

SỞ LAO ĐỘNG - THƯƠNG BINH VÀ XÃ HỘI HÀ NỘI
TRƯỜNG TRUNG CẤP CÔNG NGHỆ VÀ DU LỊCH HÀ NỘI



GIÁO TRÌNH
MÔN ĐƠN: QUẢN TRỊ CƠ SỞ DỮ LIỆU VỚI SQL
SERVER
NGHỀ: CÔNG NGHỆ THÔNG TIN
TRÌNH ĐỘ: TRUNG CẤP

*(Ban hành kèm theo Quyết định số: 78/QĐ-CNNDL ngày 03 tháng 06 năm 2019
của Hiệu trưởng Trường Trung cấp Công nghệ và Du lịch Hà Nội)*

Hà Nội, năm 2019

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Ngôn ngữ hỏi có cấu trúc (SQL) và các hệ quản trị cơ sở dữ liệu quan hệ là một trong những nền tảng kỹ thuật quan trọng trong công nghiệp máy tính. Cho đến nay, có thể nói rằng SQL đã được xem là ngôn ngữ chuẩn trong cơ sở dữ liệu. Các hệ quản trị cơ sở dữ liệu quan hệ thương mại hiện có như Oracle, SQL Server, Informix, DB2,... đều chọn SQL làm ngôn ngữ cho sản phẩm của mình

So với chuẩn SQL do ANSI/ISO đề xuất, bản thân các hệ quản trị cơ sở dữ liệu quan hệ thương mại lại có thể có một số thay đổi nào đó; Điều này đôi khi dẫn đến sự khác biệt, mặc dù không đáng kể, giữa SQL chuẩn và SQL được sử dụng trong các hệ quản trị cơ sở dữ liệu cụ thể. Trong giáo trình này, chúng tôi chọn hệ quản trị cơ sở dữ liệu SQL Server của hãng Microsoft để sử dụng cho các ví dụ minh họa cũng như lời giải của các bài tập.

Khi biên soạn, nhóm biên soạn đã dựa trên kinh nghiệm thực tế giảng dạy, tham khảo đồng nghiệp, tham khảo các giáo trình hiện có và cập nhật những kiến thức mới có liên quan để phù hợp với nội dung chương trình đào tạo và phù hợp với mục tiêu đào tạo, nội dung được biên soạn gắn với nhu cầu thực tế.

Nội dung giáo trình được biên soạn với lượng thời gian đào tạo 60 giờ gồm có:

Bài 01 MĐ16-01: Tổng quan về SQL Server

Bài 02 MĐ16-02: Các thành phần cơ bản của SQL server

Bài 03 MĐ16-03: Giới thiệu một số công cụ trong SQL server

Bài 04 MĐ16-04: Phát biểu cơ bản T-SQL

Bài 05 MĐ16-05: Thành phần khác trong SQL server

Bài 06 MĐ16-06: Quản lý bảo mật và người dùng

Mặc dù đã cố gắng tổ chức biên soạn để đáp ứng được mục tiêu đào tạo nhưng không tránh được những thiếu sót. Rất mong nhận được sự đóng góp ý kiến của các thầy, cô và bạn đọc để nhóm biên soạn sẽ điều chỉnh hoàn thiện hơn.

Hà Nội, ngày tháng năm 2019

Biên soạn

Khoa Công Nghệ Thông Tin

MỤC LỤC

LỜI GIỚI THIỆU	2
MỤC LỤC	3
GIÁO TRÌNH MÔN HỌC/MÔ ĐUN	8
BÀI 1: GIỚI THIỆU VỀ SQL SERVER	10
Mã bài: MD 16- 01	10
1. Tổng quan SQL Server	10
1.1 Giới thiệu.....	10
1.1.1 <i>SQL là ngôn ngữ cơ sở dữ liệu quan hệ</i>	10
1.1.2 <i>Vai trò của SQL</i>	11
1.2. Sự quan trọng và cần thiết của chúng trong lập trình ứng dụng.....	12
2. Các thành phần quan trọng trong SQL Server	15
2.1. Relational Database Engine - Lõi của SQL Server.....	15
2.2. Replication - Cơ chế tạo bản sao (Replica):	15
2.3. Data Transformation Service (DTS).....	15
2.4. Analysis Service - Một dịch vụ phân tích dữ liệu	15
2.5. English Query	16
2.6. Meta Data Service	16
2.7. SQL Server Books Online	16
2.8 SQL Server Tools	16
3. Cài đặt SQL Server Express	16
3.1 Yêu cầu phần cứng	16
3.2 Các bước cài đặt SQL Server 2019.....	16
3.3. Cài đặt SQL Server Management Studio (SSMS)	30
Bài tập thực hành của học viên	34
Hướng dẫn thực hiện.....	34
Những trọng tâm cần chú ý:.....	35
BÀI 2: CÁC THÀNH PHẦN CƠ BẢN CỦA SQL SERVER	36
Mã bài: MD16-02	36
1. Khái niệm cơ bản về mô hình quan hệ	36
1.1 Mô hình dữ liệu quan hệ.....	36
1.2 Bảng (Table).....	37
1.3 Khoá của bảng.....	37
1.4 Mối quan hệ và khoá ngoài	38
1.5 Câu lệnh SQL	38
1.6 Quy tắc sử dụng tên trong SQL.....	40
2. Cấu trúc của SQL SERVER	40
2.1 Các cơ sở dữ liệu hệ thống của SQL SERVER	40
2.2. Cấu trúc vật lý của một SQL Server Database	41
2.3 Nguyên tắc hoạt động của Transaction Log Trong SQL Server.....	42
2.4 Cấu Trúc Logic Của Một SQL Server Database	44
Bài tập thực hành của học viên	44
Hướng dẫn thực hiện.....	44
Những trọng tâm cần chú ý:.....	45
Bài mở rộng và nâng cao	45

Yêu cầu đánh giá kết quả học tập	45
BÀI 3: GIỚI THIỆU MỘT SỐ CÔNG CỤ TRONG SQL SERVER.....	46
Mã bài: MD16-03	46
1. Sử dụng công cụ Enterprise manager.....	46
1.1. SQL enterprise manager.....	46
1.2. SQL server services manager.....	46
1.2.1 Các dịch vụ của SQL Server	46
1.2.2. Khởi động, tạm ngưng, dừng các dịch vụ của SQL Server	47
1.3 Các công cụ của MS SQL	47
1.3.1 Trình Enterprise Manager	47
1.3.2 Làm việc với công cụ Enterprise Manager.....	48
1.3.2.1 Tạo một CSDL mới.....	48
1.3.2.2 Xóa một CSDL	49
1.3.2.3 Sửa tham số.....	49
1.4 Bảng dữ liệu	49
1.4.1 Thiết kế bảng dữ liệu.....	49
1.4.2 Các khóa.	52
1.4.3 Xóa bảng	55
1.4.4 Nhập dữ liệu vào bảng	55
1.4.5 Tạo, sửa ràng buộc, khóa	55
1.5. Khóa Index.....	56
1.6 Tạo sơ đồ (Diagrams).....	57
1.7 Khung nhìn – View	59
1.7.1 Khái niệm khung nhìn.....	59
1.7.2 Tạo khung nhìn	60
1.8 Thủ tục lưu trữ	60
1.8.1 Khái niệm thủ tục lưu trữ và hàm.....	60
1.8.2 Phân loại thủ tục lưu trữ.....	61
1.8.3 Thiết lập thủ tục lưu trữ.....	61
1.8.4 Sửa, xóa thủ tục.....	62
1.9 TRIGGER.....	62
1.9.1 Khái niệm trigger	62
1.9.2 Những trường hợp sử dụng Trigger	62
1.9.3 Đặc điểm của Trigger.....	62
1.9.4 Tạo Trigger	63
1.10. Tạo liên kết cơ sở dữ liệu dữ liệu	63
1.10.1 Tạo liên kết từ Access.....	63
1.10.2 Tạo Server liên kết – Linked Server.....	65
1.11 / Dettach và Attach cơ sở dữ liệu	66
1.11.1 Copy tập tin của CSDL.	66
1.11.2 Dettach cơ sở dữ liệu.	66
1.11.3 Attach tập tin CSDL vào Instance.....	67
1.12 Import và Export cơ sở dữ liệu.....	67
1.12.1 Import – Nhập dữ liệu.....	67
1.12.2 Export – xuất dữ liệu.....	69

1.13 Back up và Restore dữ liệu.....	70
1.13.1 Back up.....	70
1.12.2. Restore.....	74
2. Cách dùng công cụ Query Analyzer để tạo ra các CSDL.....	80
2.1 Giới thiệu.....	80
2.2. Khởi động Query Analyzer	80
2.3. Thành phần chính của Query Analyzer	81
3. Thiết lập một số cấu hình về Client network utility	81
3.1 Giới thiệu về mô hình Client/Server và các hệ quản trị CSDL phục vụ cho mô hình Client/Server.....	81
3.2. Các đặc trưng của mô hình Client/server.....	81
3.3. Các thành phần quan trọng trong SQL Server	82
3.3.1 Relational Database Engine - Cái lõi của SQL Server.....	82
3.3.2 Replication - Cơ chế tạo bản sao	82
3.3.3 Data Transformation Service (DTS).....	83
3.3.4 Analysis Service	83
3.3.5. English Query - Một dịch vụ truy vấn	83
3.3.6. Meta Data Service	83
3.3.7. SQL Server Books Online	83
Bài tập thực hành của học viên.....	83
Hướng dẫn thực hiện	84
Những trọng tâm cần chú ý:	84
Bài mở rộng và nâng cao	85
BÀI 4: PHÁT BIỂU CƠ BẢN T-SQL	87
Mã bài: MD16-04	87
1. Cú pháp các câu lệnh T-SQL	87
1.1 Identifiers (định danh)	87
1.2. Biểu thức	88
1.3. Comments (Chú thích).....	88
1.4. Lệnh USE	88
1.5. Một số hàm của T-SQL	88
1.5.1 Các hàm về chuỗi ký tự.....	88
1.5.2 Các hàm ngày tháng	88
1.5.3 Các hàm số học.....	89
2 Data Definition Language (DDL):	90
2.1 Lệnh tạo cơ sở dữ liệu – Create database.....	90
2.1.1 Cách 1: Dùng Query Analyzer	90
2.2. Lệnh tạo bảng – CREATE TABLE	91
2.2.1 Cú pháp	91
2.2.3 Cột có khả năng tính toán - Cột ảo (Trường ảo)	92
2.3 Lệnh ALTER.....	92
2.3.1 ALTER DATABASE.....	92
2.3.2. Sửa đổi cấu trúc bảng – ALTER TABLE.....	93
2.4. Lệnh xóa – DROP	93
3. Manipulation Language (DML):	93

3.1. Lệnh truy vấn dữ liệu - <i>SELECT</i>	93
3.2 Truy vấn sử dụng phép nối bảng	100
3.2.1 Kết nối theo chiều ngang.....	100
3.3.2 Kết nối theo chiều dọc – Dùng phép hợp (<i>UNION</i>).....	102
3.3 Bổ sung dữ liệu	103
3.3.1 Bổ sung từng dòng dữ liệu với lệnh <i>INSERT</i>	103
3.3.2 Bổ sung nhiều dòng dữ liệu từ bảng khác	103
3.4 Cập nhật dữ liệu	103
3.5 Xóa dữ liệu.....	104
4 Variables (Biến) trong SQL Server	104
4.1 Khai báo biến	104
4.2 Gán giá trị cho biến.....	105
5.1. Phát biểu <i>IF ...ELSE</i>	105
5.2 Phát biểu <i>CASE</i>	105
5.3 Phát biểu <i>WHILE</i>	106
Bài tập thực hành của học viên	107
Hướng dẫn thực hiện.....	113
Những trọng tâm cần chú ý:.....	115
Bài mở rộng và nâng cao.....	115
Yêu cầu đánh giá kết quả học tập.....	119
BÀI 5: THÀNH PHẦN KHÁC TRONG SQL SERVER	121
Mã bài: MD16-05	121
1. Giới thiệu về View.....	121
1.1 Khái niệm về View	121
1.2. Ưu điểm của View	122
1.3 Cách dùng view để lọc dữ liệu.....	123
1.3.1 Cách dùng view.....	123
1.3.2 Tạo khung nhìn - Khung nhìn như bộ lọc	124
1.4. Các hạn chế của view.....	125
1.5. Cách cập nhật dữ liệu vào view.....	125
1.5.1 cập nhật dữ liệu vào view.....	125
1.5.2. Sửa đổi khung nhìn	127
1.5.3. Xoá khung nhìn.....	128
1.5.4 Đổi tên View	128
2.STORE PROCEDURE	129
2.1 Thủ tục lưu trữ (Stored procedure).....	129
2.2. Hàm do người dùng định nghĩa (User Defined Function-UDF).....	132
3 Trigger.....	136
3.1 Các đặc điểm của trigger.....	136
3.2 Các trường hợp sử dụng trigger	136
3.3 Khả năng sau của trigger.....	136
3.4 Định nghĩa trigger	136
3.5 Kích hoạt trigger dựa trên sự thay đổi dữ liệu trên cột.....	139
3.6 Sử dụng trigger và Giao tác (<i>TRANSACTION</i>).....	139
3.7. DDL TRIGGER	140

3.8. Enable/ Disable TRIGGER	141
Bài tập thực hành của học viên	142
Hướng dẫn thực hiện.....	142
Những trọng tâm cần chú ý:.....	145
Bài mở rộng và nâng cao	145
Yêu cầu đánh giá kết quả học tập.....	146
BÀI 6 QUẢN LÝ BẢO MẬT VÀ NGƯỜI DÙNG	147
1. Khái niệm cơ bản về bảo mật.....	147
2 Các chế độ bảo mật	148
2.1 Windows Authentication	148
2.2 SQL Server Authentication	148
2.3 Thiết lập chế độ chứng thực Windows và SQL Server.....	148
3. Quản lý người dùng	149
3.1 Tạo tài khoản đăng nhập cho người dùng (Login).....	149
3.1.1 Thực hiện bằng giao diện	149
3.2 Cấp phát quyền truy cập vào CSDL	151
4. Quyền người dùng.....	151
4.1 Tạo người sử dụng từ giao diện	151
4.2 Tạo người sử dụng từ lệnh.....	152
4.1 Cấp phát quyền.....	155
4.2 Thu hồi quyền.....	157
Bài tập thực hành của học viên	158
Hướng dẫn thực hiện.....	159
Những trọng tâm cần chú ý:.....	160
Bài mở rộng và nâng cao	160
Yêu cầu đánh giá kết quả học tập.....	161
BÀI TẬP TỔNG HỢP	162

GIÁO TRÌNH MÔN HỌC/MÔ ĐUN

Tên môn học/mô đun: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU (MS SQL SERVER)

Mã môn học/mô đun: MD 16

Vị trí, tính chất, ý nghĩa và vai trò của môn học/mô đun:

- Vị trí: Môn học được bố trí sau khi sinh viên học xong mô đun: Học chung, Lập trình căn bản, Mạng máy tính, Cấu trúc dữ liệu, Nhập môn lập trình Web, Cơ sở dữ liệu, ...

- Tính chất của mô đun: Là mô đun kỹ thuật cơ sở bắt buộc trong chương trình đào tạo cao đẳng Ứng dụng phần mềm

- Ý nghĩa và vai trò: Mô đun được bố trí cho sinh viên học vào đầu năm thứ ba là môn chuyên môn nghề.

- Vai trò: Giáo trình “Hệ Quản trị cơ sở dữ liệu (MS SQL Server)” nhằm cung cấp cho sinh viên những kiến thức cơ bản về quản trị và khai thác CSDL

Mô đun Hệ Quản trị cơ sở dữ liệu (MS SQL Server) là mô đun đóng vai trò quan trọng trong các môn học, mô đun đào tạo nghề áp dụng trong việc xây dựng, quản lý và cấp quyền khi cần cho các mô đun lập trình CSDL.

Mục tiêu của mô đun:

- Về kiến thức:

- + Mô tả các thành phần hệ quản trị cơ sở dữ liệu, các khái niệm về cơ sở dữ liệu quan hệ hướng đối tượng và cơ sở dữ liệu quan hệ, ngôn ngữ MS SQL.
- + Trình bày các kiến trúc của hệ quản trị cơ sở dữ liệu MS SQL Server, cách làm việc và tương tác giữa các thành phần kiến trúc trong hệ thống.
- + Trình bày Kết nối hệ thống mạng để sử dụng hệ thống cơ sở dữ liệu.
- + Thực hiện thành thạo các thao tác quản trị tài khoản người dùng và tài khoản nhóm đối với hệ thống MS SQL Server.
- + Trình bày các bước cấu hình và giải quyết các vấn đề thường xảy ra trên mạng khi sử dụng truy cập cơ sở dữ liệu.
- + Trình bày các bước Bảo vệ tài nguyên dữ liệu trên các hệ thống MS SQL Server.

- Về kỹ năng:

- + Thực hành các thành phần hệ quản trị cơ sở dữ liệu và cơ sở dữ liệu quan hệ, ngôn ngữ MS SQL.
- + Thực hiện các kiến trúc của hệ quản trị cơ sở dữ liệu MS SQL Server, cách làm việc và tương tác giữa các thành phần kiến trúc trong hệ thống.
- + Thực hiện Kết nối hệ thống mạng để sử dụng hệ thống cơ sở dữ liệu.
- + Thực hiện thành thạo các thao tác quản trị tài khoản người dùng và tài khoản nhóm đối với hệ thống MS SQL Server.
- + Thiết lập cấu hình và giải quyết các vấn đề thường xảy ra trên mạng khi sử dụng truy cập cơ sở dữ liệu.
- + Bảo vệ tài nguyên dữ liệu trên các hệ thống MS SQL Server.

- Về năng lực tự chủ và trách nhiệm:

- + Nghiêm túc, tỉ mỉ trong quá trình tiếp cận với công cụ mới
- + Chủ động sáng tạo tìm kiếm các ứng dụng quản trị CSDL.
- + Bố trí làm việc khoa học đảm bảo an toàn cho người và phương tiện học tập.

Nội dung của môn học/mô đun:

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành, thí nghiệm, thảo luận, bài tập	Kiểm tra*
1	Tổng quan về SQL Server	4	2	2	
2	Các thành phần cơ bản của SQL server	4	2	2	
3	Giới thiệu một số công cụ trong SQL server	8	4	4	
4	Phát biểu cơ bản T-SQL	20	10	9	1
5	Thành phần khác trong SQL server	16	8	7	1
6	Quản lý bảo mật và người dùng	8	4	3	1
Cộng		60	30	27	3

BÀI 1: GIỚI THIỆU VỀ SQL SERVER

Mã bài: MĐ 16- 01

Mục tiêu:

- Hiểu được lịch sử phát triển và sự cần thiết của SQL SERVER trong thời đại ngày nay;
- Xác định được các cấu trúc CSDL cơ sở nhằm đảm bảo thao tác dữ liệu hiệu quả.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Tổng quan SQL Server

1.1 Giới thiệu

Ngôn ngữ hỏi có cấu trúc (SQL), có tiền thân là SEQUEL, là một ngôn ngữ được IBM phát triển và sử dụng trong hệ cơ sở dữ liệu thử nghiệm có tên là System/R vào năm 1974, chính thức được ANSI/ISO công nhận là một chuẩn ngôn ngữ sử dụng trong cơ sở dữ liệu quan hệ vào năm 1986. Cho đến hiện nay, SQL đã được sử dụng phổ biến trong các hệ quản trị cơ sở dữ liệu thương mại và có vai trò quan trọng trong những hệ thống này.

Vậy thực sự SQL là gì? Tại sao nó lại quan trọng trong các hệ quản trị cơ sở dữ liệu? SQL có thể làm được những gì và như thế nào? Nó được sử dụng ra sao trong các hệ quản trị cơ sở dữ liệu quan hệ? Nội dung của chương này sẽ cung cấp cho chúng ta cái nhìn tổng quan về SQL và một số vấn đề liên quan.

Phiên bản đầu tiên của Microsoft SQL Server ra đời đầu tiên vào năm 1989 cho các hệ điều hành chạy 16 bit với SQL Server phiên bản 1.0 và tiếp tục phát triển cho tới ngày nay.

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới được phát hành bởi Microsoft. Đây là một phần mềm chuyên dùng để thiết kế truy xuất dữ liệu theo yêu cầu của các phần mềm khác. Hay đơn giản chúng được cài làm server máy chủ nào đó.

Đối với người dùng là học sinh, sinh viên mới học lập trình thì đây chính là công cụ giúp bạn làm quen với cơ sở dữ liệu. Các môn học liên quan đến cơ sở dữ liệu, truy vấn, tạo bảng, kết nối chúng với nhau... Đặc biệt với lập trình.Net, khi kết nối với cơ sở dữ liệu local thì đều phải dùng đến phần mềm này. Ví dụ khi bạn sử dụng Winform để lập trình một chương trình nào có có kết nối CSDL.

Chủ yếu người dùng sẽ thao tác với SQL Management Studio. Đây chính là một phần mềm nhỏ trong bộ SQL Server, nó cung cấp giao diện cho người dùng thao tác, thực hiện các câu lệnh truy vấn, xây dựng CSDL... Cái này là cái mà người dùng cần thao tác đến nhiều nhất. Tuy nhiên nếu thiếu bất kì thành phần nào trong bộ SQL server thì phần mềm sẽ không hoạt động được. Và cái chúng ta cần cài đặt đó là: Microsoft SQL Server 2019 express.

1.1.1 SQL là ngôn ngữ cơ sở dữ liệu quan hệ

Mục tiêu: Hiểu được ngôn ngữ SQL là gì.

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

Tên gọi ngôn ngữ hỏi có cấu trúc phần nào làm chúng ta liên tưởng đến một công cụ (ngôn ngữ) dùng để truy xuất dữ liệu trong các cơ sở dữ liệu. Thực sự mà nói, khả

năng của SQL vượt xa so với một công cụ truy xuất dữ liệu, mặc dù đây là mục đích ban đầu khi SQL được xây dựng nên và truy xuất dữ liệu vẫn còn là một trong những chức năng quan trọng của nó. SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị cơ sở dữ liệu cung cấp cho người dùng bao gồm:

- Định nghĩa dữ liệu: SQL cung cấp khả năng định nghĩa các cơ sở dữ liệu, 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: Với SQL, người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các cơ sở dữ liệu.
- 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 cơ sở dữ liệu
- Đảm bảo toàn vẹn dữ liệu: SQL định nghĩa các ràng buộc toàn vẹn trong cơ sở dữ liệu 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.

Như vậy, có thể nói rằng SQL là một ngôn ngữ hoàn thiện được sử dụng trong các hệ thống cơ sở dữ liệu và là một thành phần không thể thiếu trong các hệ quản trị cơ sở dữ liệu. Mặc dù SQL không phải là một ngôn ngữ lập trình như C, C++, Java,... song các câu lệnh mà SQL cung cấp có thể được nhúng vào trong các ngôn ngữ lập trình nhằm xây dựng các ứng dụng tương tác với cơ sở dữ liệu.

Khác với các ngôn ngữ lập trình quen thuộc như C, C++, Java,... SQL là ngôn ngữ có tính khai báo. Với SQL, người dùng chỉ cần mô tả các yêu cầu cần phải thực hiện trên cơ sở dữ liệu mà không cần phải chỉ ra cách thức thực hiện các yêu cầu như thế nào. Chính vì vậy, SQL là ngôn ngữ dễ tiếp cận và dễ sử dụng.

1.1.2 Vai trò của SQL

Bản thân SQL không phải là một hệ quản trị cơ sở dữ liệu, nó không thể tồn tại độc lập. SQL thực sự là một phần của hệ quản trị cơ sở dữ liệu, nó xuất hiện trong các hệ quản trị cơ sở dữ liệu với vai trò ngôn ngữ và là công cụ giao tiếp giữa người sử dụng và hệ quản trị cơ sở dữ liệu.

Trong hầu hết các hệ quản trị cơ sở dữ liệu quan hệ, SQL có những vai trò như sau:

- SQL là ngôn ngữ hỏi có tính tương tác: Người sử dụng có thể dễ dàng thông qua các trình tiện ích để gửi các yêu cầu dưới dạng các câu lệnh SQL đến cơ sở dữ liệu và nhận kết quả trả về từ cơ sở dữ liệu
- SQL là ngôn ngữ lập trình cơ sở dữ liệu: Các lập trình viên có thể nhúng các câu lệnh SQL vào trong các ngôn ngữ lập trình để xây dựng nên các chương trình ứng dụng giao tiếp với cơ sở dữ liệu
- SQL là ngôn ngữ quản trị cơ sở dữ liệu: Thông qua SQL, người quản trị cơ sở dữ liệu có thể quản lý được cơ sở dữ liệu, định nghĩa các cấu trúc lưu trữ dữ liệu, điều khiển truy cập cơ sở dữ liệu,...
- SQL là ngôn ngữ cho các hệ thống khách/chủ (client/server): Trong các hệ thống cơ sở dữ liệu khách/chủ, SQL được sử dụng như là công cụ để giao tiếp giữa các trình ứng dụng phía máy khách với máy chủ cơ sở dữ liệu.
- SQL là ngôn ngữ truy cập dữ liệu trên Internet: Cho đến nay, hầu hết các máy chủ Web cũng như các máy chủ trên Internet sử dụng SQL với vai trò là ngôn ngữ để tương tác với dữ liệu trong các cơ sở dữ liệu.
- SQL là ngôn ngữ cơ sở dữ liệu phân tán: Đối với các hệ quản trị cơ sở dữ liệu phân tán, mỗi một hệ thống sử dụng SQL để giao tiếp với các hệ thống khác trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.

- SQL là ngôn ngữ sử dụng cho các cổng giao tiếp cơ sở dữ liệu: Trong một hệ thống mạng máy tính với nhiều hệ quản trị cơ sở dữ liệu khác nhau, SQL thường được sử dụng như là một chuẩn ngôn ngữ để giao tiếp giữa các hệ quản trị cơ sở dữ liệu.

1.2. Sự quan trọng và cần thiết của chúng trong lập trình ứng dụng

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.

Hệ quản trị cơ sở dữ liệu quan hệ là hệ quản trị cơ sở dữ liệu phổ biến nhất hiện nay và được hỗ trợ bởi nhiều nhà cung cấp phần mềm. Tính hiệu quả của các ứng dụng phụ thuộc vào chất lượng của việc tổ chức dữ liệu. Những cái tiến trong kỹ thuật và xử lý cơ sở dữ liệu đưa đến các cơ hội sử dụng thông tin một cách linh hoạt và hiệu quả khi dữ liệu được tổ chức và lưu trữ trong các cấu trúc quan hệ. Hệ quản trị cơ sở dữ liệu là một thành công trong lĩnh vực thương mại.

Khả năng của hệ quản trị cơ sở dữ liệu

Có hai khả năng chính cho phép phân biệt các hệ quản trị cơ sở dữ liệu với các kiểu hệ thống lập trình khác:

- Khả năng quản lý dữ liệu tồn tại lâu dài: đặc điểm này chỉ ra rằng có một cơ sở dữ liệu tồn tại trong một thời gian dài, nội dung của cơ sở dữ liệu này là các dữ liệu mà hệ quản trị CSDL truy nhập và quản lý.

- Khả năng truy nhập các khối lượng dữ liệu lớn một cách hiệu quả.

Ngoài hai khả năng cơ bản trên, hệ quản trị CSDL còn có các khả năng khác mà có thể thấy trong hầu hết các hệ quản trị CSDL đó là:

- Hỗ trợ ít nhất một mô hình dữ liệu hay một sự trừu tượng toán học mà qua đó người sử dụng có thể quan sát dữ liệu.

- Đảm bảo tính độc lập dữ liệu hay sự bất biến của chương trình ứng dụng đối với các thay đổi về cấu trúc trong mô hình dữ liệu.

- Hỗ trợ các ngôn ngữ cao cấp nhất định cho phép người sử dụng định nghĩa cấu trúc dữ liệu, truy nhập dữ liệu và thao tác dữ liệu.

- Quản lý giao dịch, có nghĩa là khả năng cung cấp các truy nhập đồng thời, đúng đắn đối với CSDL từ nhiều người sử dụng tại cùng một thời điểm. Bài giảng Hệ quản trị CSDL

- Điều khiển truy nhập, có nghĩa là khả năng hạn chế truy nhập đến các dữ liệu bởi những người sử dụng không được cấp phép và khả năng kiểm tra tính đúng đắn của CSDL.

- Phục hồi dữ liệu, có nghĩa là có khả năng phục hồi dữ liệu, không làm mất mát dữ liệu với các lỗi hệ thống.

