $y^x$
LOG( <i>x</i> )	: Logarit tự nhiên của <i>x</i>
EXP( <i>x</i> )	: Hàm mũ cơ số <i>e</i> của <i>x</i> : $e^x$ .
SIGN( <i>x</i> )	: Lấy dấu của số <i>x</i> (-1: $x < 0$ , 0: $x = 0$ , +1: $x > 0$ )
ROUND( <i>x</i> , <i>n</i> )	: Làm tròn tới <i>n</i> số lẻ.
CEILING( <i>x</i> )	: Số nguyên nhỏ nhất nhưng lớn hơn <i>x</i>
FLOOR( <i>X</i> )	: Số nguyên lớn nhất nhưng nhỏ hơn <i>x</i>
... và các hàm lượng giác: SIN, COS, TAN, ASIN, ACOS, ATAN ...	

**d- Các hàm xử lý chuỗi ký tự:**

ASCII( <i>ch</i> )	: Mã ASCII của ký tự <b>ch</b>
CHAR( <i>n</i> )	: Ký tự có mã ASCII là <b>n</b>
LOWER( <i>str</i> )	: Trả về chuỗi chữ thường
UPPER( <i>str</i> )	: Trả về chuỗi chữ hoa
LTRIM( <i>str</i> )	: Trả về chuỗi không có dấu cách bên trái
RTRIM( <i>str</i> )	: Trả về chuỗi không có dấu cách bên phải
LEFT( <i>str</i> , <i>n</i> )	: Lấy <i>n</i> ký tự phía trái của dãy <i>str</i> .
RIGHT( <i>str</i> , <i>n</i> )	: Lấy <i>n</i> ký tự phía phải của dãy <i>str</i> .
SUBSTRING( <i>str</i> , <i>start</i> , <i>n</i> )	: Lấy <i>n</i> ký tự của dãy <i>str</i> kể từ vị trí <i>start</i> trong dãy.
REPLACE( <i>str1</i> , <i>str2</i> , <i>str3</i> )	: thay thế tất cả <i>str2</i> trong <i>str1</i> bằng <i>str3</i> .
STUFF( <i>str1</i> , <i>start</i> , <i>n</i> , <i>str2</i> )	: Thay thế <i>n</i> ký tự trong <i>str1</i> từ vị trí <i>start</i> bằng chuỗi <i>str2</i> .

STR( x, len [, Dec]): Chuyển số x thành chuỗi.

**e- Các hàm xử lý ngày tháng và thời gian:**

GETDATE(): Cho ngày tháng năm hiện tại (Oracle: SYSDATE)

DAY(dd): Cho số thứ tự ngày trong tháng của biểu thức ngày dd.

MONTH(dd): Cho số thứ tự tháng trong năm của biểu thức ngày dd.

YEAR(dd): Cho năm của biểu thức ngày dd.

DATEPART(datepart, date) DATEADD(datepart,number, date) DATEDIFF(datepart, date1, date2)	Datepart	Abbreviations
	Year	yy, yyyy
	Quarter	qq, q
	Month	mm, m
	Day of year	dy, y
	Day of Month	dd, d
	Week of year	wk, ww
	Weekday	dw
	Hour	hh
	Minute	mi, n
	Second	ss, s
	Millisecond	Ms

**f- Các hàm chuyển đổi kiểu giá trị:**

**CAST**(expression AS data\_type[(length)])

**CONVERT** (data\_type[(length)], expression [, style])

Style : Dạng thức kiểu ngày mà bạn muốn khi chuyển đổi dữ liệu kiểu datetime hoặc smalldatetime tới kiểu ký tự (nchar, nvarchar, char, varchar, nchar, or nvarchar), Hoặc dạng chuỗi mà bạn muốn khi chuyển dữ liệu kiểu số (float, real, money, or smallmoney) sang kiểu ký tự(nchar, nvarchar, char, varchar, nchar, or nvarchar).

Trong bảng, 2 cột bên trái biểu diễn dạng giá trị datetime hoặc smalldatetime chuyển sang character. Cộng thêm 100 cho giá trị style để được dạng năm 4 chữ số.



Without century (yy)	With century (yyyy)	Standard	Input/Output**
-	0 or 100 (*)	Default	mon dd yyyy hh:miAM (or PM)
1	101	USA	mm/dd/yy
2	102	ANSI	yy.mm.dd
3	103	British/French	dd/mm/yy
4	104	German	dd.mm.yy
5	105	Italian	dd-mm-yy
6	106	-	dd mon yy
7	107	-	mon dd, yy
8	108	-	hh:mm:ss
-	9 or 109 (*)	Default+milliseconds	mon dd yyyy hh:mi:ss:mmmAM (or PM)
10	110	USA	mm-dd-yy

Ví dụ: `Select Conver(Char(8), GetDate(), 2)` -- *kết quả : 04.09.16*

Giá trị Style để chuyển kiểu **float** hay **real** sang kiểu ký tự.

- 0 (the default): Tối đa 6 chữ số, dùng trong ký hiệu khoa học.
- 1 luôn luôn 8 chữ số. Luôn dùng trong ký hiệu khoa học.
- 2 luôn luôn 16 chữ số. Luôn dùng trong ký hiệu khoa học..

In the following table, the column on the left represents the style value for **money** or **smallmoney** conversion to character data.

0 (the default) Không dấu phẩy phân cách hàng ngàn; có 2 chữ số thập phân.

Ví dụ: 4235.98.

1 Có dấu phân cách hàng ngàn và 2 chữ số thập phân;

Ví dụ: 3,510.92.

2 Không dấu phẩy phân cách hàng ngàn, ; có 4 chữ số thập phân.

Ví dụ: 4235.9819.

**g- Một số hàm hệ thống:**

ISDATE(variable | column name): Kiểm tra dạng ngày hợp lệ. Trả về 1 nếu hợp lệ và 0 nếu không hợp lệ.

ISNUMERIC(variable | column name): Kiểm tra dạng số hợp lệ. Trả về 1 nếu hợp lệ và 0 nếu không hợp lệ.

ISNULL(expression, value) : Trả về giá trị value nếu expression có giá trị NULL, ngược lại trả về giá trị của expression. Giá trị trả về cùng kiểu với expression.

NULLIF(exp1, exp2) : Trả về giá trị NULL nếu exp1 = exp2.

COALESCE(exp1, exp2, ..., expN) : trả về biểu thức khác NULL đầu tiên.

@@ROWCOUNT: Trả về số dòng (kiểu integer) trả về bởi phát biểu cuối cùng.

Ví dụ 1: cập nhật dữ liệu với UPDATE và dùng @@ROWCOUNT để xác định số dòng đã được thay đổi.

```
UPDATE authors SET au_lname = 'Jones'
```

```
WHERE au_id = '999-888-7777'
```

```
IF @@ROWCOUNT = 0
```

```
print 'Warning: No rows were updated'
```

**@@ERROR** : Trả về mã lỗi (integer) nếu có của phát biểu T-SQL cuối cùng. Trả về số 0 nếu không có lỗi.

Ví dụ 2: Dùng @@ERROR để kiểm tra vi phạm ràng buộc (error #547) trong phát biểu cập nhật.

```
USE pubs
```

```
GO
```

```
UPDATE authors SET au_id = '172 32 1176'
```

```
WHERE au_id = "172-32-1176"
```

```
IF @@ERROR = 547
```

```
print "Vi phạm ràng buộc"
```

**3- Where clause:**

Chọn lựa những mẫu tin theo điều kiện.

**Cú pháp:** WHERE <search\_condition> | <old\_outer\_join>

<old\_outer\_join> ::= column\_name { \*= | =\* } column\_name

**3.1 Các phép toán so sánh:** <, <=, !<, >, >=, !>, =, <> hay !=.

Kết quả phép so sánh là giá trị lôgic (*True* hoặc *False*).

**3.2 Các phép toán lôgic:** NOT, AND (*conjunction*), OR (*disjunction*).

Kết quả các phép toán lôgic là một giá trị lôgic.

**3.3 Các phép toán phạm vi:**

IS [NOT] NULL

[NOT] IN (< danh sách giá trị >)

[NOT] BETWEEN <Min> AND <Max>

[NOT] LIKE 'Mẫu văn bản' --Dùng ký hiệu thay thế là dấu % và dấu (\_)

[NOT] EXISTS(SubQuery) : Trả về True nếu tồn tại ít nhất 1 mẫu tin.

**3.4 <phép so sánh> [<lượng từ>] (SubQuery):**

- <Phép so sánh> có thể là các phép so sánh số học (>, >=, <, <=, <>, =) hoặc phép toán tập hợp IN, LIKE hoặc NOT LIKE.
- <Lượng từ> có thể là ALL, ANY (hoặc SOME). Phép so sánh = ANY có thể được thay tương đương bằng phép toán IN; phép so sánh <> ALL có thể thay tương đương bằng phép toán NOT IN.

**3.5 Phép kết ngoài (outer join):** column\_name { \*= | =\* } column\_name

Trong đó, phép toán (\*= ) Left outer join và phép toán (=\*) Right outer join.

**Chú ý:** Để định lại thứ tự các phần của kiểu ngày phù hợp với hằng kiểu ngày trong biểu thức điều kiện. Sử dụng lệnh : **SET DATEFORMAT mdy | dmy | ymd | ydm | myd | dym**

**4- Order by clause:**

Mệnh đề ORDER BY dùng sắp xếp kết quả tìm được.

Cú pháp mệnh đề này là:

ORDER BY <têncột>|<biểuthức> [ASC | DESC], <têncột>|<biểuthức> [ASC | DESC], ...

## 5- Tổng hợp dữ liệu:

### 5.1 Các hàm tổng hợp:

**AVG**( [Distinct] Column\_name)

**Count**( \* )

**Count**([Distinct] Column\_name)

**Max**(Column\_name)

**Min**(Column\_name)

**Sum**([Distinct] Column\_name)

**Ví dụ:** Cho biết số mặt hàng đã bán trong ngày cuối cùng:

```
SELECT COUNT( DISTINCT MaMH) FROM HoaDon, CTHD
WHERE HoaDon.NgayHD IN (SELECT MAX(NgayHD) FROM HoaDon)
AND HoaDon.MaHD = CTHD.MaHD
```

### 5.2 Group by clause:

**Cú pháp:** GROUP BY [ALL] < danh sách cột khóa phân nhóm >

Phân nhóm mẫu tin theo giá trị của các cột làm chuẩn phân nhóm, mỗi nhóm dữ liệu trả về một dòng tổng hợp.

Mệnh đề GROUP BY ALL trả về tất cả các nhóm, kể cả những nhóm không thỏa mãn điều kiện của mệnh đề WHERE.

**Chú ý:** Mệnh đề GROUP BY phải chứa tất cả các cột không tổng hợp có trong mệnh đề SELECT.

### 5.3 Having clause:

Cú pháp: **HAVING** < điều kiện chọn nhóm >

Dùng chỉ định những dòng tổng hợp xuất hiện phải thỏa mãn điều kiện chỉ định.

## 6- Toán tử union:

Dùng kết hợp các kết quả của 2 hay nhiều truy vấn vào cùng một kết quả.

```
SELECT < danh sách các cột >
FROM < bảng nguồn >
```

[WHERE <điều kiện chọn dòng> ]

[GROUP BY <danh sách cột khóa phân nhóm>]

[HAVING <điều kiện chọn nhóm> ]

**UNION [ALL]**

SELECT <danh sách các cột>

FROM <bảng nguồn>

[WHERE <điều kiện chọn dòng> ]

[GROUP BY <danh sách cột khóa phân nhóm>]

[HAVING <điều kiện chọn nhóm> ]

[ORDER BY <danh sách cột khóa sắp xếp> [ASC | DESC ]]

Chú ý :

- Các kết quả truy vấn phải cùng số cột, cùng thứ tự và cùng kiểu dữ liệu tương ứng từng cột.
- Bảng kết quả có tên cột được tạo từ Select đầu tiên.
- Mệnh đề ORDER BY chỉ cho phép đứng cuối trong lệnh UNION
- Từ khóa ALL : dùng chỉ định hiển thị cả những dòng trùng dữ liệu. Nếu không có từ khóa ALL thì chỉ hiện các dòng phân biệt.

#### **IV- Data manipulation language (DML)**

##### **1- Chèn mẫu tin mới :**

###### **a- Chèn trực tiếp một mẫu tin mới:**

Cú pháp: **INSERT INTO** <table name> [( column list)] **VALUES** (value list)

###### **b- Chèn dữ liệu từ các bảng :**

Cú pháp: **INSERT INTO** <table name> [(column list)] <SELECT Statement>

##### **2- Sửa dữ liệu:**

Cú pháp 1:

**UPDATE** <table name> **SET** {<column name> = <value>} [,...n ]

[FROM { < table\_source > } [ ,...n ] ]

[WHERE <conditions>;

Cú pháp 2:

UPDATE <table name> SET {<column name> = <Select statement>}[,...n]

[FROM { < table\_source > } [ ,...n ] ]

[WHERE <conditions>;

Ví dụ :

- Cập nhật số tồn và gấp đôi đơn giá của các mặt hàng có mã loại hàng bằng 1

UPDATE MatHang SET SoTon = IsNull(SoTon, 0) + 100 , DonGia = DonGia \* 2

WHERE MALH = 1

- Cập nhật số tồn của các mặt hàng đã bán trong ngày

UPDATE MatHang SET SoTon = SoTon - SL

FROM CTHD, HoaDon , MatHang

WHERE CTHD.SoHD = HoaDon.SoHD

And CTHD.MaMH = MatHang.MaMH

AND HoaDon.NgayHD = (SELECT MAX(HoaDon.NgayHD) FROM HoaDon)

- Cập nhật tiền Hóa đơn của các hóa đơn bán trong ngày

UPDATE HoaDon

SET TienHD = (SELECT SUM(SL\*DGBan) FROM CTHD

WHERE HoaDon.MaHD = CTHD.MaHD)

Where HoaDon.NgayHD IN (SELECT MAX(NgayHD) FROM HoaDon)

- Tăng đơn giá của 10 mặt hàng có đơn giá thấp.

UPDATE MatHang SET DonGia = DonGia \* 1.1

FROM MatHang,

(SELECT TOP 1 \* FROM MatHang ORDER BY DonGia) AS t1

WHERE MatHang.MaMH = t1.MaMH

Hay có thể bỏ tên table MatHang trong mệnh đề FROM

UPDATE MatHang SET DonGia = DonGia \* 1.1

FROM (SELECT TOP 1 \* FROM MatHang ORDER BY DonGia) AS  
t1 WHERE MatHang.MaMH = t1.MaMH

### 3- Xóa mẫu tin:

Cú pháp 1: **DELETE FROM** <Table Name> [**WHERE** <conditions>];

Cú pháp 2: **DELETE** <Table Name> **FROM** <Join Table>  
**[WHERE** <conditions>];

Ví dụ :

- Xóa hóa đơn có mã hóa đơn bằng 1

Delete From CTHD WHERE MAHD = 1 And MaMH = 5

- Xóa các CTHD của các hóa đơn đã bán trong ngày

Delete CTHD

FROM CTHD, HoaDon

WHERE CTHD.SoHD = HoaDon.SoHD

AND HoaDon.NgayHD = (SELECT MAX(HoaDon.NgayHD) FROM HoaDon)

- Xóa MatHang đơn giá thấp.

DELETE MatHang

FROM MatHang,

(SELECT TOP 1 \* FROM MatHang ORDER BY DonGia) AS t1

WHERE MatHang.MaMH = t1.MaMH

Hay có thể bỏ tên table MatHang trong mệnh đề FROM

DELETE MatHang

FROM (SELECT TOP 1 \* FROM MatHang ORDER BY DonGia) AS t1

WHERE MatHang.MaMH = t1.MaMH

### 4- Tạo mới một bảng với các bộ giá trị lấy từ CSDL:

Cú pháp: **SELECT** <Column list> **INTO** <new table name> ....

## Bài tập

Hãy Restore CSDL bài tập chương 2 sau đó thực hiện các bài tập sau:

**Bài 1:** Sử dụng CSDL ở bài 1 thực hiện các yêu cầu sau:

- Sử dụng lệnh INSERT để đưa dữ liệu vào các bảng NHOMSACH, NHANVIEN, DANHMUCSACH.

**Bảng NhomSach**

	Mã nhóm	Tên nhóm
+	N001	Tin học
+	N002	Khoa học kỹ thuật
+	N003	Anh văn
+	N004	Y học Đông-Tây
+	N005	Chính trị-Pháp lý
+	N006	Văn học

**Bảng NhanVien**

	Mã nhân viên	Họ lót	Tên	Phái	Ngày sinh	Địa chỉ
+	NV001	Nguyễn Ngọc	Mai	Nữ	5/21/1975	23 CMT8, Q3
+	NV002	Lê Văn	Thành	Nam	5/23/1970	322 Hai Ba Trung, Q1
+	NV003	Nguyễn Thị	Thảo	Nữ	1/1/1975	44/5 Cao Thang, Q10
+	NV004	Nguyễn Văn	Mãi	Nam	12/6/1975	32 Dien Bien Phu, Q1
+	NV005	Nguyễn Bích	Chi	Nữ	10/20/1976	34 Tran Hung Dao, Q5
+	NV006	Ngô Quang	Minh	Nam	8/19/1978	56 Le Hong Phong Q11
+	NV007	Lê Ngọc	Thìn	Nam	7/23/1975	5 Lê Lợi Q1
+	NV008	Trần Thu	Uyên	Nữ	10/15/1975	1/10 Pham Ngoc Thach Q3
+	NV009	Mai Hoa	Đào	Nữ	5/3/1976	5 Le Lai QgoVap
+	NV010	Nguyễn Thảo	Anh	Nữ	2/5/1977	14 Nguyen Trai Q11

**Bảng DanhMucSach**

	Mã sách	Tên sách	Tác giả	Mã nhóm	Đơn giá	Số lượng tồn
+	S001	Tin Học A	Lê Anh	N001	25000	50
+	S002	Tin Học B	Trần Văn Một	N001	22000	40
+	S003	PreTest B	Nguyễn Hạnh	N003	2500	80
+	S004	Thuật Toán	Lê Thị Hương	N002	27500	10
+	S005	Nhiếp Ảnh	Trần Châu	N002	25000	40
+	S006	Thuốc Cổ Truyền	Lý Hải Nam	N004	13000	50
+	S007	Kỹ nghệ lạnh	Trần Cao	N002	32000	80
+	S008	Món quà Giáng Sinh	O.Henry	N006	100000	10
+	S009	Chờ đợi một tình yêu	Đoàn Thạch Biền	N006	135000	10
+	S010	Trí tuệ Nhân Tạo	Cao Tuấn	N001	28000	10



2. Dùng Enterprise Manager để nhập dữ liệu cho 2 bảng HOADON và CHITIETHOADON

**Bảng Hoadon**

	Mã hóa đơn	Ngày bán	Mã nhân viên
+	1	7/15/2008	NV001
+	10	7/20/2008	NV001
+	2	7/16/2008	NV001
+	3	7/16/2008	NV001
+	4	7/16/2008	NV001
+	5	7/16/2008	NV001
+	6	7/16/2008	NV001
+	7	7/18/2008	NV001
+	8	7/18/2008	NV001
+	9	7/20/2008	NV001

**Bảng Chitiethoadon**

4	S003	12
4	S004	2
4	S005	15
4	S007	11
4	S009	1
5	S007	11
5	S009	6
5	S010	1
6	S003	2
6	S005	6
6	S009	2
7	S003	8
7	S008	1
7	S010	2
8	S002	5
8	S005	9

**Bài 2: Sử dụng câu lệnh Insert into...values...để:**

Nhập dữ liệu cho các CSDL bài 2, bài 3, bài 4, bài 5, bài 6, bài 7 ứng với mỗi table nhập 5 mẫu tin .

**Bài 3:**

- Dùng chức năng Import/Export để đưa (nhớ kiểm tra kết quả sau mỗi lần thực hiện):

1. Tất cả các thông tin nhân viên có trong bảng Nhanvien ở CSDL QLTV ra thành tập tin NhanVien.txt.

2. Dữ liệu của các bảng Nhomsach, Nhanvien, DanhMucsach trong CSDLQLTV vào tập tin QUANLYSACH.MDB. Lưu ý: Tập tin QUANLYSACH.MDB phải tồn tại trên đĩa trước khi thực hiện Export.

3. Dữ liệu các bảng HOADON, CHITIETHOADON trong CSDL QLTV ra thành tập tin HD.XLS

- Dùng chức năng detach / attach; backup/restore để:

1. Thực hiện chức năng detach để xuất cơ sở dữ liệu ở trên.

2. Dùng chức năng attach để lấy lại cơ sở dữ liệu.

3. Tạo một file backup cho cơ sở dữ liệu.

4. Khôi phục lại cơ sở dữ liệu từ file backup trên.

**Bài 4:** Sử dụng lệnh ALTER TABLE và lệnh Truncate Table thực hiện tại CSDL QLTV để thực hiện các công việc sau:

1. Sao chép bảng nhân viên thành bảng ChamCongNhanVien
2. Chèn thêm cột NgayCong number(3) vào bảng ChamCongNhanVien
3. Chèn thêm cột DonGiaGiaCong number(5) vào bảng ChamCongNhanVien
4. Xóa các cột HoLot, Ten, Phai, NgaySinh, DiaChi trong bảng ChamCongNhanVien
5. Thêm vào bảng Hoadon cột NGAYLAPHD, kiểu char(10).
6. Hiệu chỉnh kiểu dữ liệu cho cột vừa thêm thành kiểu datetime.
7. Đổi tên cơ sở dữ liệu thành HOTENSV\_SOMAY
8. Đổi tên bảng CHAMCONGNHANVIEN thành BANGLUONG.
9. Xóa BANGLUONG khỏi cơ sở dữ liệu.
10. Detach cơ sở dữ liệu vừa tạo
11. Dùng lệnh Drop Table ... để xóa bảng Hoadon khỏi CSDL. Bạn có thực hiện được không? Giải thích.
12. Dùng lệnh Alter Table...Drop Constraint để xóa ràng buộc giữa bảng Hoadon và Chitiethoadon.
13. Xóa dữ liệu bảng Hoadon bằng lệnh Truncate table...

14. Xóa CSDL hiện tại.

15. Attach lại cơ sở dữ liệu bằng file .mdf ở trên

**Bài 5:** Sử dụng lệnh INSERT, UPDATE, DELETE thực hiện tại CSDL QLTV để thực hiện các công việc sau:

1. Cập nhật đơn giá 30000 cho mã sách S001
2. Cập nhật số lượng tồn = 50% số lượng tồn hiện có cho những cuốn sách thuộc mã nhóm N002
3. Cập nhật mã nhóm N001 trong bảng NHOMSACH thành N100? Bạn có cập nhật được hay không?. Vì sao?.
4. Tăng đơn giá các danh mục sách lên 3%
5. Giảm 10% đơn giá cho nhóm sách Anh văn.
6. Thêm vào cột GIABAN trong bảng CHITIETHOADON. Cập nhật giá bán bằng với đơn giá trong bảng DANHMUCSACH.
7. Thêm vào bảng NHOMSACH một bộ ('N007', 'Sức khỏe và đời sống')
8. Thêm vào bảng CHITIETHOADON với số hóa đơn 10, mã sách S001 với số lượng 5 và S002 với số lượng 20.
9. Thêm vào bảng DANHMUCSACH với mã sách S011, tên sách là 'Cánh đồng bất tận', tác giả 'Nguyễn Ngọc Tư', mã nhóm N006, đơn giá 45000, số lượng tồn 5.
10. Thêm vào bảng DANHMUCSACH với mã sách S011, tên sách là 'Hỏi đáp sức khỏe', tác giả 'Đỗ Hồng Ngọc', mã nhóm N008, đơn giá 25000, số lượng tồn 50. Bạn có thêm được không? Hãy quan sát lỗi (nếu có) và giải thích?.Hãy thực hiện công việc tiếp theo để thêm được bộ đã cho.
11. Thêm vào bảng Hoa don 10 bộ dữ liệu tùy ý, do các nhân viên có mã là NV002, NV003, NV004 lập.(lưu ý các ràng buộc dữ liệu, khóa chính, khóa ngoại).
12. Thêm vào bảng Chitiethoadon 30 bộ tùy ý.
13. Xóa nhóm sách 'N007' trong bảng NHOMSACH
14. Xóa Hoadon số 9 trong bảng Hoadon. Bạn có thực hiện được hay không?. Vì sao?