Sự bùng nổ của các loại hình dịch vụ cùng sự phát triển của hệ thống ứng dụng thông tin nghiệp vụ đòi hỏi cần phải có những máy chủ với khả năng xử lý mạnh, những hệ quản trị cơ sở dữ liệu có thể đáp ứng được những yêu cầu khắt khe về hiệu năng, tính an toàn và tính sẵn sàng dữ liệu. Đó chính là lý do hệ quản trị cơ sở dữ liệu ra đời và được sử dụng rất phổ biến trong các hệ thống thông tin của các tổ chức, doanh nghiệp lớn.

- **Các Big Data Cluster và PolyBase**

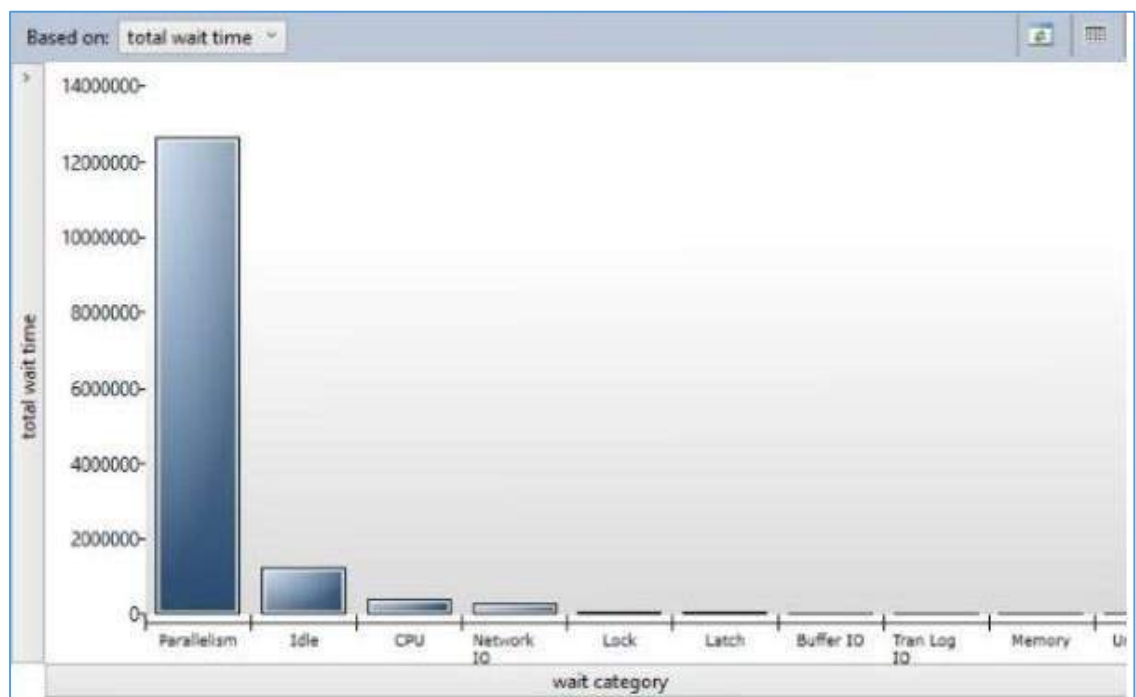
Đối với các Big Data Cluster vốn dĩ có rất nhiều công nghệ rất thú vị trong nền tảng này. Nó cho phép chạy Microsoft SQL Server trên Kubernetes. Song song với đó nó kết nối trực tiếp Microsoft SQL Server với Apache Spark. Mặc dù đây là một sự phát triển mang tính cách mạng, nhưng nó cũng đồng nghĩa với việc thực tế doanh nghiệp áp dụng sẽ phải thay đổi mô hình rất lớn.

Thành phần thú vị hơn của bộ tính năng này là một cải tiến lớn đối với PolyBase. Nó cho phép bạn kết nối Microsoft SQL Server với Oracle, MongoDB và Teradata. Đồng thời cung cấp kết nối ODBC với các nguồn dữ liệu khác (ví dụ danh sách SharePoint chẳng hạn). Microsoft gọi đây là "ảo hóa dữ liệu". PolyBase cũng đã được tự động thêm vào Microsoft SQL Server phiên bản chuẩn. Như vậy có nghĩa là sẽ có nhiều người dùng có thể tiếp cận được với nó hơn.

- **Xử lý truy vấn thông minh**

Microsoft SQL Server 2016 đã giới thiệu một tính năng tuyệt vời được gọi là Query Store. Tính năng này cho phép bạn có thông tin thời gian chạy và kế hoạch thực thi cho tất cả các truy vấn của mình. Đồng thời, với Query Store, nó cũng cho phép quản trị viên có thể nhanh chóng cô lập và tìm ra nguyên nhân gốc rễ của vấn đề hiệu suất.

Microsoft SQL Server đã xây dựng tính năng thông minh này bằng cách điều chỉnh truy vấn tự động cho các kế hoạch thực thi hội quy về hiệu suất. Microsoft SQL Server 2019 đã thay đổi dung lượng bộ nhớ được cấp cho truy vấn để sắp xếp và so sánh. Đồng thời nó cũng giới thiệu hàm vô hướng nội tuyến cho các biểu bảng. Cả hai tính năng đều là các mẫu chống hiệu suất đã tồn tại trong các phiên bản Microsoft SQL Server trước đó.



Hình 1.1 Đo lường hiệu suất sử dụng Query Store

- **Giám sát thông minh hơn**

Tính năng Query Store vốn dĩ mang lại nhiều lợi ích cho người dùng. Nhưng trên một số hệ thống, nó gây ra một số vấn đề về tranh chấp tài nguyên. Điều này đặc biệt có tác động đối với những khách hàng chạy SQL động. Microsoft SQL Server 2019 giới thiệu một tùy chọn cho phép bạn tùy biến việc thu thập dữ liệu để hạn chế vấn đề này.

Ngoài ra, Microsoft SQL Server cũng cập nhật một số cải tiến xung quanh việc thu thập kế hoạch thực thi. Những cập nhật này cho phép bạn nắm bắt các kế hoạch thực thi thực tế, giúp bạn kiểm tra được có bao nhiêu hàng được truy vấn sử dụng. Từ những thông số đó bạn có thể so sánh chúng với những dữ liệu ước tính mà trình tối ưu hóa truy vấn đã sử dụng. Đòi lại, điều này có thể giúp bạn nhanh chóng cô lập một vấn đề về hiệu suất.

- **Tối ưu hóa hiệu suất**

Microsoft SQL Server từ lâu đã gặp vấn đề khi nó cố gắng chèn các bản ghi tuần tự vào một bảng với khối lượng rất lớn. Vấn đề này xảy ra là do sự cạnh tranh trong bộ nhớ. Vì chỉ có một luồng có thể truy cập một trang dữ liệu tại một thời điểm nhất định. Tính năng OLTP trong bộ nhớ được thiết kế để khắc phục vấn đề này. Tuy nhiên yếu điểm của nó là việc nó không tương thích 100% với tất cả các loại dữ liệu nên không thể sử dụng ở mọi nơi. SQL Server 2019 giới thiệu một tính năng tùy chọn được gọi là "Optimi_for_sequential_key". Nó giúp giảm ảnh hưởng của vấn đề kể trên.

Một cải tiến hiệu suất khác đến từ không gian làm việc tạm thời của Microsoft SQL Server là TempDB. Tận dụng tính năng OLTP trong bộ nhớ, giúp loại bỏ những vấn đề bị tắc nghẽn ở lớp dữ liệu.

- ***Phục hồi sơ sở dữ liệu nhanh hơn***

Việc khôi phục dữ liệu đối với quản trị viên đôi khi giống như một cơn ác mộng vậy. Điều này là do các giao dịch có trong nhật ký giao dịch cơ sở dữ liệu nhưng lại không có trong bản sao lưu cơ sở dữ liệu được dùng để khôi phục.

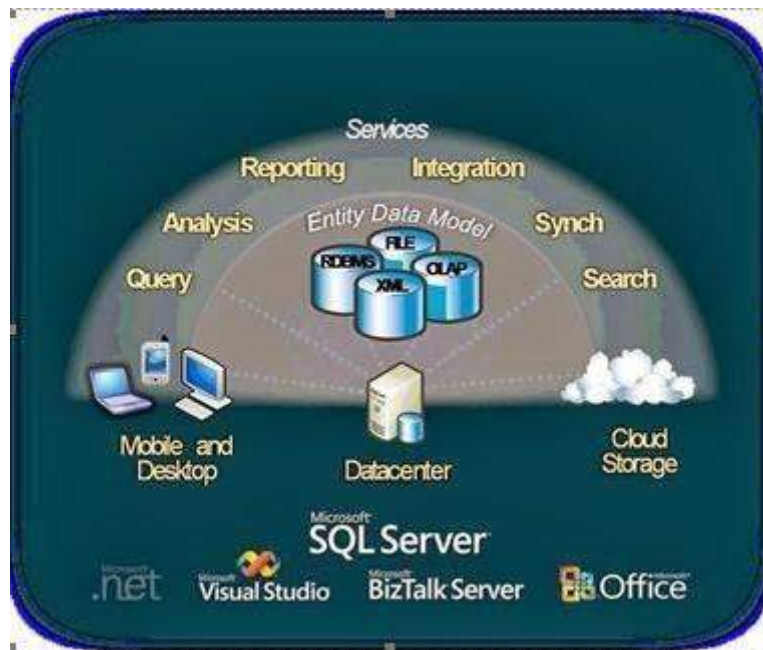
Với Microsoft SQL Server 2019, họ đã thay đổi quá trình này bằng cách triển khai một danh sách các phiên bản để cho phép các giao dịch được cập nhật lại nhanh hơn nhiều sau khi khôi phục hoặc khởi động lại máy chủ. Bằng việc đưa ra tính năng này, trong phiên bản Microsoft SQL Server 2019, mỗi lần khôi phục mẫu mất 90 giây đã giảm xuống chỉ còn 1 giây.

- ***Một số tính năng khác***

Cuối cùng, còn khá nhiều tính năng “nhỏ nhất” nhưng hữu ích khác có trong phiên bản Microsoft SQL Server 2019 lần này. Trong đó có thể kể đến như tạo chỉ mục hoặc mã hoá. Tính năng mã hóa cũng đã được cải tiến với tên gọi là mã hóa an toàn. Nó cho phép công cụ cơ sở dữ liệu hoạt động trên dữ liệu được mã hóa trong T-SQL trong khi dữ liệu đó không bị người dùng hoặc quản trị viên nhìn thấy..

- ***Thông tin trong toàn bộ doanh nghiệp***

SQL Server 2019 cung cấp một cơ sở hạ tầng có thể mở rộng, cho phép quản lý các báo cáo, phân tích với bất kỳ kích thước và sự phức tạp nào, bên cạnh đó nó cho phép người dùng dễ dàng hơn trong việc truy cập thông tin thông qua sự tích hợp sâu hơn với Microsoft Office. Điều này cho phép CNTT đưa được thông tin của doanh nghiệp rộng khắp trong tổ chức. SQL Server 2019 tạo những bước đi tuyệt vời trong việc lưu trữ dữ liệu, cho phép người dùng hợp nhất các trung tâm dữ liệu vào một nơi lưu trữ dữ liệu tập trung của toàn doanh nghiệp.



Hình 1.2. Nền tảng dữ liệu của SQL Server

2. Các thành phần quan trọng trong SQL Server

2.1. Relational Database Engine - Lõi của SQL Server:

Đây là một engine có khả năng chứa data ở các quy mô khác nhau dưới dạng table và support tất cả các kiểu kết nối (data connection) thông dụng của Microsoft như ActiveX Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC). Ngoài ra nó còn có khả năng tự điều chỉnh (tune up) ví dụ như sử dụng thêm các tài nguyên (resource) của máy khi cần và trả lại tài nguyên cho hệ điều hành khi một user log off.

2.2. Replication - Cơ chế tạo bản sao (Replica):

Giả sử bạn có một database dùng để chứa dữ liệu được các ứng dụng thường xuyên cập nhật. Một ngày đẹp trời bạn muốn có một cái database giống y hệt như thế trên một server khác để chạy báo cáo (report database) (cách làm này thường dùng để tránh ảnh hưởng đến performance của server chính). Vấn đề là report server của bạn cũng cần phải được cập nhật thường xuyên để đảm bảo tính chính xác của các báo cáo. Bạn không thể dùng cơ chế back up and restore trong trường hợp này. Thế thì bạn phải làm sao? Lúc đó cơ chế replication của SQL Server sẽ được sử dụng để bảo đảm cho dữ liệu ở 2 database được đồng bộ (synchronized). Replication sẽ được bàn kỹ trong bài 12

2.3. Data Transformation Service (DTS)

Một dịch vụ chuyển dịch data Nếu bạn làm việc trong một công ty lớn trong đó data được chứa trong nhiều nơi khác nhau và ở các dạng khác nhau cụ thể như chứa trong Oracle, DB2 (của IBM), SQL Server, Microsoft Access. Bạn chắc chắn sẽ có nhu cầu di chuyển data giữa các server này (migrate hay transfer) và không chỉ di chuyển bạn còn muốn định dạng (format) nó trước khi lưu vào database khác, khi đó bạn sẽ thấy DTS giúp bạn giải quyết công việc trên dễ dàng như thế nào. DTS sẽ được bàn kỹ trong bài 8.

2.4. Analysis Service - Một dịch vụ phân tích dữ liệu

Dữ liệu (Data) chứa trong database sẽ chẳng có ý nghĩa gì nhiều nếu như bạn không thể lấy được những thông tin (Information) bổ ích từ đó. Do đó Microsoft cung cấp cho bạn một công cụ rất mạnh giúp cho việc phân tích dữ liệu trở nên dễ dàng và

hiệu quả bằng cách dùng khái niệm hình khối nhiều chiều (multi-dimension cubes) và kỹ thuật "đào mỏ dữ liệu" (data mining) sẽ được chúng tôi giới thiệu trong bài 13.

2.5. English Query:

Đây là một dịch vụ giúp cho việc query data bằng tiếng Anh "trơn" (plain English).

2.6. Meta Data Service:

Dịch vụ này giúp cho việc chứa đựng và "xào nấu" Meta data dễ dàng hơn. Thế thì Meta Data là cái gì vậy? Meta data là những thông tin mô tả về cấu trúc của data trong database như data thuộc loại nào String hay Integer..., một cột nào đó có phải là Primary key hay không....Bởi vì những thông tin này cũng được chứa trong database nên cũng là một dạng data nhưng để phân biệt với data "chính thống" người ta gọi nó là Meta Data. Phần này chắc là bạn phải xem thêm trong một thành phần khác của SQL Server sắp giới thiệu sau đây là SQL Server Books Online vì không có bài nào trong loạt bài này nói rõ về dịch vụ này cả.

2.7. SQL Server Books Online

Cho dù bạn có đọc các sách khác nhau dạy về SQL server thì bạn cũng sẽ thấy books online này rất hữu dụng và không thể thiếu được (cho nên Microsoft mới hào phóng đính kèm theo SQL Server).

2.8 SQL Server Tools

- Enterprise Manager: Đây là một công cụ cho ta thấy toàn cảnh hệ thống CSDL một cách rất trực quan. Nó rất hữu ích đặc biệt cho người mới học và không thông thạo lắm về SQL.

- Query Analyzer: Đối với một DBA giỏi thì hầu như chỉ cần công cụ này là có thể quản lý cả một hệ thống database mà không cần đến những thứ khác. Đây là một môi trường làm việc khá tốt vì ta có thể đánh bất kỳ câu lệnh SQL nào và chạy ngay lập tức đặc biệt là nó giúp cho chúng ta debug mấy cái Stored procedure dễ dàng.

- SQL Profiler: Nó có khả năng chụp tất cả các sự kiện hay hoạt động diễn ra trên một SQL Server và lưu lại dưới dạng text file rất hữu dụng trong việc kiểm soát hoạt động của SQL Server.

- Ngoài một số công cụ trực quan như trên chúng ta cũng thường hay dùng osql và bcp (bulk copy) trong command prompt.

3. Cài đặt SQL Server Express

3.1 Yêu cầu cấu hình cài đặt

3.1.1 Yêu cầu phần cứng

- **CPU:** bộ vi xử lý x64: AMD Opteron, AMD Athlon 64, Intel Xeon với hỗ trợ Intel EM64T, Intel Pentium IV với hỗ trợ EM64T..
- **Tốc độ CPU:** Tối thiểu yêu cầu là 1,4 GHz. Khuyến nghị là 2.0 GHz hoặc nhanh hơn.
- **Ram:** Đối với phiên bản Express tối thiểu 1 GB. Tất cả các phiên bản khác tối thiểu 4 GB. Và chúng có khả năng mở rộng được khi kích thước cơ sở dữ liệu tăng lên nhằm đảm bảo hiệu suất.
- **Ổ đĩa trống:** Ít nhất 10GB.
- **.NET Framework:** Phiên bản 3.5 SP1.

3.1.2 Yêu cầu Hệ điều hành khi cài đặt SQL Server 2019

- Các định dạng hệ thống tệp được đề xuất khi cài đặt SQL Server 2019 là NTFS và ReFS. Cài đặt SQL Server trên máy tính có hệ thống tệp FAT32 vẫn được hỗ

trợ. Nhưng không được khuyến nghị vì nó kém an toàn hơn hệ thống tệp NTFS hoặc ReFS.

- Không cố cài đặt SQL Server 2019 trên ổ đĩa nén, mã hóa hoặc chỉ đọc vì quá trình thiết lập sẽ chặn cài đặt.
- Lưu ý là việc cài đặt SQL Server 2019 sẽ không hỗ trợ cho phiên bản 32 bit nữa. Tất cả các tính năng của SQL Server 2019 đều được hỗ trợ trên hệ thống x64 (64-bit). Nhưng chỉ một nhóm nhỏ một số tính năng sau được hỗ trợ trên hệ thống x86 (32-bit).
- Trong khi cài đặt SQL Server, Windows Installer tạo các tệp tạm thời trên ổ đĩa hệ thống. Trước khi bạn chạy Thiết lập để cài đặt hoặc nâng cấp SQL Server, hãy đảm bảo rằng bạn có ít nhất 6 GB dung lượng đĩa khả dụng trên ổ đĩa hệ thống cho các tệp này.
- Yêu cầu dung lượng đĩa cứng thực tế khi cài đặt SQL Server 2019 phụ thuộc vào cấu hình hệ thống của bạn và các tính năng mà bạn quyết định cài đặt. Theo ước lượng tất cả các tính năng để hoàn tất việc cài đặt SQL Server 2019 sẽ khoảng 8,3 GB.
- Không cài đặt SQL Server trên Bộ điều khiển miền (Domain controller).
- SQL Server 2019 Enterprise Edition và Web Edition được hỗ trợ trên Trung tâm dữ liệu Windows Server 2019, Windows Server 2019 Standard, Windows Server 2019 Essentials, Windows Server 2016 Datacenter, Windows Server 2016 Standard, Windows Server 2016 Essentials. Nó không được hỗ trợ trên Windows 10 và Windows 8.
- SQL Server 2019 Developer Edition, Standard Edition và Express Edition được hỗ trợ trên các hệ điều hành sau: Trung tâm dữ liệu Windows Server 2019, Windows Server 2019 Standard, Windows Server 2019 Essentials, Windows Server 2016 Datacenter, Windows Server 2016 Standard, Windows Server 2016 Essentials; Windows 10 Enterprise, Windows 10 Professional, Windows 10 Home. Windows 10 TH1 1507 trở lên được hỗ trợ. Windows 8 không được hỗ trợ.
- Trước khi cài đặt SQL Server 2019, Hãy chắc chắn dịch vụ Công cụ quản lý Windows đang chạy. Bạn có thể kiểm tra tác vụ này tại (Bảng điều khiển -> Công cụ quản trị -> Dịch vụ).
- Thay đổi cấu hình tường lửa của bạn để cho phép truy cập SQL Server.
- Tài khoản người dùng để cài đặt SQL Server 2019 phải có quyền quản trị cao nhất.

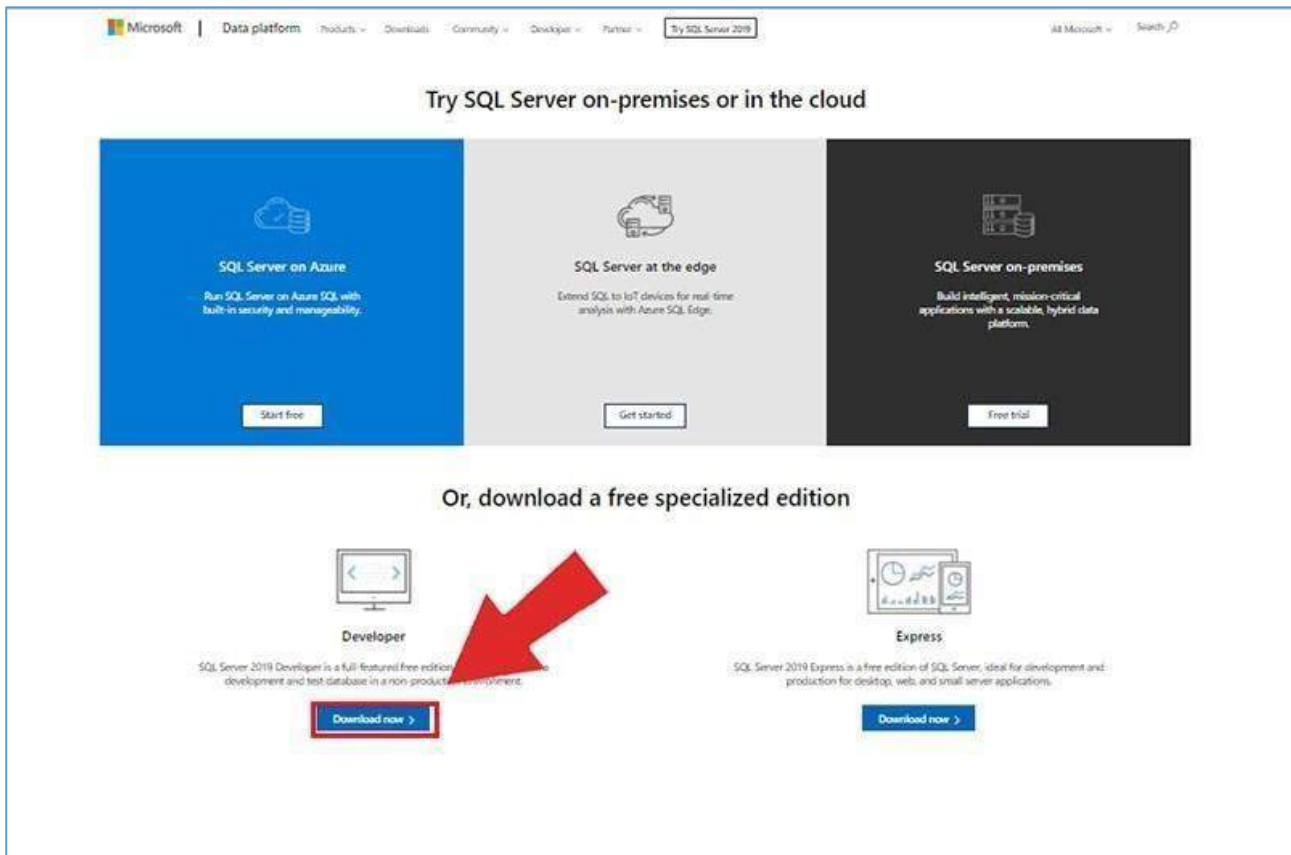
3.2 Các bước cài đặt SQL Server 2019

Trước khi cài đặt Microsoft SQL Server bạn cần chú ý

- Các bạn nên cài SQL Server trước khi cài Microsoft Visual Studio để tránh xung đột.
- Các bạn phải cài .Net Framework 3.5 trước khi mở file cài đặt SQL Server để tránh bị lỗi.
- Các bạn phải cài SQL Server 2019 trước khi cài Red Gate SQL(SQL Toolbelt) nhé.

Bước 1: Truy cập trang chủ để tải SQL Server 2019 [tại đây](#).

Sau đó ấn **nút download** ở phần developer để tải phần mềm về



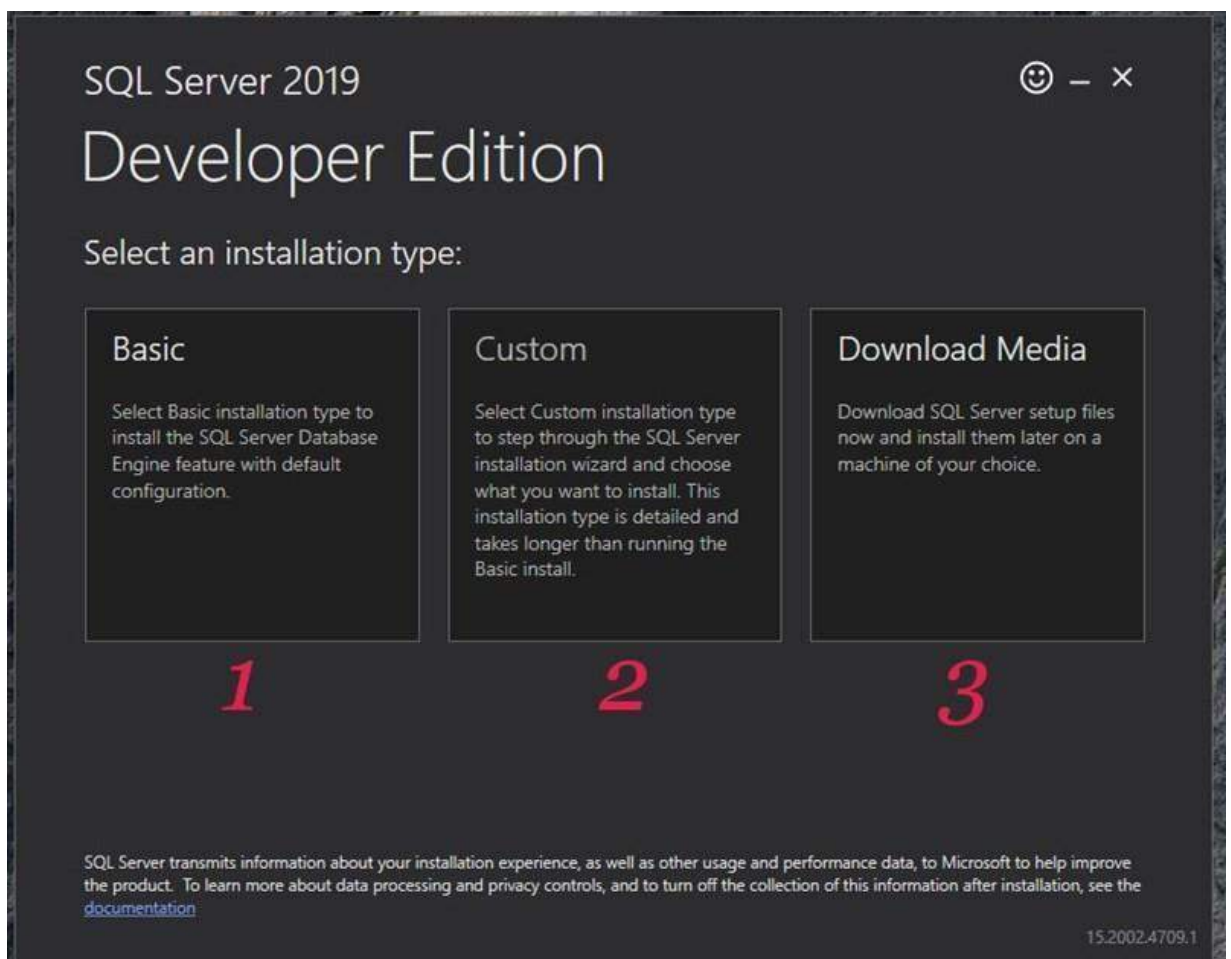
Hình 1.3 Tải SQL Server

Bước 2: Mở ứng dụng đã tải xong ra

Name	Date modified	Type	Size
Today (1)			
SQL2019-SSEI-Dev		M Application	5,808 KB
Earlier this week (1)			
UCBrowser_V7.0.185.1002_windows_pf10...	12/8/2020 4:47 PM	Application	1,662 KB
Last week (11)			
v1	12/4/2020 3:44 PM	JPG File	627 KB
Be_Vietnam	12/4/2020 9:21 AM	WinRAR ZIP archive	620 KB
128381857_697825477835930_6592331914...	12/3/2020 6:24 PM	JPG File	16 KB
128869501_1174932156255893_701580714...	12/3/2020 6:24 PM	JPG File	22 KB
128850278_2976720375882436_806686959...	12/3/2020 6:24 PM	JPG File	18 KB
128144142_1326367891041144_519702111...	12/3/2020 6:24 PM	JPG File	21 KB
128362364_377765840197454_3284141650...	12/3/2020 6:24 PM	JPG File	45 KB
128368128_1257539014622538_548071240...	12/3/2020 5:56 PM	JPG File	101 KB
128551042_3674759862617045_770762505...	12/3/2020 5:55 PM	JPG File	209 KB
128409698_294017095314737_3643618174...	12/3/2020 5:55 PM	JPG File	82 KB