**Bài 6:** Sử dụng CSDL QLTV tạo query sử dụng lệnh SELECT với các dạng liên kết:

1. Inner Join
2. Left Outer Join
3. Right Outer Join

Kết quả hiển thị là thông tin của các cột sau:

MAHD	NGAYBAN	SOLUONG	MANHANVIEN

2. So sánh kết quả sau mỗi lần thực hiện dạng kết nối.

**Bài 7:** Sử dụng lệnh SELECT, UPDATE, DELETE thực hiện tại CSDL QLTV để thực hiện các công việc sau:

1. Liệt kê danh mục sách theo thứ tự mã sách.
2. Liệt kê nhân viên theo thứ tự tên.
3. Liệt kê những nhân viên nam theo thứ tự tên.
4. Liệt kê những nhân viên có tên là Mai.
5. Liệt kê những nhân viên có tên bắt đầu bằng ký tự t
6. Liệt kê những nhân viên có họ lót là thị hay văn
7. Liệt kê những nhân viên sinh năm 1975
8. Liệt kê những nhân viên sinh vào tháng 9
9. Liệt kê những cuốn sách có tên tác giả bắt đầu là Nguyễn
10. Liệt kê những sách có số lượng tồn < 100
11. Liệt kê những quyển sách có đơn giá từ 14000 đến 20000
12. Liệt kê những cuốn sách thuộc nhóm N001 và N002
13. Liệt kê những sách có đơn giá  $\geq 10000$  và số lượng tồn  $> 100$
14. Liệt kê những cuốn sách thuộc nhóm N001 và số lượng tồn từ 10 đến 20
15. Liệt kê hóa đơn theo thứ tự tăng dần của MaNV, nếu trùng mã nhân viên thì xếp theo ngày bán.
16. Hiển thị danh sách những sách thuộc ngành tin học gồm: mã sách, tên sách, mã nhóm.

17. Liệt kê sách thuộc nhóm tin học có số lượng tồn <10
18. Liệt kê các danh mục sách và tiền tồn vốn, xếp theo thứ tự giảm dần của tiền tồn. bảng kết quả gồm mã sách, tên sách, nhóm sách, tác giả, đơn giá, số lượng, tiền vốn = đơn giá x số lượng tồn
19. Danh sách các hóa đơn ứng với tổng tiền của từng hóa đơn
20. Danh sách các hóa đơn có ngày bán là ngày 15/7/2008
21. Danh sách các sách đã được bán, ứng với tổng số lượng, thành tiền
22. Danh sách các hóa đơn bán trong 20/7/2008, ứng với tổng số lượng, thành tiền
23. Danh sách các sách không bán được.
24. Danh sách các nhân viên chưa lập hóa đơn nào.
25. Danh sách các sách có số lượng bán nhiều nhất.
26. Danh sách các nhân viên ứng với tổng số tiền hóa đơn mà nhân viên ấy lập
27. Thống kê thành tiền ứng với mỗi nhóm sách và mỗi ngày
28. Cho biết nhân viên nào bán được nhiều sách nhất và số lượng là bao nhiêu.
29. Cho biết những nhân viên nào có cùng ngày sinh.
30. Cho biết nhân viên nào có tuổi lớn nhất.
31. Cho biết nhân viên nào có tuổi nhỏ nhất.
32. Cập nhật thông tin cho cột NGAYLAPHD trong bảng HOADON chính là thông tin của cột NGAYBAN.
33. Thêm vào bảng NHANVIEN cột LƯƠNG, kiểu float. Cập nhật lương=1000000 cho những nhân viên có lập hóa đơn.
34. Thêm vào cột PHUCAP trong bảng NHANVIEN. Cập nhật PHUCAP =50%\*LUONG cho những nhân viên bán được số lượng sách >100.
35. Thêm vào cột NGAYVAOLAM, kiểu datetime. Cập nhật ngày vào làm cho nhân viên = Năm sinh +18.
36. Cập nhật phụ cấp cho những nhân viên có thâm niên từ 5 năm trở lên (thâm niên được tính = năm hiện tại – năm vào làm).

**Bài 8:** Sử dụng lệnh SELECT, UPDATE, DELETE để thực hiện các công việc sau:

(Từ 1..6 sử dụng CSDL QLKH, từ 7..13 sử dụng CSDL QLNV, từ 14..28 sử dụng CSDL QLDT)

1. Cho biết kết quả cuối khoá của các học viên do giáo viên Trần văn Ngà dạy trong khoá học có mã số là 73
2. Cho biết danh sách lớp học là số lượng thật sự của lớp đó
3. Cho biết tên lớp học và số lượng học viên xếp loại khá hoặc giỏi trong lớp đó
4. Cho biết tên lớp học có sĩ số thật sự .
5. Cho biết họ tên địa chỉ của những học viên là giáo viên đồng thời là lớp trưởng .
6. Cho biết họ tên giáo viên và số lượng lớp mà người đó dạy .
7. Liệt kê những nhân viên nào làm thâm niên làm việc lâu nhất
8. Cho biết họ tên ,tên phòng ban và thu nhập hàng năm của các nhân viên
9. Liệt kê những nhân viên có chức vụ cao nhất của công ty
10. Liệt kê những nhân viên có chức vụ thấp nhất trong công ty
11. Liệt kê phòng ban , số lượng nhân viên của phòng ban có nhân viên nhiều nhất
12. Liệt kê tất cả các nhân viên có số lượng cao hơn mức lương trung bình của tất cả các nhân viên
13. Liệt kê những nhân viên có lương thấp nhất trong mỗi phòng ban .
14. Hãy liệt kê những sinh viên thuộc khoa công nghệ thông tin khoá 2012
15. Cho biết khoa công nghệ thông tin có bao nhiêu sinh viên .
16. Cho biết họ tên sinh viên thuộc khoa có mã là CNTT có nhiều hơn 3 môn dưới điểm trung bình .
17. Cho biết sinh viên của khoa nào ,khóa học nào phải học nhiều nhất trong toàn trường .
18. Cho biết những môn học nào hiện nay có nhiều sinh viên đang học nhất (Là những môn học mà các sinh viên đã học nhưng chưa thi ).
19. Cho biết những môn học nào thuộc khoa công nghệ thông tin khoá 2010 hiện nay chưa có sinh viên nào đăng ký học .

20. Cho biết những môn học nào được dạy nhiều hơn một khoa
21. Cho biết những môn học nào là những môn học chung cho toàn trường khoá 2005.  
Biết rằng môn học chung là những môn học được dạy trên tất cả các khoa.
22. Cho biết họ tên sinh viên được tốt nghiệp .
23. Hãy cho biết những môn học mà sinh viên có mã SV01 còn nợ so với chương trình đào tạo .
24. Liệt kê những sinh viên thuộc khoa công nghệ thông tin khoá 2010 học với số tín chỉ cao nhất .
25. Hãy cho biết điểm trung bình tích lũy của sv có tên Nguyễn Thanh Nam là bao nhiêu?
26. Liệt kê những sinh viên thuộc khoa CNTT khoá 2010 có điểm trung bình tích lũy là cao nhất .
27. Cho biết những sinh viên nào thuộc khoa công nghệ thông tin khoá 2010 có điểm môn lập trình căn bản là cao nhất ?
28. Cho biết sinh viên nào thuộc khoa cơ khí khoá 2010 hiện nay chưa tích lũy được tín chỉ nào?

## Chương IV : TẠO VIEW (BẢNG ẢO)

### I- Khái niệm :

Khung nhìn (View) là một bảng ảo, có cấu trúc như một bảng, khung nhìn không lưu trữ dữ liệu mà dữ liệu của nó được tạo ra khi sử dụng, khung nhìn là đối tượng thuộc CSDL. Khung nhìn được tạo ra từ câu lệnh truy vấn dữ liệu (lệnh Select), truy vấn từ một hoặc nhiều bảng dữ liệu

### II- Cách tạo và sử dụng khung nhìn :

#### 1- Tạo View bằng T-SQL:

```
CREATE VIEW ViewName [(ColName1, ...)]  
    [WITH ENCRYPTION]  
AS <Phát biểu select>  
    [WITH CHECK OPTION]
```

#### Ví dụ:

```
USE BanHang  
CREATE VIEW vwTienHD (MaHD, NgayLap, TienHD)  
    AS SELECT a.MaHD, NgayHD, SUM(SL*DG)  
        FROM HoaDon a, CTHD b  
        GROUP BY a.MaHD, NgayHD  
        HAVING SUM(SL*DG) > 0  
  
Go  
EXEC sp_helptext vwTienHD  
SELECT * FROM vwTienHD
```

```
Ví dụ:      CREATE VIEW vwTienHD (MaHD, NgayLap, TienHD)  
            AS  
            SELECT a.MaHD, NgayHD, SUM(SL*DG)  
                FROM HoaDon a, CTHD b WHERE Year(NgayHD) =2004  
                GROUP BY a.MaHD, NgayHD HAVING SUM(SL*DG) > 0  
  
            Union  
            SELECT a.MaHD, NgayHD, SUM(SL*DG)  
                FROM HoaDon a, CTHD b WHERE Year(NgayHD) =2005  
            GROUP BY a.MaHD, NgayHD HAVING SUM(SL*DG) > 0  
            EXEC SP_HelpText vwTienHD
```



## 2- Các hạn chế khi tạo Views

- Views không thực hiện trên các bảng tạm.
- Lệnh CREATE VIEW không thể kết hợp với các phát biểu T-SQL khác trong một gói
- Views không có hơn 1024 columns.
- Không chứa mệnh đề INTO.

## 3- Mã hóa View:

Để tránh trường sao chép định nghĩa View, bạn có thể mã hóa View bằng cách thêm từ khóa WITH ENCRYPTION trước từ khóa AS.

Ví dụ: **CREATE VIEW** vwTienHD (MaHD, NgayLap, TienHD)

**WITH ENCRYPTION**

**AS**

SELECT a.MaHD, NgayHD, SUM(SL\*DG)

FROM HoaDon a, CTHD b

GROUP BY a.MaHD, NgayHD

HAVING SUM(SL\*DG) > 0

GO

EXEC SP\_HELPTEXT vwTienHD

Kết quả là: *The Object comments have been encryption.*

*Chú ý: View đã mã hóa vẫn có thể sửa hoặc xóa nó.*

## 4- Sử dụng View để thay đổi dữ liệu:

Bạn có thể sử dụng View trong các phát biểu Insert, Update, Delete để thay đổi dữ liệu trên Table nguồn của View.

### • Các hạn chế :

- Không thể sử dụng lệnh INSERT hay DELETE trên View có sử dụng phát biểu INNER JOIN. (Trừ khi có sử dụng INSTEAD OF Trigger)
- Không thể sử dụng lệnh INSERT nếu View có khai báo cột tính toán hoặc chứa những cột có ràng buộc NOT NULL mà không khai báo giá trị mặc định

Ví dụ : **CREATE VIEW** vwCTHD (MaHD, MaMH, SoLuong)

**AS**

SELECT MaHD, MaSP, SL

```
FROM CTHD
```

```
Where MaHD = 10248
```

```
Go
```

```
Insert vwCTHD(MaHD,MaMH,SoLuong) Values(10248,1,20)
```

```
Insert vwCTHD(MaHD,MaMH,SoLuong) Values(10249,1,20)
```

- Bạn có thể ràng buộc dữ liệu được cập nhật phải thỏa mãn điều kiện Where của View bằng cách khai báo thêm Từ khóa : **WITH CHECK OPTION**

```
CREATE VIEW vwCTHD (MaHD, MaMH, SoLuong)
```

```
AS
```

```
SELECT MaHD, MaSP, SL
```

```
FROM CTHD
```

```
Where MaHD = 10248
```

```
WITH CHECK OPTION
```

```
Go
```

```
UPDATE vwCTHD SET soluong = soluong + 10
```

```
INSERT vwCTHD(MaHD,MaMH,SoLuong) VALUES(10248,1,20)
```

```
/* lệnh chèn sau đây sẽ bị lỗi : The attempted insert or update failed because  
the target view either specifies WITH CHECK OPTION or spans a  
view that specifies WITH CHECK OPTION and one or more rows  
resulting from the operation did not qualify under the CHECK  
OPTION constraint.
```

```
INSERT vwCTHD(MaHD,MaMH,SoLuong) VALUES(10249,1,20)
```

## 5- Thay đổi định nghĩa View:

Sử dụng phát biểu ALTER VIEW để thay đổi định nghĩa của View nhưng vẫn duy trì giấy phép cho View. Nếu bạn xóa View, sau đó tạo lại nó, bạn phải cấp lại các giấy phép cho nó.

### Cú pháp

```
ALTER VIEW ViewName [(column , ...)]
```

```
[WITH ENCRYPTION]
```

```
AS
```

```
select_statement
```

```
[WITH CHECK OPTION]
```

## 6- Xóa View:

**DROP VIEW <view\_name>**

Ví dụ:

USE SieuThi

Go

IF EXISTS (SELECT TABLE\_NAME FROM  
**INFORMATION\_SCHEMA.VIEWS**

WHERE TABLE\_NAME = 'vwCTHD')

DROP VIEW vwCTHD

GO

CREATE VIEW vwCTHD (MaHD, MaMH, SoLuong)

AS

SELECT MaHD, MaSP, SL

FROM CTHD

Where MaHD = 10248

## Bài tập

Hãy Restore các CSDL đã tạo ở chương 1 sau đó thực hiện các bài tập sau

**Bài 1:** Sử dụng CSDL QLBH tạo các view sau:

1. Cho biết loại hàng nào có mặt hàng nhiều nhất
2. Cho biết những nhân viên có thâm niên lâu nhất
3. Cho biết những mặt hàng bán chạy nhất trong năm nay.
4. Cho biết những loại hàng nào chưa có mặt hàng
5. Cho biết những nhân viên nào chưa từng lập hóa đơn
6. Cho biết những nhân viên nào lập hóa đơn nhiều nhất trong tháng 1 năm 2012
7. Cho biết thành tiền của hóa đơn số 9
8. Cho biết khách hàng Trần Thị Thanh Thúy đã đến mua hàng trong tháng 1 năm 2012 bao nhiêu lần
9. Cho biết khách hàng Lê Thị Cẩm Duyên đã tốn bao nhiêu tiền để mua hàng .
10. Cho biết những mặt hàng thuộc loại đường .
11. Cho biết ngày nào hóa đơn được lập ra nhiều nhất

**Bài 2:** Sử dụng CSDL QLTV tạo các view sau:

1. Tạo một view có tên là VWSACH\_1, chính là danh sách các cuốn sách, được sắp xếp theo tên sách.
2. Lần lượt tạo view có tên là VWSACH\_2, VWSACH\_3 thông tin giống như VWSACH\_1 nhưng có thêm từ khóa WITH ENCRYPTION.

3. Thực hiện các yêu cầu sau: Nhớ kiểm tra và cho nhận xét.

- Dùng lệnh INSERT...VALUES... chèn mẫu tin sau vào bản NHOMSACH:

MaNhom	TenNhom
N007	Kỹ thuật trồng trọt

Dùng lệnh Insert ... values chèn mẫu tin sau thông qua VWSACH\_1. Bạn có chèn được hay không?. Vì sao?

MaSach	TenSach	TacGia	MaNhom	DonGia	SLTon
S111	Đèn không hắt nắng	Dzunichi Watanabe (Cao Xuân Hạo dịch)	N006	55000	45

- Dùng lệnh ALTER để hiệu chỉnh VWSACH\_1 không có sắp xếp theo tên sách. Sau đó, chèn mẫu tin trên vào bảng danh mục sách thông qua view này.

- Chèn thêm các mẫu tin sau vào bảng danh mục sách thông qua VWSACH\_1 đã hiệu chỉnh ở trên:

MaSach	TenSach	TacGia	MaNhom	DonGia	SLTon
S112	Kỹ thuật trồng hoa phong lan	Nguyễn Lâm Hùng	N007	45000	35
S113	Kỹ thuật chăm sóc hoa mai	Lê Xuân A	N007	35000	15
S114	Kỹ thuật chăm sóc cây cam	Trần Ha	N007	24000	12

- Dùng sp\_helptext tênview để lần lượt xem text của các view VWSACH\_1, VWSACH\_2, VWSACH\_3

- Dùng lệnh Alter Table để hiệu chỉnh kiểu dữ liệu của cột DONGIA thành kiểu money, có hiệu chỉnh được không? Tại sao? Muốn hiệu chỉnh được thì phải như thế nào? Bạn thử thực hiện.

4. Tạo view có tên là VWSACH\_4, chỉ có những cuốn sách thuộc nhóm Kỹ thuật trồng trọt, với lựa chọn WITH CHECK OPTION.

5. Dùng câu lệnh INSERT ... VALUES ... để chèn mẫu tin sau vào bảng danh mục sách thông qua viewVWSACH\_4. Bạn có chèn được không? Tại sao?

MaSach	TenSach	TacGia	MaNhom	DonGia	SLTon
S115	Quê Mẹ	Thanh Tịnh	N006	35000	7

6. Bạn hãy chèn mẫu tin trên thông qua VWSACH\_1.

7. Dùng lệnh DELETE thông qua VWSACH\_1 để xóa mẫu tin vừa chèn ở trên.

## CHƯƠNG V: T-SQL NÂNG CAO

### I- Khai báo và sử dụng biến

#### 1- Biến cục bộ

- Là một đối tượng có thể chứa giá trị thuộc một kiểu dữ liệu nhất định, tên biến bắt đầu bằng một ký tự @.
- Biến cục bộ có giá trị trong một query batch hoặc trong một thủ tục thường trú (stored procedure) hoặc hàm (function).
- Khai báo biến cục bộ bằng lệnh declare: cung cấp tên biến và kiểu dữ liệu:

**Declare** tên\_biến Kiểu\_dữ\_liệu

**Ví dụ:**

Declare @MaSinhVien char(10)

Declare @HoTen nvarchar(30)

Declare @Sum float, @Count int

- Để gán giá trị cho một biến cục bộ dùng lệnh set. Giá trị gán cho biến phải phù hợp với kiểu dữ liệu của biến.

**Set** tên\_biến = giá\_trị

**Set** tên\_biến = tên\_biến

**Set** tên\_biến = biểu\_thức

**Set** tên\_biến = kết\_quả\_truy\_vấn

**Ví dụ:**

Set @MaLop = 'TH2001'

Set @SoSV = (select count (\*) from SinhVien)

Set @MaLop = 'TH'+Year(@NgayTuyenSinh)

Đưa kết quả truy vấn vào biến:

SV(MaSV: int; HoTen: nvarchar(30), Tuoi int)

Select @Var1 = HoTen, @Var1 = Tuoi from SV

where MaSV = 1

Lưu ý: nếu câu truy vấn trả về nhiều dòng, các biến chỉ nhận giá trị tương ứng của dòng đầu tiên.

## 2- Biến toàn cục

- Là các biến hệ thống do SQL Server cung cấp, tên biến bắt đầu bằng 2 ký tự @
- SQL tự cập nhật giá trị cho các biến này, người sử dụng không thể gán giá trị trực tiếp.
- Một số biến hệ thống thường dùng
  - + @@error: thông báo mã lỗi, nếu @@error = 0: thao tác thực hiện thành công.
  - + @@rowcount: cho biết số dòng bị ảnh hưởng bởi lệnh cuối (insert, update, delete).
  - + @@trancount: cho biết số giao dịch đang hoạt động trên kết nối hiện tại.
  - + @@fetch\_status: cho biết thao tác lấy dữ liệu từ cursor có thành công không.

## II- Cấu trúc điều khiển

### 1- Lệnh If...else

- Chức năng: xét điều kiện để quyết định những lệnh T-SQL nào sẽ được thực hiện
- Cú pháp:

If biểu\_thức\_điều\_kiện

Lệnh| Khối\_lệnh

[Else Lệnh| Khối\_lệnh]

Khối lệnh là một hoặc nhiều lệnh nằm trong cặp từ khóa begin...end

**Ví dụ:** xét 2 lược đồ quan hệ (LĐQH)

**HocPhan**( MaHP, TenHP, SiSo)

**DangKy**( MaSV, MaHP)

Viết lệnh để thêm một đăng ký mới cho sinh viên có mã số 001 vào học phần HP01 (giả sử học phần này đã tồn tại trong bảng HocPhan).

Declare @SiSo int

select @SiSo = SiSo from HocPhan

where MaHP=' HP01'

if @SiSo < 50

Begin

```
insert into DANG_KY(MaSV, MaHP) values('001', 'HP01')  
  
print N'Đăng ký thành công'  
  
End Else  
  
print N'Học phần đã đủ SV'
```

## 2- Lệnh While

- Chức năng: thực hiện lặp lại một đoạn lệnh T-SQL khi điều kiện còn đúng.
- Cú pháp:

While biểu\_thức\_điều\_kiện

Lệnh| Khối lệnh

- Có thể sử dụng Break và Continue trong khối lệnh của while

Break: thoát khỏi vòng while hiện hành.

Continue : trở lại đầu vòng while, bỏ qua các lệnh sau đó.

**Ví dụ:** Xét lược đồ quan hệ SinhVien(MaSV int, HoTen nvarchar(30))

Viết lệnh xác định một mã sinh viên mới theo qui định: mã sinh viên tăng dần, nếu có chỗ trống thì mã mới xác định sẽ chèn vào chỗ trống đó. Chẳng hạn, nếu trong bảng SinhVien đã có các mã sinh viên 1, 2, 3, 7 mã sinh viên mới là 4.

Giải:

Declare @STT int

Set @STT = 1

While exists(select \* from SV where MaSV = @STT)

set @STT = @STT+1

Insert into SV(MaSV, HoTen) values(@STT, 'Nguyen Van A')

## 3- Lệnh Case

- Chức năng: kiểm tra một dãy các điều kiện và trả về kết quả phù hợp với điều kiện đúng. Lệnh case được sử dụng như một hàm trong câu select.
- Cú pháp: Có hai dạng:

**Dạng 1 (simple case):**

Case *Biểu\_thức\_đầu\_vào*

When *Giá\_trị* then *kết\_quả*

[...n]

[ Else *kết\_quả\_khác*]

End

**Dạng 2 (searched case):**

Case

When *biểu\_thức\_điều\_kiện* then *kết\_quả*

[...n]

[ Else **kết\_quả\_khác**]

End

**Ví dụ:** xét LĐQH NHAN\_VIEN( MaNV, HoTen, NgaySinh, CapBac, Phai)

Cho biết những nhân viên đến tuổi nghỉ hưu biết rằng tuổi về hưu của nam là 60, của nữ là 55).

Giải:

```
select * from NHAN_VIEN
      where datediff(yy, NgaySinh, getdate()) > =
```

```
Case Phai
  when 'Nam' then 60
  when 'Nu' then 55
```

End

Cho biết mã NV, họ tên và loại nhân viên (cấp bậc <=3: bình thường, cấp bậc = null: chưa xếp loại, còn lại: cấp cao).