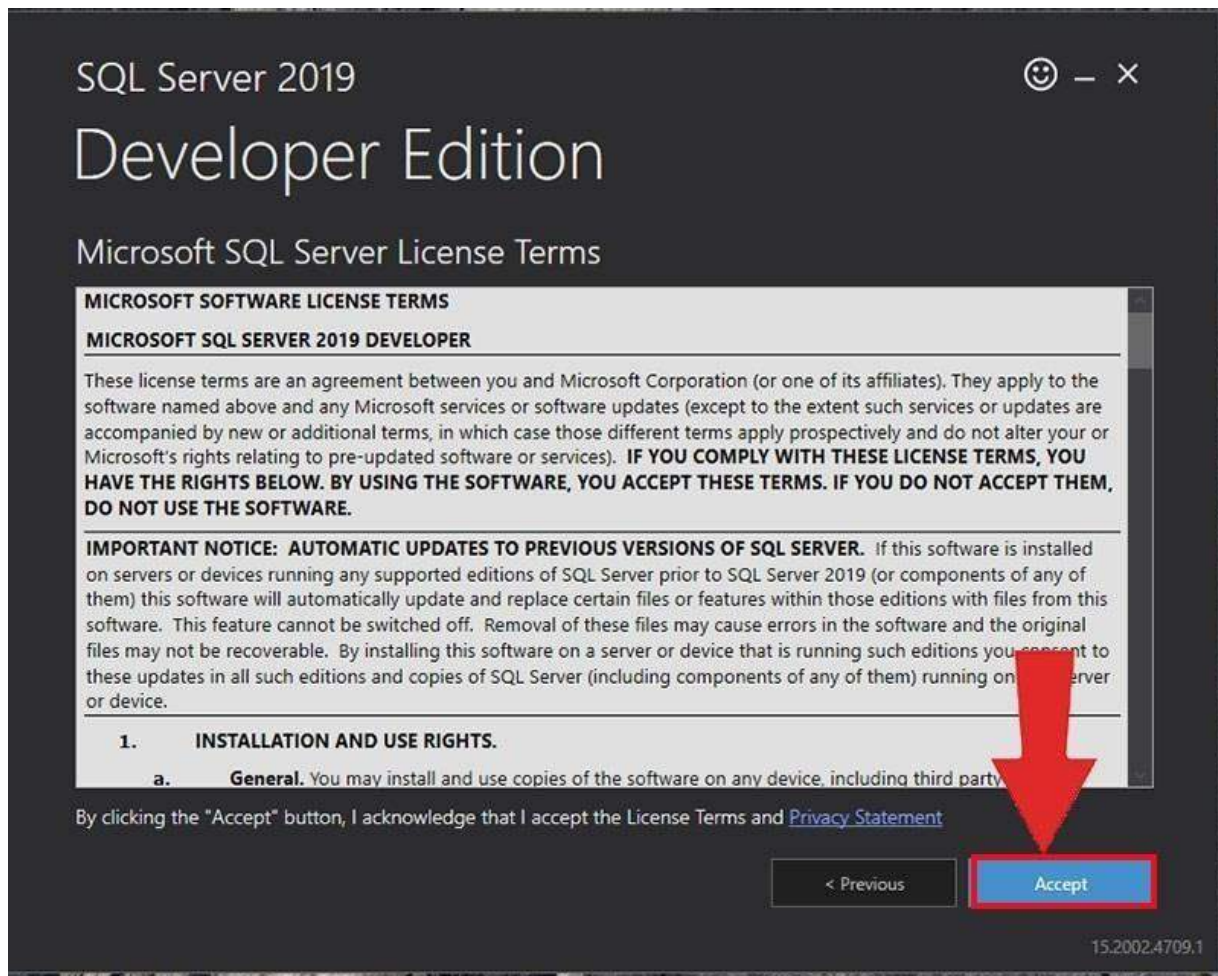
Hình 1.4 Mở ứng dụng

Bước 3: Sau khi mở ứng dụng ta có 3 lựa chọn tùy mục đích khác nhau cho người dùng:



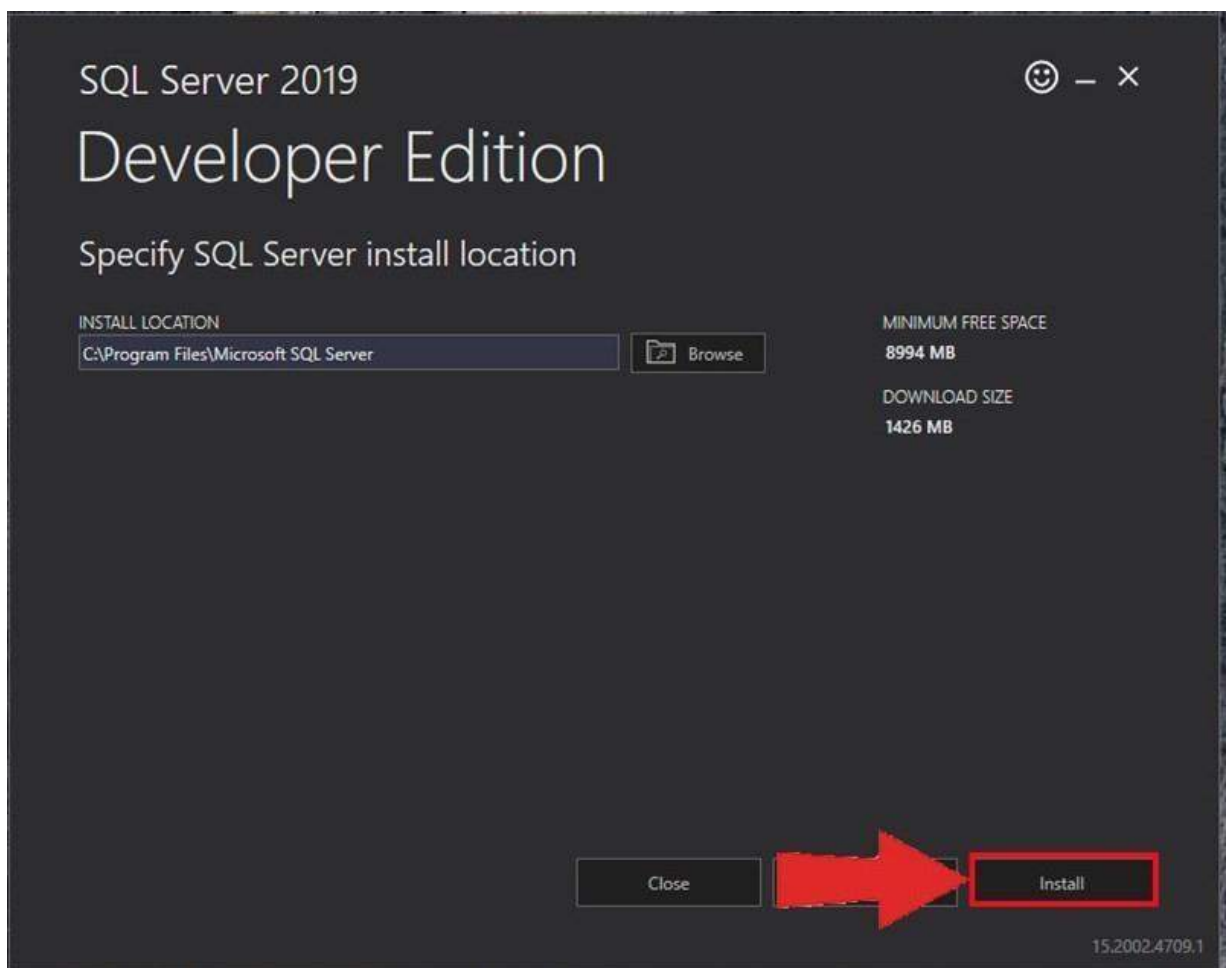
Hình 1.6 lựa chọn cài đặt

1. **Basic:** đây là tùy chọn đơn giản nhất cho người dùng, tại đây ứng dụng sẽ tự động cài đặt các chức năng cơ bản cho bạn.
2. **Custom:** đây là phần cài đặt cho các bạn muốn sử dụng chuyên sâu hơn, khi chọn bạn sẽ được tự cài đặt các cấu hình của phần mềm.
3. **Download Media:** khi chọn vào mục này, hệ thống sẽ tải về cho bạn một file cài đặt offline nhằm mục đích cài được trên nhiều thiết bị khác nhau mà không cần load lại từ đầu.



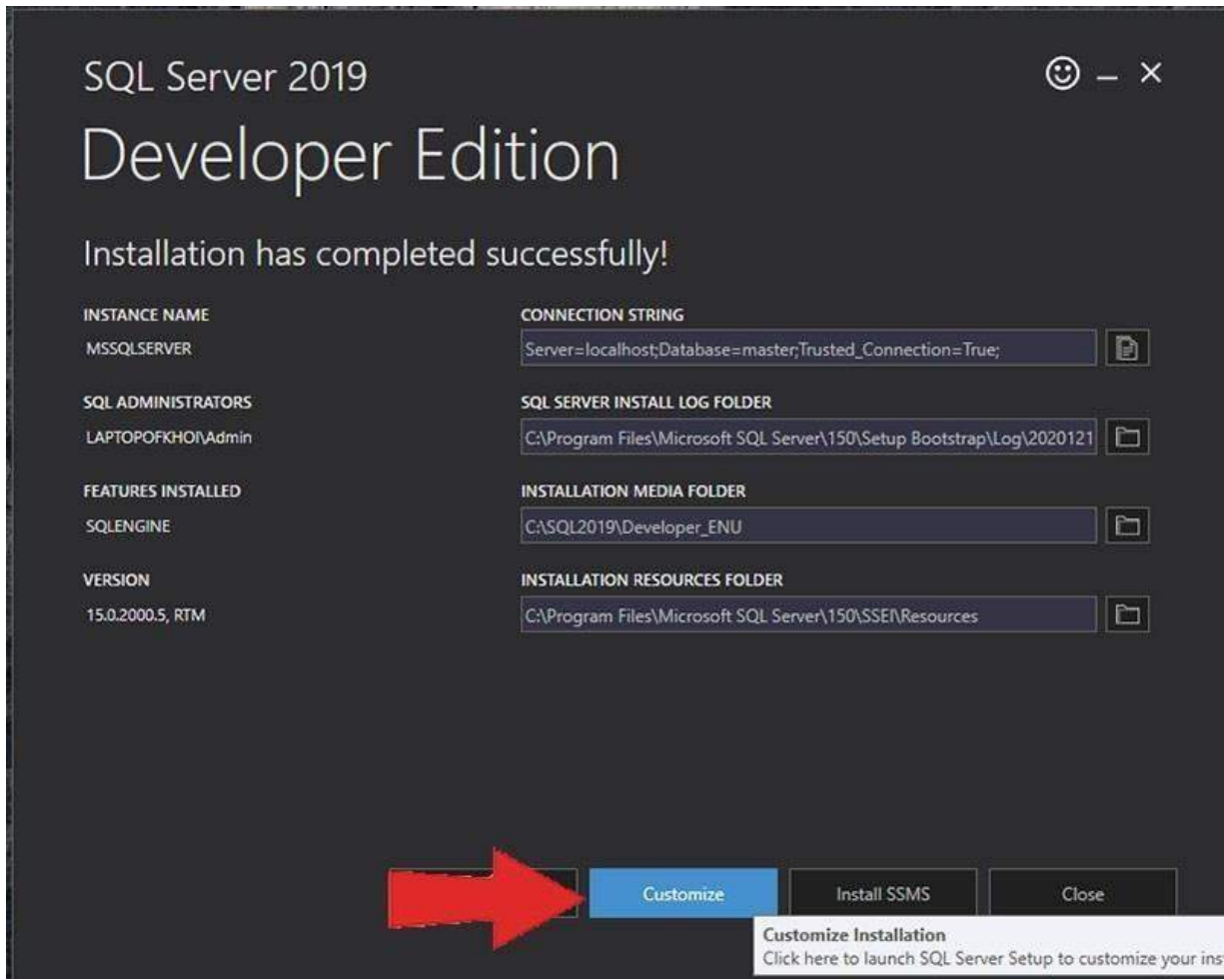
Hình 1.7 Chấp nhận cài đặt

Trong bài hướng dẫn này mình sẽ chọn chế độ **Basic**, các bạn hãy chọn basic và ấn **Accept** sau đó ấn nút **Install** để tiến hành cài đặt.



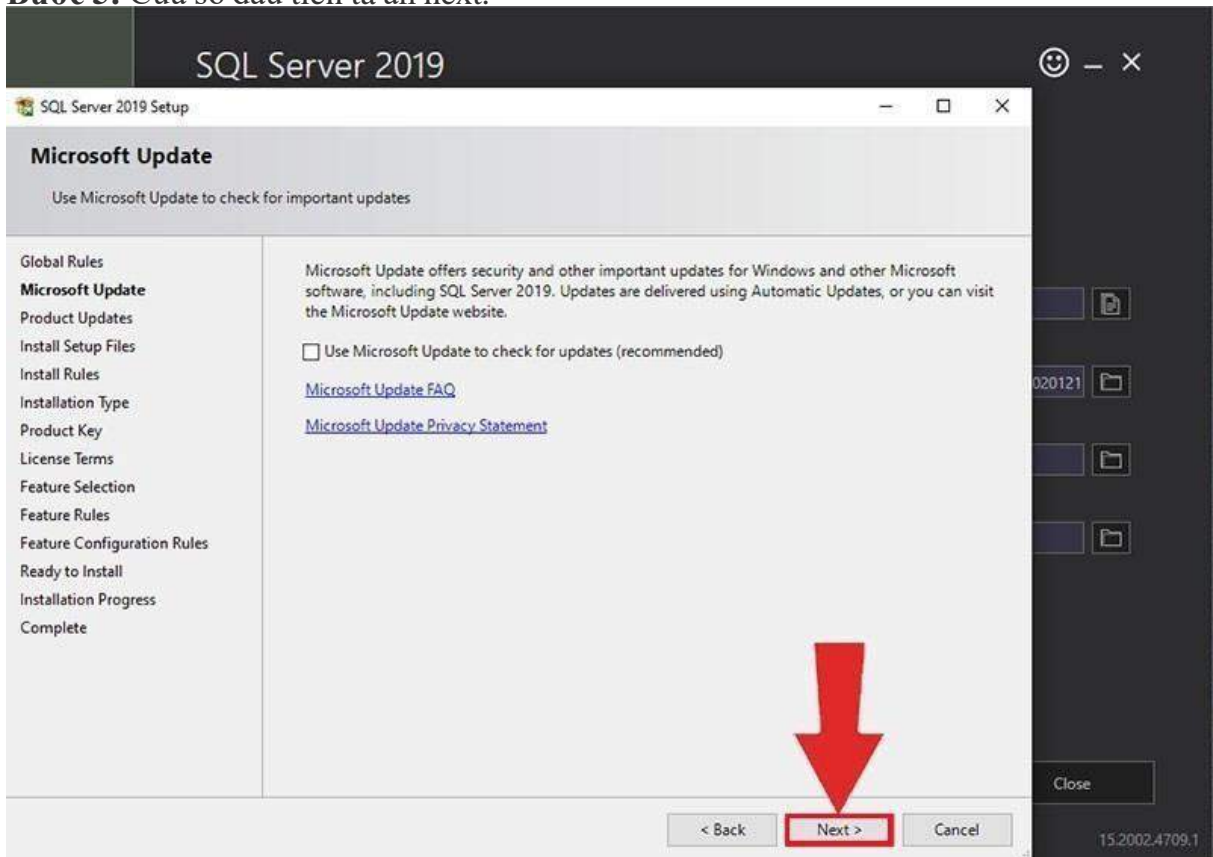
Hình 1.8 Bắt đầu cài đặt

Bước 4: Tại đây ta ấn nút **Customize** để bắt đầu **setting** cấu hình để sử dụng.



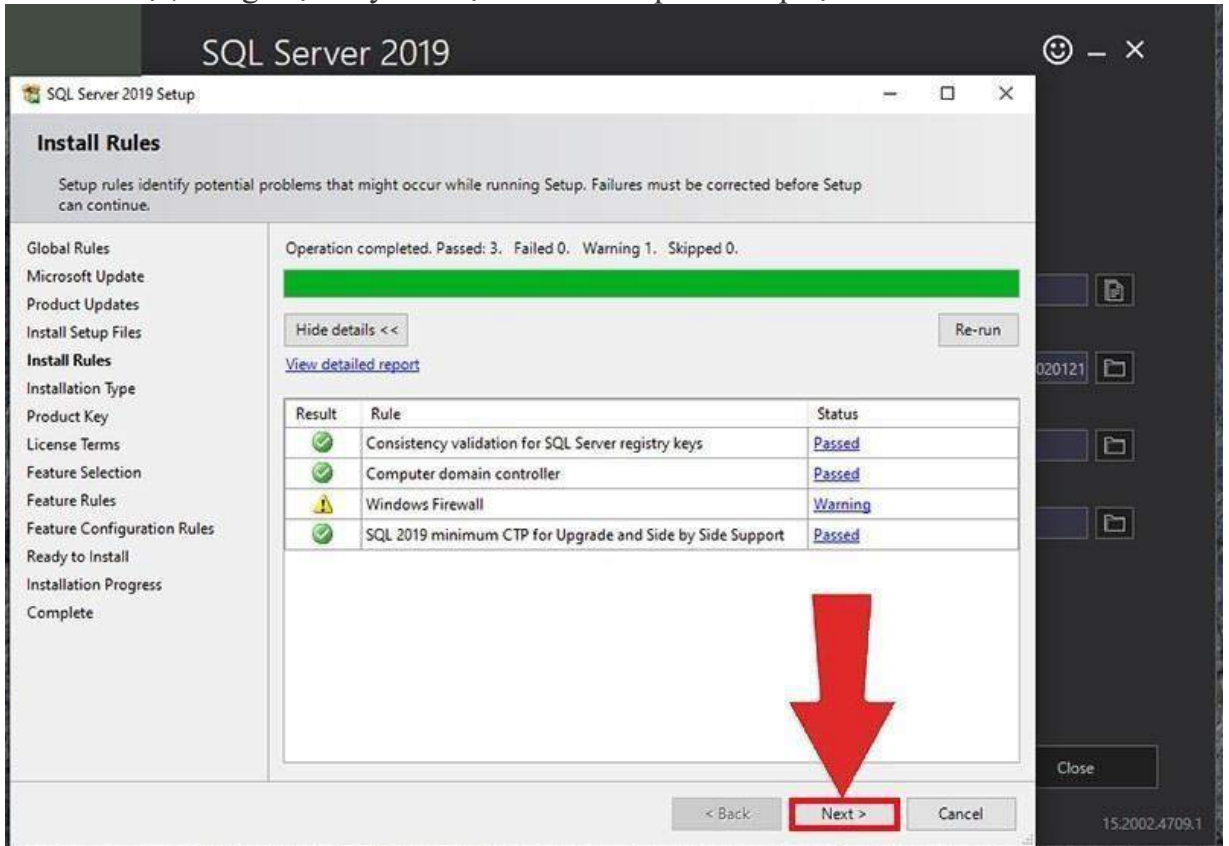
Hình 1.9 Cài đặt tự cấu hình

Bước 5: Cửa sổ đầu tiên ta ấn next.



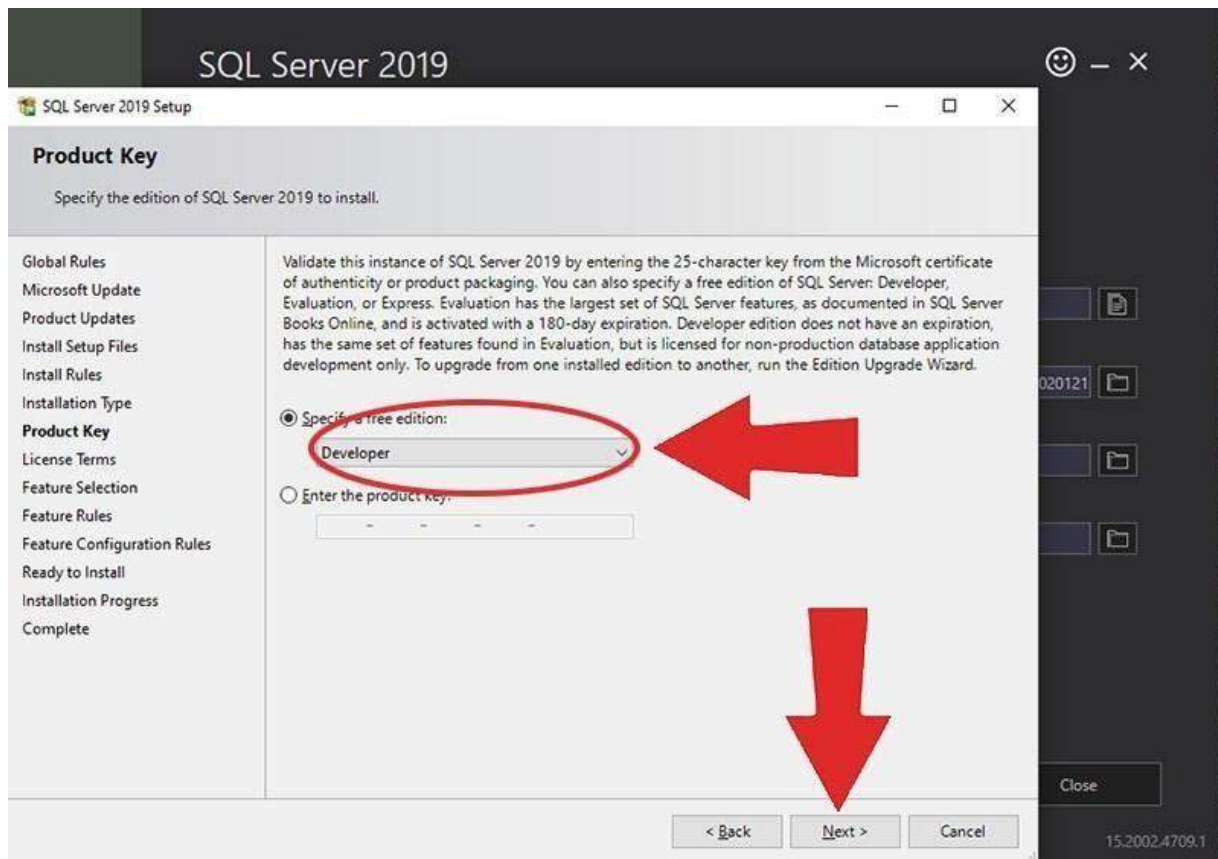
Hình 1.10 Cửa sổ cập nhật khi cài đặt

Bước 6: Hệ thống sẽ kiểm tra xem các mục đã đạt yêu cầu chưa, dấu tích xanh là đã ổn và thường thì mục Firewall sẽ màu vàng cảnh báo là nó có thể ảnh hưởng đến quá trình cài đặt, riêng mục này các bạn có thể bỏ qua và tiếp tục bấm next.



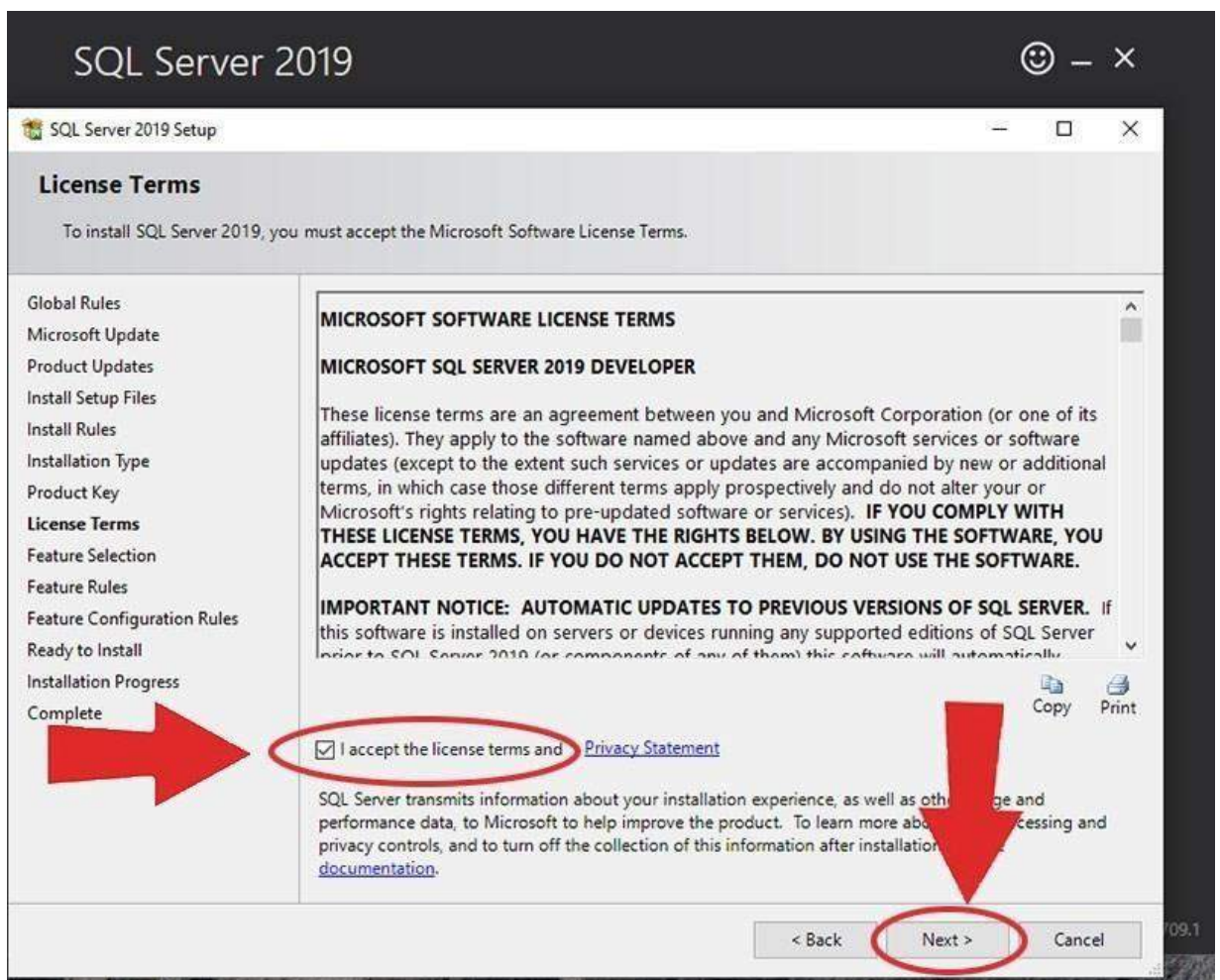
Hình 1.11 Thông báo chính sách cài đặt

Tiếp theo, ở mục **Product key** bạn phải chọn mục **developer** để có thể sử dụng miễn phí phần mềm này và ấn next để tiếp tục.



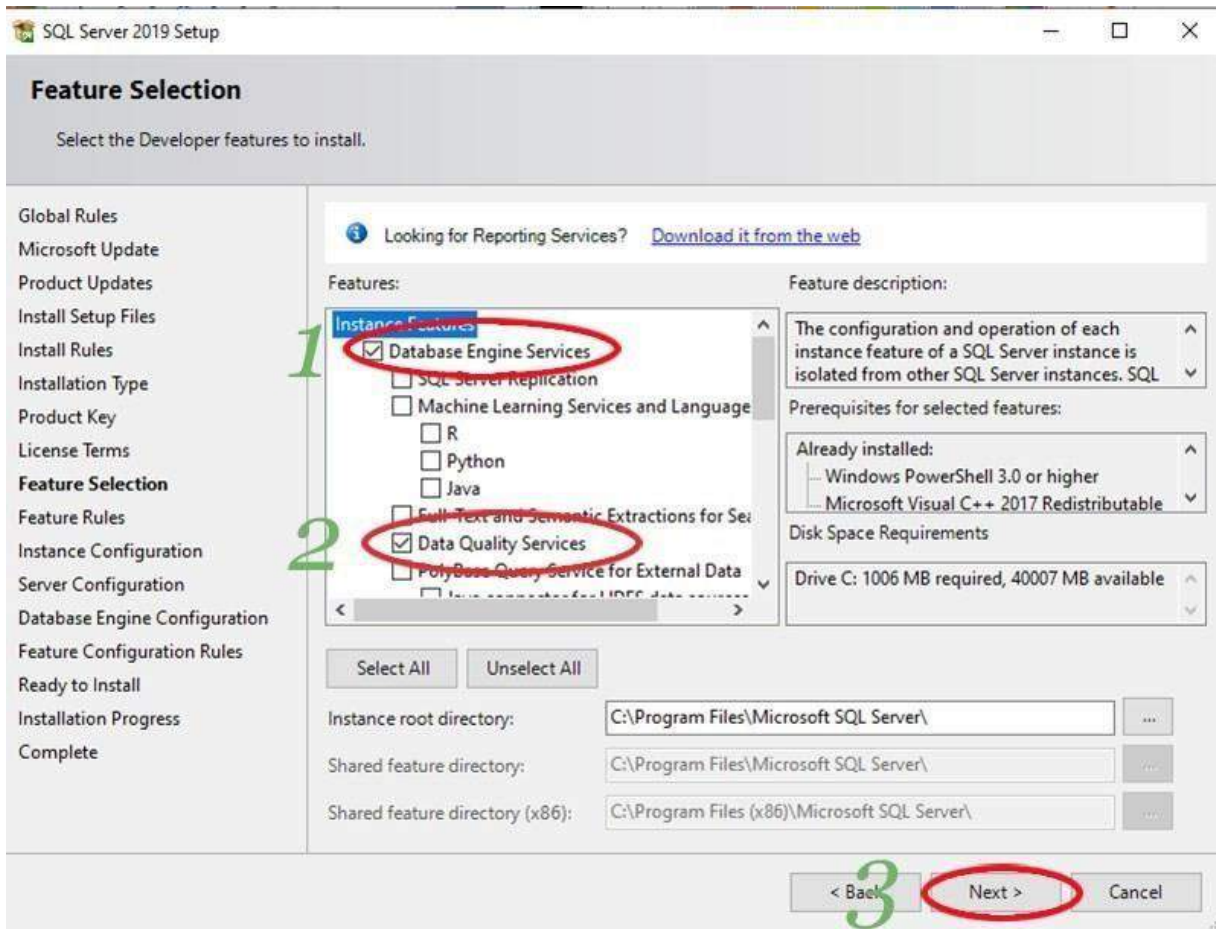
Hình 1.12 Chọn phiên bản cài đặt

Ở mục License Terms ta chọn **I accept the license terms and Privacy Statement** và nhấn next

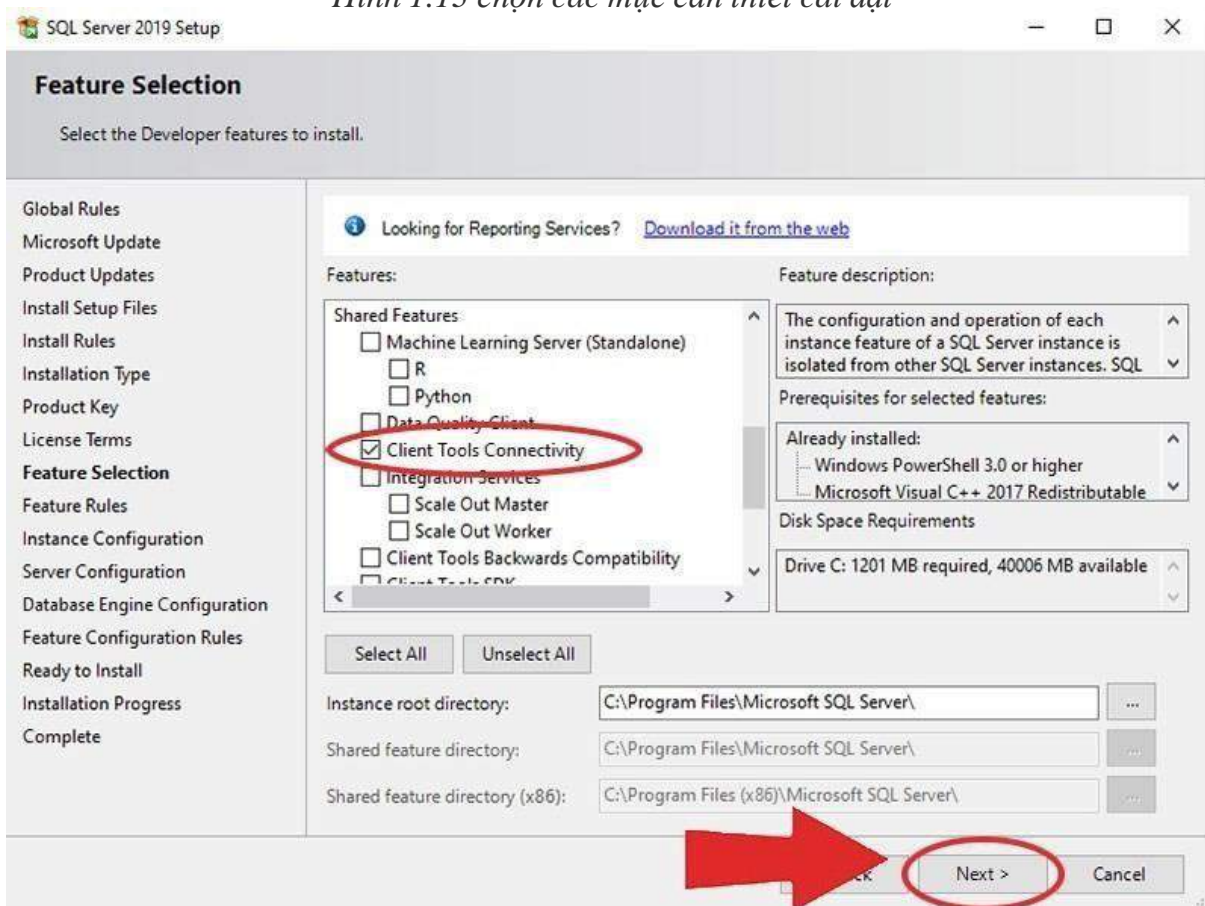


Hình 1.13 Đồng ý License cài đặt

Ở mục Features Selection ta nên chọn **Database Engine Services, Data Quality Client và Client Tools Connectivity** để ứng dụng cung cấp đủ packages và chức năng cho bạn học đầy đủ về môn cơ sở dữ liệu và bấm next.

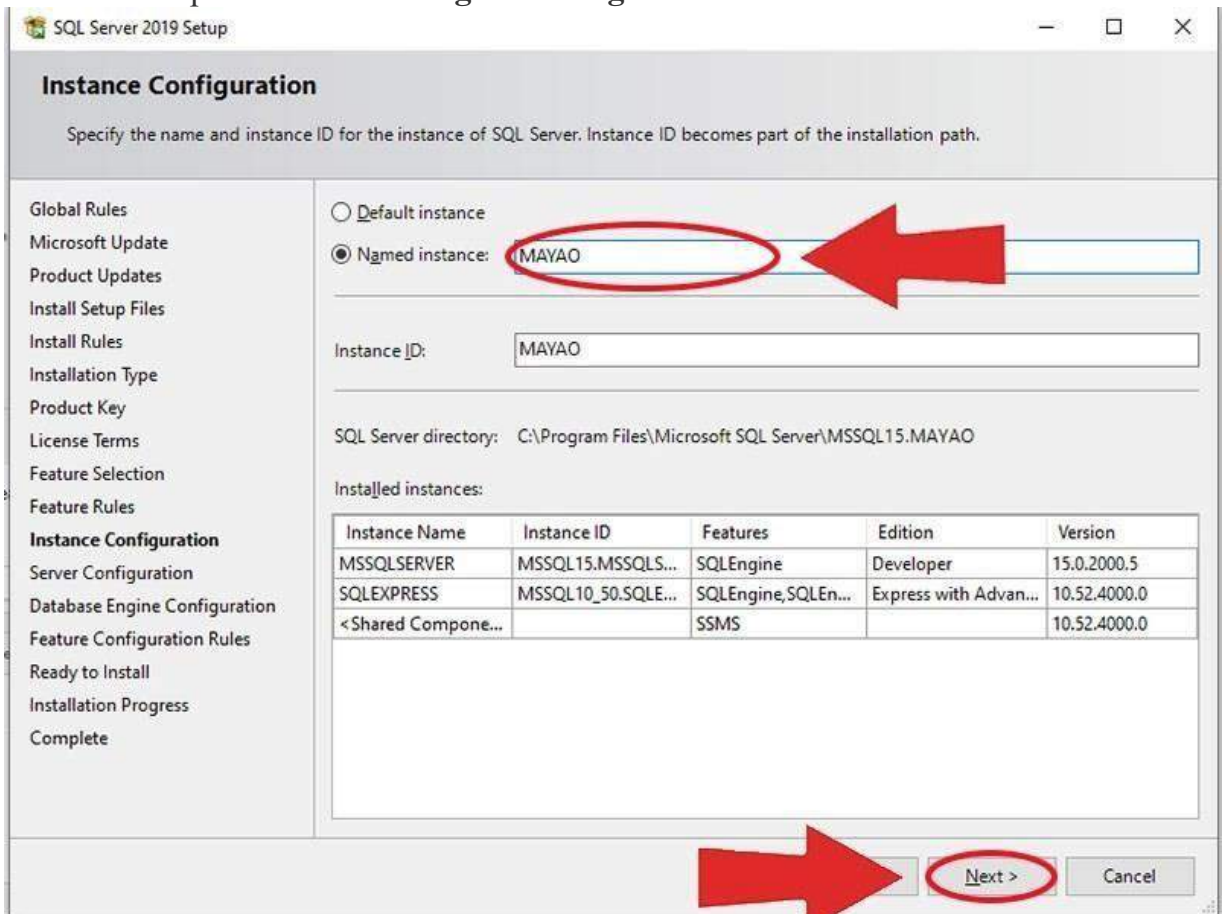


Hình 1.13 chọn các mục cần thiết cài đặt



Hình 1.14 cài đặt tool trên máy trạm

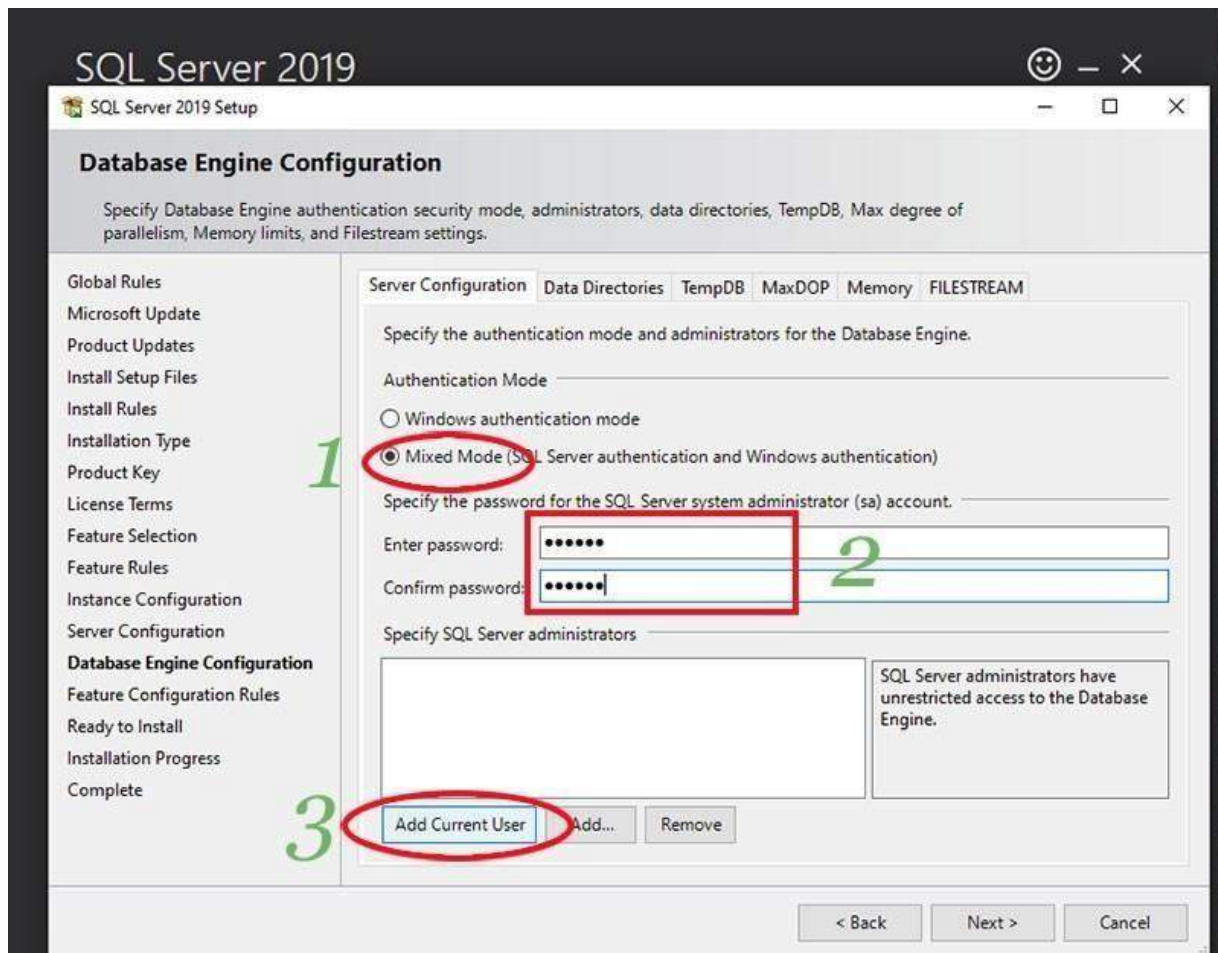
Bước 7: Đến với mục **Feature Configuration Rules**, tới đây bạn có thể đặt tên cho **Instance** (tên các bạn có thể đặt tùy ý, không dấu, không khoảng trắng) sau đó nhấn next cho đến phần **Database Engine Configuration**.



Hình 1.15 Xác định máy chủ

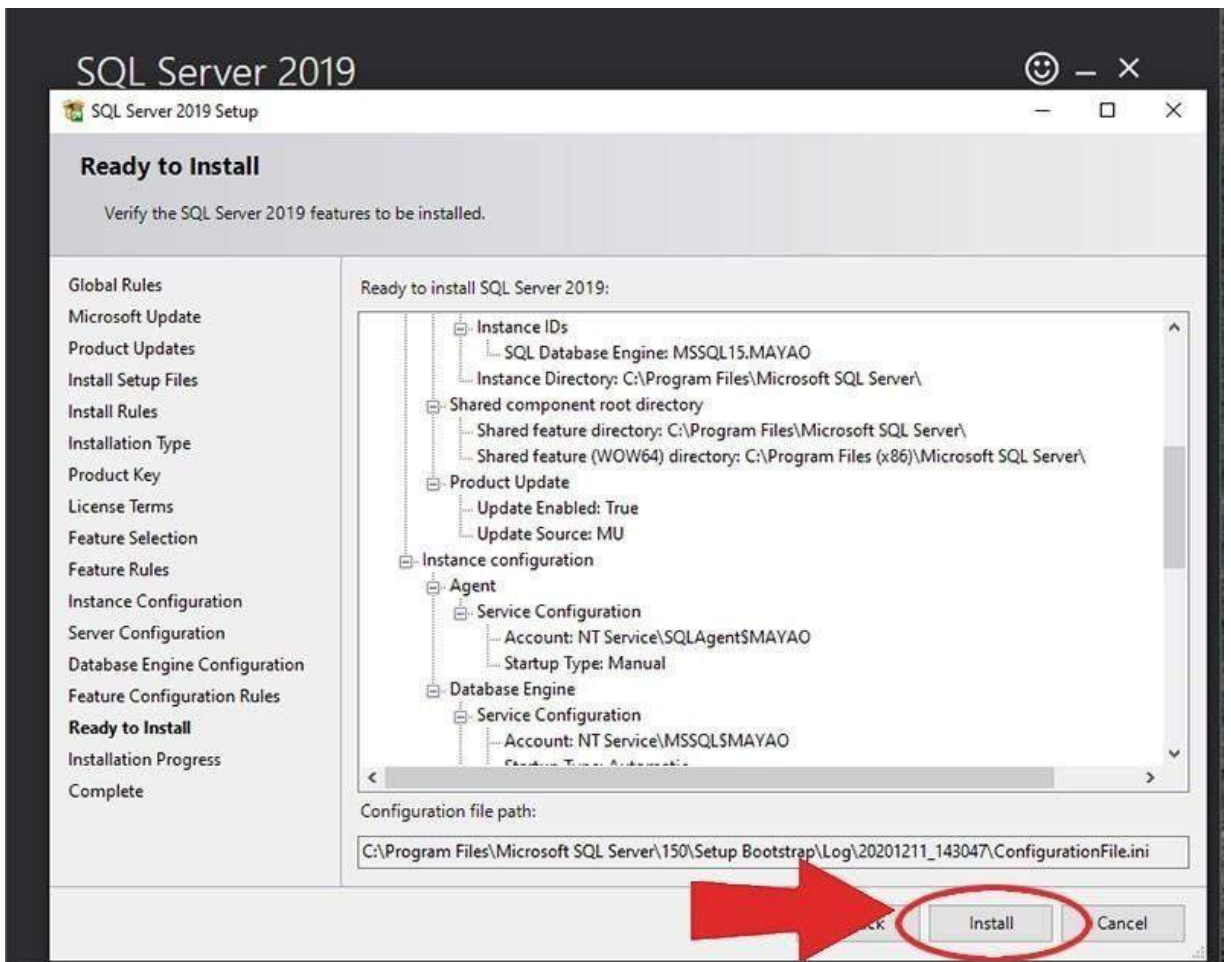
Bước 8: Ở mục này, ta bấm chọn chức năng **Mix Mode**. Đây là chức năng **bảo mật** cho cơ sở dữ liệu của bạn.

Tiếp theo các bạn **nhập mật khẩu** cho tài khoản **supper admin(sa)** của bạn. Và cuối cùng nhấn nút **Add current User** để thêm tài khoản. Cuối cùng bạn nhấn next cho đến mục **Ready to Install**.

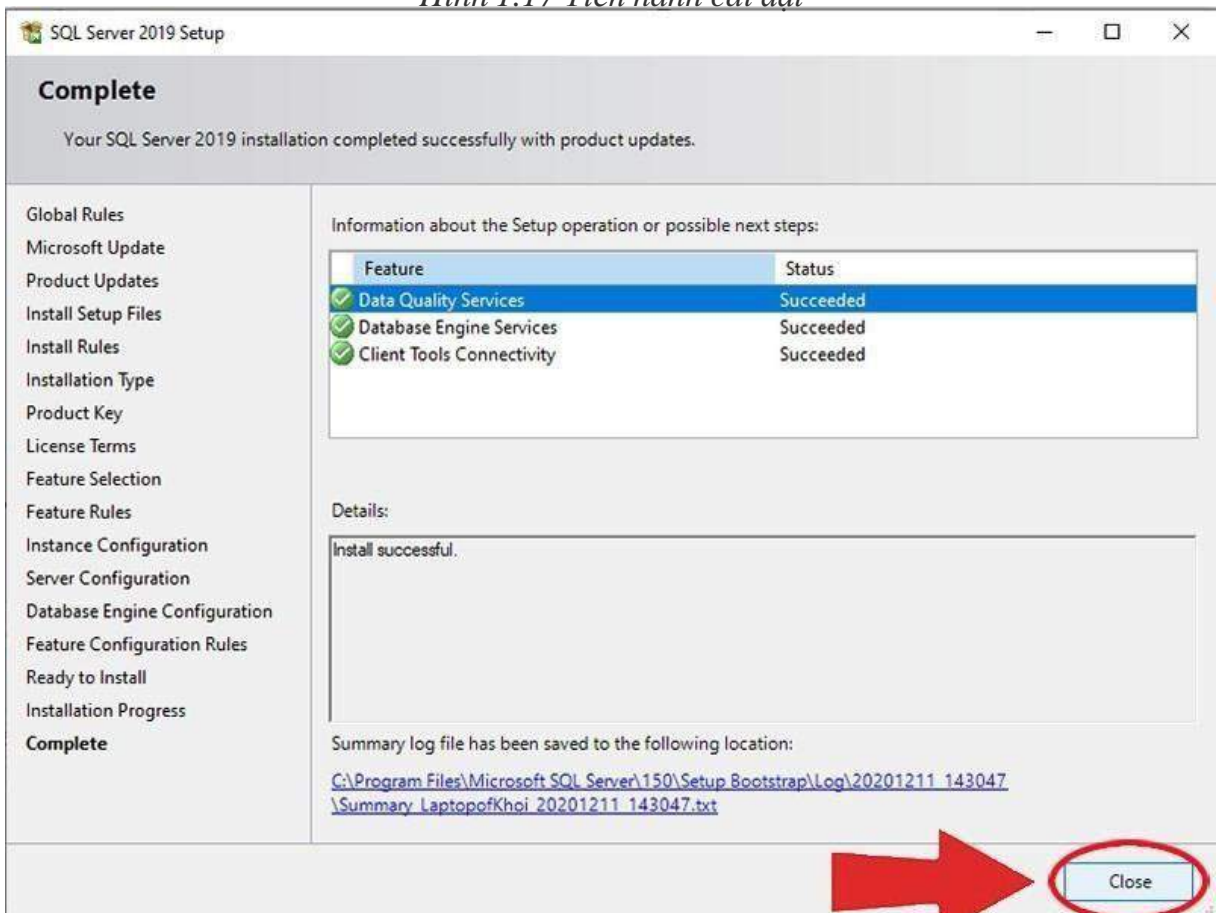


Hình 1.16 Xác định Password

Bước 9: Đây là phần cuối cùng. Bạn chỉ cần nhấn **Install** và đợi đến khi file cài đặt xong và bấm **close** để hoàn tất.



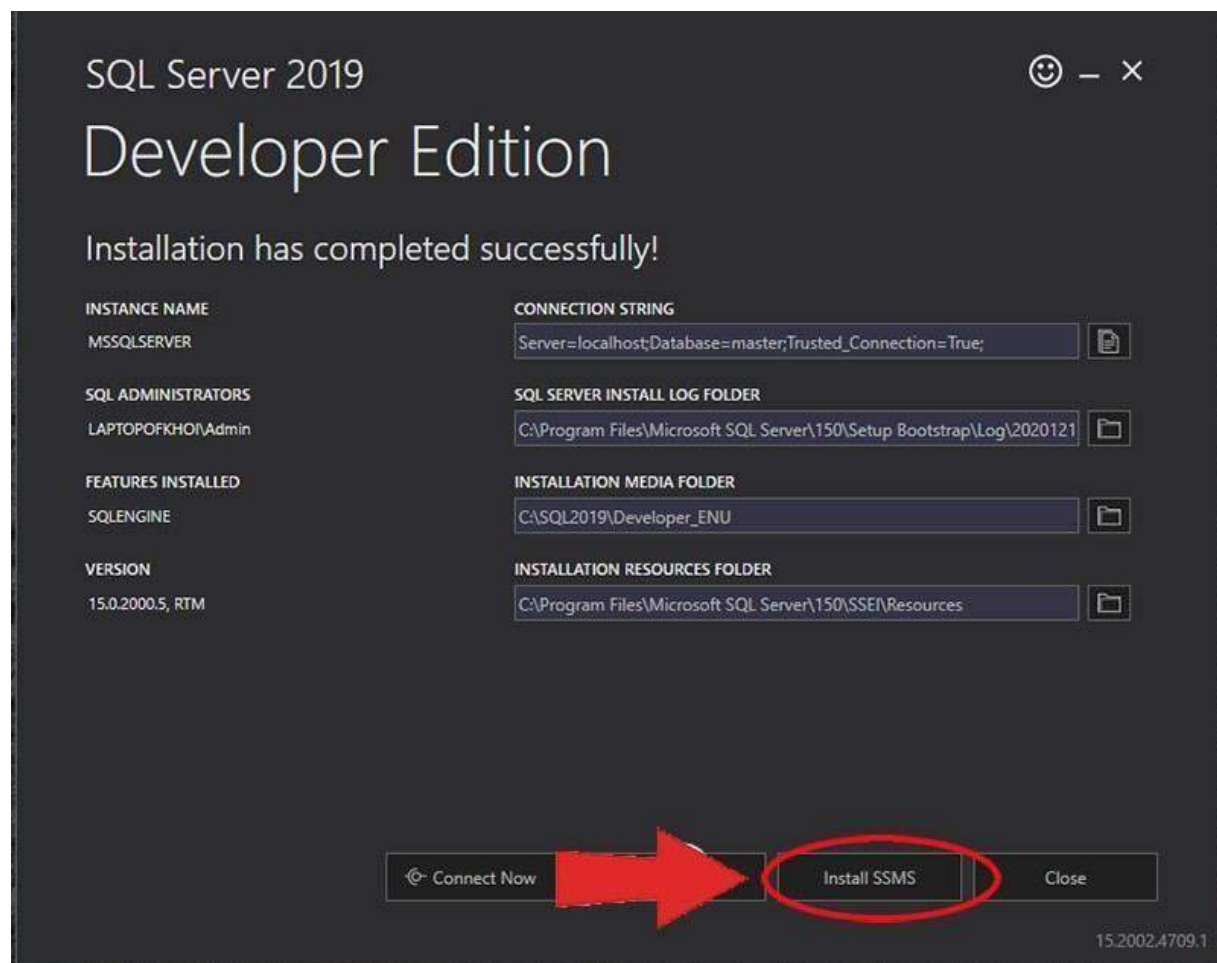
Hình 1.17 Tiến hành cài đặt



Hình 1.18 Hoàn thành cài đặt

3.3. Cài đặt SQL Server Management Studio (SSMS)

Bước 1: Ở màn hình Developer Edition chọn nút **Install SSMS**. Hệ thống sẽ tự động đưa bạn đến link tải. <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>



Hình 1.19 Tiên hành cài đặt SSMS

Bước 2: Ấn vào “Download SQL Server Management Studio (SSMS)” để tải file SSMS về

Version
SQL Server 2019

Filter by title

Download SSMS

Release Notes

- > Overview
- > Quickstarts
- > Tutorials
- > Concepts
- > How-to
- > References
- > Resources
- > SqlPackage.exe
- > SQL Server Profiler
- > Visual Studio native helpers

Download PDF

Download SQL Server Management Studio (SSMS)

10/27/2020 • 5 minutes to read

Applies to: SQL Server (all supported versions) Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases. Use SSMS to deploy, monitor, and upgrade the data-tier components used by your applications, and build queries and scripts.

Use SSMS to query, design, and manage your databases and data warehouses, wherever they are - on your local computer, or in the cloud.

Download SSMS

[Download SQL Server Management Studio \(SSMS\)](#)

SSMS 18.7.1 is the latest general availability (GA) version of SSMS. If you have a previous GA version of SSMS 18 installed, installing SSMS 18.7.1 upgrades it to 18.7.1.

Hình 1.20 Tải để cài đặt SSMS

Bước 3: Mở file vừa tải về, bấm **Install** để cài giao diện cho phần mềm

RELEASE 18.7.1

Microsoft SQL Server Management Studio with Azure Data Studio

Welcome. Click "Install" to begin.