Giải:

```
Select MaNV, HoTen, 'Loai' = Case
  when CapBac <= 3 then 'Bình Thuong'
  when CapBac is null then 'Chua xep loai'
  else 'Cấp Cao' End
From NhanVien
```



### III- Thủ tục thường trú (Stored Procedures)

#### 1- Khái niệm

Thủ tục thường trú (Stored Procedures - SP) chứa các lệnh T\_SQL. Tương tự như một thủ tục trong các ngôn ngữ lập trình, SP trong SQL Server có thể truyền tham số, có tính tái sử dụng. Các thủ tục này được dịch và lưu trữ thành một đối tượng trong CSDL.

##### Ý nghĩa:

- Tính tái sử dụng, tính uyển chuyển nhờ hệ thống tham số.
- Khi biên dịch SP, các lệnh trong của nó được tối ưu hóa nó sao cho thực thi hiệu quả nhất. Kết quả tối ưu hóa được lưu bền vững. Khi gọi thực thi thủ tục không cần biên dịch và tối ưu hóa lại lời gọi thủ tục tiết kiệm thời gian và tài nguyên hơn khối lệnh tương đương thân thủ tục.
- Trong ứng dụng triển khai theo môi trường client/server, client gửi lời gọi SP lên server thì chiếm đường truyền ít hơn rất nhiều lần so với việc gửi khối lệnh tương đương trong thân thủ tục. Giảm khối lượng thông tin trao đổi khi ứng dụng gửi yêu cầu thực hiện công việc về cho server do đó tránh nghẽn đường truyền, giảm trì trệ.
- Đóng gói chỉ các thao tác cho phép trên CSDL vào các SP và quy định truy xuất dữ liệu phải thông qua SP. Ngoài ra còn có thể phân quyền trên SP. Hỗ trợ tốt hơn cho việc đảm bảo an toàn (security) cho CSDL.
- SP giúp cho việc kết xuất báo biểu bằng Crystal Report trở nên đơn giản và hiệu quả hơn rất nhiều so với việc kết xuất dữ liệu trực tiếp từ các bảng và khung nhìn.

#### 2- Khai báo và sử dụng thủ tục

##### Cú pháp khai báo:

```
Create {proc | procedure} procedure_name  
{Parameter_name DataType [=default] [output] }[,...n]  
As  
{ khối lệnh }  
Go
```

##### Lưu ý:

Tên tham số đặt theo qui tắc như tên biến cục bộ.

Giá trị trả về của SP dùng một (hay một số) tham số output.

Ví dụ:

- Xây dựng SP cho biết danh sách sinh viên của một lớp có mã cho trước

```
Create proc DS_Lop @MaLop varchar(10)
As
    Select SV.MaSV, SV.HoVaTen, SV.NgaySinh
    From SinhVien SV where SV.Lop = @MaLop Go
```

- Xây dựng SP tính toán giá trị cho đơn hàng có mã cho trước với quan hệ DonHang như sau:

**DonHang**(MaDH, SoLuong, DonGia, ThueSuat, ChietKhau, ThanhTien)

```
Create proc TongTien @MaDH varchar(10)
As
    Declare @ThanhTien float
    Declare @TienThue float
    Declare @TienChietKhau float
    Declare @DonGia float, @SoLuong int
    Set @SoLuong = (select SoLuong from DonHang where Ma = @MaDH)
    Set @DonGia = (select DonGia from DonHang where Ma = @MaDH)
    Set @TienThue = (select ThueSuat from DonHang where Ma = @MaDH)
    Set @TienChietKhau = (select ChietKhau from DonHang
                        where Ma = @MaDH)

    Set @ThanhTien = @DonGia* @SoLuong
    Set @TienThue = @ThanhTien* @TienThue/100
    Set @ThanhTien = @ThanhTien + @TienThue
    Set @TienChietKhau = @ThanhTien* @TienChietKhau/100
    Set @ThanhTien = @ThanhTien - @TienChietKhau
    Update DonHang
    set ThanhTien = @ThanhTien
    where Ma = @MaDH
Go
```

- Viết thủ tục thêm một đăng ký của sinh viên vào một học phần (tổng quát ví dụ trong phần If ... else)

```
Create procedure usp_ThemDangKy
```

```
@MaSV char(5), @MaHP char(5), @SiSo int = 0 output
```

As

```
select @SiSo = SiSo from HocPhan
```

```
where MaHP= @MaHP if @SiSo < 50
```

Begin

```
insert into DANG_KY(MaSV, MaHP) values(@MaSV, @MaHP)
```

```
set @SiSo = @SiSo+1
```

End

Go

- Xây dựng SP tính điểm trung bình và xếp loại cho sinh viên thuộc lớp cho trước.

Giả sử có các quan hệ như sau:

SinhVien (MaSV, HoTen, DTB, XepLoai, Lop)

MonHoc (MaMH, TenMH)

KetQua (MaMH, MaSV, LanThi, Diem) trong đó:

- Điểm thi chỉ tính lần thi sau cùng.
- Xếp loại: Xuất sắc [9, 10], Giỏi [8, 8.9], Khá [7, 7.9], Trung bình [5.0, 6.9], Yếu [0,4.9].
- Kết quả xuất dạng tham số output, không ghi xuống CSDL.

Giải

```
Create proc XepLoaiSV @MaSV varchar(10), @DTB float out put,
```

```
@XL nvarchar(20) out put
```

As

```
Set @DTB = (Select avg(Diem) from KetQua Kq
```

```
Where MaSV = @MaSV
```

```
and not exists (select * from KetQua Kq1
```

```
where Kq1.MaSV = @MaSV
```

and Kq1.MaMH=Kq.MaMH  
and Kq1.LanThi > Kq.LanThi))

If @DTB >= 9

Set @XL = N'Xuất sắc'

Else if @DTB >= 8 Set @XL  
= N'Giỏi'

Else if @DTB >= 7 Set  
@XL = N'Khá'

Else if @DTB >= 5  
Set @XL = N'Trung bình'

Else

Set @XL = N'Yếu'

Go

### **Cú pháp gọi thực hiện thủ tục:**

EXEC| EXECUTE

{ [ @return\_status = ] procedure\_name

{ [ @parameter\_name = ] value [ OUTPUT ] } [ ,...n ]

### **Lưu ý:**

- Khi gọi thực hiện SP, dùng từ khóa Exec và cần truyền đủ tham số với kiểu dữ liệu phù hợp và thứ tự chính xác như khai báo trong định nghĩa SP.
- Có thể truyền giá trị cho tham số đầu vào (input) là một hằng hoặc một biến đã gán giá trị, không truyền được một biểu thức.
- Để nhận được giá trị kết quả (thông qua tham số đầu ra), cần truyền vào một biến và có từ khóa output.

### **Ví dụ:**

- Gọi thủ tục usp\_ThemDangKy:

Exec usp\_ThemDangKy '001', 'HP01'

hoặc

```
Exec usp_ThemDangKy @MaHP = 'HP01', @MaSV = '001'
```

- Gọi thủ tục usp\_ThemDangKy có nhận kết quả đầu ra:

```
Declare @SiSo int
```

```
Exec usp_ThemDangKy '001', 'HP01', @SiSo output
```

```
Print @SiSo
```

- Gọi thủ tục usp\_ThemDangKy có nhận kết quả đầu ra và kết quả trả về từ thủ tục :

```
Declare @SiSo int, @KetQua int
```

```
Exec @KetQua = usp_ThemDangKy '001', 'HP01', @SiSo output
```

- Gọi thực hiện thủ tục xếp loại sinh viên:

```
Declare @MaSinhVien varchar(10)
```

```
Declare @DiemTB varchar(10)
```

```
Declare @XepLoai varchar(10)
```

```
Set @MaSinhVien = '0712345'
```

```
Exec XepLoaiSV @MaSinhVien, @DiemTB out put, @XepLoai out put
```

```
Exec XepLoaiSV '0713478', @DiemTB out put, @XepLoai out put
```

### **Sửa thủ tục**

Thay từ khóa Create trong lệnh tạo thủ tục bằng từ khóa Alter.

### **Xóa thủ tục**

```
Drop {procedure|proc} procedure_name
```

Ví dụ: Drop procedure usp\_ThemDangKy

## **3- Stored procedure hệ thống**

- Là những thủ tục do SQL Server cung cấp sẵn để thực hiện các công việc: quản lý CSDL, quản lý người dùng, cấu hình CSDL,...
- Các thủ tục này có tên bắt đầu bằng “sp\_” Khi xây dựng thủ tục, tránh đặt tên thủ tục bắt đầu với “sp\_”.

## IV- Kiểu dữ liệu cursor

### 1- Khái niệm Cursor

- Là một cấu trúc dữ liệu ánh xạ đến một tập các dòng dữ liệu kết quả của một câu truy vấn (select).
- Cho phép duyệt tuần tự qua tập các dòng dữ liệu và đọc giá trị từng dòng.
- Thẻ hiện của cursor là 1 biến, nhưng tên biến này không bắt đầu bằng '@'.
- Vị trí hiện hành của cursor có thể được dùng như điều kiện trong mệnh đề where của lệnh update hoặc delete: cho phép cập nhật/xoá dữ liệu (dữ liệu thật sự trong CSDL) tương ứng với vị trí hiện hành của cursor.

### 2- Khai báo và sử dụng Cursor

#### a- Khai báo Cursor

Có thể sử dụng cú pháp chuẩn SQL 92 hoặc cú pháp T\_SQL mở rộng.

– Cú pháp SQL 92 chuẩn:

```
Declare cursor_name [Insensitive] [Scroll] Cursor
For select_statement
[ For {Read only| Update [of column_name [,...n] ] } ]
```

– Cú pháp T\_SQL mở rộng

```
Declare cursor_name Cursor
[ Local | Global ]
[ Forward_only| Scroll]
[ Static| Dynamic]
[ Read_only]
For select_statement
[ For Update [ of column_name [,...n] ] ]
```

Lưu ý: Tên cursor trong các cách khai báo không bắt đầu bằng ký tự “@”.

Ý nghĩa các tham số tùy chọn trong khai báo:

- o Insensitive/ static: nội dung của cursor không thay đổi trong suốt thời gian tồn tại, trong trường hợp này cursor chỉ là read only.

- Dynamic: trong thời gian tồn tại, nội dung của cursor có thể thay đổi nếu dữ liệu trong các bảng liên quan có thay đổi.
- Local: cursor cục bộ, chỉ có thể sử dụng trong phạm vi một khối (query batch) hoặc một thủ tục/ hàm.
- Global: cursor toàn cục, có thể sử dụng trong một thủ tục/hàm hay một query batch bất kỳ hoặc đến khi bị hủy một cách tường minh.
- Forward\_only: cursor chỉ có thể duyệt một chiều từ đầu đến cuối.
- Scroll: có thể duyệt lên xuống cursor tùy ý (duyệt theo đa chiều).
- Read only: chỉ có thể đọc từ cursor, không thể sử dụng cursor để update dữ liệu trong các bảng liên quan (ngược lại với “for update...” ).

Mặc định khi khai báo cursor nếu không chỉ ra các tùy chọn thì cursor có các tính chất:

- Global
- Forward\_only
- Read only hay “for update” tùy thuộc vào câu truy vấn
- Dynamic

#### **b- Duyệt cursor**

Dùng lệnh Fetch để duyệt tuần tự qua cursor theo cú pháp:

Fetch

[ [Next| Prior| First| Last| Absolute n| Relative n]

From ] Tên\_cursor

[Into Tên\_biến [,...n] ]

- Mặc định: fetch next.
- Đối với cursor dạng forward\_only, chỉ có thể fetch next.
- Biến hệ thống @@fetch\_status cho biết lệnh fetch vừa thực hiện có thành công hay không, giá trị của biến này cơ sở để biết đã duyệt đến cuối cursor hay chưa.

#### **c- Quy trình sử dụng Cursor**

- Khai báo cursor.

- “Mở” cursor bằng lệnh Open

Open tên\_cursor

- Khai báo các biến tạm để chứa phần tử hiện hành (đang được xử lý) của cursor:

Các biến tạm phải cùng kiểu dữ liệu với các trường tương ứng của phần tử trong cursor. Có n trường trong phần tử của cursor thì phải có đủ n biến tạm tương ứng.

- Fetch (next,...) cursor để chuyển đến vị trí phù hợp:

- Có thể đưa các giá trị của dòng hiện hành vào các biến thông qua mệnh đề into của lệnh fetch.
- Nếu không có mệnh đề into, các giá trị của dòng hiện hành sẽ được hiển thị ra cửa sổ kết quả (result pane) sau lệnh fetch.
- Có thể sử dụng vị trí hiện tại như là điều kiện cho mệnh đề where của câu delete/ update (nếu cursor không là read\_only).

- Lặp lại việc duyệt và sử dụng cursor, có thể sử dụng biến @@fetch\_status để biết đã duyệt qua hết cursor hay chưa.

- @@FETCH\_STATUS = 0 : lấy dữ liệu thành công;
- @@FETCH\_STATUS < 0 : không lấy được dữ liệu.

- Đóng cursor bằng lệnh Close

Close Tên\_cursor

Lưu ý: Sau khi đóng, vẫn có thể mở lại nếu cursor chưa bị hủy.

- Hủy cursor bằng lệnh deallocate

Deallocate Tên\_cursor

Ví dụ: xét hai LĐQH

SINHVIEN (MaSV, HoTen, MaKhoa)

KHOA(MaKhoa, TenKhoa)

Duyệt và đọc giá trị từ cursor

Cập nhật lại giá trị MaSV = Viết tắt tên Khoa + MaSV hiện tại cho tất cả sinh viên:

```
declare cur_DSKhoa cursor for select MaKhoa, TenKhoa from Khoa
```



```
open cur_DSKhoa

declare @MaKhoa int, @TenKhoa varchar(30), @TenTat varchar(5)

fetch next from cur_DSKhoa into @MaKhoa, @TenKhoa

while @@fetch_status = 0

begin

    -- xác định tên tắt của Khoa dựa vào @TenKhoa...
    update SinhVien set MaSV = @TenTat+MaSV

        Where MaKhoa = @MaKhoa

    fetch next from cur_DSKhoa into @MaKhoa, @TenKhoa

end

Close cur_DSKhoa

Deallocate cur_DSKhoa
```

**d- Dùng cursor để xác định dòng cập nhật**

```
declare cur_DSKhoa cursor scroll for select MaKhoa, TenKhoa from Khoa
open cur_DSKhoa

fetch absolute 2 from cur_DSKhoa
if ( @@fetch_status = 0)

    update Khoa

        set TenKhoa = 'aaa'

        where current of cur_DSKhoa

    Close cur_DSKhoa

    Deallocate cur_DSKhoa
```

**3- Biến cursor**

- Ta có thể khai báo một biến kiểu cursor và gán cho nó tham chiếu đến một cursor đang tồn tại.
- Biến cursor có thể được xem như là con trỏ cursor.
- Biến cursor là một biến cục bộ.

– Biến cursor sau khi gán giá trị được sử dụng như một cursor thông thường.

Ví dụ :

```
Declare @cur_var cursor
```

```
set @cur_var = my_cur    -- my_cur là một cursor đang tồn tại hoặc:
```

```
Declare @cur_var cursor
```

```
set @cur_var = cursor for select_statement
```

Kết hợp cursor với stored procedure

Xây dựng SP tính điểm trung bình và xếp loại cho sinh viên thuộc lớp cho trước. Giả sử có các quan hệ như sau:

SinhVien (MaSV, HoTen, DTB, XepLoai, Lop)

MonHoc (MaMH, TenMH)

KetQua (MaMH, MaSV, LanThi, Diem)

Biết rằng

Điểm thi chỉ tính lần thi sau cùng

Xếp loại: Xuất sắc [9, 10], Giỏi [8, 8.9], Khá [7, 7.9], Trung bình [5.0, 6.9], Yếu [0, 4.9].

Kết quả ghi xuống CSDL, đồng thời xuất ra tổng số sinh viên xếp loại giỏi của lớp đó.

- Phân tích ví dụ:

- Lớp cần xét có nhiều sinh viên, từng sinh viên cần được xử lý thông qua 3 bước:

Tính điểm trung bình cho sinh viên, điểm trung bình phải là điểm của lần thi sau cùng. Có thể tái sử dụng thủ tục XepLoaiSVLop.

Dựa vào điểm trung bình của sinh viên để xác định xếp loại.

Cập nhật điểm và xếp loại vào bảng sinh viên.

- Mọi sinh viên đều lặp lại 3 bước trên.

Từ phân tích trên ta thấy:

Cần xử lý nhiều phần tử (các sinh viên).

Mỗi phần tử xử lý tương đối phức tạp (truy vấn, tính toán, gọi thủ tục khác, điều kiện rẽ nhánh, cập nhật dữ liệu, ...).

Cách xử lý các phần tử là như nhau.

⇒ Sử dụng cursor là thích hợp

Cursor chứa các sinh viên của lớp cần xét, chỉ cần chứa mã sinh viên là được.

- **Xây dựng thủ tục**

Create procedure XepLoaiSVLop

@Lop nvarchar(10), @SoSVGioi int out

As

Declare @DTB float

Declare @XepLoai nvarchar(20)

Declare @MaSV nvarchar(10)

Declare cur\_SV cursor

For (select MaSV from SinhVien where Lop=@Lop)

Open cur\_SV

Fetch Next from cur\_SV into @MaSV

While @@FETCH\_STATUS = 0

Begin

Exec XepLoaiSV @MaSV, @DTB output, @XepLoai output

Update SinhVien set DTB = @DTB, XepLoai=@XepLoai

Where MaSV = @MaSV

Fetch Next from cur\_SV into @MaSV

End

Close cur\_SV

Deallocate cur\_SV

Set @SoSVGioi = (select count(\*) from sinhvien

where lop = @Lop and XepLoai = N'Giỏi')

Go

## V- Hàm người dùng (User Defined Functions)

### 1- Khái niệm hàm người dùng

- Giống stored procedure:
  - Mã lệnh có thể tái sử dụng.
  - Chấp nhận các tham số input.
  - Biên dịch một lần và từ đó có thể gọi khi cần.
- Khác stored procedure:
  - Chấp nhận nhiều kiểu giá trị trả về (chỉ một giá trị trả về).
  - Không chấp nhận tham số output.
  - Khác về cách gọi thực hiện.
- Có thể xem hàm người dùng thuộc về 3 loại tùy theo giá trị trả về của nó:
  - Giá trị trả về là kiểu dữ liệu cơ sở (int, varchar, float, datetime...).
  - Giá trị trả về là Table có được từ một câu truy vấn.
  - Giá trị trả về là table mà dữ liệu có được nhờ tích lũy dần sau một chuỗi thao tác xử lý và insert.

### 2- Khai báo và sử dụng

#### a- Khai báo hàm người dùng

Loại 1: Giá trị trả về là kiểu dữ liệu cơ sở

```
Create function func_name  
  
    ( {parameter_name  DataType [= default ] } [...n])  
  
returns DataType  
  
As  
  
Begin  
  
    ...  
  
Return {value | variable | expression} End
```

#### Ví dụ:

```
Create function SoLonNhat
```

```
(@a int,@b int,@c int) return int
```

As

Begin

```
declare @max int    set @max =  
@a    if @b > max set @max = @b  
if @c > max set @max = @c    return  
@max
```

End

Loại 2: Giá trị trả về là một bảng có được từ một câu truy vấn

```
Create function func_name
```

```
( {parameter_name DataType [= default ] } [...n]) returns
```

Table

As

```
Return [ ( ]select_statement [ ) ]
```

Go

**Ví dụ:** Viết hàm in danh sách các mặt hàng của một mã đơn hàng cho trước

```
Create function DanhSachMatHang
```

```
( @MaDonHang varchar(10) ) returns Table
```

As

```
Return
```

```
(Select MH.TenHang,MH.DonGia
```

```
From ChiTietDH CT, MatHang MH Where
```

```
CT.MaDH = @MaDonHang and
```

```
CT.MaMH = MH.MaMH)
```

Go

Loại 3: Giá trị trả về là một bảng mà dữ liệu có được nhờ tích lũy dần sau một chuỗi thao tác xử lý và insert.

```
Create function func_name  
  
    ( {parameter_name  DataType [= default ] } [...n])  
returns TempTab_name Table(Table_definition)  
  
As  
  
Begin  
  
    ...  
  
Return  
  
End  
  
Go
```

**Ví dụ:**

```
Create function DanhSachLop ()  
  
returns @DS Table(@MaLop varchar(10),@SoSV int)  
  
As  
  
--các xử lý insert dữ liệu vào bảng DS  
  
return  
  
Go
```

Lưu ý: Trong thân hàm không được sử dụng các hàm hệ thống bất định (Built-in nondeterministic functions ), bao gồm :

- GETDATE
- GETUTCDATE
- NEWID
- RAND
- TEXTPTR
- @@TOTAL\_ERRORS, @@CPU\_BUSY, @@TOTAL\_READ,  
@@IDLE, @@TOTAL\_WRITE, @@CONNECTIONS ...

### **b- Sử dụng hàm người dùng**

Các hàm người dùng được sử dụng trong câu truy vấn, trong biểu thức... phù hợp kiểu dữ liệu trả về của nó.

Ví dụ:

- `Select dbo.SoLonNhat(3,5,7)`
- `Select * from DanhSachLop()`

Lưu ý:

- Nếu dùng giá trị mặc định của tham số, phải dùng từ khóa default.
- Khi gọi hàm loại 1 (trả về giá trị cơ bản), phải có tên owner của hàm đi kèm (ví dụ `dbo.SoLonNhat(5,8,-10)`).

### **c- Thay đổi hàm người dùng**

Thay từ khóa create trong các lệnh tạo hàm bằng từ khóa alter

### **d- Xóa hàm người dùng**

`Drop function tên_hàm_cần_xóa`

**Ví dụ:**

`Drop function DanhSachMatHang`

## **3- Các hàm hệ thống**

Ngoài các hàm do người dùng định nghĩa, SQL Server còn cung cấp các hàm xây dựng sẵn của hệ thống. Các hàm này cung cấp tiện ích như xử lý chuỗi, xử lý thời gian, xử lý số học...

Sinh viên tìm hiểu thêm về các hàm này trong Books on-line và các tài liệu tham khảo.

- Để tạo hàm hệ thống cần tiến hành theo các bước sau:

- Tạo hàm trong cơ sở dữ liệu Master
- Tên hàm bắt đầu bởi `fn_functionName`
- Thay đổi chủ nhân của hàm bằng thủ tục `sp_changeobjectowner` như sau:

`EXEC sp_changeobjectowner 'fn_functionName' , 'system_function_schema'`

Ví dụ: Tạo hàm hệ thống thực chuyển đổi một biến kiểu ngày tháng sang kiểu chuỗi.

--Tạo hàm fn\_doingay

```
create function fn_doingay(@ngay datetime)
returns char(10)
as
begin
    return convert(nchar(10),@ngay,103)
end
```

--Thay đổi chủ nhân của hàm

```
EXEC sp_changeobjectowner 'fn_doingay' , 'system_function_schema'
```

Sau lệnh này hàm fn\_doingay có thể dùng được cho CSDL bất kỳ.

```
Select manv, hoten, fn_doingay(ngaysinh)
From nhanvien
```

## **VI- Triggers và cài đặt ràng buộc dữ liệu**

### **1- Giới thiệu**

- Trigger là một loại stored procedure đặc biệt có các đặc điểm sau:
  - Tự động thực hiện khi có thao tác insert, delete hoặc update trên dữ liệu.
  - Thường dùng để kiểm tra các ràng buộc toàn vẹn của CSDL hoặc các qui tắc nghiệp vụ.
  - Một trigger được định nghĩa trên một bảng, nhưng các xử lý trong trigger có thể sử dụng nhiều bảng khác.
- Xử lý của trigger thường cần sử dụng đến hai bảng tạm:
  - Inserted: chứa các dòng vừa mới được thao tác insert/ update thêm vào bảng.
  - Deleted: chứa các dòng vừa mới bị xóa khỏi bảng bởi thao tác update/delete.