Location:

C:\Program Files (x86)\Microsoft SQL Server Management Studio 18 Change

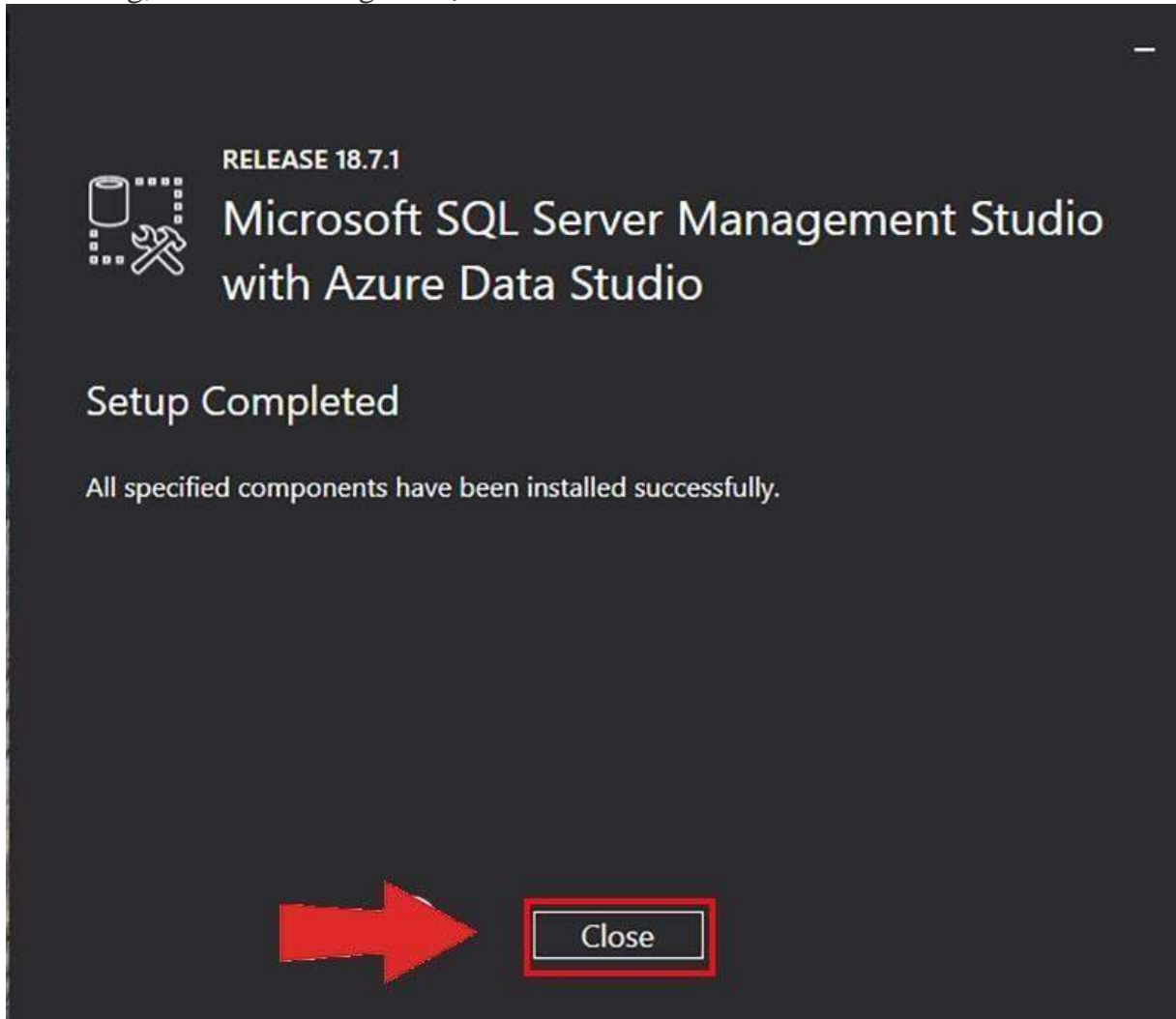
By clicking the "Install" button, I acknowledge that I accept the [Privacy Statement](#) and the License Terms for [SQL Server Management Studio](#) and [Azure Data Studio](#)

SQL Server Management Studio transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the [documentation](#)

Install Close

Hình 1.21 Bắt đầu cài đặt SSMS

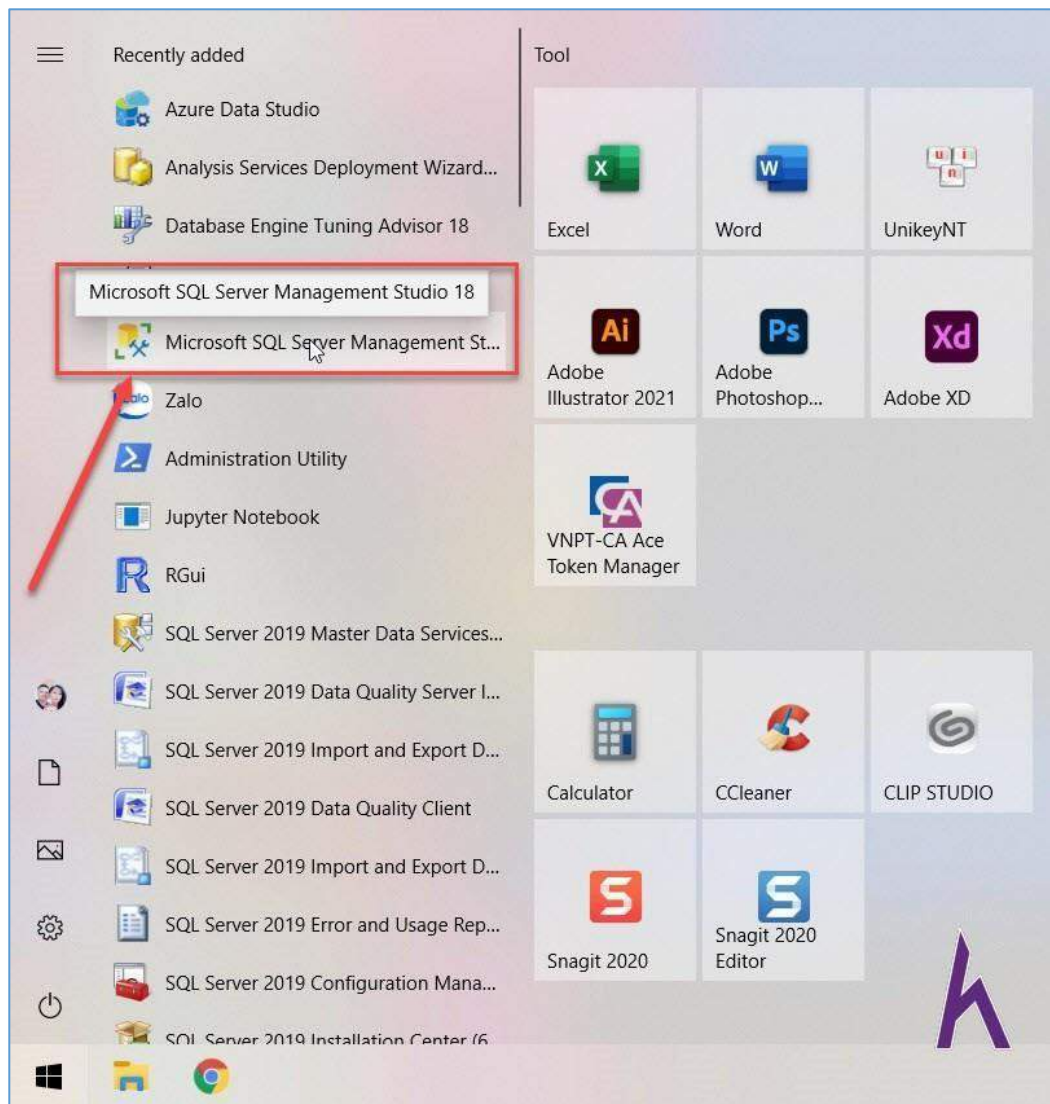
Cuối cùng, sau khi cài xong ta chọn **Close** để hoàn tất.



Hình 1.21 Hoàn tất cài đặt SSMS

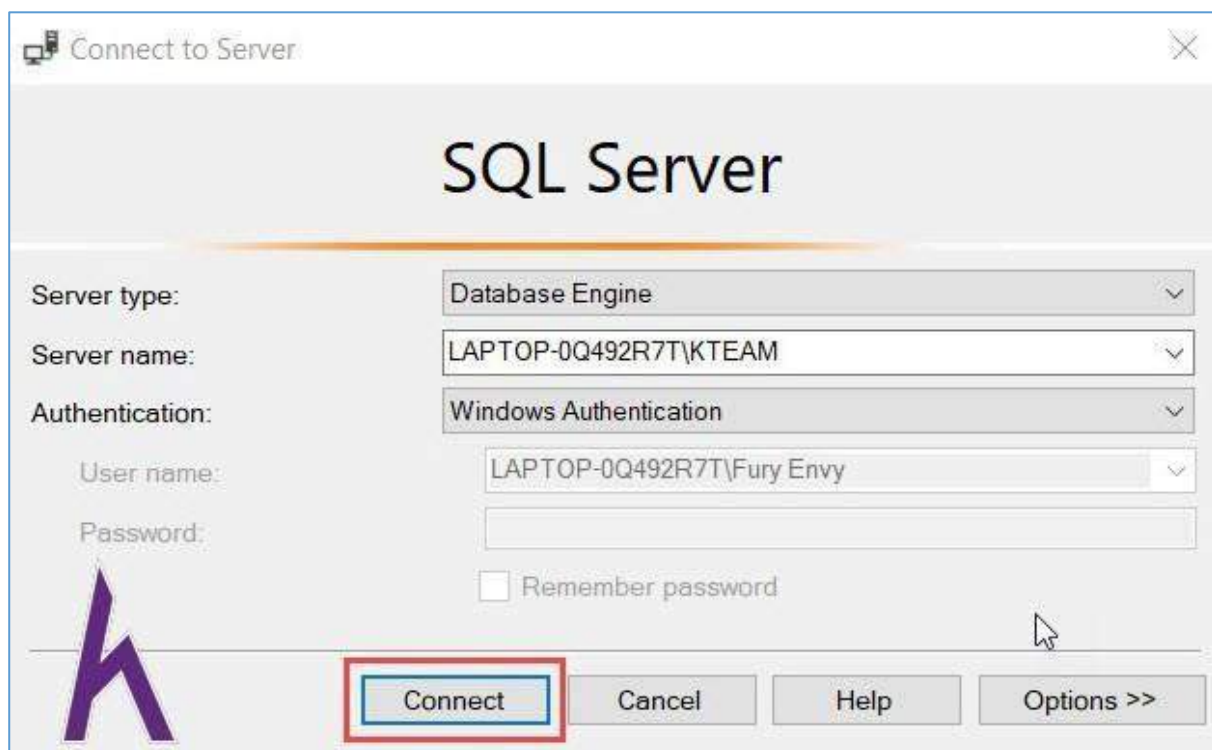
Bạn khởi động [Microsoft SQL Server Management Studio 201](#) lên để tiếp tục.

- Thiết lập tài khoản theo [Windows Authentication](#)(Đăng nhập theo tài khoản của máy).
 - Phần Server type: bạn chọn [Database Engine](#).
 - Phần Server name: bạn chọn tên máy của bạn.
 - Phần Authentication: bạn chọn [Windows Authentication](#).



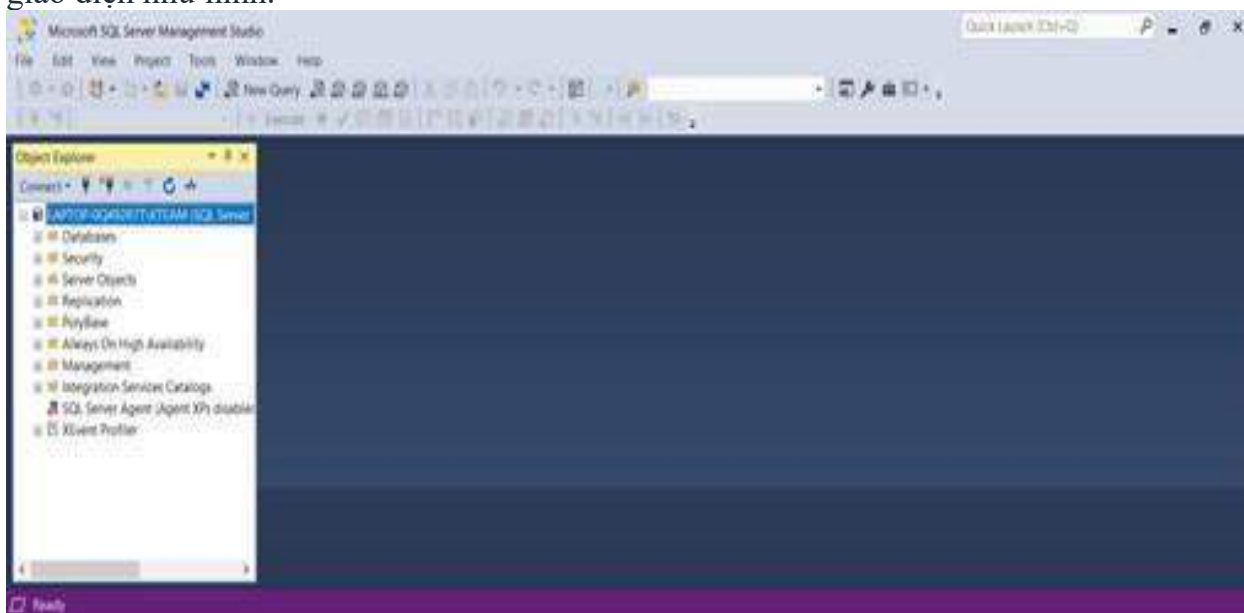
Hình 1.22 Mở SQL server 2019

Sau khi thiết lập bạn nhấn **Connect** để kết nối. Mình khuyên dùng tài khoản theo **SQL Server Authentication** thì sẽ bảo mật tốt hơn. Khi kết nối thành công ta sẽ được giao diện như hình.



Hình 1.23 Kết nối đến máy chủ SQL Server

Sau khi thiết lập bạn nhấn **Connect** để kết nối. Mình khuyên dùng tài khoản theo **SQL Server Authentication** thì sẽ bảo mật tốt hơn. Khi kết nối thành công ta sẽ được giao diện như hình.



Hình 1.24 Kết nối thành công đến máy chủ

Bài tập thực hành của học viên

1. Trình bày vai trò và sự cần thiết của SQL Server
2. Trình bày kiến trúc của SQL Server
3. Cài đặt SQL Server 2019

Hướng dẫn thực hiện

1. Trình bày vai trò của SQL Server, tham khảo mục 1.1.2 trong bài học trên
2. Trình bày sự cần thiết của SQL Server, tham khảo mục 1.2 trong bài học trên
3. Cài đặt SQL Server 2019, tham khảo mục 3 trong bài học trên

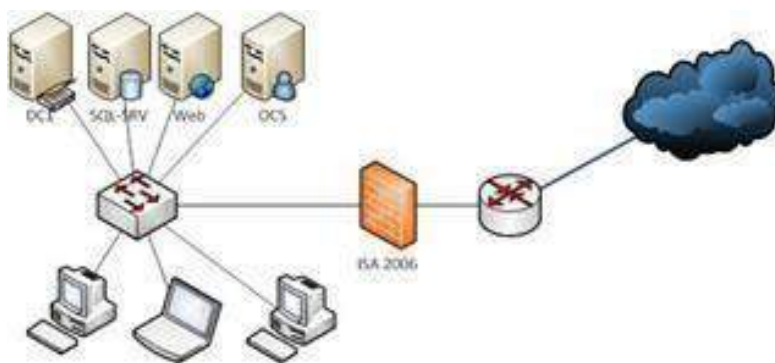
Những trọng tâm cần chú ý:

- Trình bày được lịch sử phát triển và đặc điểm của MS SQL Server
- Cấu hình máy phải đảm bảo yêu cầu cài đặt của SQL SERVER 2019
- Trình bày vai trò và sự cần thiết của SQL Server.
- Trình bày yêu cầu thiết bị phần cứng để cài đặt SQL Server
- Thao tác đúng các bước cài đặt SQL Server 2019.

Bài mở rộng và nâng cao

Cài đặt trên máy chủ SQL-SRV dịch vụ MS SQL Server 2019 để làm Database server lưu trữ dữ liệu tập trung cho hệ thống mạng của công ty

Mô hình



Hình 1: Sơ đồ tổng quát hệ thống mạng

Yêu cầu đánh giá kết quả học tập

Nội dung

- Về kiến thức:
 - + Trình bày vai trò và sự cần thiết của SQL Server
 - + Trình bày được lịch sử phát triển và đặc điểm của MS SQL Server
 - + Trình bày các bước cài đặt MS SQL Server với tên Instance Named là Default
 - + Trình bày các bước cài đặt MS SQL với tất cả các component và chứng thực với cơ chế local.
 - + Trình bày được các bước cài đặt SQLServer 2019
- Về kỹ năng:
 - + Thao tác thành thạo quản lý trên hệ thống.
 - + Thực hiện đúng các bước cài đặt SQLServer 2019
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

Phương pháp

- Về kiến thức: Đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp.
- Về kỹ năng: Đánh giá kỹ năng thực hành thực hiện các thao tác cài đặt SQLServer 2019.
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

BÀI 2: CÁC THÀNH PHẦN CƠ BẢN CỦA SQL SERVER

Mã bài: MD16-02

Mục tiêu:

- Hiểu được mô hình quan hệ;
- Hiểu được cấu trúc CSDL trong SQL SERVER như: Master, Model, msdl, Tempdb, pubs, tempdb, Northwind.

Nội dung chính:

1. Khái niệm cơ bản về mô hình quan hệ

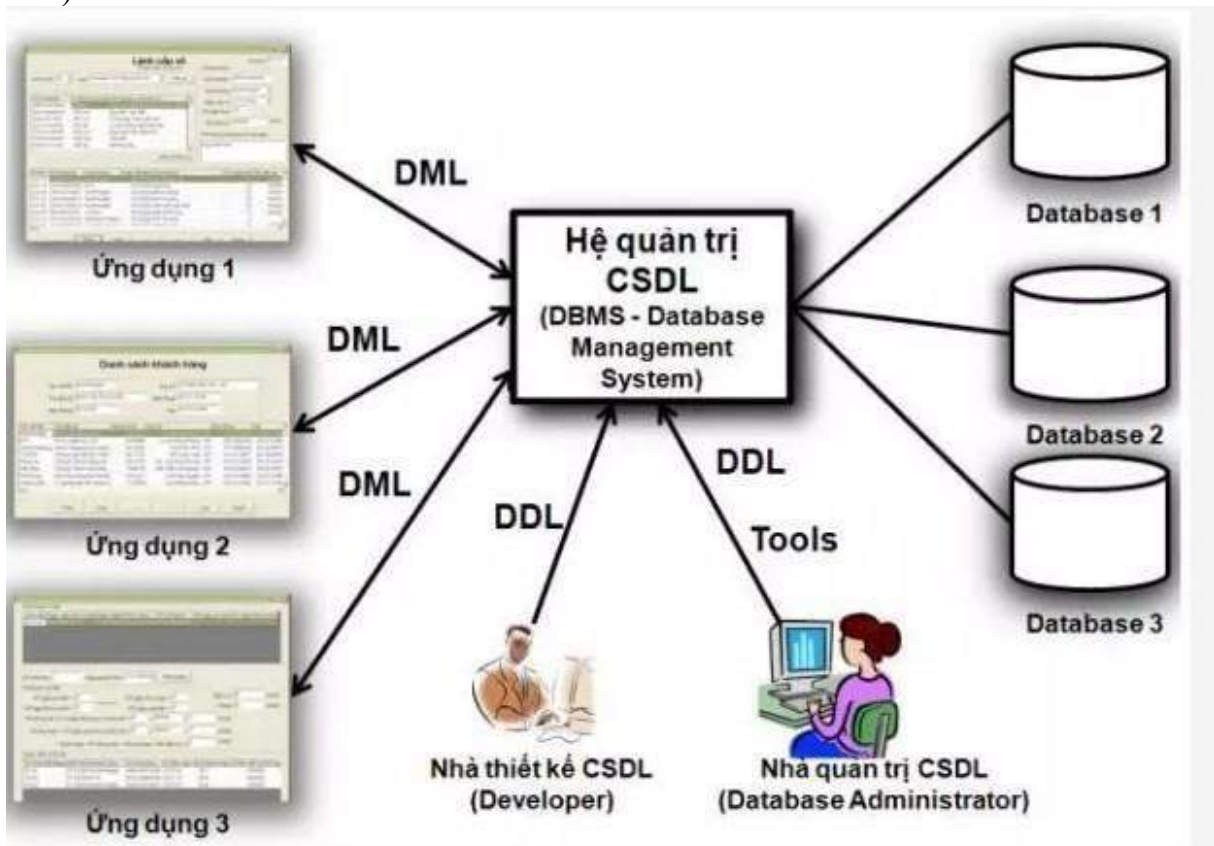
Mục tiêu: Hiểu được thế nào là mô hình quan hệ.

Trong hầu hết các cơ sở dữ liệu hiện nay, RDBMS không những lưu trữ dữ liệu mà còn quản trị hệ cơ sở dữ liệu bằng cách kiểm soát những dữ liệu nào được nhập vào và những kiểu dữ liệu nào có thể truy xuất ra khỏi hệ thống. Nếu muốn tất cả dữ liệu đều an toàn thì cần phải sử dụng đến hệ thống lưu trữ.

RDBMS cho phép lưu trữ dữ liệu cùng với những nguyên tắc ràng buộc dữ liệu do người dùng hay hệ thống định nghĩa, trong chương này chúng ta sẽ xem xét những thành phần của SQL Server, Kiểu dữ liệu, và các loại dữ liệu quan hệ

1.1 Mô hình dữ liệu quan hệ

Mô hình dữ liệu quan hệ được Codd đề xuất năm 1970 và đến nay trở thành mô hình được sử dụng phổ biến trong các hệ quản trị cơ sở dữ liệu thương mại. Nói một cách đơn giản, một cơ sở dữ liệu quan hệ là một cơ sở dữ liệu trong đó tất cả dữ liệu được tổ chức trong các bảng có mối quan hệ với nhau. Mỗi một bảng bao gồm các dòng và các cột: mỗi một dòng được gọi là một bản ghi (bộ) và mỗi một cột là một trường (thuộc tính).



Hình 2.1 Mô hình dữ liệu quan hệ

- Vai trò và lợi ích của việc sử dụng cơ sở dữ liệu trong quản lý thông tin dữ liệu của người dùng: từ một kịch bản thực tế, giảng viên cần chỉ ra các cách tiếp cận lưu trữ

và quản lý dữ liệu (tệp và CSDL), nêu các đặc điểm (điểm mạnh và hạn chế) của từng cách tiếp cận.

- Vai trò của hệ quản trị cơ sở dữ liệu trong việc khai thác và quản trị cơ sở dữ liệu
- Các khái niệm cơ bản về mô hình dữ liệu quan hệ: bảng/quan hệ, bản ghi/dòng dữ liệu, thuộc tính (attributes), miền giá trị (domain), khóa (primary key, foreign key)
- Tìm hiểu khái quát về các mô hình truy nhập dữ liệu SQL Server: mô hình tập trung, mô hình khách/chủ, mô hình hướng dịch vụ. Giảng viên có thể tổ chức thảo luận về các đặc điểm, ưu/nhược điểm của từng mô hình kiến trúc này. Ở đây, giảng viên cũng nên đưa ra các gợi ý về khả năng áp dụng và lựa chọn áp dụng các mô hình kiến trúc này tùy từng điều kiện phát triển ứng dụng cụ thể.
- Tìm hiểu khái quát về kiến trúc bên trong của SQL Server: các thành phần (components) và dịch vụ (services). Ở đây, giảng viên cũng nên làm rõ các thành phần cơ bản (bắt buộc) đối với hệ quản trị cơ sở dữ liệu (cụ thể là: quản lý lưu trữ, quản lý truy vấn và quản trị giao dịch) và các thành phần thêm vào cung cấp giá trị gia tăng cho xử lý dữ liệu trên SQL Server (ví dụ như dịch vụ tích hợp dữ liệu, dịch vụ phân tích dữ liệu, dịch vụ tạo báo cáo, ...)
- Tìm hiểu khái quát về các ngôn ngữ lập trình hỗ trợ trong SQL Server: Transact-SQL (hay T-SQL) và CLR. Giảng viên nên chỉ rõ sự khác nhau giữa các ngôn ngữ lập trình này và phạm vi ứng dụng hiệu quả.
- Một vài nét về quá trình phát triển và tiến hóa của hệ quản trị cơ sở dữ liệu SQL Server. Ở đây, giảng viên nên nhấn mạnh vào các khác biệt cơ bản giữa các “thế hệ” quản trị cơ sở dữ liệu: Thế hệ 2 là mở rộng của thế hệ 1 về khả năng định nghĩa kiểu dữ liệu của người dùng và một số dịch vụ phân tích dữ liệu. Thế hệ 3 (SQL Server 2014 và SQL Server 2019) là bước nhảy vọt nhờ tích hợp CLR bên trong lõi của hệ quản trị cơ sở dữ liệu SQL Server.
- Nêu khái quát các đặc điểm nổi trội của SQL Server 2019. Các dịch vụ hỗ trợ của SQL Server cho phép bảo vệ dữ liệu mọi nơi, mọi lúc.

1.2 Bảng (Table)

Như đã nói ở trên, trong cơ sở dữ liệu quan hệ, bảng là đối tượng được sử dụng để tổ chức và lưu trữ dữ liệu. Một cơ sở dữ liệu bao gồm nhiều bảng và mỗi bảng được xác định duy nhất bởi tên bảng. Một bảng bao gồm một tập các dòng và các cột: mỗi một dòng trong bảng biểu diễn cho một thực thể. (mỗi một dòng trong bảng SINHVIEN tương ứng với một sinh viên); và mỗi một cột biểu diễn cho một tính chất của thực thể (chẳng hạn cột TENKHOA trong bảng KHOA biểu diễn cho tên của các khoa được lưu trữ trong bảng).

Như vậy, liên quan đến mỗi một bảng bao gồm các yếu tố sau:

- Tên của bảng: được sử dụng để xác định duy nhất mỗi bảng trong cơ sở dữ liệu.
- Cấu trúc của bảng: Tập các cột trong bảng. Mỗi một cột trong bảng được xác định bởi một tên cột và phải có một kiểu dữ liệu nào đó (chẳng hạn TENKHOA trong bảng KHOA ở hình 1.1 có kiểu là CHAR). Kiểu dữ liệu của mỗi cột qui định giá trị dữ liệu có thể được chấp nhận trên cột đó.
- Dữ liệu của bảng: Tập các dòng (bản ghi) hiện có trong bảng.

1.3 Khoá của bảng

Trong một cơ sở dữ liệu được thiết kế tốt, mỗi một bảng phải có một hoặc một tập các cột mà giá trị dữ liệu của nó xác định duy nhất một dòng trong một tập các dòng của bảng. Tập một hoặc nhiều cột có tính chất này được gọi là khoá của bảng.

	makhoa	tenkhoa	dienthoai
1	KHOA01	Khoa Công nghệ thông tin	05113.940.675
2	KHOA02	Khoa Cơ bản	05113.940.676
3	KHOA03	Khoa Cơ khí	05113.940.677
4	KHOA04	Khoa Du lịch	05113.940.678
5	KHOA05	Khoa Điện	05113.940.679
6	KHOA06	Khoa Kinh tế	05113.940.670
7	KHOA07	Khoa May - TT thời trang	05113.940.671
8	KHOA08	Khoa Sư phạm nghề	05113.940.672

Việc chọn khoá của bảng có vai trò quan trọng trong việc thiết kế và cài đặt các cơ sở dữ liệu quan hệ. Các dòng dữ liệu trong một bảng phải có giá trị khác nhau trên khoá. Bảng MONHOC trong hình dưới đây có khoá là cột MAMONHOC

Một bảng có thể có nhiều tập các cột khác nhau có tính chất của khoá (tức là giá trị của nó xác định duy nhất một dòng dữ liệu trong bảng). Trong trường hợp này, khoá được chọn cho bảng được gọi là khoá chính (primary key) và những khoá còn lại được gọi là khoá phụ hay là khoá dự tuyển (candidate key/unique key).

1.4 Mỗi quan hệ và khoá ngoài

Các bảng trong một cơ sở dữ liệu không tồn tại độc lập mà có mối quan hệ mật thiết với nhau về mặt dữ liệu. Mối quan hệ này được thể hiện thông qua ràng buộc giá trị dữ liệu xuất hiện ở bảng này phải có xuất hiện trước trong một bảng khác. Mối quan hệ giữa các bảng trong cơ sở dữ liệu nhằm đảm bảo được tính đúng đắn và hợp lệ của dữ liệu trong cơ sở dữ liệu.

MaLop	TenLop	SiSo	MaKhoa
11T1D	Cao đẳng Sửa chữa máy tính	22	KHOA01
11T1E	Cao đẳng Lập trình máy tính	55	KHOA01
11T1F	Cao đẳng Quản trị mạng	45	KHOA01
11T6A	Trung cấp Quản trị mạng	55	KHOA01
11T6B	Trung cấp Sửa chữa máy tính	15	KHOA01
12CK...	Cao đẳng Cơ khí động lực	28	KHOA03
12DL...	Cao đẳng Hướng dẫn du lịch	65	KHOA04
12D...	Cao đẳng Điện tử	25	KHOA05
12KT...	Cao đẳng Kế toán doanh ng...	65	KHOA06

	makhoa	tenkhoa	dienthoai
1	KHOA01	Khoa Công nghệ thông tin	05113.940.675
2	KHOA02	Khoa Cơ bản	05113.940.676
3	KHOA03	Khoa Cơ khí	05113.940.677
4	KHOA04	Khoa Du lịch	05113.940.678
5	KHOA05	Khoa Điện	05113.940.679
6	KHOA06	Khoa Kinh tế	05113.940.670
7	KHOA07	Khoa May - TT thời trang	05113.940.671
8	KHOA08	Khoa Sư phạm nghề	05113.940.672

Hình 2.2 Quan hệ 2 bảng trong một cơ sở dữ liệu

Mối quan hệ giữa các bảng trong một cơ sở dữ liệu thể hiện đúng mối quan hệ giữa các thực thể trong thế giới thực, mối quan hệ giữa hai bảng LOP và KHOA không cho phép một lớp nào đó tồn tại mà lại thuộc vào một khoa không có thật.