Lưu ý: update = delete dòng chứa giá trị cũ + insert dòng chứa giá trị mới – Inserted và deleted là các bảng trong bộ nhớ chính:

- Cục bộ cho mỗi trigger.



- Có cấu trúc giống như bảng (table) mà trigger định nghĩa trên đó
  - Chỉ tồn tại trong thời gian trigger đang xử lý.
- Nếu thao tác insert/ delete/ update thực hiện trên nhiều dòng, trigger cũng chỉ được gọi một lần Bảng inserted/ deleted có thể chứa nhiều dòng.

## 2- Sử dụng Trigger

### a- Khai báo trigger

#### – Cú pháp:

```
Create trigger tên_trigger  
On {tên_bảng|tên_view}  
{For| After| Instead of } { [delete] [,] [insert] [,] [update] }  
As  
  
{ các lệnh T-sql }  
  
Go
```

Trong đó:

#### For | After:

- Trigger được gọi thực hiện sau khi thao tác delete/ insert/ update tương ứng đã được thực hiện thành công:
  - Các dòng mới được thêm chứa đồng thời trong bảng dữ liệu và bảng inserted.
  - Các dòng bị xoá chỉ nằm trong bảng deleted (đã bị xoá khỏi bảng dữ liệu).
- Có thể xử lý quay lui thao tác đã thực hiện bằng lệnh rollback transaction.

#### Instead of:

- Trigger được gọi thực hiện thay cho thao tác delete/ insert/ update tương ứng:
  - Các dòng mới được thêm chỉ chứa trong bảng inserted.
  - Các dòng bị chỉ định xoá nằm đồng thời trong bảng deleted và bảng dữ liệu (dữ liệu không bị xoá).

- Trigger Instead of thường được dùng để xử lý cập nhật trên khung nhìn.

**Lưu ý:**

- Lệnh tạo trigger phải là lệnh đầu tiên trong một query batch.
- Trên một bảng có thể định nghĩa nhiều trigger for/after cho mỗi thao tác nhưng chỉ có thể định nghĩa một trigger instead of cho mỗi thao tác.
- Không thể định nghĩa trigger instead of update/ delete trên bảng có cài đặt khóa ngoại dạng update cascade/ delete cascade.
- Trong thân trigger, có thể sử dụng hàm Update(tên\_cột) để kiểm tra xem việc cập nhật được thực hiện trên cột nào.

Update(tên\_cột) = true : có thực hiện cập nhật trên cột tên\_cột

**b- Sử dụng trigger cài đặt một số loại ràng buộc**

**Ví dụ 1-** Ràng buộc liên thuộc tính – liên quan hệ

Cho CSDL:

DatHang(MaPDH, NgayDH,...)

GiaoHang(MaPGH, MaPDH, NgayGH,...)

Ràng buộc: Ngày giao hàng không thể nhỏ hơn ngày đặt hàng tương ứng Bảng tầm ảnh hưởng:

	Thêm	Xóa	Sửa
DatHang	-	-	+ (NgayDH)
GiaoHang	+	-	+ (NgayGH, MaPDH)

Cần cài đặt trigger cho thao tác sửa trên bảng DatHang, và thêm/sửa trên bảng Giaohang

***Trigger cho thao tác thêm và sửa trên giao hàng:***

Create trigger tr\_GH\_ins\_upd\_NgayGH On

GIAOHANG for insert, update As

if update(MaPDH) or update (NgayGH)

if exists(select \* from inserted i, DatHang d

```
where i.MaPDH = d.MaPD and
      i.NgayGH<d.NgayDH)

begin

      raiserror (N'Ngày GH không thể nhỏ hơn ngày DH',0,1)

rollback tran

end
```

*Bài tập: Trigger cho thao tác sửa trên đặt hàng.*

**Ví dụ 2** – Ràng buộc toàn vẹn liên bộ

Xét LĐQH: KetQua(MASV, MAMH, LANTHI, DIEM)

Tạo trigger kiểm tra RBTV: “Sinh viên chỉ được thi tối đa 2 lần cho một môn học”

- Xác định bảng tầm ảnh hưởng:

	Insert	Delete	Update
KetQua	+	-	+ (MASV, MAMH)

- Cài đặt trigger dựa trên bảng tầm ảnh hưởng:

```
-- Tao trigger ung voi thao tac insert tren bang KetQua

Create trigger trg_KetQua_insert

on KetQua

for insert

as

      declare  @SoLanThi  int

select @SoLanThi=count(*)

      from KetQua K, inserted I

      where I.MaSV=K.MaSV and I.MaMH = K.MaMH

      if @SoLanThi > 2

begin
```

```
raiserror('Số lần thi phai <= 2', 0,1)
```

```
rollback transaction
```

```
end
```

*Bài tập: Tạo trigger ứng với thao tác Update trên bảng KetQua.*

**Ví dụ 3:** Trigger cho việc thực hiện một thao tác cập nhật dữ liệu nào đó.

Cho quan hệ

CHI\_TIET\_HOA\_DON(MaHD,STT, MaMH, SoLuong, DonGia, ThanhTien)

Viết trigger thực hiện cập nhật giá trị của ThanhTien khi thêm một chi tiết hóa đơn

```
Create Trigger trgTINHHTIEN_CTHD
```

```
On CHI_TIET_HOA_DON
```

```
For Insert, update
```

```
As
```

```
Begin
```

```
Update CHI_TIET_HOA_DON
```

```
Set ThanhTien=SoLuong*DonGia
```

```
Where exists (select *
```

```
From Inserted I
```

```
Where CHI_TIET_HOA_DON.MAHD=I.MAHD and
```

```
CHI_TIET_HOA_DON.STT=I.STT
```

## Bài tập

### Phần thủ tục:

1. Giải phương trình bậc nhất
2. In ra màn hình các số nguyên từ 1->N
3. In ra màn hình các số nguyên chẵn từ 1->N
4. Giải phương trình bậc 2
5. Tính giai thừa một số N
6. Tính biểu thức :  $S=1+2/2!+3/3!+\dots+N/N!$
7. Tổng các số từ 1-N

Sử dụng CSDL QLTV để trả lời các câu hỏi sau:

8. Viết thủ tục có tham số dạng input dùng để chèn một mẫu tin vào bảng Nhanvien.
9. Viết thủ tục dùng để xóa thông tin một cuốn sách khi biết Masach.
10. Viết thủ tục dùng để xóa 1 hóa đơn khi biết SoHD. Lưu ý trước khi xóa mẫu tin trong hóa đơn thì phải xóa các mẫu tin của các chi tiết hóa đơn đó. Nếu không xóa được hóa đơn thì cũng không được phép xóa chi tiết hóa đơn của hóa đơn đó.
11. Viết thủ tục có tham số dùng để cập nhật dữ liệu mới của một mẫu tin nào đó khi biết Masach trong bảng Danhmucsach.
12. Viết một thủ tục dùng để lấy về tổng số lượng đơn đặt hàng và đơn giá trung bình của của một nhóm sách nào đó khi biết mã nhóm.

### Phần hàm:

1. Tính giai thừa một số N
2. Tính tổng các số từ 1-N
3. Tổng các số nguyên chẵn từ 1-N
4. Tính giá trị biểu thức :  $S=1+2/2!+3/3!+\dots+N/N!$
5. Truyền vào tháng và năm cho biết tháng đó có bao nhiêu ngày
6. Tính X lũy thừa Y với X là số thực và Y là số nguyên

7. Tìm ước số chung lớn nhất của hai số nguyên a,b
8. Tìm bội số chung nhỏ nhất của hai số nguyên a,b
9. Kiểm tra N có phải là số nguyên tố không?

### Sử dụng CSDL QLTV

10. Viết hàm trả về tổng thành tiền của một hóa đơn nào đó. Thanh tien = (SốLượng \* Đơn giá) - ChiếtKhấu.
11. Viết hàm trả về chiết khấu dựa vào số lượng lập hoá đơn và theo quy định sau:  
 Chiết khấu = 5%\* GIABAN nếu Soluong <5  
 Chiết khấu = 6%\* GIABAN nếu 10>Soluong >=5  
 Chiết khấu = 8%\* GIABAN nếu 15>Soluong>=10  
 Chiết khấu =10%\* GIABAN nếu Soluong>=15
12. Viết hàm trả về danh sách các hoá đơn đã lập của nhân viên nào đó. Thông tin gồm: SoHD, Soluong,Dongia, Thanh tien, MaNV.

### Phần Cursor

1. Cho database sau:

MatHang(MaHang,TenHang)

Ban(MaHang,Ngay,MaKH,SL,Dgia)

Thống kê số tiền bán của các mặt hàng theo dạng sau:

MaHang	TenHang	TongTien

tblBan :

MaHang	SL	DonGia	Ngay	MaKH
S	30	10	12/12/2015	1
S	70	10	13/12/2015	1
D	100	20	14/12/2015	2
G	200	50	15/12/2015	3

tblMatHang

MaMH	TenHang
S	Sữa
D	Đường
G	Gạo
K	Khoai
T	Trứng

Hướng dẫn : Tạo table Thống kê có dạng theo table MatHang sau đó thêm trường tổng tiền

2. Cho database sau:

KS(MaKS,TenKS,DiaChi,NamXD)

Khach(SoCMND,HoTen,DiaChi,Phai,NS)

Phong(MaKS,MaP,TenP)

Thue(MaKS,MaP,SoCMND,NgayDen,NgayDi)

Dùng con trỏ thống kê theo dạng sau :

TenKS	Phong	Số Lược Thuê

### Phần Trigger

Cho cơ sở dữ liệu QLNS như sau:

**ChucVu** (MaCV, TenCV)

**NhanVien** (MaNV, HoTen, NgaySinh, QueQuan, GioiTinh, MaDV, MaCV)

**DonVi** (MaDV, TenDV)

**Luong** ( MaNV, LuongChinh, HeSo, BacLuong, PhuCap, BHYT, BHXH, TongLuong)

1. Tạo trigger thêm 1 nhân viên mới
2. Tạo trigger khi thay đổi mã chức vụ ở bảng chức vụ thì các bảng có liên quan sẽ thay đổi theo.

3. Tạo trigger khi xóa xóa nhân viên ở bảng nhân viên thì các bảng ở các bảng khác sẽ xóa theo.
4. Tạo trigger để khi Insert và Update vào bảng lương sẽ tự động tính các trường sau:

- $\text{PhuCap} = 30\% \text{Luong}$ .
- $\text{BHYT} = 5\% \text{Luong}$
- $\text{BHXH} = 10\% \text{Luong}$
- $\text{TongLuong} = \text{Luong} + \text{PhuCap} - \text{BHYT} - \text{BHXH}$

Trong đó:  $\text{Luong} = \text{LuongChinh} * \text{HeSo} * \text{BacLuong}$



## CHƯƠNG 6: BẢO MẬT VÀ AN TOÀN DỮ LIỆU

### I- Bảo mật trong hệ quản trị cơ sở dữ liệu

#### 1- Khái niệm cơ bản về bảo mật

Nhằm bảo vệ hệ thống CSDL không bị xâm nhập, người quản trị cơ sở dữ liệu phải quyết định cho phép hay không cho phép người dùng truy cập và thao tác trên cơ sở dữ liệu dựa vào nhiệm vụ của người dùng trên hệ CSDL. Người quản trị thường dựa trên nền tảng lý thuyết bảo mật của hệ cơ sở dữ liệu đa người dùng, nhằm tìm ra phương pháp bảo mật theo đúng với nhu cầu của bảo mật dữ liệu.

Với mục đích tăng tính bảo mật dữ liệu, SQL Server hỗ trợ các tính năng cho phép người quản trị thiết lập cơ chế bảo vệ cơ sở dữ liệu trong môi trường đa người dùng, bao gồm các yếu tố chính sau:

- Vai trò của người dùng trong hệ thống và cơ sở dữ liệu.
- Quyền sử dụng các ứng dụng cơ sở dữ liệu trong SQL Server.
- Quyền tạo và sửa đổi cấu trúc các đối tượng CSDL.
- Quyền truy cập, xử lý dữ liệu.

Khi đăng nhập vào một hệ thống CSDL đa người dùng, người sử dụng cần phải cung cấp UserID (tài khoản) và Password (mật khẩu). Dựa trên UserID hệ thống có khả năng kiểm soát tất cả các hành vi của người sử dụng trên CSDL SQL Server.

Để thực hiện được chức năng này, người quản trị CSDL cần phải thiết lập các quyền xử lý và truy cập vào CSDL khi tạo ra UserID, ngoài ra còn có một số thuộc tính khác của SQL Server như quyền backup dữ liệu, trao đổi dữ liệu với các ứng dụng CSDL khác, ...

Khi nói đến bảo mật, người quản trị cần quan tâm đến các thông tin sau của người dùng:

- Một người dùng chỉ có một UserID và một mật khẩu.
- Thời gian có hiệu lực của mật khẩu.
- Giới hạn chiều dài của mật khẩu.
- Giới hạn người sử dụng theo license hay mở rộng.
- Thông tin về người sử dụng.

Khi tạo người sử dụng, tên tài khoản cần rõ ràng, dễ hiểu để gọi nhớ, và không cho phép các ký tự đặc biệt, không nên có khoảng trắng.

## 2- Lựa chọn bảo mật

Khi tạo ra một người dùng (login user) trong SQL Server, có 3 cách để tăng tính bảo mật cho người sử dụng đó:

- Giao tiếp với hệ điều hành: sử dụng UserID và Password của hệ điều hành Windows để đăng nhập SQL Server. Với loại bảo mật này, người dùng truy cập vào mạng và có thể sử dụng CSDL SQL Server, đồng thời một người dùng có UserID và Password có thể sử dụng tài nguyên trên mạng.
- Bảo mật chuẩn: với loại này, người sử dụng có UserID và Password tách rời với hệ điều hành mạng, ứng với loại bảo mật này người sử dụng chỉ có hiệu lực trong CSDL SQL Server, không thể sử dụng tài nguyên trên mạng.
- Tổng hợp cả hai trường hợp trên: một số người dùng sử dụng quyền sử dụng trên hệ điều hành và SQL Server, một số khác chỉ sử dụng quyền truy cập vào SQL Server.

**Lưu ý:** Tài khoản người dùng có giá trị trên SQL Server hiện hành, khi sang một SQL Server khác phải tạo ra tài khoản người dùng trên server đó.

## 3- Tạo Tài Khoản Đăng Nhập (Login Account):

### a- Thêm Tài khoản với chế độ xác nhận SQL Server:

```
SP_ADDLOGIN [ @login = ] <'Tênđăngnhậ'>  
            [ , @password = ] <'password'>  
            [ , @defdb = ] <'TênCSDL'>  
            [ , @deflanguage = ] <'Ngônngữ'>  
            [ , @sid =] <'Mã nhận dạng Login'>  
            [ , @encryptopt = 'skip_encryption' ]
```

- @defdb : tên CSDL mặc định được mở khi đăng nhập. Nếu không chỉ định mặc định là **CSDL Master**.
- @deflanguage : Ngôn ngữ mặc định
- @sid : varbinary(16) : Mã nhận dạng của hệ thống. Nếu không chỉ định, hệ thống tự tạo một mã số mới.
- @encryptopt varchar(20): Mặc định Password sẽ được mã hóa khi lưu trong các table hệ thống. Ngoại trừ gán giá trị **skip\_encryption** cho tham số này.

- Thủ tục trả về giá trị 0 nếu thành công, ngược lại trả về giá trị 1.

**Ví dụ:** Tạo tài khoản tên 'user01', mật khẩu '01' vào CSDL 'QLDeTai'

```
Sp_AddLogin 'user01', '01', 'QLDeTai'
```

- b- Cấp quyền kết nối cho User hay nhóm User của Windows kết nối đến SQL Server:**

```
SP_GRANTLOGIN [@loginname =] 'login'
```

'Login': <Domain>|<Computer name>\<tên nhóm>

**Ví dụ:** Thêm tài khoản cho user Windows NT [Server4\User01] kết nối đến SQL Server.

```
EXEC sp_grantlogin 'Server4\User01'
```

Hay EXEC sp\_grantlogin [Server4\User01]

*Chú ý: Chỉ được thực hiện bởi những thành viên có vai trò **sysadmin** và **securityadmin**.*

- 4- Thay đổi mật khẩu:**

```
Sp_Password [ @old = ] <'oldPW'>
```

```
[ , @new = ] <'newPW'>
```

```
[ , @LoginName = ] <'login'>
```

- 5- Cấp quyền truy xuất CSDL hiện hành cho Login:**

```
Sp_GrantDBAccess [ @loginname = ] 'Tên đăng nhập'
```

```
[ [ , @name_in_db = ] 'TênUser' ]
```

**Ví dụ:** Use QLDeTai

```
Go
```

```
sp_GrantDBAccess 'user01', 'Anh'
```

- 6- Xóa quyền truy xuất CSDL hiện hành:**

```
Sp_RevokeDBAccess [ @name_in_db = ] 'TênUser'
```

**Ví dụ:** Sp\_RevokeDBAccess 'Anh'

- 7- Cấp quyền thực thi trên CSDL:**

Bao gồm các quyền: Select, Insert, Update, Delete, Reference, Excecute.

```
GRANT ALL | <quyền> [...]
```

**ON** <TênTable|View>[(<têncột>,...)] | <tênSP>

**TO** <tên Login hoặc Role> [...]

**[WITH GRANT OPTION]**

**[AS <tên Role> ]**

- **All** : Cấp tất cả các quyền thực thi Select, Insert, Update, Delete hay Reference trên table hay view; quyền Excecute cho SP.
- **TO** <tên Login hoặc Role> : Khi cấp quyền cho nhóm hay user của Windows NT, phải chỉ định: <Domain>|<Computer name>\<tênnhóm>.

Để cấp quyền cho nhóm cục bộ Windows NT built-in, dùng từ khóa BUILTIN thay thế tên domain hoặc computer name.

Quyền thực thi đã cấp cho role **Public** được áp dụng cho tất cả users trong CSDL. Quyền thực thi đã cấp cho user **Guest** được sử dụng cho tất cả Users không được phân quyền truy xuất trên CSDL.

**Ví dụ:** USE QLDeTai

```
Grant All On DeTai To User01
```

```
GO
```

```
GRANT SELECT ON DeTai TO public
```

```
GO
```

```
GRANT INSERT, UPDATE, DELETE ON DeTai TO Mary, [Corporate\BobJ]
```

```
GO
```

- **WITH GRANT OPTION** : Cho phép Login cấp quyền đã chỉ định trên đối tượng cho Login khác.
- **AS {group | role}** : Được dùng khi quyền thực thi trên một đối tượng đã cấp cho nhóm hoặc role, và một User của nhóm hoặc Role muốn cấp quyền thực thi cho User khác không là thành viên của nhóm hoặc Role.

**Ví dụ:** table NhatKy được tạo bởi user Lac. Lac cấp quyền SELECT table NhatKy cho Role BanBe với mệnh đề WITH GRANT OPTION để các user thành viên của Role BanBe có thể nhường quyền này cho các user khác không thuộc Role BanBe.

User Hong, là thành viên của BanBe, muốn nhường quyền SELECT table NhatKy cho user Khoa, không là thành viên của role BanBe.

```
/* User Lac */
```

```
GRANT SELECT ON NhatKy TO BanBe WITH GRANT OPTION
```

```
/* User Hong */
```

```
GRANT SELECT ON NhatKy TO Khoa AS BanBe
```

## 8- Từ Chối quyền thực thi trên CSDL:

**DENY** ALL | <quyền> [,...]

[ ( *column* [ ,...*n* ] ) ] ON { *table* | *view* }

| ON *table* | *view* [ ( *column* [ ,...*n* ] ) ]

| ON *stored\_procedure*

**TO** <tên Login hoặc Role> [,...] [Cascade]

**Ví dụ:** USE QLDeTai

```
GO
```

```
GRANT SELECT ON DeTai TO public
```

```
DENY SELECT, INSERT, UPDATE, DELETE
```

```
ON DeTai TO Mary, John, Tom
```

## 9- Xóa bỏ quyền thực thi đã cấp hoặc từ chối trên CSDL:

**REVOKE** [ GRANT OPTION FOR ]

{ ALL | *permission* [ ,...*n* ] }

[ ( *column* [ ,...*n* ] ) ] ON { *table* | *view* }

| ON { *table* | *view* } [ ( *column* [ ,...*n* ] ) ]

| ON { *stored\_procedure* | *extended\_procedure* }

| ON { *user\_defined\_function* }

**TO** | **FROM** *security\_account* [ ,...*n* ]

[ CASCADE ]

[ AS { *group* | *role* } ]

**Ví dụ :** Xóa bỏ quyền Select đã từ chối cho User Mary trên table DeTai.

```
REVOKE SELECT ON DeTai TO Mary
```

## 10- Cấp quyền tạo đối tượng trong CSDL:

**GRANT ALL** | <lệnh> [,...]

**TO** <tên Login hoặc Role> [,...]

Bao gồm các lệnh: CREATE DATABASE; CREATE DEFAULT;

CREATE PROCEDURE; CREATE RULE; CREATE TABLE; CREATE VIEW;

BACKUP DATABASE; BACKUP LOG

**Ví dụ:** GRANT CREATE DATABASE, CREATE TABLE

TO Mary, John, [Corporate\BobJ]

## 11- Từ chối quyền tạo đối tượng trên csdl:

**DENY** { ALL | *statement* [ ,...*n* ] } **FROM** *security\_account* [ ,...*n* ]

**Ví dụ:** DENY CREATE TABLE FROM Joe, [Corporate\BobJ]

## 12- Xóa bỏ quyền tạo đối tượng đã cấp hoặc từ chối trên CSDL:

**REVOKE** { ALL | *statement* [ ,...*n* ] } **FROM** *security\_account* [ ,...*n* ]

**Ví dụ :** REVOKE CREATE TABLE FROM Joe, [Corporate\BobJ]

## 13- Vai Trò Của User Trong SQL Server

SQL Server hỗ trợ 2 nhóm roles:

- Vai trò trên SQL Server: Chứa các quyền quản trị SQL Server
- Vai trò trên Database: Chứa các quyền quản lý và thực thi trên các đối tượng của CSDL. Ngoài các vai trò được cung cấp bởi SQL Server có thể tạo thêm vai trò khác.

### a- Tạo Vai trò trên CSDL:

**SP\_ADDROLE** [@rolename =] 'role' [ , [@ownername =] 'owner']

**Ví dụ:** Trong CSDL QLDeTai thêm role 'QuanLy'

SP\_ADDROLE 'QuanLy'

Go

GRANT SELECT ON DeTai TO QuanLy

**b- Thêm User vào Role:**

```
SP_ADDROLEMEMBER [@rolename =] 'role',
                  [@membername =] 'UserName'
```

**Ví dụ:** Cấp quyền truy cập CSDL QLDeTai cho User và thêm vai trò ‘QuanLy’ cho User.

```
USE QLDeTai

GO

EXEC SP_GRANTDBACCESS 'Server4\User01', 'Hong'

GO

EXEC SP_ADDROLEMEMBER 'QuanLy', 'Hong'
```

**14- Các quyền truy cập trên các đối tượng trong một CSDL:**

Quyền	Diễn giải
SELECT	Cho phép người sử dụng nhìn thấy dữ liệu, nếu người sử dụng có quyền này thì họ chỉ có thể thực thi những phát biểu select để truy vấn dữ liệu trên các bảng hay các view được cho phép.
INSERT	Cho phép người sử dụng thêm dữ liệu, nếu người sử dụng có quyền này, họ có thể thực hiện phát biểu Insert, đối với một số hệ thống CSDL khác, muốn thực thi phát biểu Insert, người sử dụng phải có quyền Select, trong SQL Server quyền Insert không liên quan đến quyền truy vấn Select.
UPDATE	Quyền này cho phép người sử dụng chỉnh sửa dữ liệu bằng phát biểu Update.
DELETE	Quyền này cho phép người sử dụng xóa dữ liệu bằng phát biểu Delete.
REFERENCE	Cho phép người sử dụng thêm dữ liệu vào bảng có khóa ngoại bằng phát biểu Insert, trong SQL Server quyền Insert không liên quan đến quyền truy vấn Select.
EXECUTE	Quyền này cho phép người sử dụng thực thi các thủ tục (SP) trong CSDL.

**15- Vai trò của người sử dụng trong SQL Server và cơ sở dữ liệu**

### *Vai trò trên SQL Server*

<b>Vai trò (Role)</b>	<b>Diễn giải</b>
sysadmin	Có các quyền tương đương với sa.
serveradmin	Cấu hình một số tham số và tắt server.
setupadmin	Bị giới hạn bởi một số chức năng liên kết server và khởi động một số thủ tục.
securityadmin	Quản lý người dùng và tạo CSDL.
processadmin	Được phép dừng các giao tác đang thực hiện trên CSDL và một số quá trình thực hiện khác của SQL Server.
dbcreator	Được phép tạo CSDL.
Diskadmin	Quản lý các tập tin liên quan đến CSDL SQL Server.

### *Vai trò trên CSDL*

<b>Vai trò</b>	<b>Diễn giải</b>
db_owner	Với vai trò này, người sử dụng (NSD) thuộc nhóm sở hữu CSDL mới có thể truy cập vào CSDL.
db_accessadmin	Thực hiện các chức năng giống như securityadmin.
db_datareader	NSD được phép select trên các bảng dữ liệu của các người dùng khác trong CSDL.
db_datawriter	NSD được phép insert, update, delete trên các bảng dữ liệu của các người dùng khác trong CSDL.
db_ddladmin	NSD có thể thêm hay chỉnh sửa các đối tượng của CSDL.
db_securityadmin	NSD có quyền tương đương với quyền của securityadmin.
db_backupoperator	NSD có thể thực hiện chức năng backup dữ liệu.
db_denydareader	Không cho phép sử dụng phát biểu SELECT trên tất cả các bảng dữ liệu của CSDL.
db_denydewriter	Không cho phép sử dụng phát biểu INSERT, UPDATE, DELETE trên tất cả các bảng dữ liệu của CSDL.

## **II- Sao lưu và khôi phục dữ liệu**

### **1- Lý do phải sao lưu và khôi phục dữ liệu**

Trong quá trình thực hiện quản trị CSDL SQL Server thì một số nguyên nhân sau đây bắt buộc bạn phải xem xét đến kỹ thuật sao lưu và khôi phục dữ liệu:



- Ổ đĩa bị hỏng (chứa các tập tin CSDL).
- Server bị hỏng.
- Nguyên nhân bên ngoài (thiên nhiên, hỏa hoạn, mất cắp,...).
- User vô tình xóa dữ liệu.
- Bị vô tình hay cố ý làm thông tin sai lệch.
- Bị hack.

## 2- Các loại sao lưu dữ liệu

Sao lưu (Backup) dữ liệu trong SQL Server gồm các loại sau:

- Full Database Backups: sao chép toàn bộ CSDL (các tập tin bao gồm các bảng, khung nhìn, các đối tượng khác).
- Differential Database Backups: sao chép những dữ liệu thay đổi trong Data file kể từ lần full backup gần nhất.
- File or file group backups: sao chép một file đơn hay file group.
- Differential File or File Group Backups: thực hiện như Differential Database nhưng copy phần dữ liệu thay đổi của file đơn hoặc file group.
- Transaction log backups: Ghi nhận tất cả các transaction chứa trong transaction log file kể từ lần transaction log backup gần nhất. Với loại sao lưu này ta có thể khôi phục dữ liệu tại một thời điểm.

## 3- Các mô hình khôi phục dữ liệu

- Full Recovery model: Là mô hình phục hồi toàn bộ hoạt động giao dịch của dữ liệu (Insert, Update, Delete, hoạt động bởi lệnh bcp, bulk insert). Với mô hình này ta có thể phục hồi dữ liệu tại một thời điểm trong quá khứ đã được lưu trong transaction log file.
- Bulk-Logged Recovery Model: Mô hình này được thực thi cho các thao tác bcp, bulk insert, create chỉ mục, writetext, updatetext, các hoạt động này chỉ nhật ký sự kiện vào log để biết mà không sao lưu toàn bộ dữ liệu, chi tiết như trong full recover. Các sự kiện Insert, Update, Delete vẫn được nhật ký và khôi phục bình thường.

- Simple Recovery Model: Với mô hình này bạn chỉ phục hồi lại thời điểm sao lưu gần nhất mà không theo thời điểm khác trong quá khứ.

**Cách đặt mô hình khôi phục:**

- Chọn CSDL.
- Nhấn nút phải chuột → Properties → Options → Recovery Model.

Xét ví dụ sau: Giả sử ta có một CSDL được backup theo chiến lược như hình vẽ bên dưới

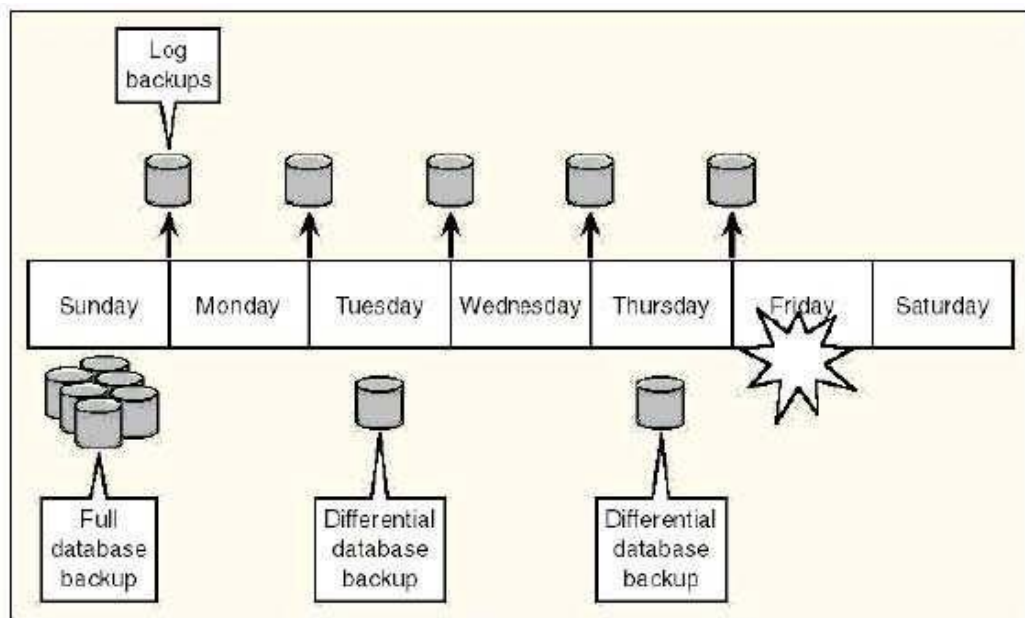
Nhìn hình ta thấy CSDL được lập lịch Full Database Backup vào ngày chủ nhật, Differential Database Backup vào ngày thứ ba và thứ năm, còn Log Database Backup vào 5 ngày trong tuần, ngày thứ sáu có sự cố với CSDL, data file bị hỏng. Vấn đề đặt ra là phải phục hồi dữ liệu và CSDL hoạt động bình thường. Ta phải làm các bước sau:

Thực hiện Backup log file (giả sử log file không bị hỏng).

Khôi phục Full Database của ngày chủ nhật.

Phục hồi Differential Database của ngày thứ năm.

Khôi phục Transaction log backup ngày thứ năm.



#### 4- Sao lưu cơ sở dữ liệu (Backup Database)

Trước khi xem xét kỹ thuật sao lưu CSDL, ta thống nhất một số thuật ngữ bằng tiếng Anh như sau:

Backup: Là quá trình copy toàn bộ hoặc một phần database, transaction log, file, file group thành lập một backup set được chứa trong backup media (disk hoặc tape) bằng cách sử dụng một backup device (tape drive name hoặc physical filename).

Backup Device: Một file vật lý hoặc một drive tape.

Backup file: Một file chứa Backup set.

Backup media: là Disk hoặc tape.

Backup set: Một bộ backup một lần backup đơn chứa trên backup media.

***Các bước thực hiện backup như sau:***

- Chọn CSDL cần backup.
- Nhấn phải chuột → Tasks → Backup
- Nhập các tham số, lựa chọn kiểu.

**5- Khôi phục dữ liệu (Restore Database)**

Là chức năng thực hiện khôi phục dữ liệu đã sao lưu, tùy theo chiến lược backup mà người quản trị có thể phục hồi đến thời điểm nào, thu được bộ dữ liệu trong quá khứ như thế nào. Khôi phục dữ liệu được thực hiện theo thứ tự backup, thông tin này được lưu trữ trong msdb. Các bước thực hiện như sau:

- Chọn mục Databases → Nhấn nút phải chuột → Tasks → Restore
- Nhập tham số, chọn mô hình khôi phục.

**Bài tập**

1. Tạo hai tài khoản đăng nhập là **nguoisudung1**, **nguoisudung2** và cho phép hai tài khoản này sử dụng cơ sở dữ liệu QuanLyNhanVien.
2. Cấp phát tất cả các quyền cho **nguoisudung1**, cấp quyền SELECT cho **nguoisudung2** trên hai bảng của cơ sở dữ liệu QuanLyNhanVien.
3. Thu hồi quyền Delete đối với **nguoisudung1** trên bảng DONVI.
4. Sao lưu cơ sở dữ liệu QuanLyNhanVien và lưu trên thư mục D:\BackUp\Data với tên QLNV.bak.
5. Xóa cơ sở dữ liệu QuanLyNhanVien khỏi hệ thống và phục hồi lại như cũ nhờ tập tin QLNV.bak.

## **TÀI LIỆU THAM KHẢO**

### **Tiếng việt**

- [1] Phạm Hữu Khang, Quản trị SQL Server 2000, NXB Thống kê, 2005.
- [2] Dương Quang Thiện, SQL Server 2000 Lập trình T-SQL, NXB Văn hoá Sài Gòn, 2007.
- [3] Vũ Tuyết Trinh, SQL Server 2008 (Slide bài giảng).
- [4] Microsoft SQL Server 2008 - Quản Trị Cơ Sở Dữ Liệu (Tập 1,2) Phạm Hữu Khang – NXB LDXH
- [5] SQL Server 2008 Books Online

### **Tiếng Anh**

- [6] Bill Hamilton, ADO.NET Cookbook, O'Reilly, 2003 (Ebook).
- [7] Ramakrishnan, R. and Gehrke, J., Database Management Systems, Third Edition, McGraw Hill, 2003.
- [8] Ramez Elmasri, Shamkant B. Navathe, Fundamentals of database systems, Addison Wesley - 4th edition, 2004.
- [9] Sumathi, S. and Esakkirajan, S., Fundamentals of Relational Database Management Systems, Springer-Verlag, 2007.

### **Internet**

Keyword: Hệ quản trị CSDL, Database management system, SQL Server 2008

## PHỤ LỤC

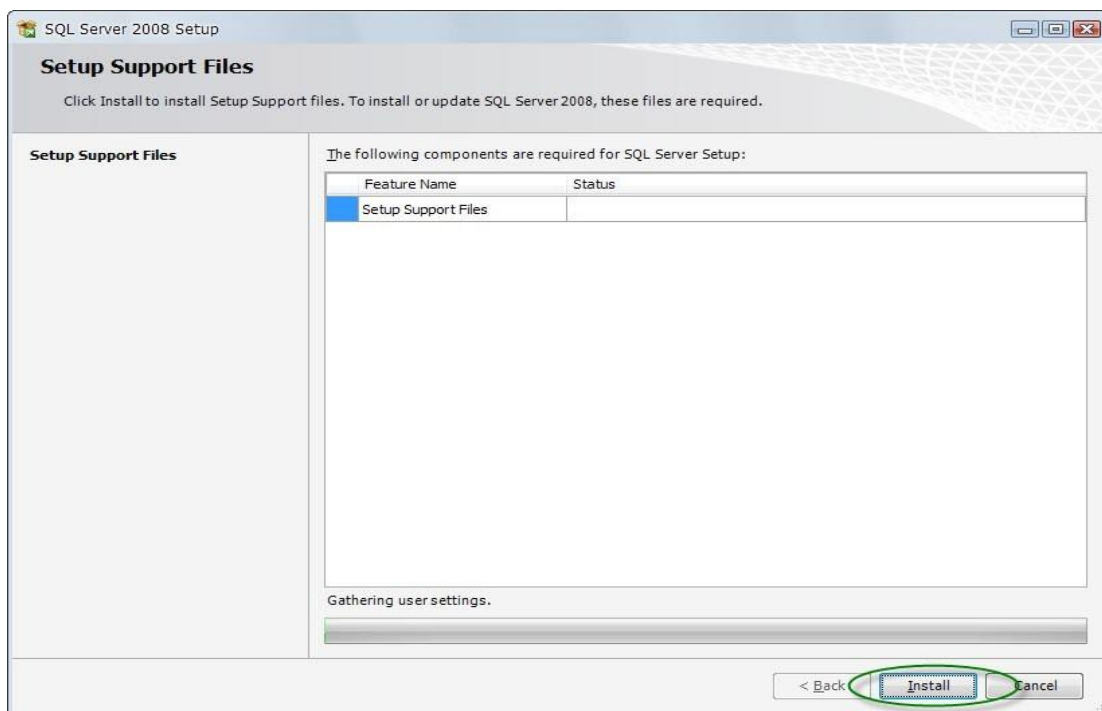
### Các bước cài đặt SQL Server 2008

**Bước 1:** Chạy file setup.exe để cài đặt, chọn Installation -> New SQL Server stand-alone ...



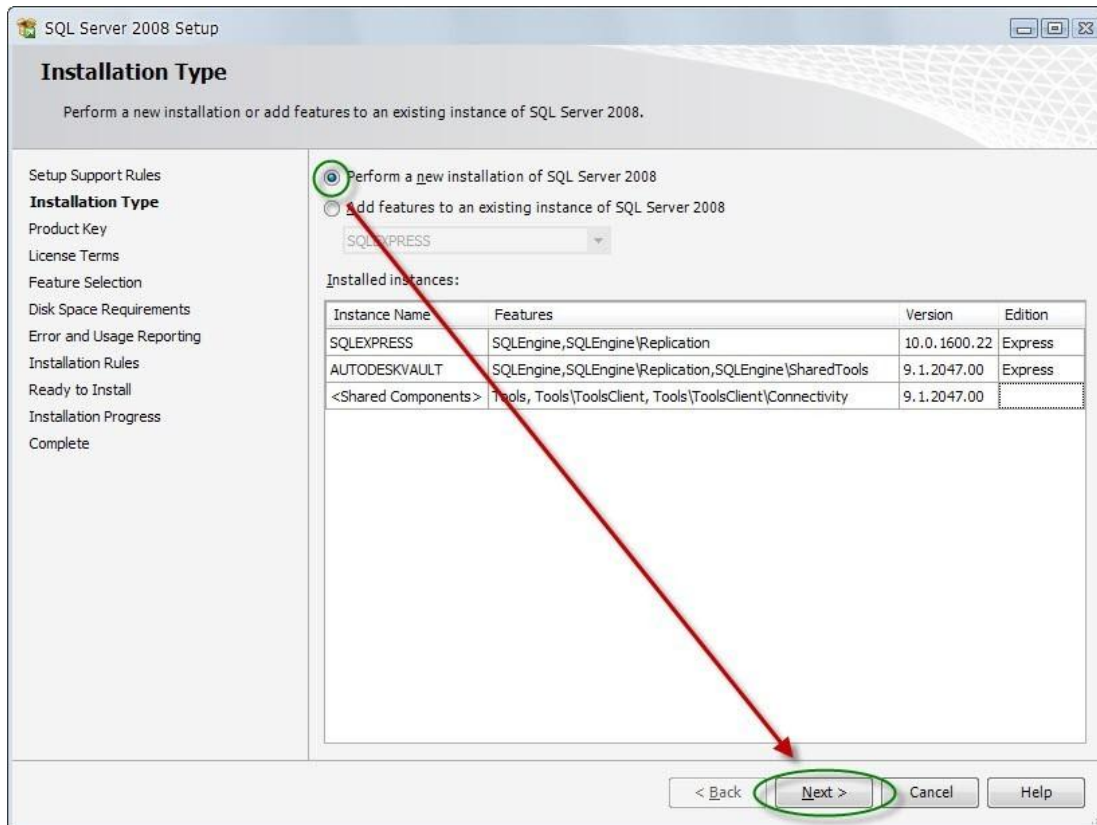
Hình 1.1. Giao diện SQL Server Installation Center