Khái niệm khoá ngoài (Foreign Key) trong cơ sở dữ liệu quan hệ được sử dụng để biểu diễn mối quan hệ giữa các bảng dữ liệu. Một hay một tập các cột trong một bảng mà giá trị của nó được xác định từ khóa chính của một bảng khác được gọi là khoá ngoài. Cột MAKHOA của bảng LOP được gọi là khoá ngoài của bảng này, khoá ngoài này tham chiếu đến khoá chính của bảng KHOA là cột MAKHOA.

1.5 Câu lệnh SQL

SQL chuẩn bao gồm khoảng 40 câu lệnh. Trong các hệ quản trị cơ sở dữ liệu khác nhau, mặc dù các câu lệnh đều có cùng dạng và cùng mục đích sử dụng song mỗi một hệ quản trị cơ sở dữ liệu có thể có một số thay đổi nào đó. Điều này đôi khi dẫn đến cú pháp chi tiết của các câu lệnh có thể sẽ khác nhau trong các hệ quản trị cơ sở dữ liệu khác nhau.

Các câu lệnh của SQL chuẩn:

Bảng 2.1: Một số câu lệnh thông dụng trong SQL

Câu lệnh	Chức năng
<i>Thao tác dữ liệu</i>	
SELECT	Truy xuất dữ liệu
INSERT	Bổ sung dữ liệu
UPDATE	Cập nhật dữ liệu
DELETE	Xoá dữ liệu
TRUNCATE	Xoá toàn bộ dữ liệu trong bảng
<i>Định nghĩa dữ liệu</i>	
CREATE TABLE	Tạo bảng
DROP TABLE	Xoá bảng
ALTER TABLE	Sửa đổi bảng
CREATE VIEW	Tạo khung nhìn
ALTER VIEW	Sửa đổi khung nhìn
DROP VIEW	Xoá khung nhìn
CREATE INDEX	Tạo chỉ mục
DROP INDEX	Xoá chỉ mục
CREATE SCHEMA	Tạo lược đồ cơ sở dữ liệu
DROP SCHEMA	Xoá lược đồ cơ sở dữ liệu
CREATE PROCEDURE	Tạo thủ tục lưu trữ
ALTER PROCEDURE	Sửa đổi thủ tục lưu trữ
DROP PROCEDURE	Xoá thủ tục lưu trữ
CREATE FUNCTION	Tạo hàm (do người sử dụng định nghĩa)
ALTER FUNCTION	Sửa đổi hàm
DROP FUNCTION	Xoá hàm
CREATE TRIGGER	Tạo trigger
ALTER TRIGGER	Sửa đổi trigger
DROP TRIGGER	Xoá trigger
<i>Điều khiển truy cập</i>	
GRANT	Cấp phát quyền cho người sử dụng
REVOKE	Thu hồi quyền từ người sử dụng
<i>Quản lý giao tác</i>	
COMMIT	Ủy thác (kết thúc thành công) giao tác
ROLLBACK	Quay lui giao tác
SAVE TRANSACTION	Đánh dấu một điểm trong giao tác
<i>Lập trình</i>	
DECLARE	Khai báo biến hoặc định nghĩa con trỏ
OPEN	Mở một con trỏ để truy xuất kết quả truy vấn
FETCH	Đọc một dòng trong kết quả truy vấn (sử dụng con trỏ)
CLOSE	Đóng một con trỏ
EXECUTE	Thực thi một câu lệnh SQL

Các câu lệnh của SQL đều được bắt đầu bởi các từ lệnh, là một từ khoá cho biết chức năng của câu lệnh (chẳng hạn SELECT, DELETE, COMMIT). Sau từ lệnh là các mệnh đề của câu lệnh. Mỗi một mệnh đề trong câu lệnh cũng được bắt đầu bởi một từ khoá (chẳng hạn FROM, WHERE,...).

Ví dụ: Câu lệnh:

```
SELECT masv,hodem,ten
FROM sinhvien
WHERE malop='10T1a'
```

dùng để truy xuất dữ liệu trong bảng SINHVIEN được bắt đầu bởi từ lệnh SELECT, trong câu lệnh bao gồm hai mệnh đề: mệnh đề FROM chỉ định tên của bảng cần truy xuất dữ liệu và mệnh đề WHERE chỉ định điều kiện truy vấn dữ liệu.

1.6 Quy tắc sử dụng tên trong SQL

Các đối tượng trong cơ sở dữ liệu dựa trên SQL được xác định thông qua tên của đối tượng. Tên của các đối tượng là duy nhất trong mỗi cơ sở dữ liệu. Tên được sử dụng nhiều nhất trong các truy vấn SQL và được xem là nền tảng trong cơ sở dữ liệu quan hệ là tên bảng và tên cột.

Trong các cơ sở dữ liệu lớn với nhiều người sử dụng, khi ta chỉ định tên của một bảng nào đó trong câu lệnh SQL, hệ quản trị cơ sở dữ liệu hiểu đó là tên của bảng do ta sở hữu (tức là bảng do ta tạo ra). Thông thường, trong các hệ quản trị cơ sở dữ liệu này cho phép những người dùng khác nhau tạo ra những bảng trùng tên với nhau mà không gây ra xung đột về tên. Nếu trong một câu lệnh SQL ta cần chỉ đến một bảng do một người dùng khác sở hữu (hiển nhiên là phải được phép) thì tên của bảng phải được viết sau tên của người sở hữu và phân cách với tên người sở hữu bởi dấu chấm:

tên_người_sở_hữu.tên_bảng

Một số đối tượng cơ sở dữ liệu khác (như khung nhìn, thủ tục, hàm), việc sử dụng tên cũng tương tự như đối với bảng.

Ta có thể sử dụng tên cột một cách bình thường trong các câu lệnh SQL bằng cách chỉ cần chỉ định tên của cột trong bảng. Tuy nhiên, nếu trong câu lệnh có liên quan đến hai cột trở lên có cùng tên trong các bảng khác nhau thì bắt buộc phải chỉ định thêm tên bảng trước tên cột; tên bảng và tên cột được phân cách nhau bởi dấu chấm.

Ví dụ dưới đây minh họa cho ta thấy việc sử dụng tên bảng và tên cột trong câu lệnh SQL

```
SELECT masv,hodem,ten,sinhvien.malop,tenlop
FROM dbo.sinhvien,dbo.lop
WHERE sinhvien.malop = lop.malop
```

2. Cấu trúc của SQL SERVER

Mục tiêu: Liệt kê được các đối tượng cấu thành SQL server.

2.1 Các cơ sở dữ liệu hệ thống của SQL SERVER

RDBMS cũng như SQL SERVER chứa đựng nhiều đối tượng bao gồm:

- Database: cơ sở dữ liệu của SQL SERVER
- Tập tin log: tập tin lưu trữ những chuyển tác của SQL
- Tables: bảng dữ liệu.
- Filegroups: tập tin nhóm
- Diagrams: sơ đồ quan hệ
- Views: Khung nhìn (hay bảng ảo) số liệu dựa trên bảng.
- Stored Procedure: Thủ tục và hàm nội
- User defined Function: Hàm do người dùng định nghĩa
- Users: Người sử dụng cơ sở dữ liệu
- Roles: Các qui định vai trò và chức năng trong hệ thống SQL SERVER
- Rules: Những qui tắc
- Defaults: Các giá trị mặc nhiên
- User defined data types: Kiểu dữ liệu do người dùng tự định nghĩa

- Full text catalogs: Tập phân loại dữ liệu Text

Cơ sở dữ liệu là đối tượng có ảnh hưởng cao nhất khi chúng ta làm việc với SQL SERVER, tuy nhiên những đối tượng con của cơ sở dữ liệu mới là thành phần chính của cơ sở dữ liệu.

Đặc điểm của SQL Server 2019 là **Multiple-Instance** nên khi nói đến một (SQL) Server nào đó là ta nói đến một Instance của SQL Server 2000, thông thường đó là Default Instance. Một Instance của SQL Server 2000 có 4 system databases và một hay nhiều user database. Các system databases bao gồm:

- **Master:** Chứa tất cả những thông tin cấp hệ thống (system-level information) bao gồm thông tin về các database khác trong hệ thống như vị trí của các data files, các login account và các thiết đặt cấu hình hệ thống của SQL Server (system configuration settings).
- **Tempdb:** Chứa tất cả những table hay stored procedure được tạm thời tạo ra trong quá trình làm việc bởi user hay do bản thân SQL Server engine. Các table hay stored procedure này sẽ biến mất khi khởi động lại SQL Server hay khi ta disconnect.
- **Model:** Database này đóng vai trò như một bảng kẽm (template) cho các database khác. Nghĩa là khi một user database được tạo ra thì SQL Server sẽ copy toàn bộ các system objects (tables, stored procedures...) từ Model database sang database mới vừa tạo.
- **Msdb:** Database này được SQL Server Agent sử dụng để hoạch định các báo động và các công việc cần làm (schedule alerts and jobs).

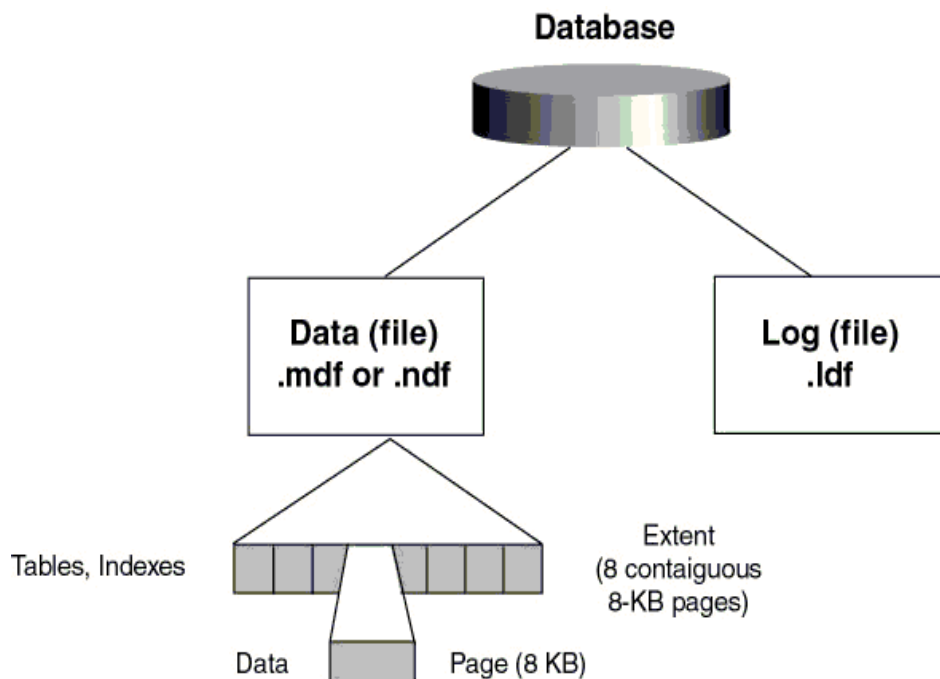
2.2. Cấu trúc vật lý của một SQL Server Database

Mỗi một database trong SQL Server được lưu trữ dưới dạng 2 tập tin:

- Tập tin dữ liệu (data file): Lưu trữ dữ liệu gồm 2 loại:
 - **Primary data file** (tên file có phần mở rộng là **.mdf**): đây là file chính chứa data và những system tables.
 - **Secondary data file** (tên file có phần mở rộng là **.ndf**): đây là file phụ thường chỉ sử dụng khi database được phân chia để chứa trên nhiều đĩa.
- Tập tin Log (Transaction log file – tên file có phần mở rộng là **.ldf**): đây là file ghi lại tất cả những thay đổi diễn ra trong một database và chứa đầy đủ thông tin để có thể roll back hay roll forward khi cần.

Ghi chú; Mỗi một database trong SQL Server có ít nhất một tập tin dữ liệu và một tập tin log:

Data trong SQL Server được chứa thành từng **Page** 8KB và 8 page liên tục tạo thành một **Extent** như hình vẽ dưới đây:



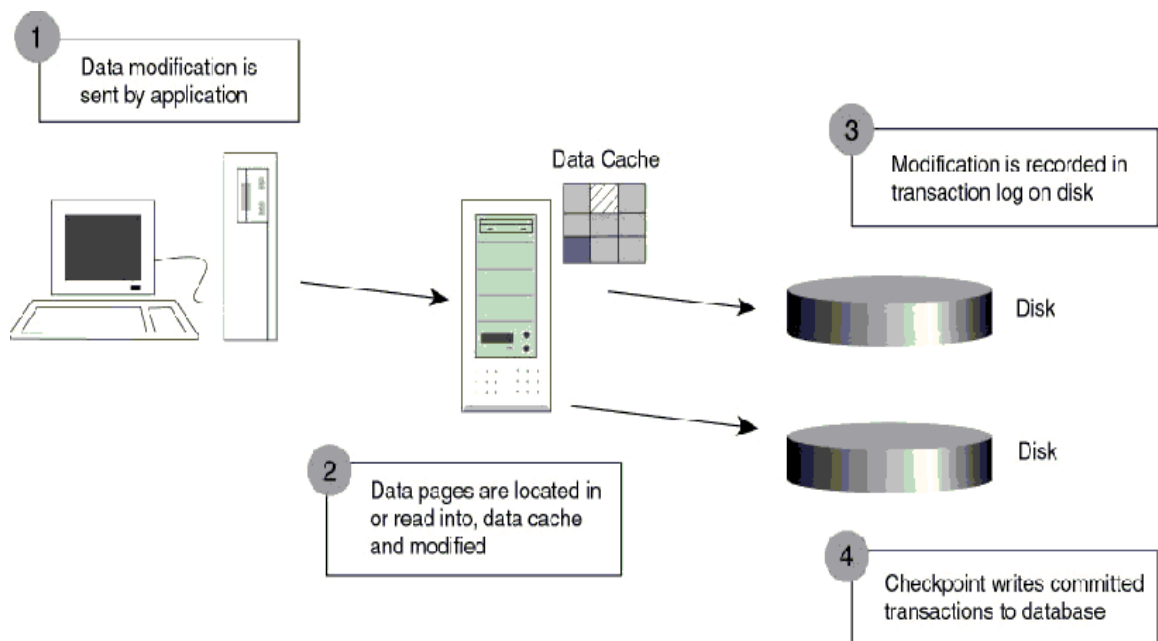
Max row size = 8060 bytes

Hình 2.3 Cấu trúc vật lý của một SQL Server Database

Trước khi SQL Server muốn lưu data vào một table nó cần phải dành riêng một khoảng trống trong data file cho table đó. Những khoảng trống đó chính là các extents. Có 2 loại Extents: **Mixed Extents** (loại hỗn hợp) dùng để chứa data của nhiều tables trong cùng một Extent và **Uniform Extent** (loại thuần nhất) dùng để chứa data của một table. Đầu tiên SQL Server dành các Page trong Mixed Extent để chứa data cho một table sau đó khi data tăng trưởng thì SQL dành hẳn một Uniform Extent cho table đó.

2.3 Nguyên tắc hoạt động của Transaction Log Trong SQL Server

Transaction log file trong SQL Server dùng để ghi lại các thay đổi xảy ra trong database. Quá trình này diễn ra như sau: đầu tiên khi có một sự thay đổi data như Insert, Update, Delete được yêu cầu từ các ứng dụng, SQL Server sẽ tải (load) data page tương ứng lên memory (vùng bộ nhớ này gọi là data cache), sau đó data trong data cache được thay đổi (những trang bị thay đổi còn gọi là *dirty-page*). Tiếp theo mọi sự thay đổi đều được ghi vào transaction log file cho nên người ta gọi là *write-ahead* log. Cuối cùng thì một quá trình gọi là **Check Point Process** sẽ kiểm tra và viết tất cả những transaction đã được committed (hoàn tất) vào đĩa cứng (flushing the page).



Hình 2.4 Nguyên tắc hoạt động của Transaction Log

Ngoài Check Point Process những dirty-page còn được đưa vào đĩa bởi một **Lazy writer**. Đây là một anh chàng làm việc âm thầm chỉ thức giấc và quét qua phần data cache theo một chu kỳ nhất định sau đó lại ngủ yên chờ lần quét tới.

Xin giải thích thêm một chút về khái niệm transaction trong database. Một transaction hay một giao dịch là một loạt các hoạt động xảy ra được xem như một công việc đơn (unit of work) nghĩa là hoặc thành công toàn bộ hoặc không làm gì cả (all or nothing). Sau đây là một ví dụ cổ điển về transaction:

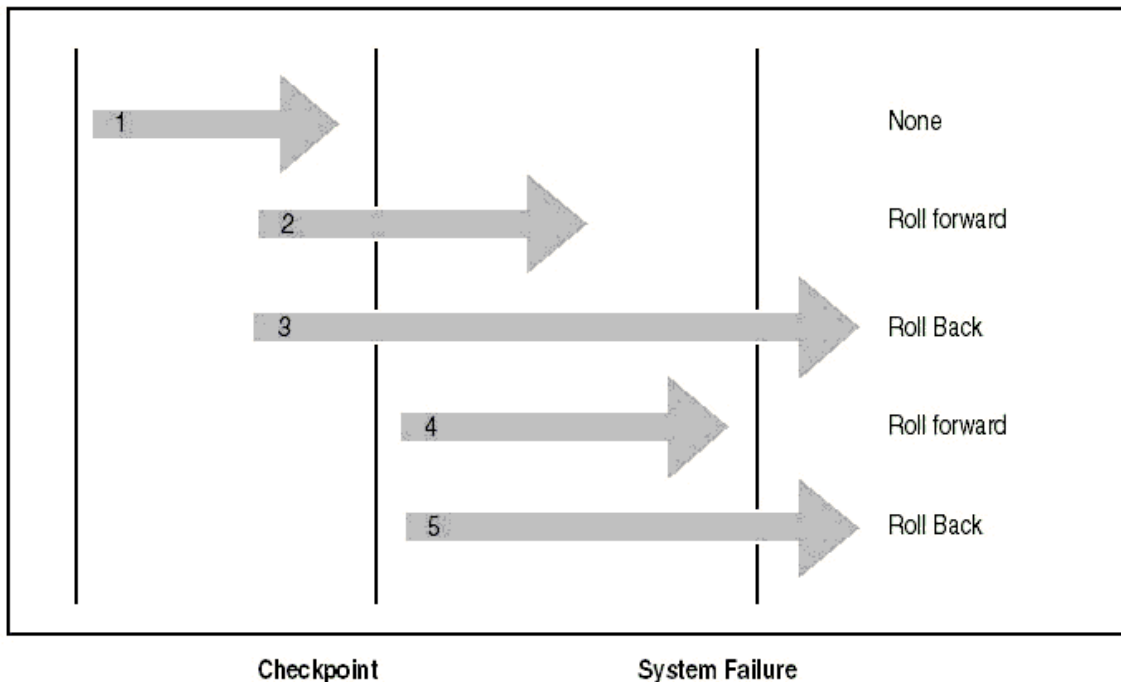
Chúng ta muốn chuyển một số tiền \$500 từ account A sang account B như vậy công việc này cần làm các bước sau:

1. Trừ \$500 từ account A
2. Cộng \$500 vào account B

Tuy nhiên việc chuyển tiền trên phải được thực hiện dưới dạng một transaction nghĩa là giao dịch chỉ được xem là hoàn tất (committed) khi cả hai bước trên đều thực hiện thành công. Nếu vì một lý do nào đó ta chỉ có thể thực hiện được bước 1 (chẳng hạn như vừa xong bước 1 thì điện cúp hay máy bị treo) thì xem như giao dịch không hoàn tất và cần phải được phục hồi lại trạng thái ban đầu (roll back).

Thế thì Check Point Process hoạt động như thế nào để có thể đảm bảo một transaction được thực thi mà không làm "đơ" database.

Transaction Recovery



Hình 2.5 Mô hình chuyển đổi

Trong hình vẽ trên, một transaction được biểu diễn bằng một mũi tên. Trục nằm ngang là trục thời gian. Giả sử một Check Point được đánh dấu vào thời điểm giữa transaction 2 và 3 như hình vẽ và sau đó sự cố xảy ra trước khi gặp một Check point kế tiếp. Như vậy khi SQL Server được restart nó sẽ dựa trên những gì ghi trong transaction log file để phục hồi data (xem hình vẽ).

Điều đó có nghĩa là SQL Server sẽ không cần làm gì cả đối với transaction 1 vì tại thời điểm Check point data đã được lưu vào đĩa rồi. Trong khi đó transaction 2 và 4 sẽ được roll forward vì tuy đã được committed nhưng do sự cố xảy ra trước thời điểm check point kế tiếp nên data chưa kịp lưu vào đĩa. Tức là dựa trên những thông tin được ghi trên log file SQL Server hoàn toàn có đầy đủ cơ sở để viết vào đĩa cứng. Còn transaction 3 và 5 thì chưa được committed (do bị down bất ngờ) cho nên SQL Server sẽ roll back hai transaction này dựa trên những gì được ghi trên log file.

2.4 Cấu Trúc Logic Của Một SQL Server Database

Mỗi Database trong SQL Server đều được tổ chức thành những objects như tables, views, stored procedures, indexes, constraints.... Những system objects trong SQL Server thường có bắt đầu bằng chữ sys hay sp. Các objects trên sẽ được nghiên cứu lần lượt trong các chương tiếp theo của giáo trình

Bài tập thực hành của học viên

1. Trình bày mô hình quan hệ của SQL Server
2. Trình bày cấu trúc của SQL Server
3. Khởi động SQL server, quan sát và ghi chép lại các thành phần của cơ sở dữ liệu hiện có trên SQL server 2019

Hướng dẫn thực hiện

1. Trình bày mô hình quan hệ của SQL Server, tham khảo mục 1 trong bài 2 trong giáo trình

2. Trình bày cấu trúc của SQL Server, tham khảo mục 2 trong bài 2 trong giáo trình
 3. Khởi động SQL server, quan sát và ghi chép lại các thành phần của cơ sở dữ liệu hiện có trên SQL server 2019
 - Tên database, tên tập tin log, tên các bảng dữ liệu.
 - Filegroups, diagrams, khung nhìn (hay bảng ảo).
 - Thủ tục và hàm nội, hàm do người dùng định nghĩa, người sử dụng cơ sở dữ liệu.
- Các qui định vai trò và chức năng trong hệ thống SQL SERVER, những qui tắc, các giá trị mặc nhiên

Những trọng tâm cần chú ý:

- Trình bày mô hình quan hệ của SQL Server
- Trình bày cấu trúc của SQL Server
- Trình bày Cấu trúc vật lý của một SQL Server Database
- Khởi động SQL server, quan sát và ghi chép lại các thành phần của cơ sở dữ liệu hiện có trên SQL server 2019
- Thực hiện đúng việc bật CLR integration và Database Mail
- Cài đặt số lượng kết nối tối đa vào SQL Server
- Sử dụng lệnh sqlcmd để liệt kê danh sách CSDL trên server

Bài mở rộng và nâng cao

Cấu hình những thông số cơ bản giúp người dùng có thể kết nối và truy cập database.

Yêu cầu đánh giá kết quả học tập

Nội dung

- Về kiến thức:
 - + Trình bày Cấu trúc vật lý của một SQL Server Database
 - + Khởi động SQL Sever Management Studio và đăng nhập vào SQL server
 - + Trình bày các bước bật CLR integration và Database Mail
 - + Trình bày các bước cài đặt số lượng kết nối tối đa vào SQL Server
 - + Trình bày được các bước kết nối và truy cập database vào SQLServer 2019
- Về kỹ năng:
 - + Thực hiện đúng việc bật CLR integration và Database Mail
 - + Cài đặt số lượng kết nối tối đa vào SQL Server
 - + Sử dụng lệnh sqlcmd để liệt kê danh sách CSDL trên server
 - + Thực hiện đúng các bước kết nối và truy cập database vào SQLServer 2019
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

Phương pháp

- Về kiến thức: Đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp.
- Về kỹ năng: Đánh giá kỹ năng thực hành thực hiện các thao tác bật CLR integration và Database Mail, Cài đặt số lượng kết nối tối đa vào SQLServer 2019.
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

BÀI 3: GIỚI THIỆU MỘT SỐ CÔNG CỤ TRONG SQL SERVER

Mã bài: MĐ16-03

Mục tiêu:

- Hiểu được và sử dụng tốt các công cụ như: Enterprise manager, Query Analyzer
- Hiểu và thiết lập được các dịch vụ mạng và một số dịch vụ khác có liên quan.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Sử dụng công cụ Enterprise manager

Mục tiêu: Trình bày được các công cụ cơ bản của SQL và công dụng của chúng.

1.1. SQL enterprise manager

Enterprise Manager còn gọi tắt là EM, là một công cụ chính dành cho nhà quản trị server và CSDL. Enterprise Manager cho phép bạn dừng và khởi động một Server, cũng như cho phép bạn thực hiện các tác vụ sau:

- Đăng ký một server.
- Cấu hình các server cục bộ hoặc từ xa.
- Cấu hình và quản lý một cài đặt với nhiều server (multiple-server).
- Cài đặt các bảo mật đăng nhập (login security), thêm các người dùng (user), các nhà quản trị hệ thống (system administrator), và các điều hành viên.
- Gán một password nhà quản trị hệ thống.
- Tạo và lập biểu cho các công việc (job).
- Tạo các cảnh báo và cấu hình giao tiếp đến nhà quản trị hệ thống thông qua e-mail.

Cài đặt và quản trị các CSDL, các bảng (table), các chỉ mục (index), các truy vấn (view), các thủ tục (stored procedure), các qui tắc (rule), các bẫy lỗi (trigger), các mặc định (default), các thiết bị dự phòng (backup device), và các vết lỗi (error log).

- Quản lý các service khác.

1.2. SQL server services manager

SQL Server Manager là một công cụ cho phép khởi động, tạm dừng hoặc dừng các dịch vụ (service) trong SQL Server. SQL Server hoạt động được thông qua các dịch vụ mà nó tự cung cấp.

1.2.1 Các dịch vụ của SQL Server

Một dịch vụ (service) là một chương trình hoặc tiến trình thực thi một chức năng đặc biệt nào đó nhằm hỗ trợ cho các chương trình khác. SQL Server cung cấp các dịch vụ sau: SQL Server Service, SQL Server Agent Service, MicroSoft Search Service và Microsoft Distributed Transaction Coordinator. Mỗi dịch vụ có chức năng và nhiệm vụ riêng, hỗ trợ cho các hoạt động của SQL Server. Nếu cài đặt một hoặc nhiều hơn thể hiện (instance) của SQL Server thì tên của service cho mỗi instance của SQL Server là MSSQL\$InstanceName, \$InstanceName là một tên của instance chúng ta chỉ định ở lần cài đặt. Ứng với mỗi SQL Server Agent service cho mỗi instance được gọi là SQLAGENT\$InstanceName. Tuy nhiên, nhiều instances của SQL Server, sẽ chỉ có một Microsoft Distributed Transaction Coordinator và Microsoft Search.

a. SQL SERVER SERVICE

Khi bạn khởi động SQL Server có nghĩa là dịch vụ SQL Server service được khởi động ở Windows NT hoặc Windows 2000. Dịch vụ này quản lý các tập tin CSDL, xử lý các câu lệnh T-SQL, định vị tài nguyên giữa các kết nối của người dùng hiện hành, đảm bảo tính nhất quán dữ liệu, và nhiều hơn nữa.

b. SQL SERVER AGENT SERVICES

SQL Server Agent hỗ trợ việc lập biểu và thực thi các công việc (job), các cảnh báo (alert), thông báo, và kế hoạch duy trì CSDL. Không có service này, công việc quản trị sẽ trở nên khó khăn nhiều. SQL Server Agent cho phép thực hiện tự động các thao tác nhằm duy trì CSDL.

Ví dụ như chúng ta có thể tạo một job để thực hiện tự động dự phòng (backup) dữ liệu mỗi đêm vào lúc 1 giờ sáng và một job khác thực hiện dự phòng transaction log mỗi 30 phút một lần.

Để kiểm tra hiệu năng hệ thống của bạn cảnh báo hiện trạng hiệu năng để báo cho bạn nếu server CPU đã hoạt động trên 90%. SQL Server Agent phải chạy để thực thi các tác vụ kiểu như vậy.

Dịch vụ này có thể được cấu hình khởi chạy một các tự động hoặc chạy một cách thủ công. Chúng ta nên cấu hình cho nó khởi động tự động để đảm bảo rằng các job, alerts, và notification sẽ có thể được thực thi.

c. MICROSOFT DISTRIBUTED TRANSACTION COORDINATOR.

Dùng quản lý các giao tác phân tán.

d. MICROSOFT SEARCH.