**Bước 2:** Chọn Ok -> Next .



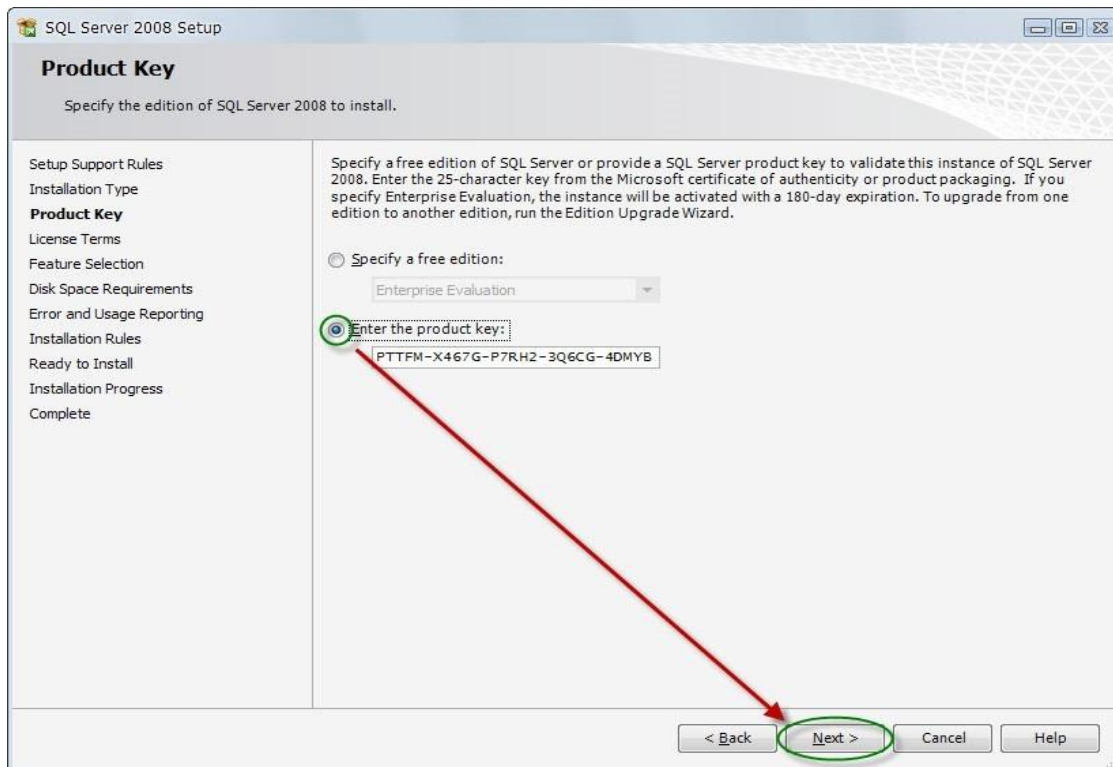
Hình 1.2. Giao diện Setup Support Files

### **Bước 3:** chọn kiểu cài đặt mới



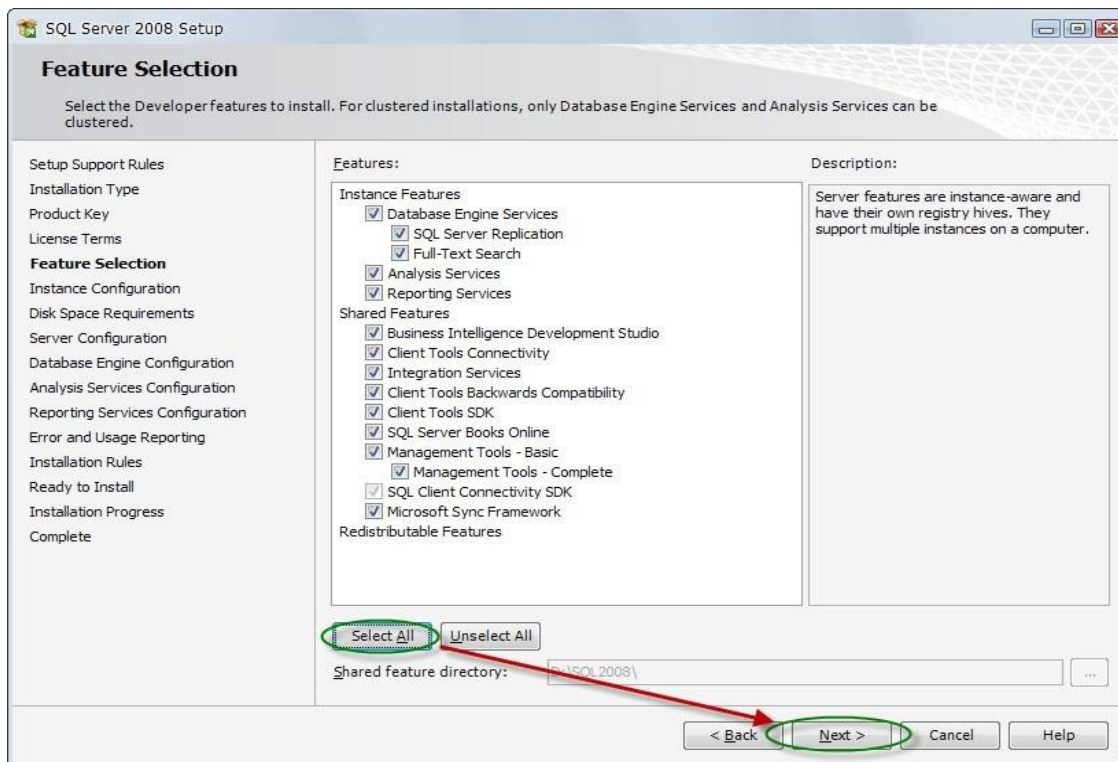
Hình 1.3. Giao diện Installation Type

### **Bước 4:** Nhập product key



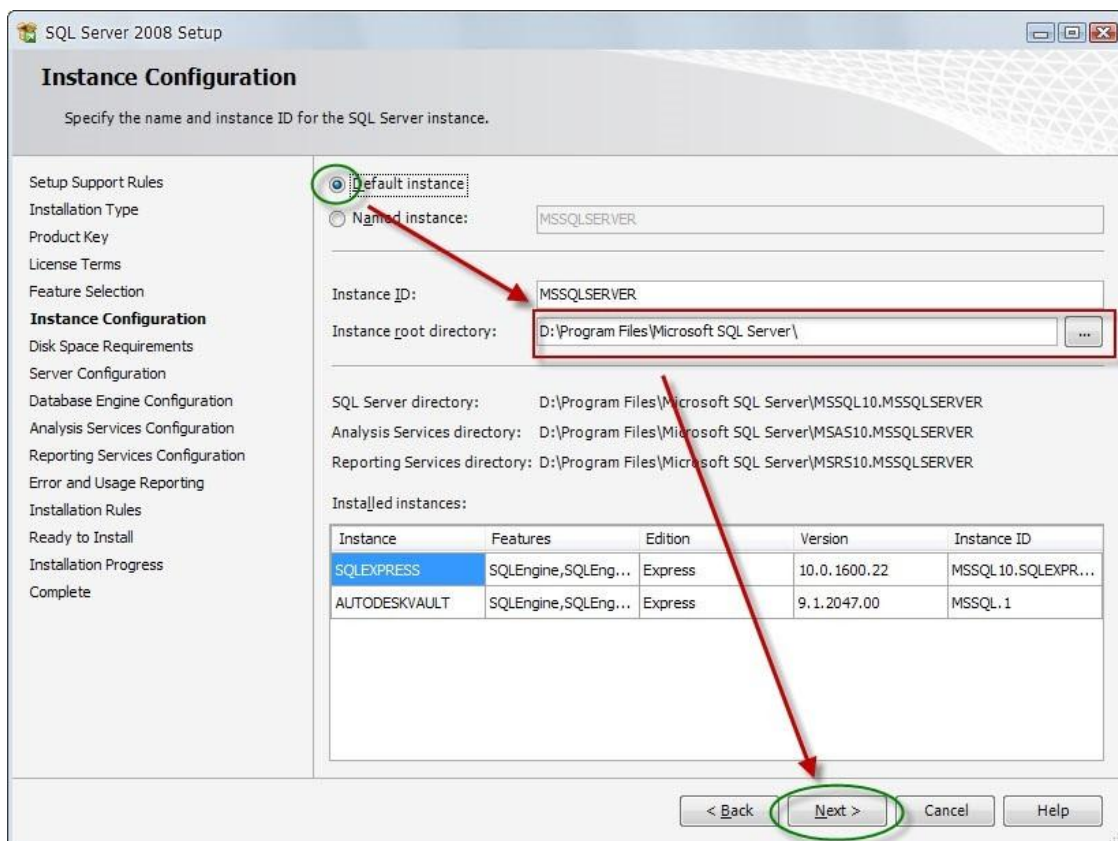
Hình 1.4. Giao diện Product Key

**Bước 5:** Sau khi đồng ý License Terms, chọn các thành phần cài đặt



Hình 1.5. Giao diện Feature Selection

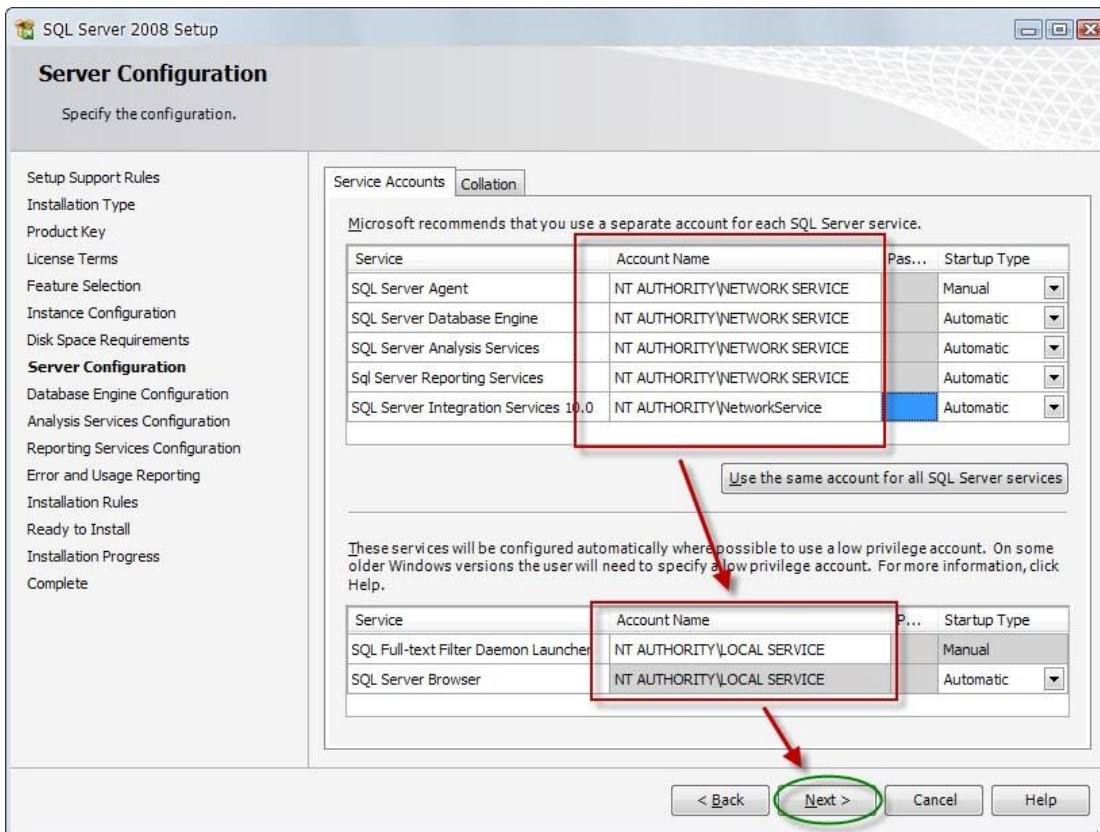
**Bước 6:** Thiết lập cài đặt chọn Default instance



Hình 1.6. Giao diện Instance Configuration

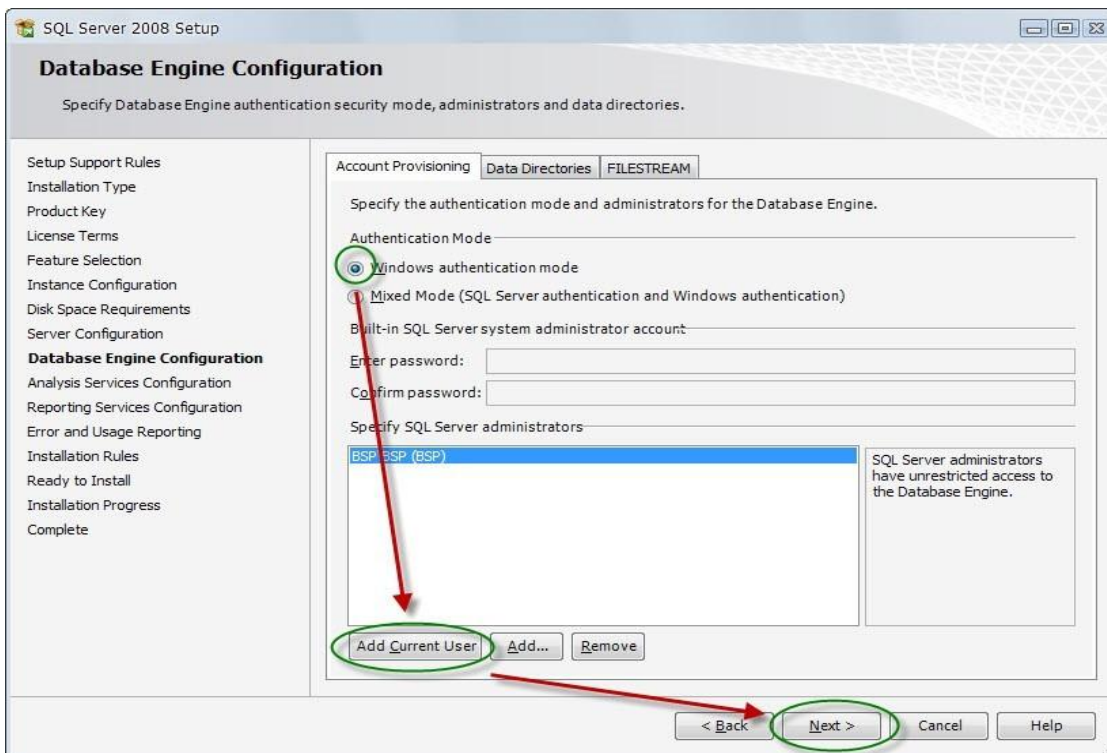


## **Bước 7:** Cấu hình server



Hình 1.7. Giao diện Server Configuration

## **Bước 8:** Cấu hình dữ liệu như sau chọn Window Authentication và Add current User

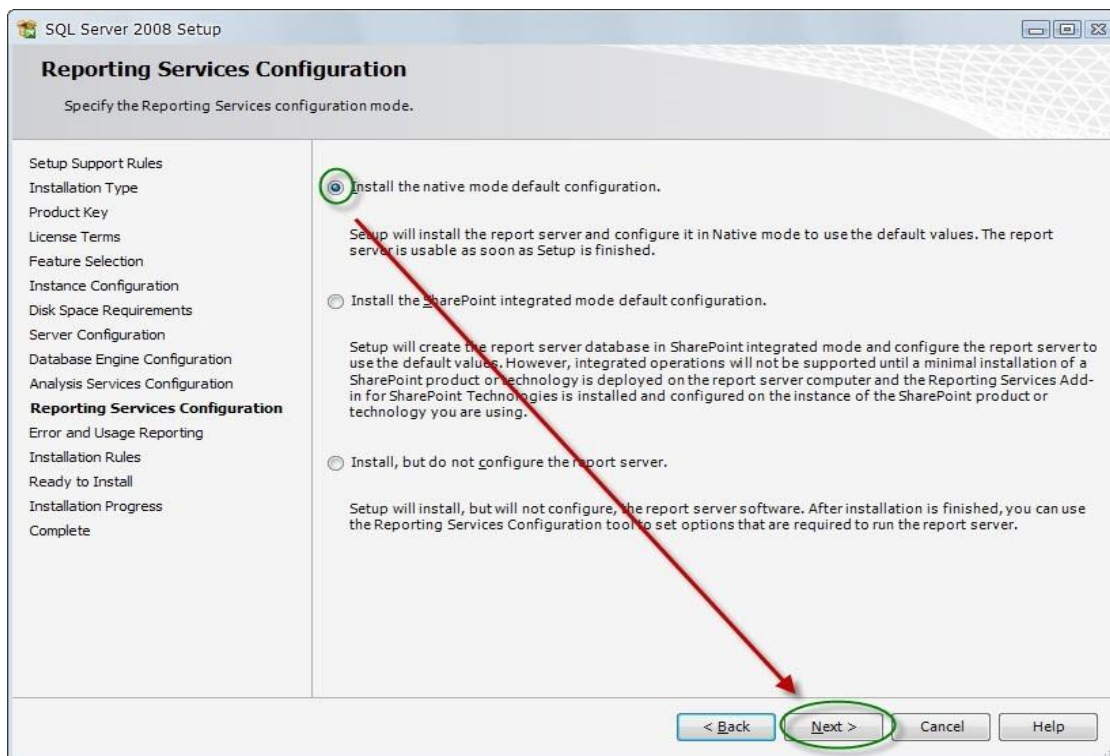


Hình 1.8. Giao diện Database Engine Configuration



**Bước 9:** Cấu hình analysis services Add Current User

**Bước 10:** Cấu hình report chọn option như hình nhấn Next, Next ... Cho đến khi hoàn tất



Hình 1.9. Giao diện Reporting Services Configuration

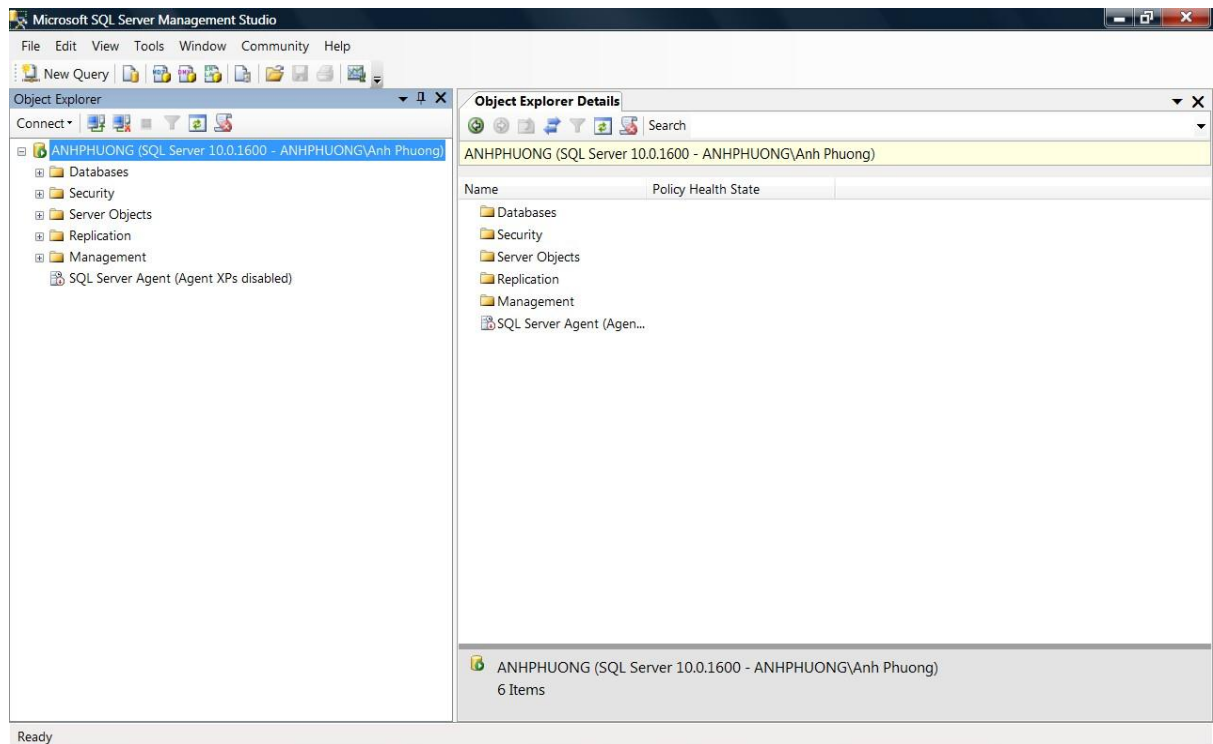
**SQL Server Management Studio**

Mở SQL Server Management Studio ta làm như sau: Vào start -> chọn program -> chọn Microsoft SQL Server 2008 -> chọn SQL Server Management Studio



Hình 1.10. Kết nối vào SQL Server

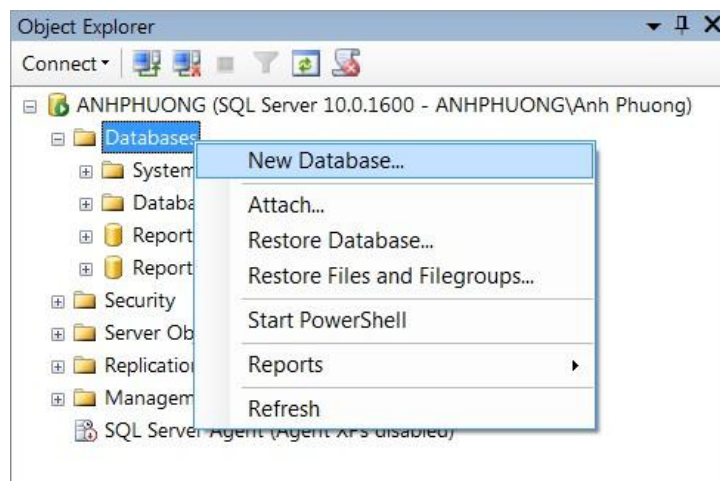
Sau khi nhấn nút Connect sẽ xuất hiện màn hình sau:



Hình 1.11. SQL Server Management Studio

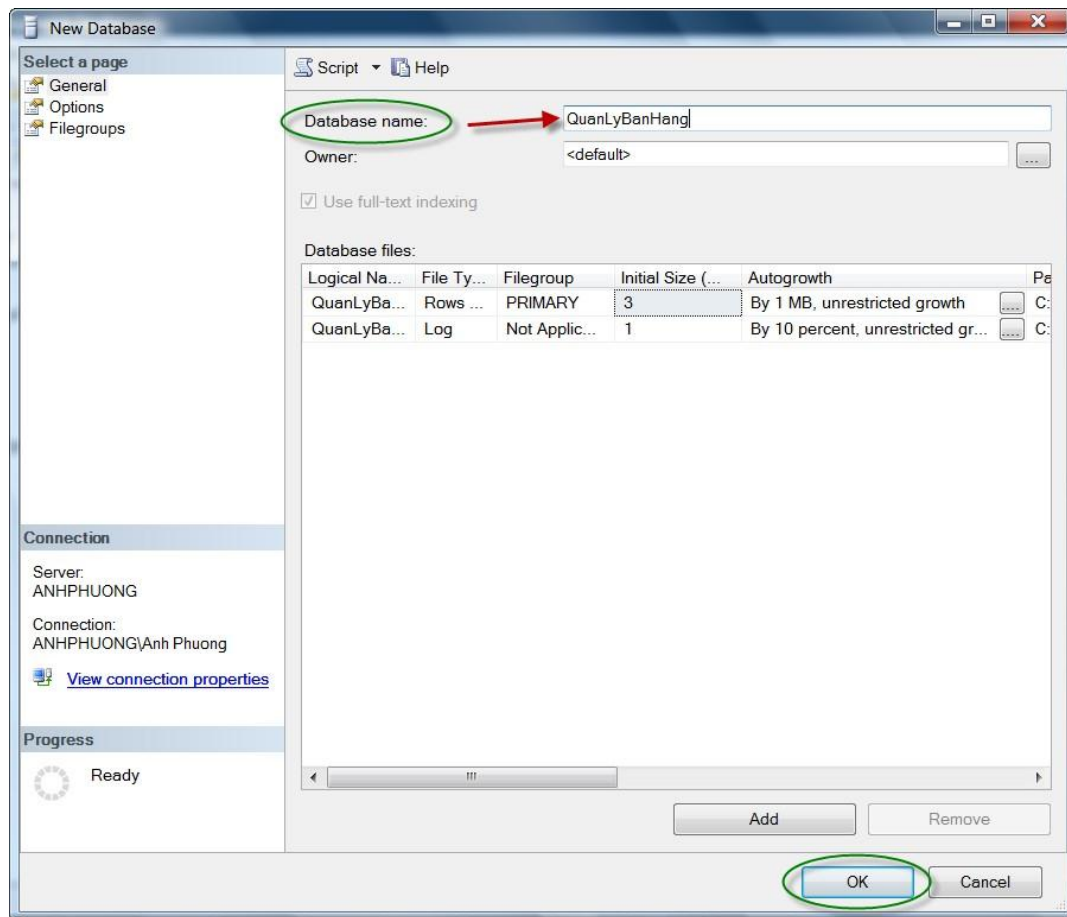
### Tạo cơ sở dữ liệu (database)

Chọn database -> Click phải -> Chọn New Database...



Hình 1.12. Hộp thoại Object Explorer

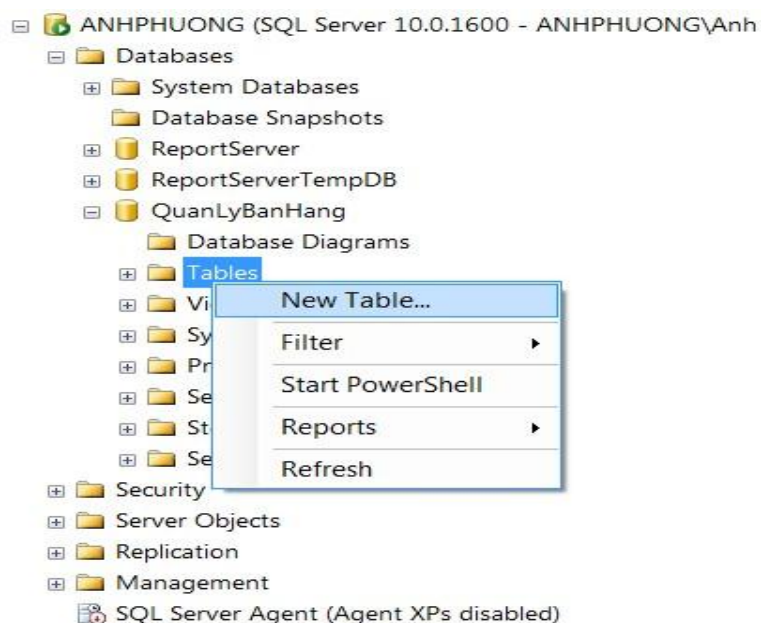
Trong hộp thoại New Database đặt tên cho database name -> Chọn OK



Hình 1.13. Giao diện New Database

### Tạo bảng (table)

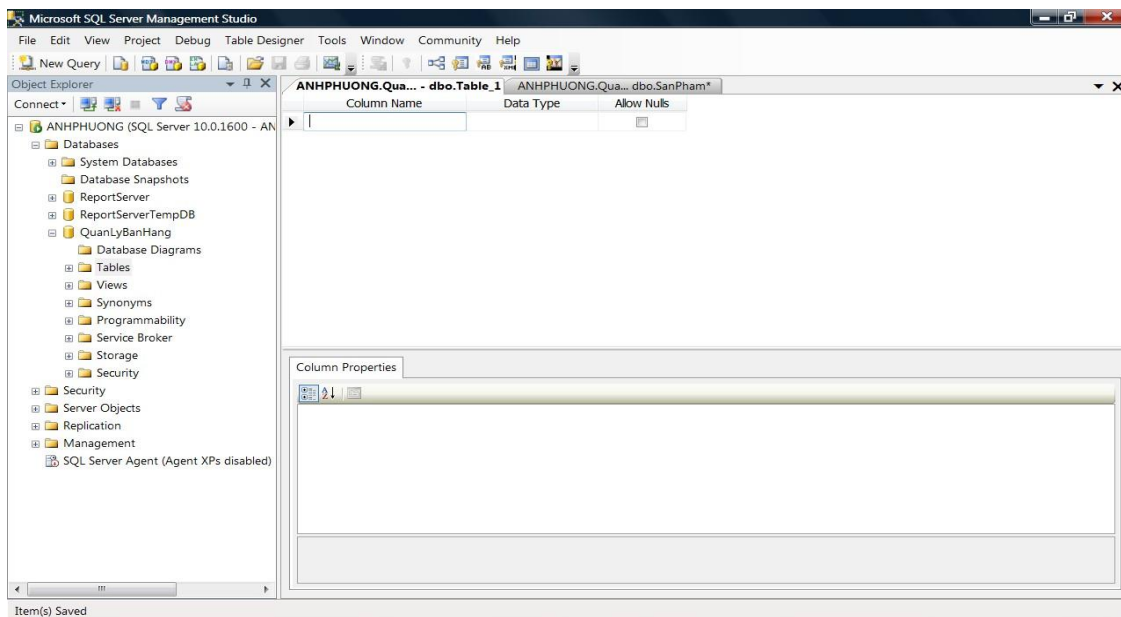
Vào database quản lý bán hàng chọn table. Sau đó click phải lên table -> Chọn New Table



Hình 1.14. Tạo bảng mới

Khi chọn New Table sẽ xuất hiện bên phải màn hình bên dưới

Sau đó ta nhập Column Name, Data Type... Nhấn Enter để nhập cột kết tiếp.




Hình 1.15. Nhập các thông tin tạo bảng



Lưu table trên thanh Standard toolbar -> chọn Save



Hình 1.16.

## Tạo quan hệ kết nối giữa các bảng (relationship)

1. **Tạo khóa chính (Primary key)** cho table trong SQL Server Management Studio, tạo cột và kiểu dữ liệu. Sau đó trên thanh toolbar, chọn nút Set Primary Key . Có thể click phải lên column chọn Set Primary Key.

	Column Name	Data Type	Allow Nulls
	MaSP	int	<input type="checkbox"/>
	TenSP	nvarchar(50)	<input checked="" type="checkbox"/>
	DonGia	float	<input checked="" type="checkbox"/>
	SoLuong	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Hình 1.17.

## 2. Tạo khóa ngoại (Foreign key) trong cửa sổ thiết kế table.

Foreign được dùng để liên kết các table lại với nhau. Cần lưu ý khi tạo foreign key là tên cột, kiểu dữ liệu giống tên cột của khóa chính mà table nó đại diện.

Ví dụ sau MaDMSP làm foreign key

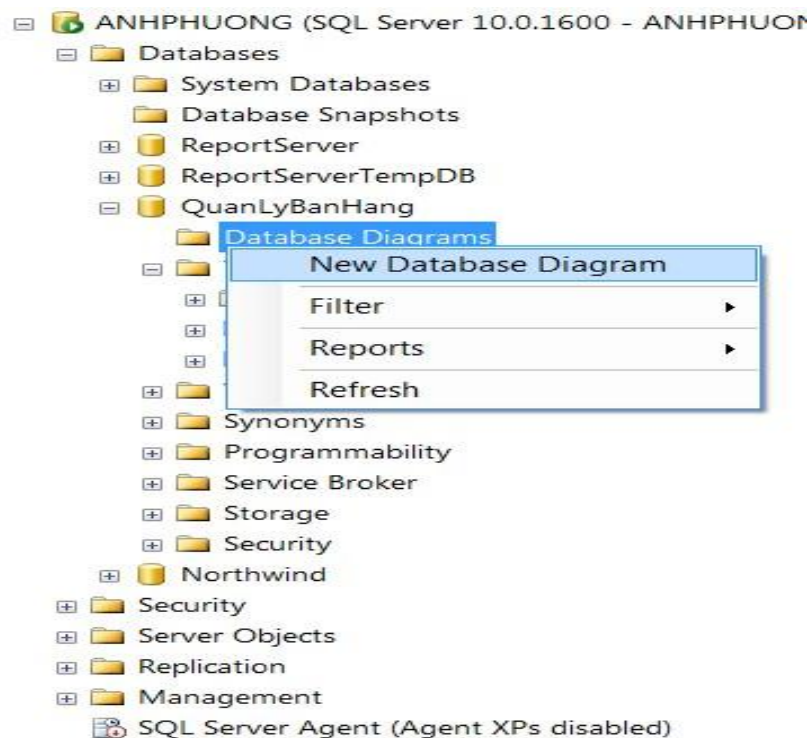
	Column Name	Data Type	Allow Nulls
🔑	MaSP	int	<input type="checkbox"/>
	TenSP	nvarchar(50)	<input checked="" type="checkbox"/>
	DonGia	float	<input checked="" type="checkbox"/>
	SoLuong	int	<input checked="" type="checkbox"/>
▶	MaDMSP	int	<input type="checkbox"/>
			<input type="checkbox"/>

Hình 1.18.

## Tạo sơ đồ (Diagrams)

Diagrams là 1 cửa sổ hiển thị mối quan hệ giữa các table của 1 database. Tạo diagram ta thực hiện như sau:

- Trong cửa sổ Object Explorer chọn tên database cần tạo -> Click phải vào Database Diagrams -> Chọn New Database Diagram



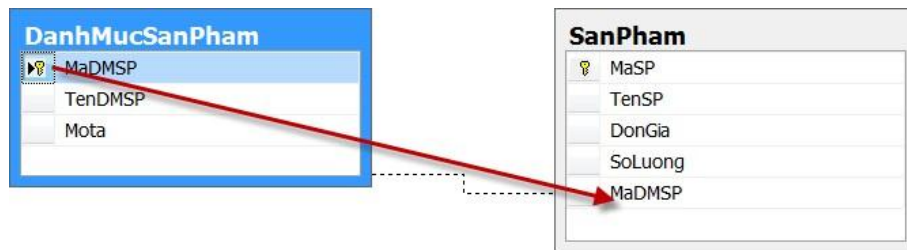
Hình 1.19.

- Sau khi chọn New Database Diagram sẽ xuất hiện hộp thoại để Add các table, sau khi add xong chọn Close.



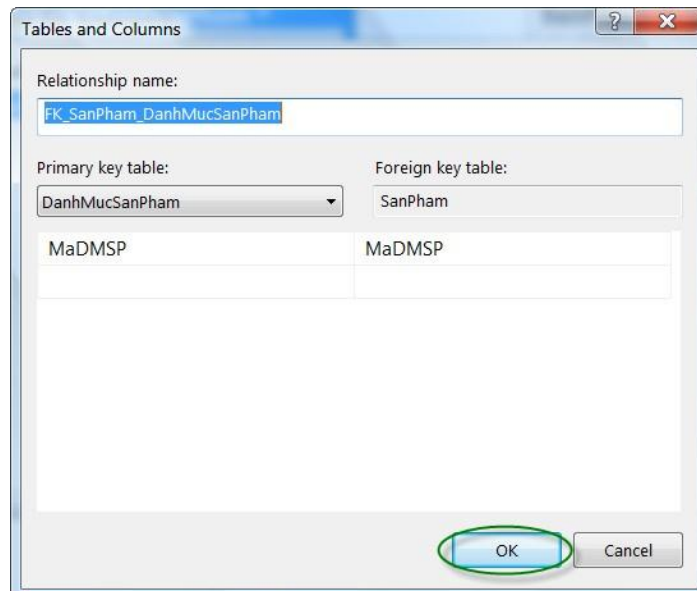
Hình 1.20.

Để thiết lập mối quan hệ giữa các table ta chọn cột dữ liệu của cột làm khóa chính trong bảng cha (parent table) và kéo nó đến khóa ngoại trong bảng con (child table)



Hình 1.21.

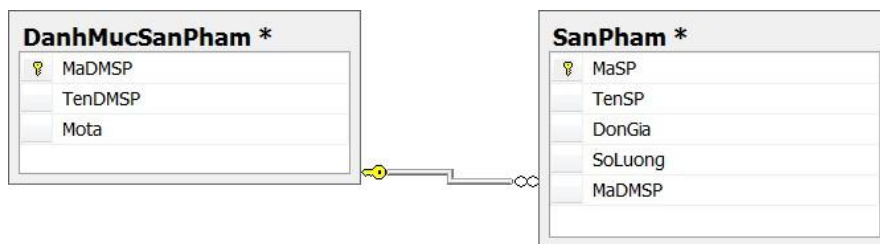
Sau khi kéo mối quan hệ cho 2 table sẽ xuất hiện hộp thoại như hình 1.22



Hình 1.22.

Khi ta chọn OK giữa 2 table sẽ xuất hiện một kết nối giữa 2 table





Hình 1.23.

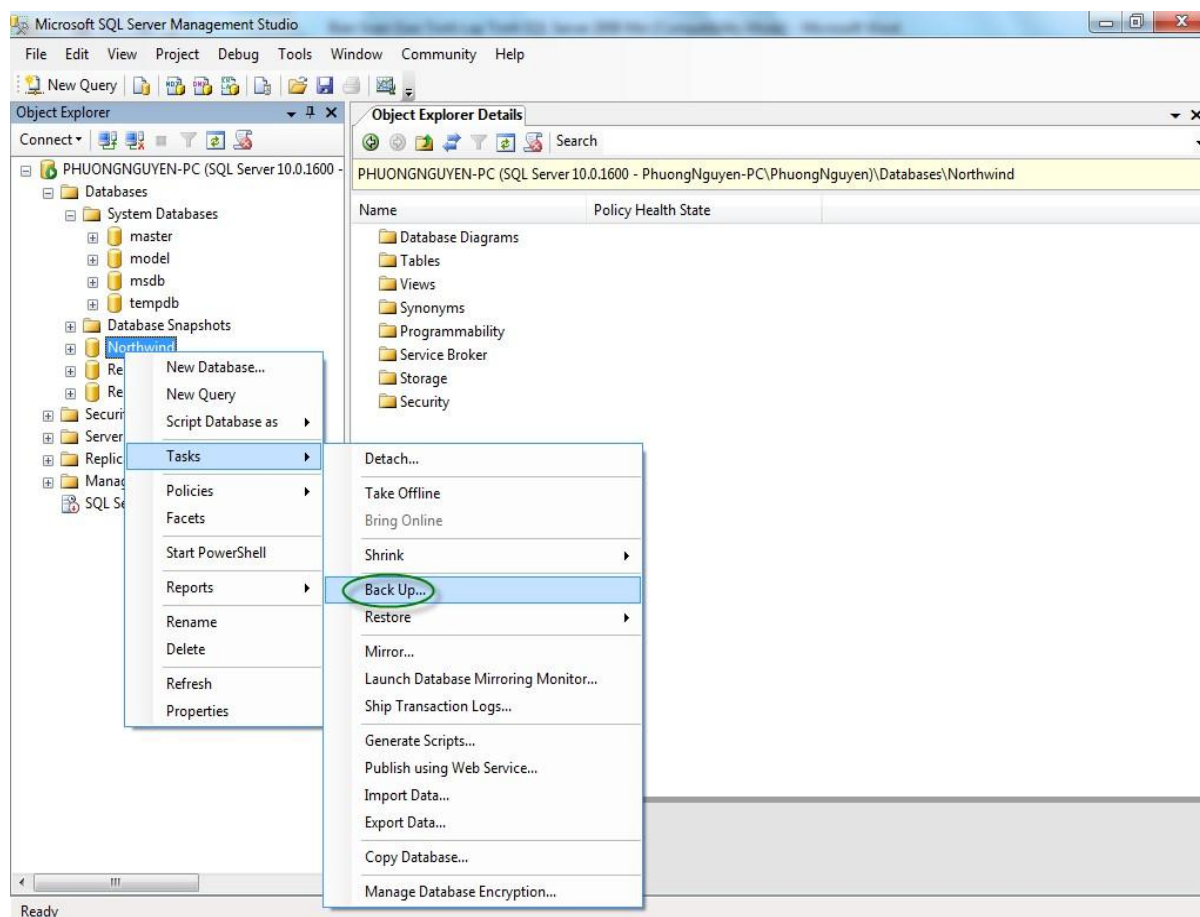
Trong cùng một cách, bạn có thể tạo mối quan hệ khác. Khi bạn đã hoàn tất, bạn có thể lưu và đóng diagram.

## Back up và Restore dữ liệu

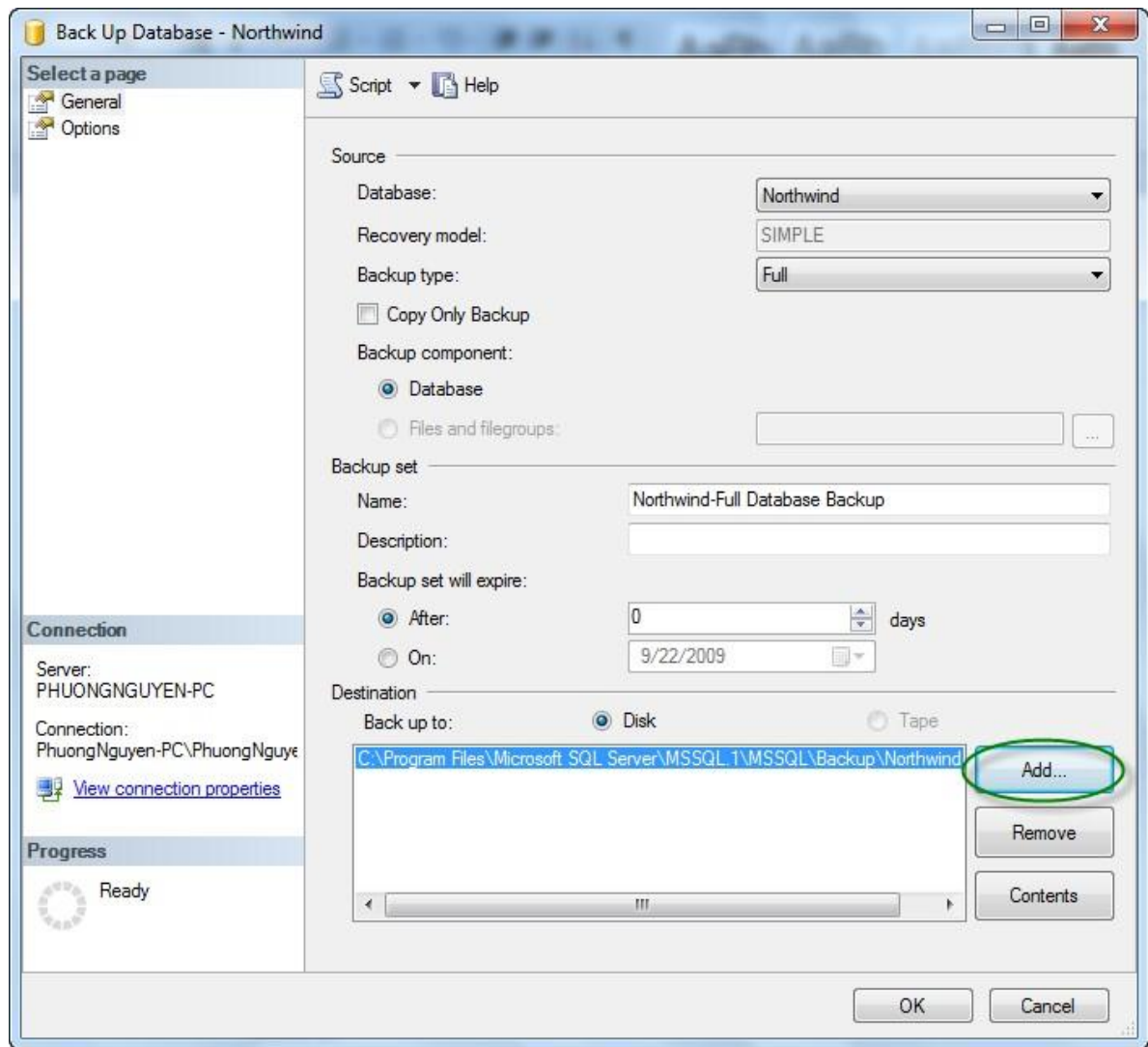
### Back up

Click phải vào database cần back up -> Chọn Tasks -> Chọn Back up...

Thực hiện các thao tác theo thứ tự các hình bên dưới



Hình 1.24.

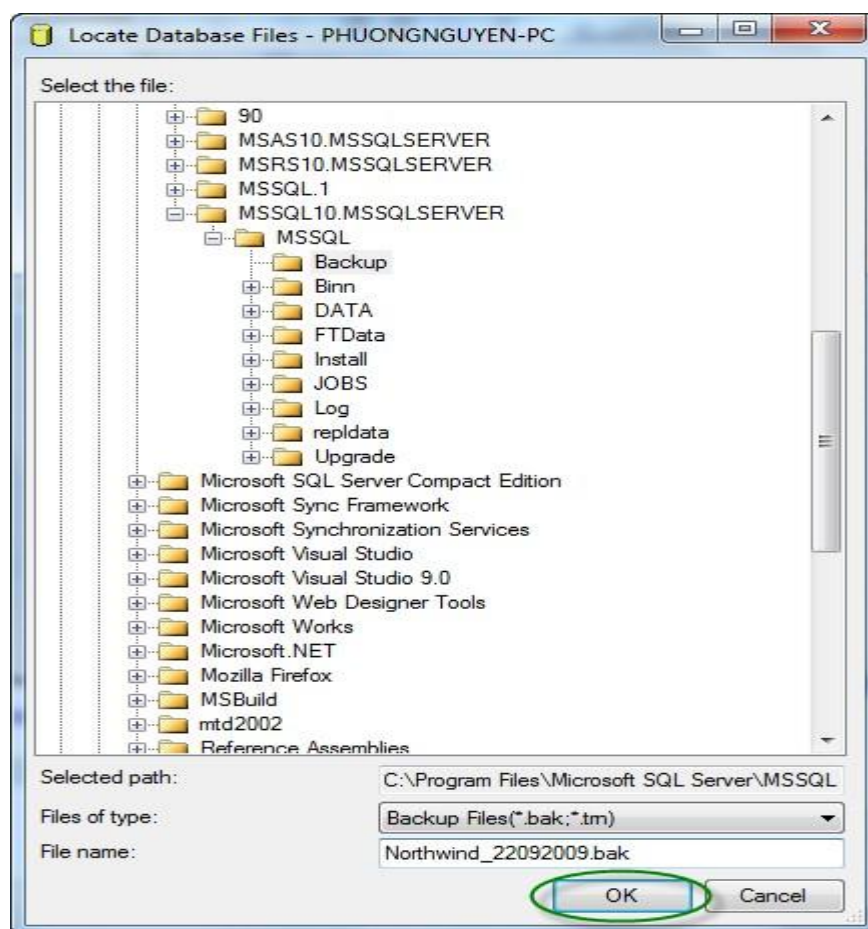


Hình 1.25.

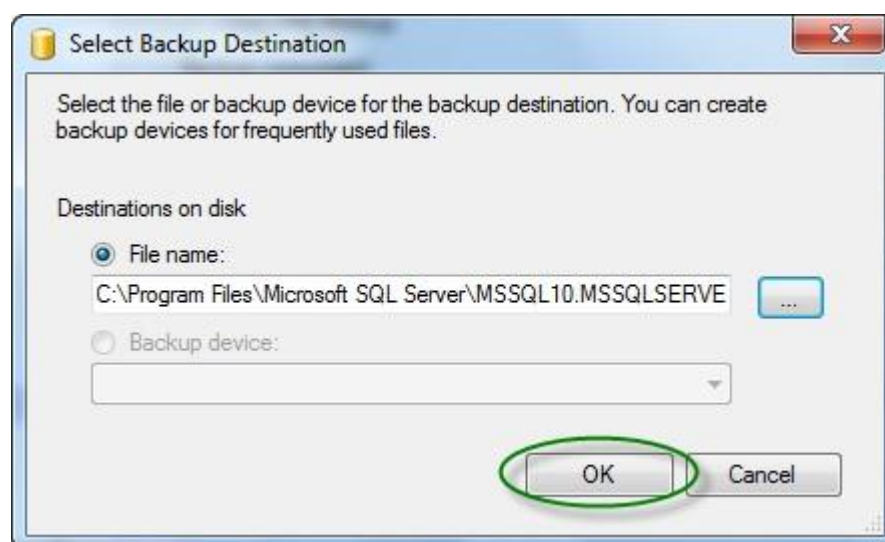


Hình 1.26.

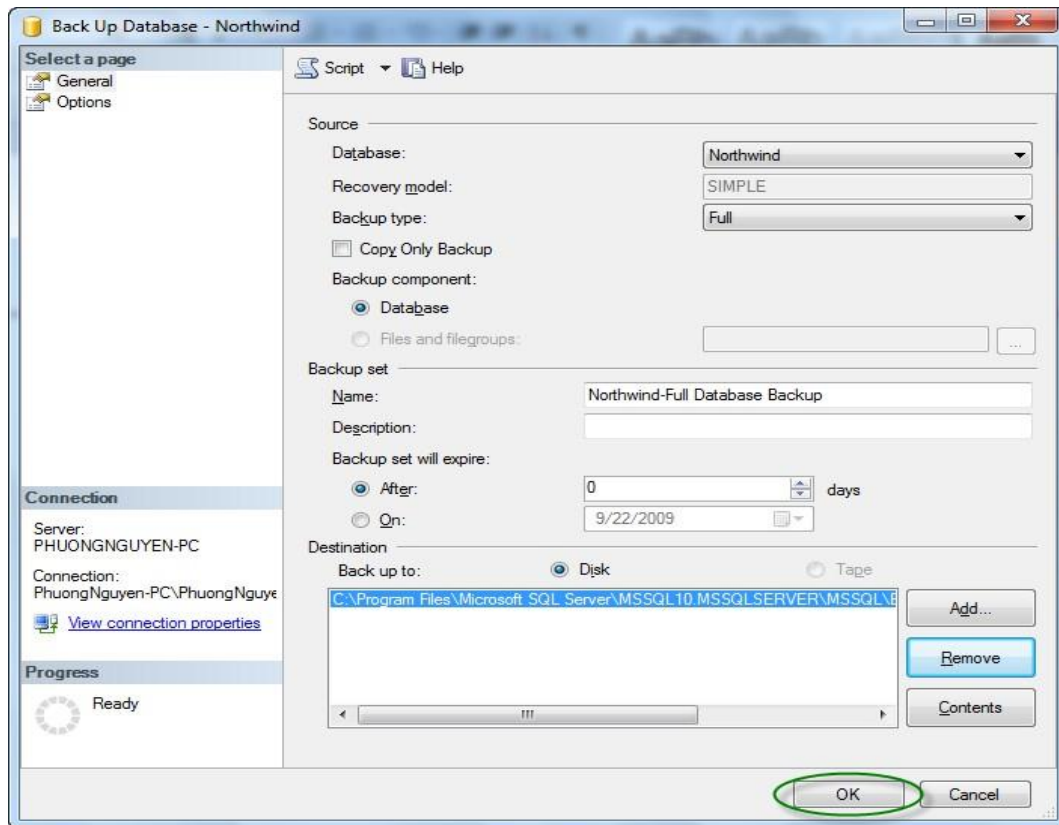




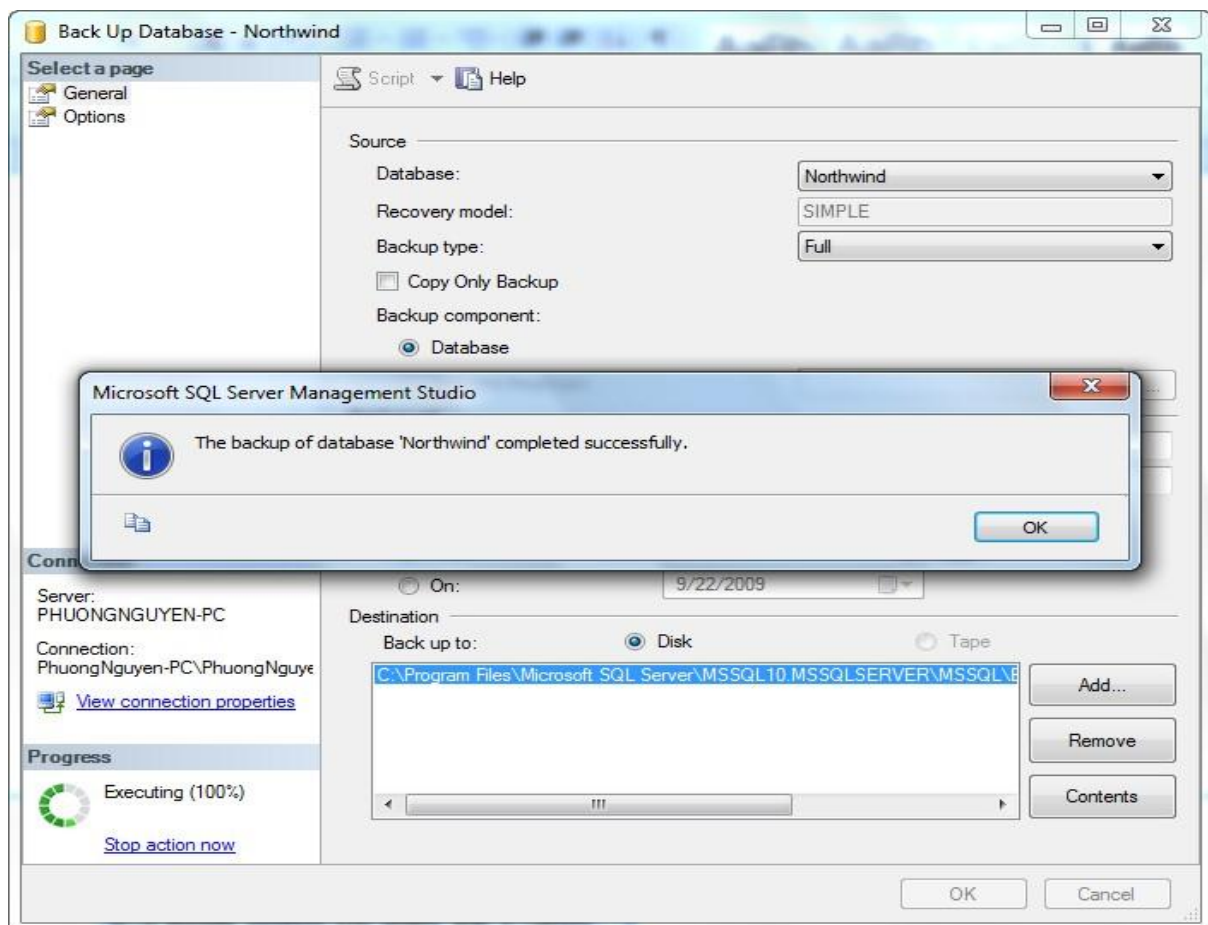
Hình 1.27.