1.2.2. Khởi động, tạm ngưng, dừng các dịch vụ của SQL Server

Để khởi động hay dừng các dịch vụ SQL Server bằng cách dùng SQL Server Service Manager thì thực hiện các bước sau:

1. Click chọn Start/Programs/Microsoft SQL Server, và chọn Service Manager để hiển thị công cụ SQL Service Manager.

2. Tên server cục bộ xuất hiện trong mục Server và các dịch vụ của SQL Server xuất hiện trong mục Service. Trong danh sách xổ xuống, chọn tên của server và dịch vụ mà bạn muốn điều khiển.

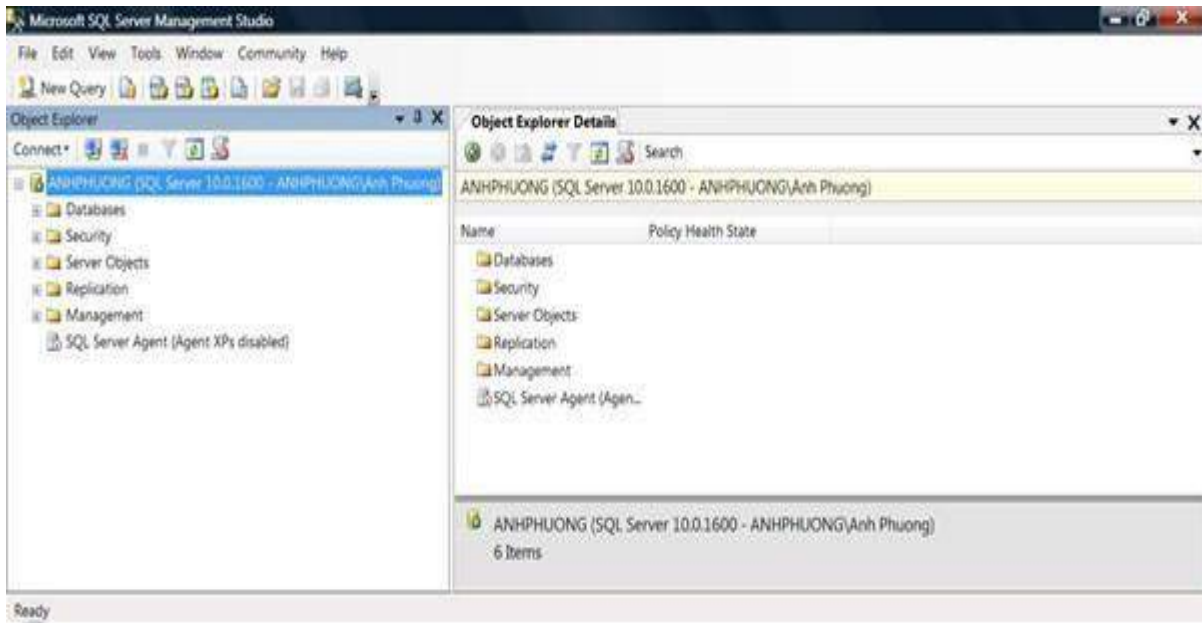
Chúng ta có thể khởi động (start) và dừng (stop) các service đang được chọn bằng cách click vào ứng tương ứng. Nếu Service ở trạng thái dừng thì click vào Start/Continue để chạy tiếp.

4. Auto-start service when OS Server: cấu hình tự động chạy dịch vụ khi hệ điều hành khởi động.

1.3 Các công cụ của MS SQL

1.3.1 Trình Enterprise Manager

Đây là môi trường trực quan sử dụng hệ thống thực đơn Enterprise Manager. Sử dụng hệ thống thực đơn (menu) để thực hiện các yêu cầu. Chủ yếu các yêu cầu tạo, thêm, xoá, sửa dữ liệu. Quản lý lịch trình Backup dữ liệu. Quản lý người dùng đang truy cập CSDL. Tạo, xóa quyền Login User. Định cấu hình cho Server. Tạo và quản lý tìm kiếm.

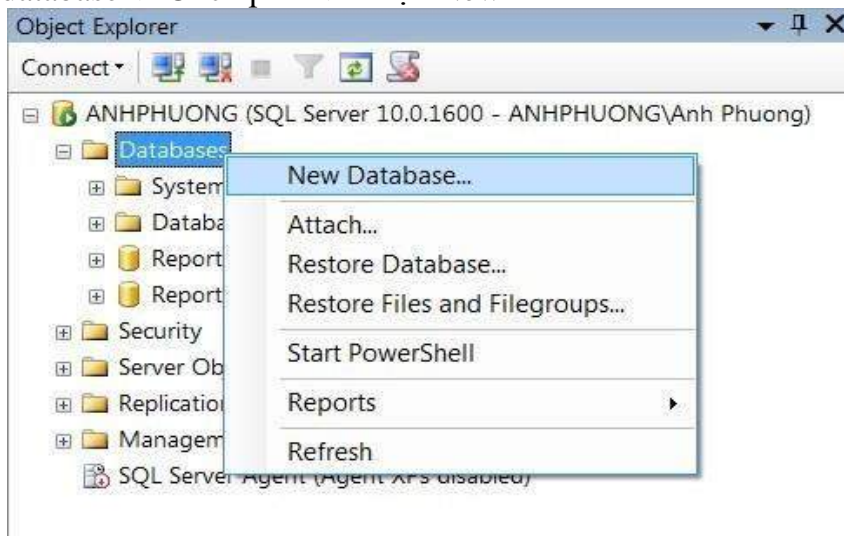


Hình 3.1: Giao diện SQL Server Enterprise Manager

1.3.2 Làm việc với công cụ Enterprise Manager

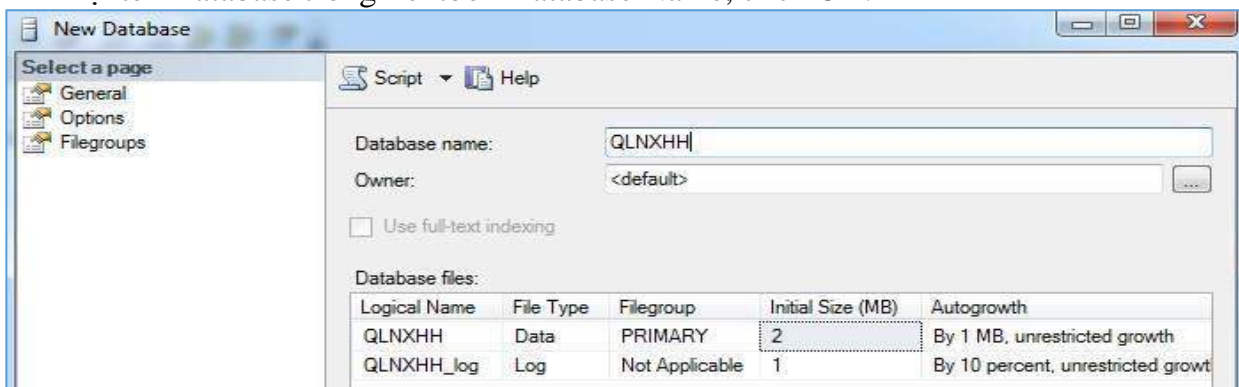
1.3.2.1 Tạo một CSDL mới

Chọn database -> Click phải -> Chọn New Database...



Hình 3.2: Giao diện SQL Server

- Đặt tên Database trong Textbox Database Name, click OK.

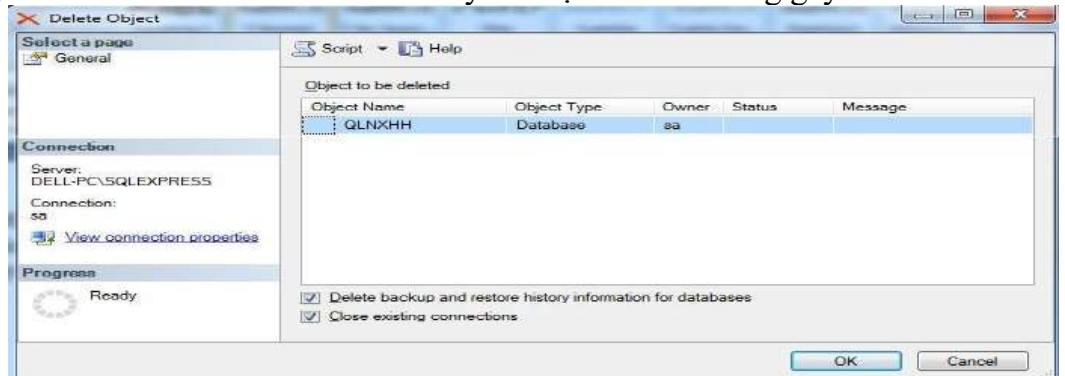


Hình 3.3: Hộp thoại New Database

- + Initial Size (MB): dung lượng khởi tạo tập tin.
- + Autogrowth: dung lượng tăng trưởng tập tin

1.3.2.2 Xóa một CSDL

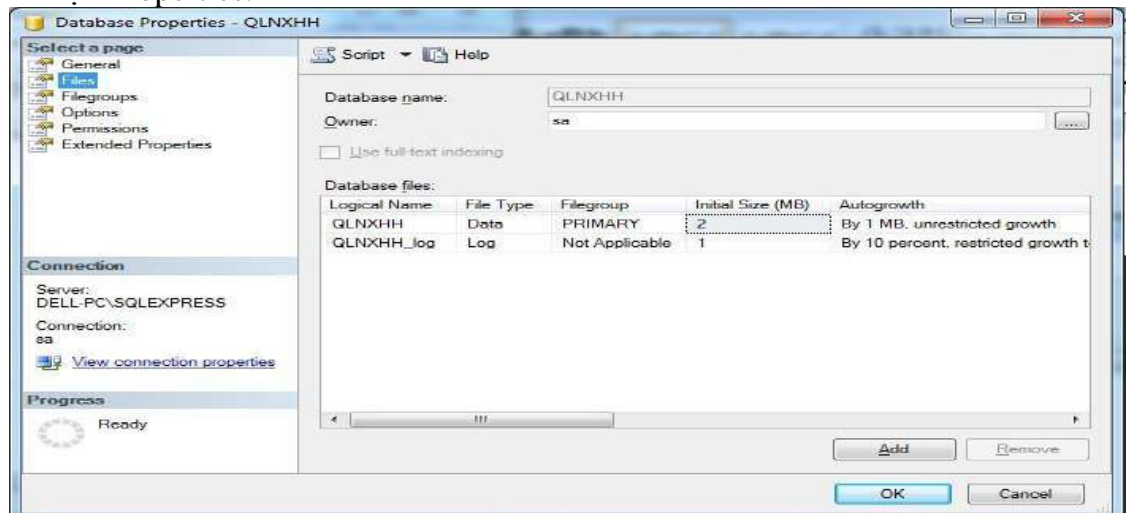
- Click phải chuột lên CSDL cần xóa -> Delete.
- Nên chọn dấu tích vào Close existing connections. Khi đó SQL Server 2014 sẽ ngắt tất cả các kết nối vào CSDL này và việc xóa sẽ không gây báo lỗi.



Hình 3.4: Hộp Thoại Delete Object

1.3.2.3 Sửa tham số.

- Chọn CSDL.
- Nhấn nút phải chuột
- Chọn Properties.



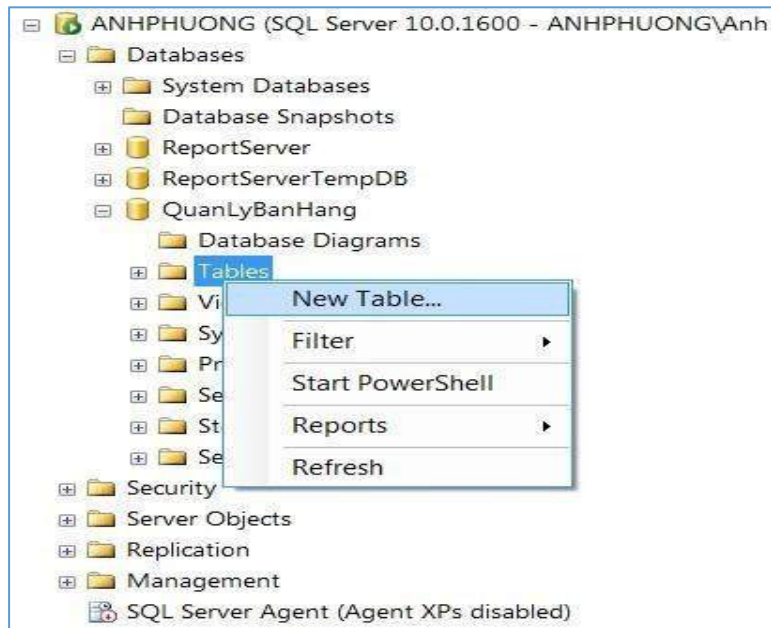
Hình 3.5:Hộp thoại Database Properties

1.4 Bảng dữ liệu

1.4.1 Thiết kế bảng dữ liệu:

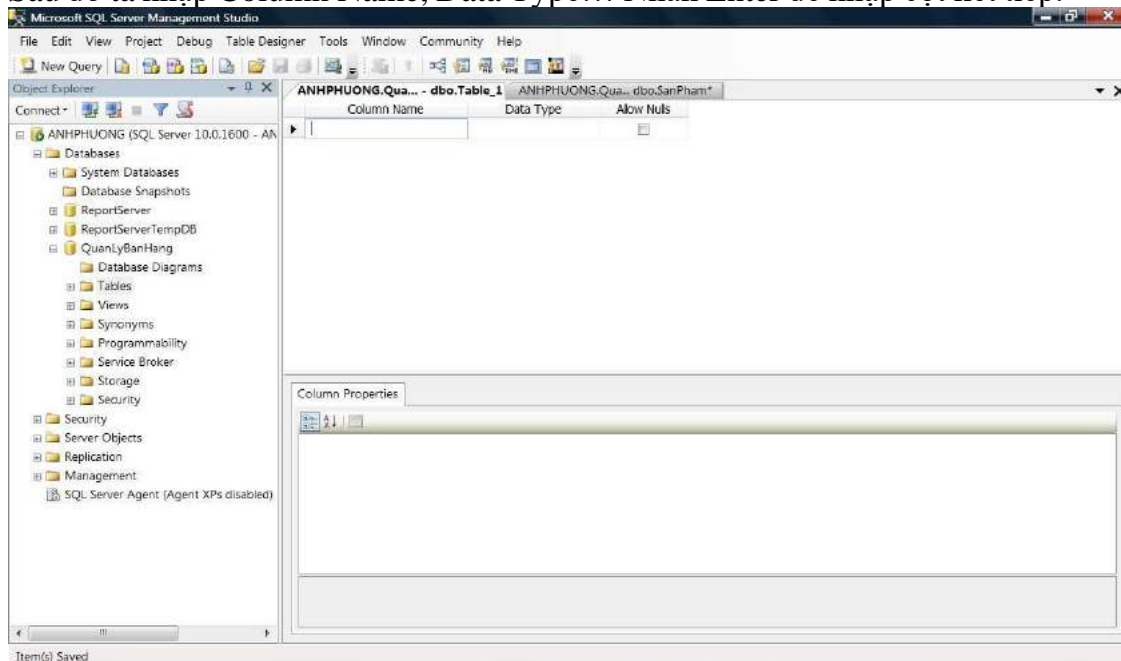
Table (bảng dữ liệu) là một thành phần cơ bản của CSDL, một CSDL được thiết kế từ một hoặc nhiều bảng dữ liệu, mỗi bảng dữ liệu được cấu trúc từ các hàng và cột dữ liệu, mỗi hàng dùng mô tả một đối tượng, vấn đề, sự kiện,... cột thể hiện thuộc tính của các đối tượng, sự kiện,... của hàng. Dữ liệu cùng cột có cùng kiểu (data type). Ngoài các hàng, cột bảng còn có các khóa, liên kết, ràng buộc,...

Chọn database -> Click phải lên table -> Chọn New Table ...



Hình 3.6. Hộp thoại Object Explorer

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 3.7. Hộp thoại thuộc tính Table

Lưu table trên thanh Standard toolbar -> chọn Save



Hình 3.7. Hộp thoại lưu table

Trước khi bắt tay vào thiết lập bảng dữ liệu trước hết ta phải xác định xem bảng sẽ xây dựng như thế nào, dựa trên một số thông tin sau:

- Kiểu dữ liệu trong bảng.

- Các cột, kiểu dữ liệu tương ứng (và độ dài nếu cần thiết).
- Cột nào cho phép giá trị NULL (là giá trị mà phần dữ liệu thuộc hàng, cột xác định không được gán giá trị nào, vì vậy nên 2 phần tử có cùng giá trị NULL là không bằng nhau).
- Giá trị ngầm định (là giá trị mà khi chưa nhập vào nó nhận giá trị này).
- Chỉ số Index, khóa chính, khóa ngoài.

a. Các kiểu dữ liệu

Char(n)	Kiểu chuỗi với độ dài cố định
Nchar(n)	Kiểu chuỗi với độ dài cố định hỗ trợ UNICODE
Varchar(n)	Kiểu chuỗi với độ dài chính xác
Nvarchar(n)	Kiểu chuỗi với độ dài chính xác hỗ trợ UNICODE
Int	Số nguyên có giá trị từ -231 đến 231 - 1
Tinyint	Số nguyên có giá trị từ 0 đến 255.
Smallint	Số nguyên có giá trị từ -215 đến 215 - 1
Bigint	Số nguyên có giá trị từ -263 đến 263-1
Numeric	Kiểu số với độ chính xác cố định.
Decimal	Tương tự kiểu Numeric
Float	Số thực có giá trị từ -1.79E+308 đến 1.79E+308
Real	Số thực có giá trị từ -3.40E + 38 đến 3.40E + 38
Money	Kiểu tiền tệ
Bit	Kiểu bit (có giá trị 0 hoặc 1)
Datetime	Kiểu ngày giờ (chính xác đến phần trăm của giây)
Smalldatetime	Kiểu ngày giờ (chính xác đến phút)
Binary	Dữ liệu nhị phân với độ dài cố định (tối đa 8000 bytes)
Varbinary	Dữ liệu nhị phân với độ dài chính xác (tối đa 8000 bytes)
Image	Dữ liệu nhị phân với độ dài chính xác (tối đa 2,147,483,647 bytes)
Text	Dữ liệu kiểu chuỗi với độ dài lớn (tối đa 2,147,483,647 ký tự)
Ntext	Dữ liệu kiểu chuỗi với độ dài lớn và hỗ trợ UNICODE (tối đa 1,073,741,823 ký tự)

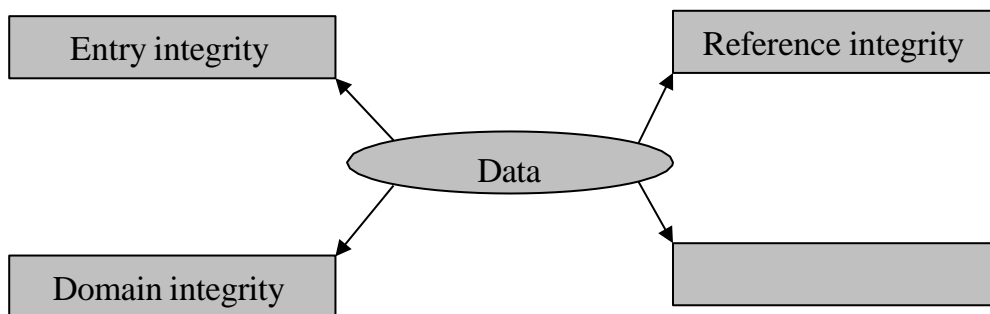
b/ Ràng buộc dữ liệu.

Để có một CSDL khi lưu trữ dữ liệu có độ tin cậy, độ chính xác cao, nhanh và thuận tiện trong khai thác dữ liệu thì toàn vẹn dữ liệu là vấn đề hết sức quan trọng. Khi ràng buộc được thiết lập, dữ liệu khi nhập vào CSDL sẽ được kiểm soát, độ tin cậy thông tin sẽ được bảo đảm.

Có nhiều kiểu ràng buộc dữ liệu, một CSDL có thể gồm một hoặc nhiều ràng buộc, ràng buộc có thể trên một bảng, trên nhiều bảng.

Toàn vẹn dữ liệu chia thành 4 loại:

- Toàn vẹn thực thể (Entry integrity): Mỗi thực thể đều được xác định theo một khóa, khi biết khóa ta hoàn toàn có thể xác định được thực thể tương ứng. Khóa như vậy coi là khóa chính.
- Toàn vẹn theo miền (Domain integrity): Là loại toàn vẹn có hiệu ứng với các cột dữ liệu trong một phạm vi nào đó, ví dụ kiểu dữ liệu cũng là một dạng của toàn vẹn miền, ràng buộc theo khóa check cũng là toàn vẹn theo miền.
- Toàn vẹn dạng tham chiếu (Referential integrity): Khi một bảng có quan hệ với một bảng khác theo một mối quan hệ, trong mối quan hệ đó sẽ có một khóa chính (như phần toàn vẹn thực thể) và một khóa ngoài, khóa ngoài sẽ là khóa tham chiếu của khóa chính, giá trị của khóa ngoài sẽ thuộc tập các giá trị của khóa chính hoặc giá trị NULL. Ràng buộc kiểu quan hệ (Relationship) gọi là toàn vẹn kiểu tham chiếu.
- Toàn vẹn do người dùng định nghĩa (User-defined integrity): Là toàn vẹn do người dùng định nghĩa, quy định dữ liệu nhập vào theo quy cách, giá trị được kiểm soát chặt chẽ, toàn vẹn kiểu này cũng có thể xây dựng trên cơ sở các toàn vẹn trước.



Hình 3.8: Mô hình Ràng buộc dữ liệu


Bốn loại toàn vẹn nói trên ta có thể thống kê tương ứng với các khóa, quy tắc, ràng buộc trong SQL Server như sau:

Kiểu toàn vẹn	Công cụ trong SQL Server
Entry integrity	<ol style="list-style-type: none"> 1. Ràng buộc Primary key 2. Ràng buộc Unique 3. Cột Identity
Domain integrity	<ol style="list-style-type: none"> 1. Giá trị ngầm định Default 2. Ràng buộc khóa ngoài Foreign Key 3. Ràng buộc Check 4. Thuộc tính NOT NULL
Referential integrity	<ol style="list-style-type: none"> 1. Ràng buộc Foreign Key 2. Ràng buộc Check
User-defined integrity	<ol style="list-style-type: none"> 1. Rules 2. Stored procedures 3. Triggers

1.4.2 Các khóa.


a. Khóa chính – Primary Key.

Là một hoặc tổ hợp nhiều cột dữ liệu xác định duy nhất trong một bảng, giá trị khóa chính luôn khác NULL.

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 .

Bạn cũng có thể click phải lên column chọn Set Primary Key.

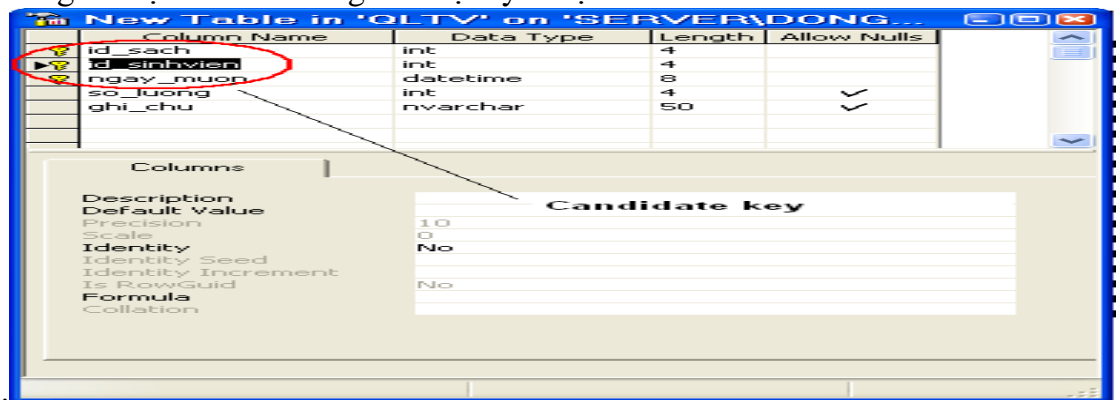
Ví dụ: **Bảng Nhân viên.**

	Column Name	Data Type	Allow Nulls
	MaNV	varchar(3)	<input type="checkbox"/>
	TenNV	nvarchar(30)	<input type="checkbox"/>
	UserName	varchar(20)	<input type="checkbox"/>
	Password	varchar(20)	<input type="checkbox"/>

Hình 3.9: Bảng nhân viên

Trong ví dụ trên bảng dữ liệu có khóa chính là một cột dữ liệu MaNV, khi cột xác định là khóa chính bên cạnh xuất hiện biểu tượng chìa khóa, thuộc tính Allow Nulls không được đánh dấu.

Ví dụ: Bảng dữ liệu lưu trữ thông tin nhật ký mượn sách



Hình 3.10: Bảng nhật ký mượn sách

Trong ví dụ trên bảng dữ liệu có khóa chính được tổ hợp từ 3 cột dữ liệu id_sach, id_sinhvien, ngay_muon, ba cột trên xác định duy nhất một sinh viên được mượn một loại sách trong một ngày (giả sử quy chế xác định như vậy), các cột tham gia khóa chính gọi là candidate key.

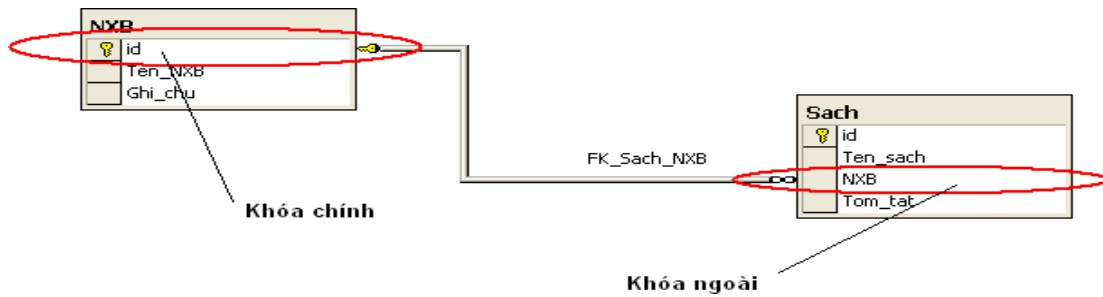
b. Khóa ngoài.

Theo chuẩn thiết kế CSDL, khi lưu trữ thông tin sách phải có một cột chứa thông tin nhà xuất bản. Một nhà xuất bản có thể xuất bản nhiều quyển sách và một quyển sách chỉ xuất bản ở một nhà xuất bản. Nên trong thiết kế ta phải có:

- + Bảng dữ liệu lưu trữ danh sách các nhà xuất bản: Có khóa chính đại diện cho nhà xuất bản.
- + Bảng dữ liệu lưu trữ sách: Có chứa thông tin nhà xuất bản.
- + Quan hệ giữa nhà xuất bản và sách: Mã khóa nhà xuất bản thuộc bảng nhà xuất bản và thông tin nhà xuất bản thuộc bảng sách, cột thông tin nhà xuất bản thuộc bảng sách tham gia quan hệ trên gọi là khóa ngoài (Foreign key).

Cột dữ liệu là khóa ngoài có thể có quan hệ với nhiều khóa chính ở nhiều bảng, một bảng có thể có nhiều khóa ngoài, khóa ngoài có thể có giá trị NULL, giá trị của khóa ngoài luôn nằm trong tập giá trị của khóa chính trong mối quan hệ đã thiết lập.

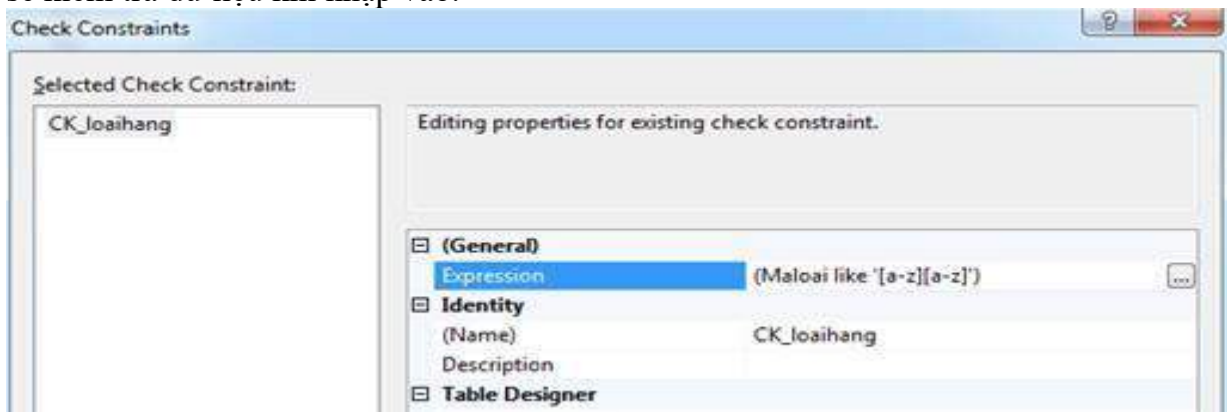
Khóa ngoài và khóa chính phải có cùng kiểu dữ liệu, cùng kích thước.