Hình 1.28.



Hình 1.29.

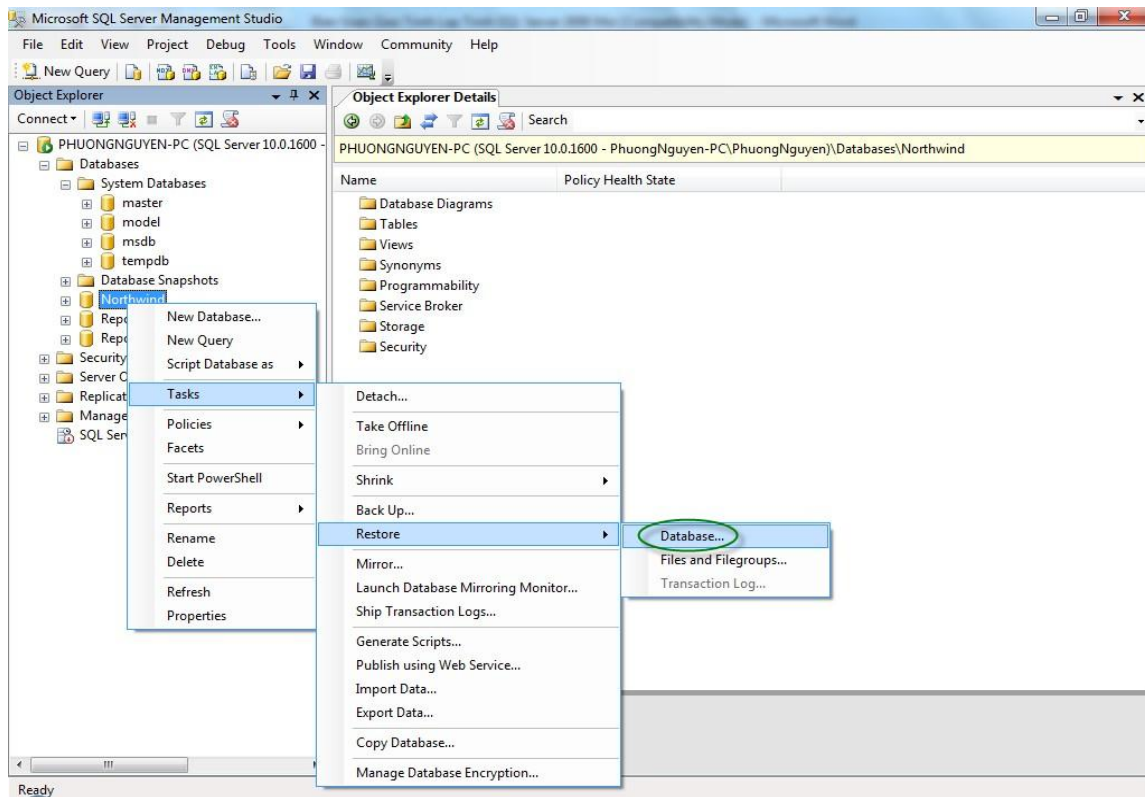


Hình 1.30.

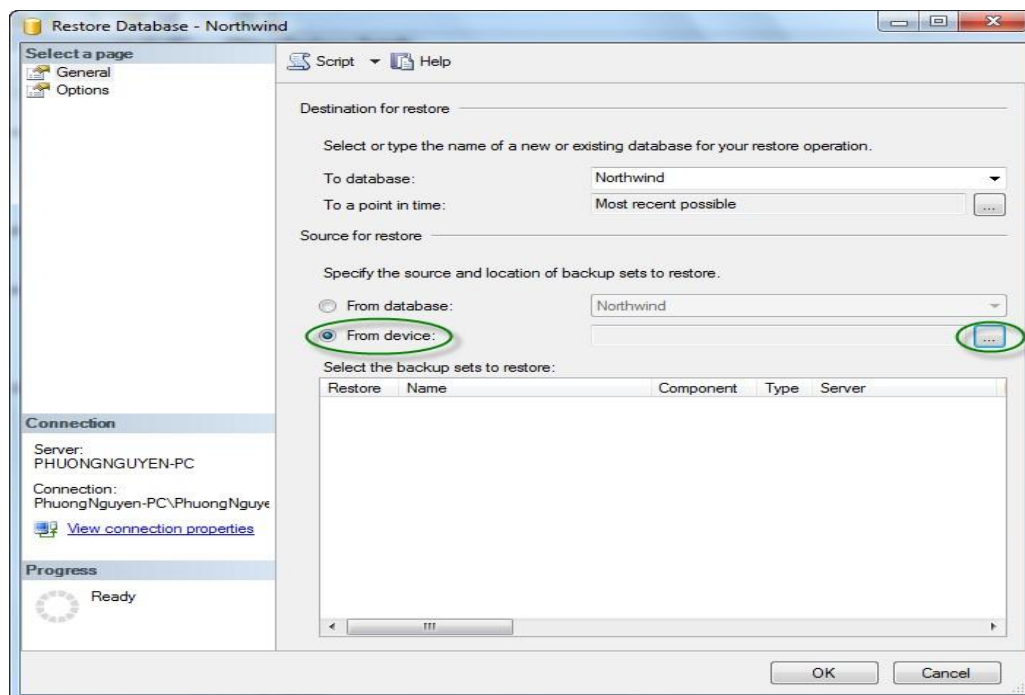
## Restore

Click phải vào database cần Restore -> Chọn Tasks -> Chọn Restore -> Database...

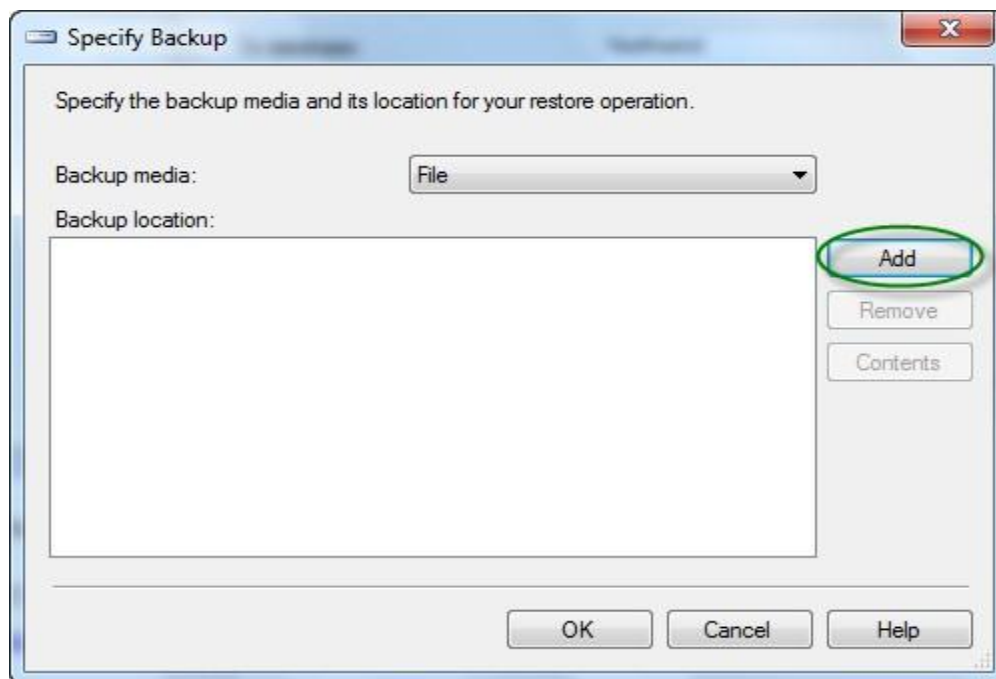
Thực hiện các thao tác theo thứ tự các hình bên dưới



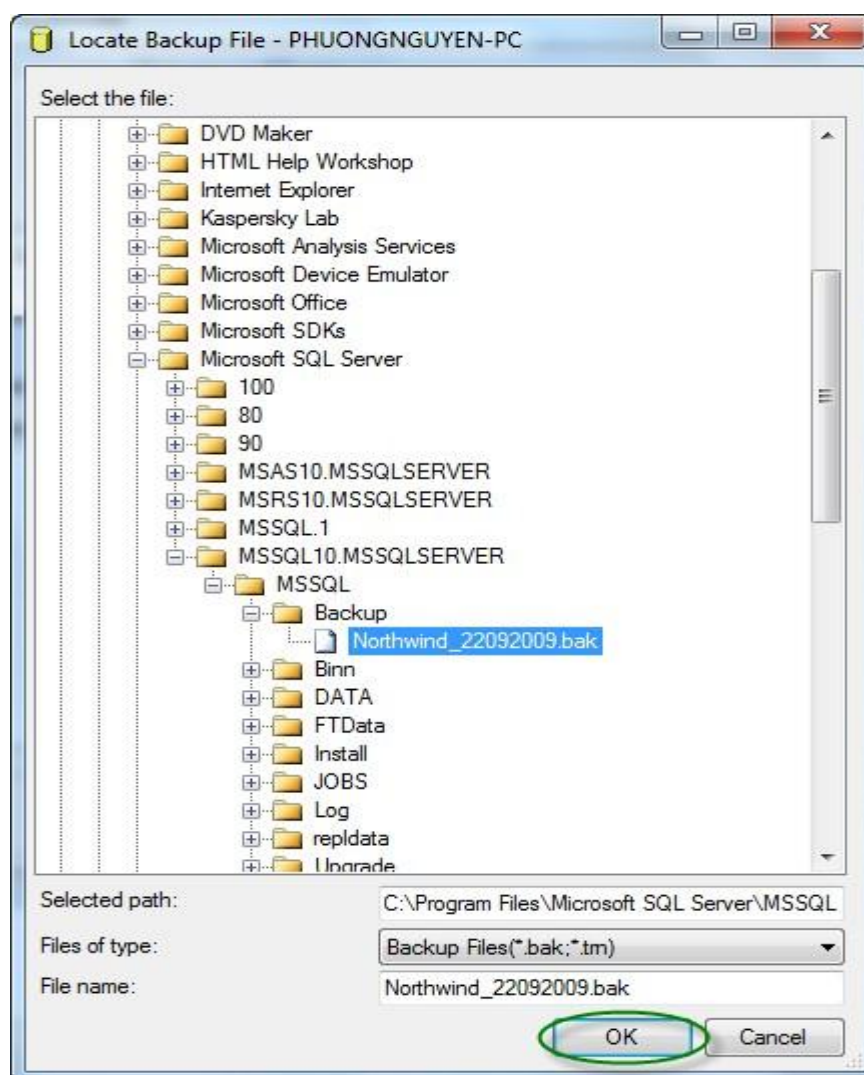
Hình 1.31.



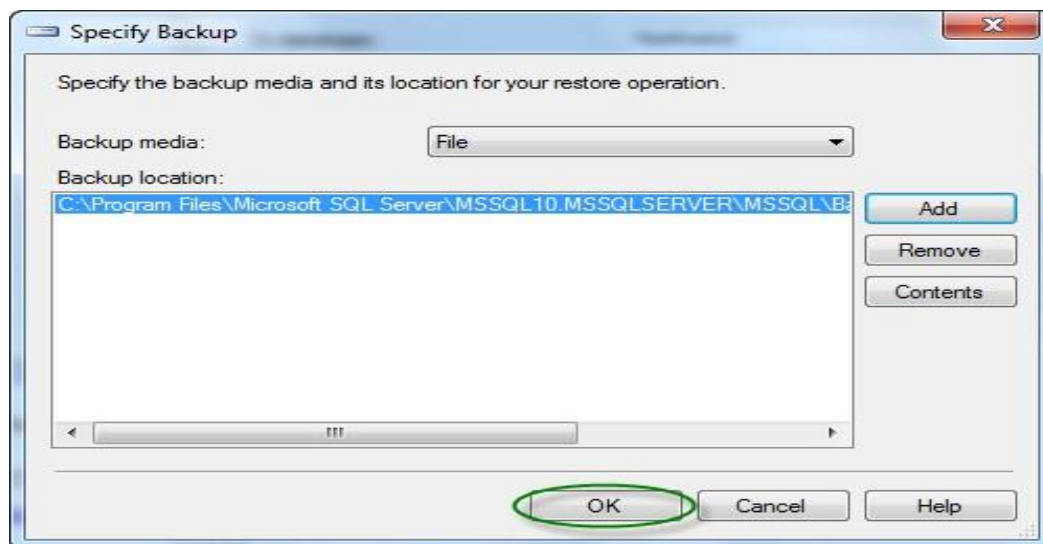
Hình 1.32.



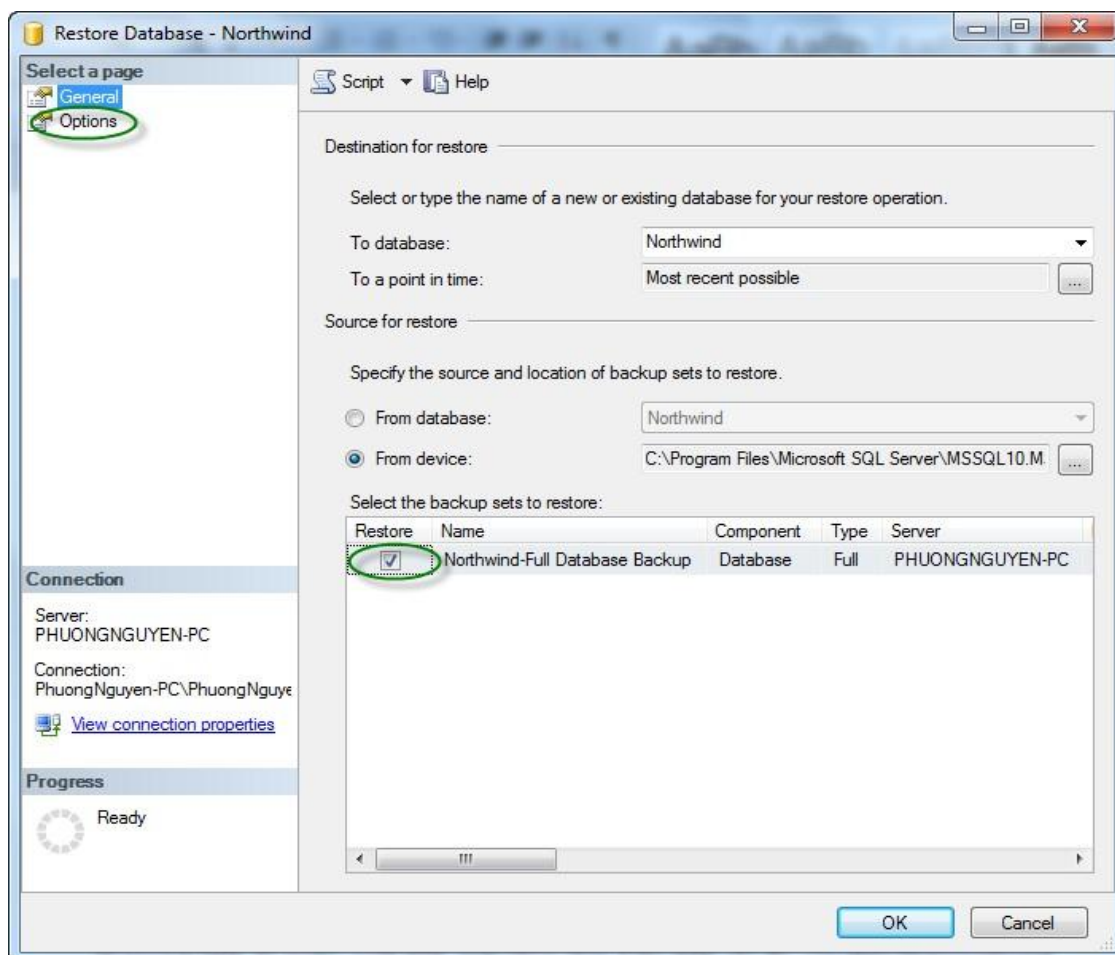
Hình 1.33



Hình 1.34

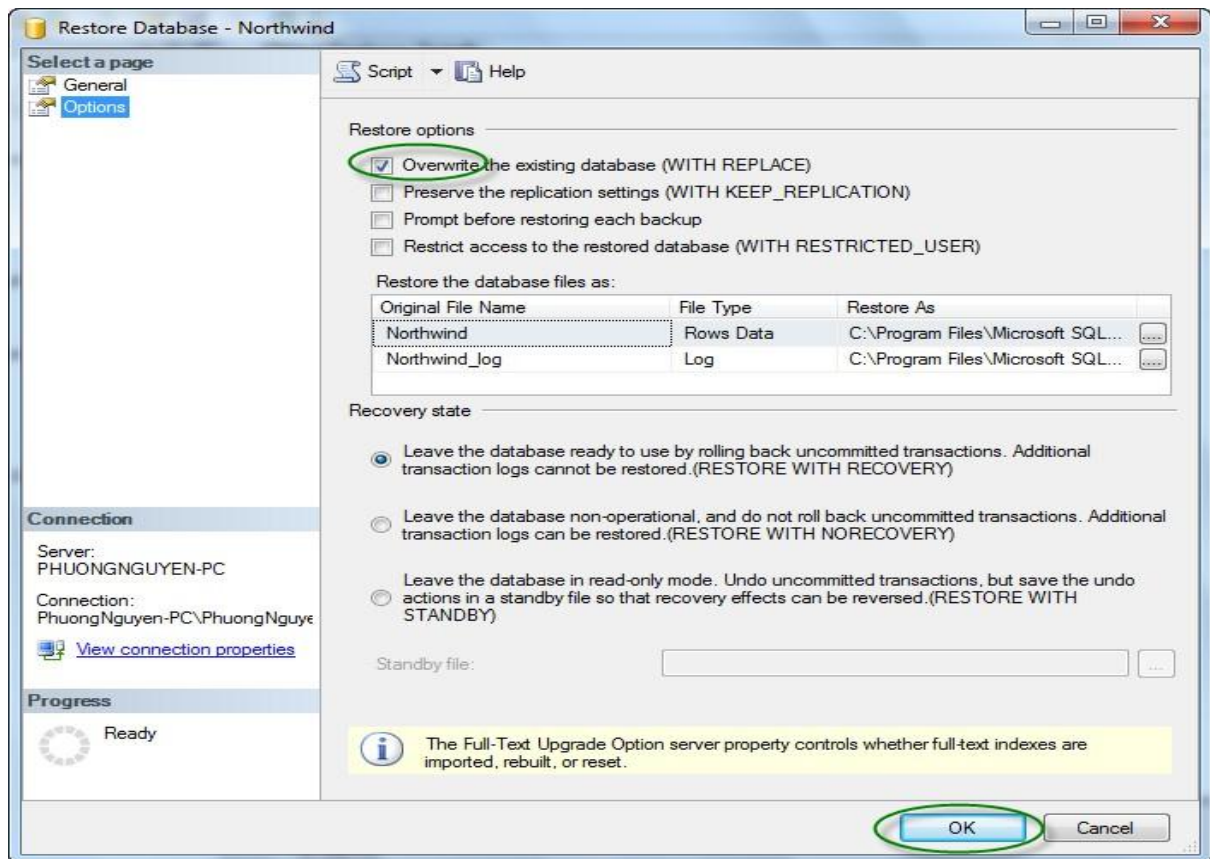


Hình 1.35

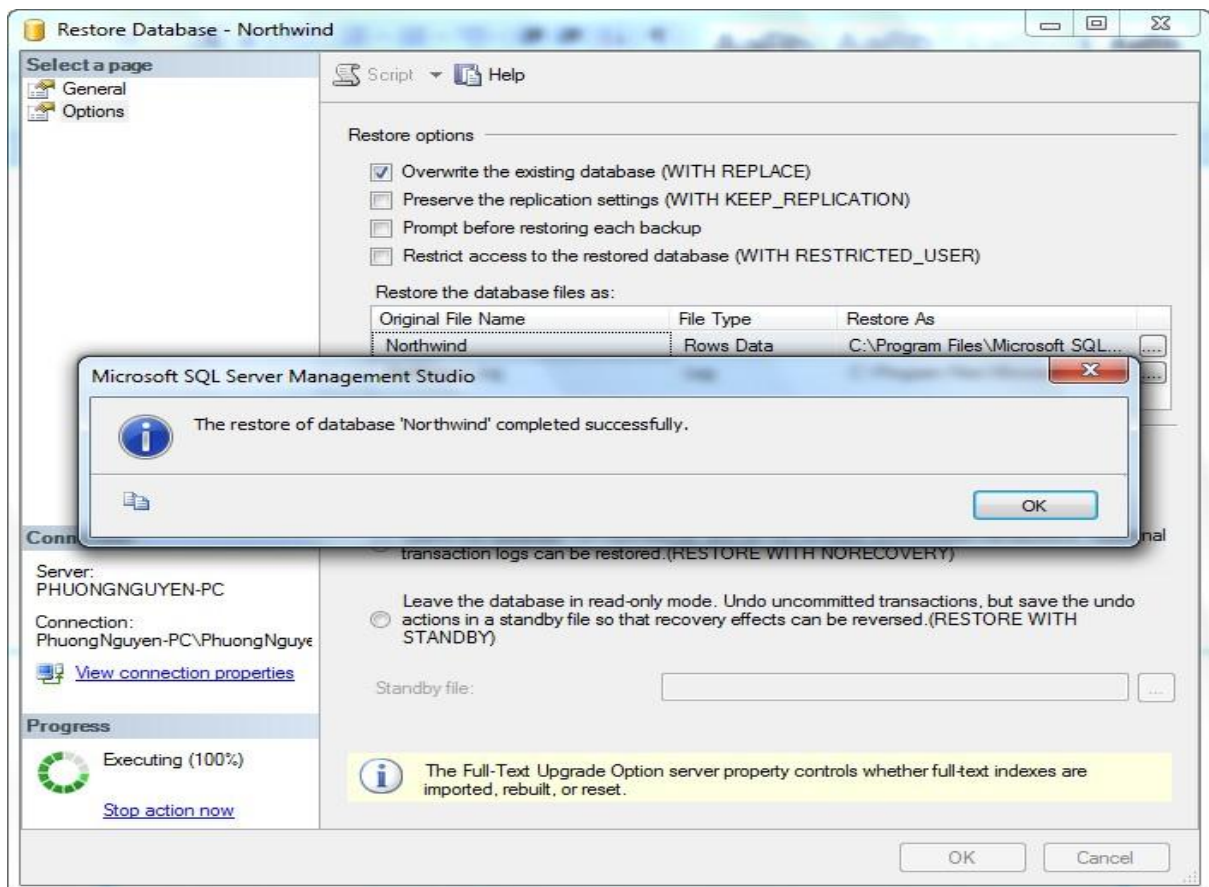


Hình 1.36





Hình 1.37



Hình 1.38