Hình 3.11: Mô phỏng khóa chính – khóa ngoại

Ràng buộc Check.

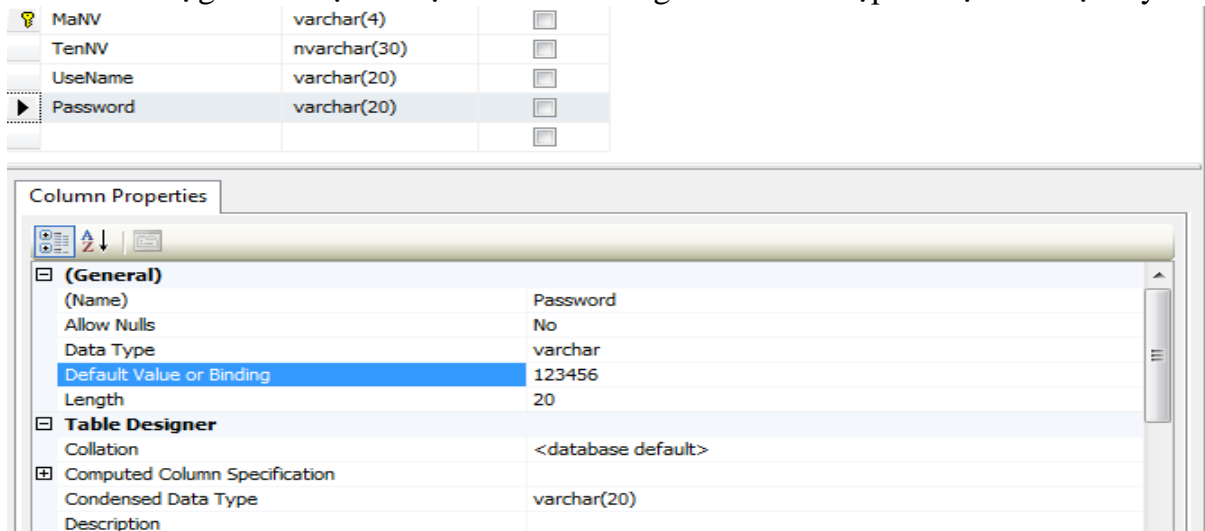
Là ràng buộc không chế dữ liệu nằm trong một phạm vi nào đó. Ràng buộc này sẽ kiểm tra dữ liệu khi nhập vào.



Hình 3.12: Hộp thoại Check Constrains

Giá trị ngầm định – Default.

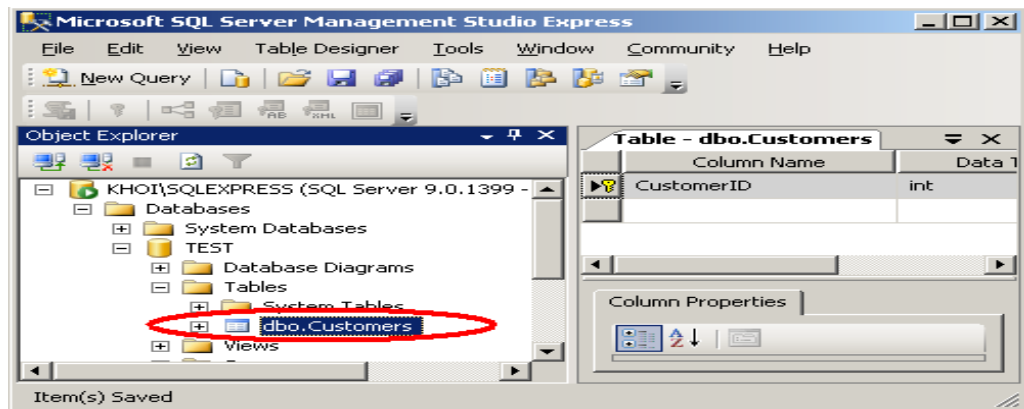
Giá trị gán cho cột dữ liệu khi thêm bản ghi và chứa nhập dữ liệu vào cột này.



Hình 3.13 Column Properties

1.4.3 Xóa bảng

Click chuột phải lên bảng muốn xóa -> Delete -> OK.



Hình 3.14 Hộp thoại xóa bảng

1.4.4 Nhập dữ liệu vào bảng.

- Chọn bảng dữ liệu
- Nhấn chuột phải -> Open Table

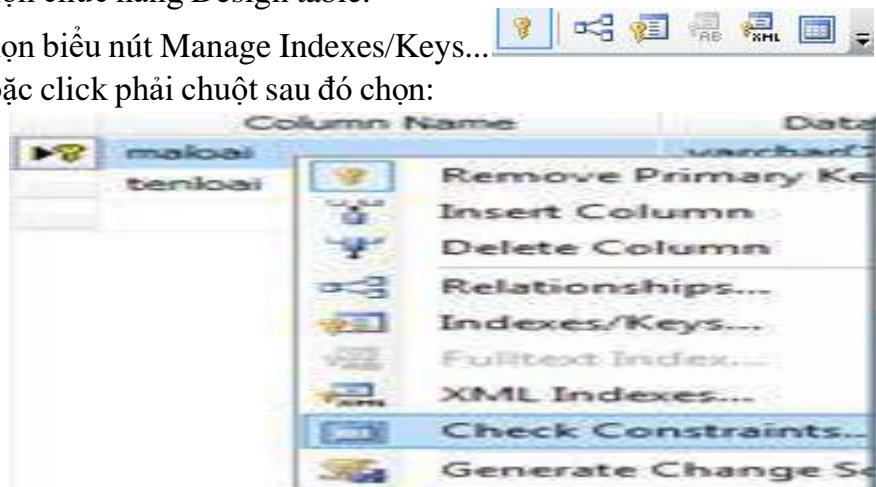
	maloi	tenloai
	BK	Bánh Kẹo
▶*	NULL	NULL

Chú ý: Việc sửa, xóa được thực hiện trực tiếp. Đối với các cột là dạng số, tăng tự động không cần nhập dữ liệu. Để lưu lại dữ liệu đã nhập bạn di chuyển con trỏ sang hàng khác.

1.4.5 Tạo, sửa ràng buộc, khóa.

Phần này nhằm thực hiện thao tác với các ràng buộc, khóa: relationship, check, unique,...

- Chọn chức năng Design table.
- Chọn biểu nút Manage Indexes/Keys...
- Hoặc click phải chuột sau đó chọn:



Hình 3.15 Hộp thoại Indexes/Keys

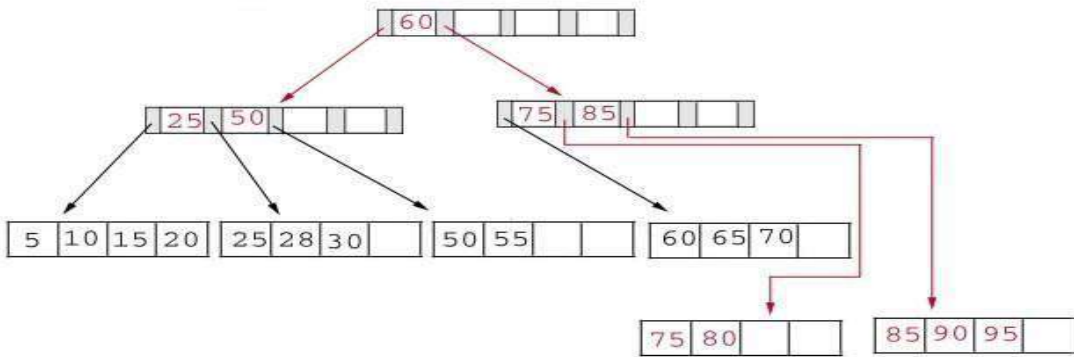
* Từ khóa Like.

Từ khóa Like được sử dụng tương tự như phép so sánh, phép Like được thực hiện cho dữ liệu kiểu chuỗi, phép Like được xem là phép so sánh theo định dạng của chuỗi, việc định dạng xác định theo một số từ khóa sau:

- % Xác định bất kỳ chuỗi ký tự nào hoặc không có ký tự nào tại vị trí.
- _ Một ký tự bất kỳ nào đó.
- [] Một ký tự nào đó nằm trong phạm vi, ví dụ [a-f].
- [^] Xác định một ký tự không thuộc phạm vi nào đó, ví dụ [^a-f].

1.5. Khóa Index.

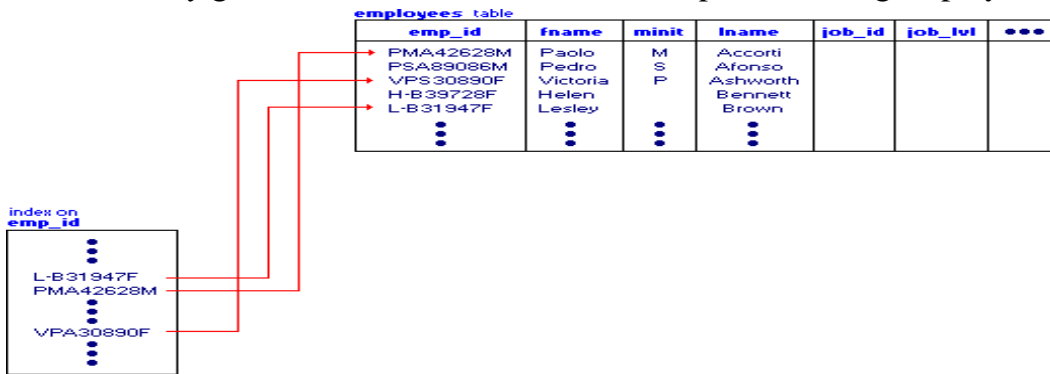
Index là một khóa quan trọng đối với CSDL đặc biệt là cơ sở dữ liệu lớn. Index được thiết lập từ một hoặc nhiều cột dữ liệu của bảng dữ liệu. các giá trị của khóa Index sẽ được sắp xếp và lưu trữ theo một danh sách (bảng khác). Mỗi giá trị trong khóa Index là duy nhất trong danh sách, mỗi giá trị khóa Index sẽ liên kết đến giá trị trong bảng dữ liệu (liên kết dạng con trỏ). Việc lưu trữ dữ liệu của bảng có khóa Index được thực hiện theo cấu trúc cây B-Tree nhằm tăng tốc độ truy xuất dữ liệu đối với ổ đĩa (thiết bị thứ cấp).



Hình 3.16 Mô hình liên kết dạng con trỏ

Khi tìm kiếm một giá trị trong cột dữ liệu, mà cột này tham gia tạo khóa Index, đầu tiên câu lệnh xác định vị trí của giá trị nằm trong khóa Index bằng phép duyệt cây, sau đó thực hiện tìm theo liên kết đến bản ghi chứa giá trị tương ứng với khóa trong bảng.

Sơ đồ ví dụ dưới đây gồm khóa Index được tạo từ cột emp_id của bảng employees.



Hình 3.17: Sơ đồ Khóa Index

Việc thiết kế khóa Index dựa trên nhu cầu truy vấn, chèn dữ liệu trên một bảng, xác định dựa vào một số tham số sau:

- + Cột thường được sử dụng làm khóa truy vấn dữ liệu (xác định cột tham gia khóa Index).
- + Tập lệnh thường sử dụng truy vấn cần tốc độ cao (xác định tập cột tham gia truy vấn).
- + Dữ liệu nhập vào bảng có khóa Index cần nhanh hơn hay truy vấn cần nhanh hơn (xác định đặt clustered hoặc nonclustered).
- + Lượng dữ liệu nhập đồng loạt nhiều hay ít (xác định tham số fillfactor).

Clustered Index.

Khi khóa đặt thuộc tính Clustered, dữ liệu của bảng sẽ được sắp xếp vật lý trên đĩa, như vậy khi thiết kế khóa dạng này dữ liệu được chèn và sẽ tìm đúng vị trí trên đĩa để lưu trữ (vùng đĩa dành cho bảng dữ liệu), chính vì vậy mà có thể xảy ra trường hợp phải

dịch chuyển danh sách các giá trị đã có ở đĩa. Những việc tạo khóa Index dạng này sẽ không cần sắp xếp giá trị ở dạng logic mà khi truy nhập đĩa đã bảo đảm dữ liệu được sắp xếp.

Bảng dữ liệu chỉ có thể tạo tối đa một khóa Lustered Index.

Nonclustered Index.

Dữ liệu Index không sắp xếp ở dạng vật lý mà chỉ sắp xếp logic, dữ liệu của bảng lưu trữ giá trị khóa Index được sắp xếp, nhanh trong nhập dữ liệu.

Unique Index.

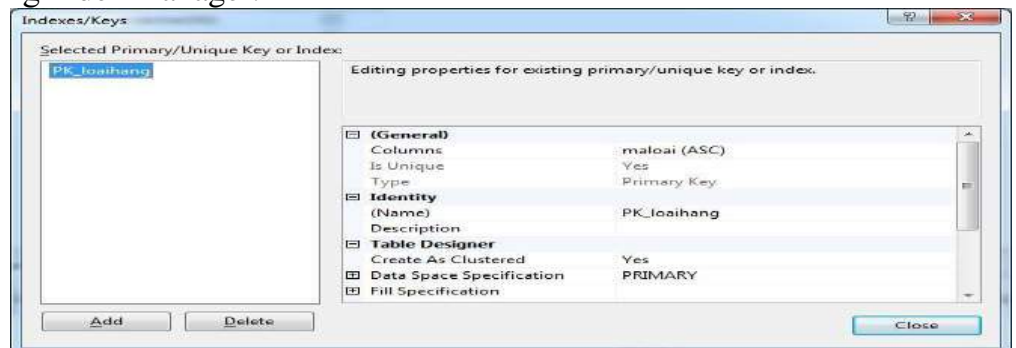
Xác định dữ liệu của cột tham gia khóa Index không lặp lại.

Fill Factor.

Khi tạo khóa Index, dữ liệu tham gia tạo khóa Index sẽ được phân theo mức của B-Tree, các mức được phân theo page dữ liệu, giá trị Fill factor xác định phần khoảng trống tối đa của page theo tỷ lệ phần trăm. Nhờ khoảng trống này mà tốc độ bố trí cấu trúc Index, tốc độ truy lục thông tin trong cây được cải thiện.

a/ Tạo khóa Index

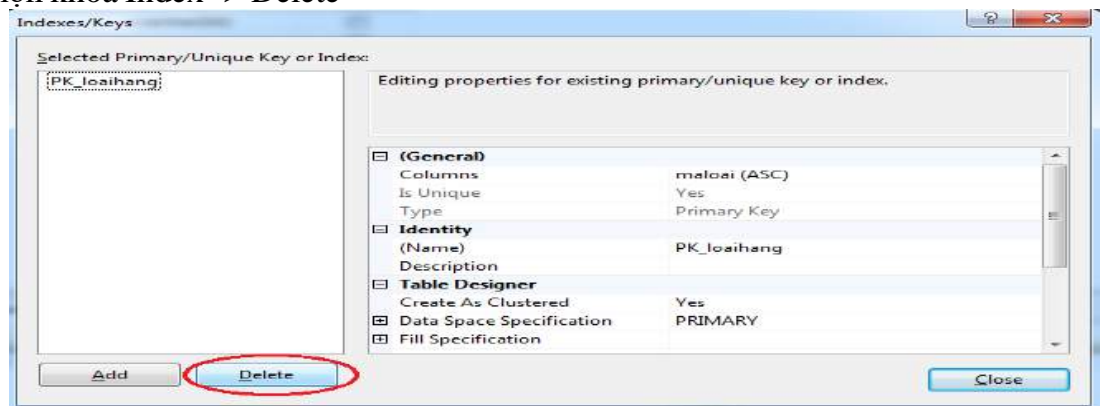
- Chọn chức năng Design table
- Vào bảng Index manager.



Hình 3.18 Hộp thoại ràng buộc

b/ Xóa khóa Index

- Vào Index manager
- Chọn khóa Index -> Delete

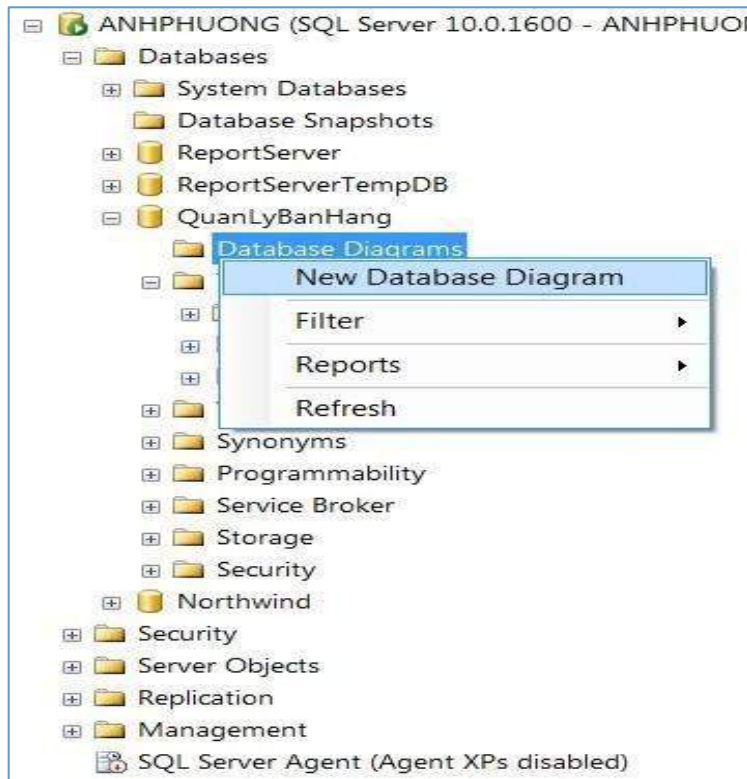


Hình 3.19 Hộp thoại xóa ràng buộc

1.6 Tạo sơ đồ (Diagrams)

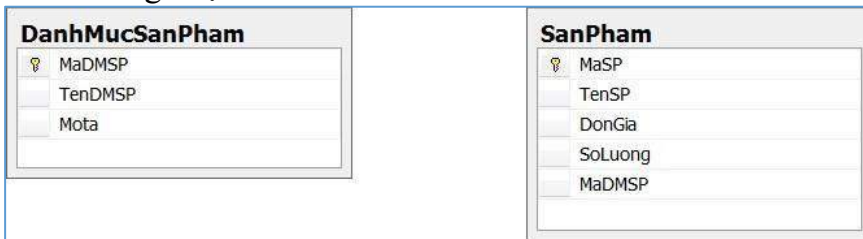
Diagrams là 1 cửa sổ hiển thị mối quan hệ giữa các table của 1 database. Tạo diagram ta thực hiện như sau:

- Trong cửa sổ Object Explorer chọn tên database cần tạo -> Click phải vào Database Diagrams -> Chọn New Database Diagram



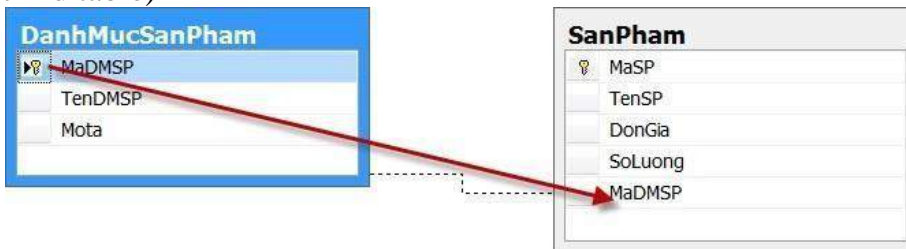
Hình 3.20 Hộp thoại Database Diagram

Sau khi chọn New Database Diagram sẽ xuất hiện hộp thoại để Add các table, sau khi add xong chọn Close.



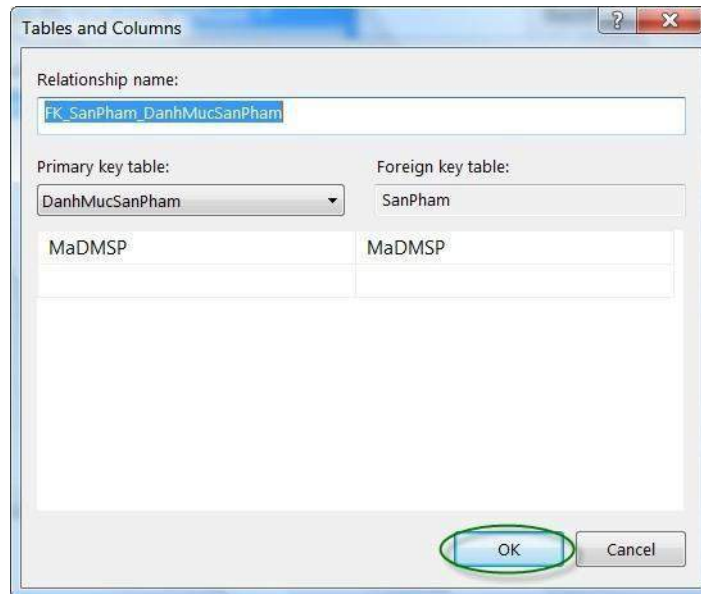
Hình 3.21 Hộp thoại chọn bảng

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)



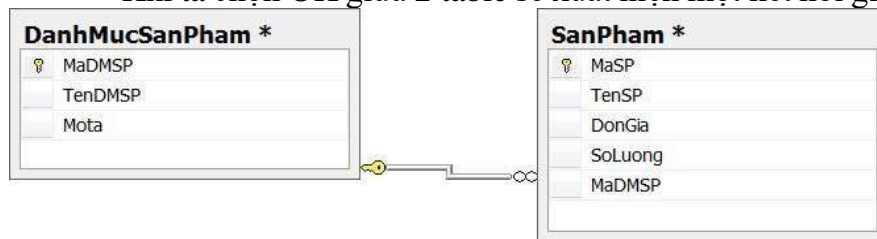
Hình 3.22. Hộp thoại tạo liên kết

Sau khi kéo mối quan hệ cho 2 table sẽ xuất hiện hộp thoại như hình 1.21.



Hình 3.23. Hộp thoại thiết lập liên kết

Khi ta chọn OK giữa 2 table sẽ xuất hiện một kết nối giữa 2 table



Hình 3.24. Thiết lập liên kết thành công

Trong cùng một cách, bạn có thể tạo mối quan hệ khác. Khi bạn đã hoàn tất, bạn có thể lưu và đóng diagram.

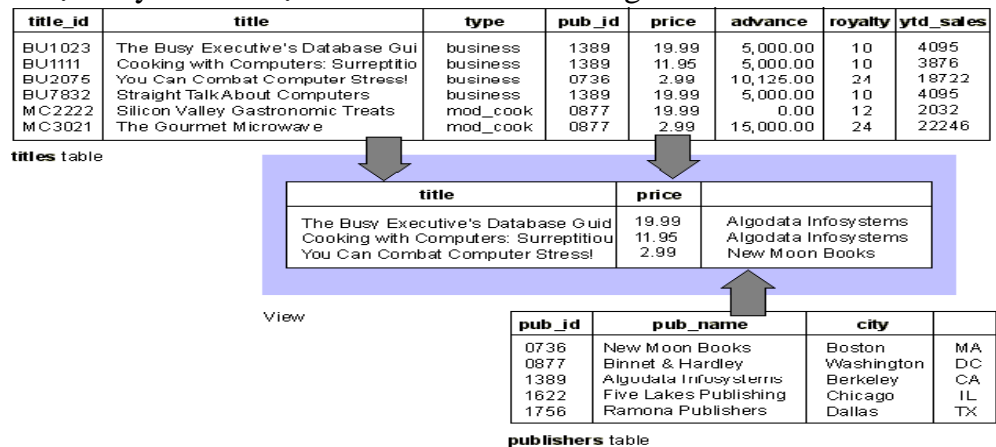
1.7 Khung nhìn – View

1.7.1 Khái niệm khung nhìn.

Khung nhìn (View) là một bảng tạm thời, có cấu trúc như một bảng, khung nhìn không lưu trữ dữ liệu mà 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. Khung nhìn được sử dụng khai thác dữ liệu như một bảng dữ liệu, chia sẻ nhiều người dùng, an toàn trong khai thác, không ảnh hưởng dữ liệu gốc.

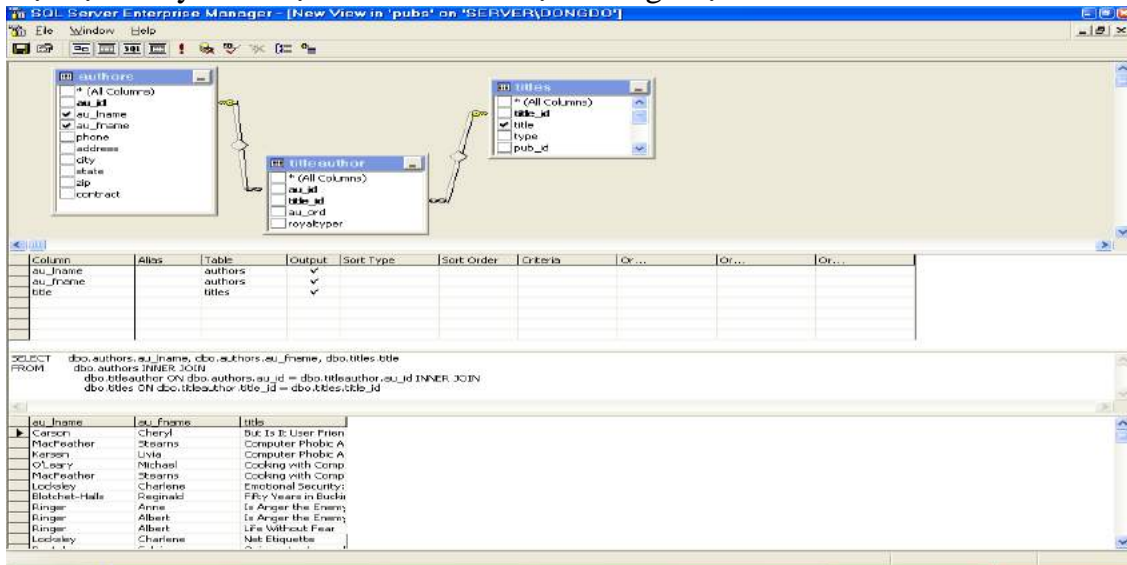
Có thể thực hiện truy vấn dữ liệu trên cấu trúc của khung nhìn.



Hình 3.25 Mô hình truy vấn bảng

1.7.2 Tạo khung nhìn.

- Chọn chức năng Views của CSDL.
- Nhấn phải chuột → New View.
- Chọn các bảng tham gia câu lệnh truy vấn dữ liệu cho View
- Soạn lệnh truy vấn hoặc đánh dấu các cột tham gia tạo View.



Hình 3.26 Cửa sổ truy vấn bảng

- Sửa đổi lệnh Select theo ý muốn.
- Ghi kịch bản -> đặt tên view.

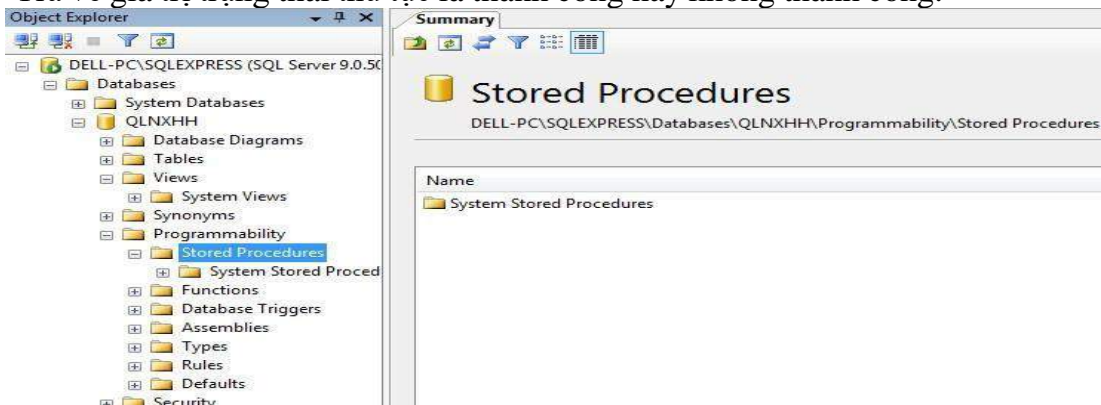
1.8 Thủ tục lưu trữ

1.8.1 Khái niệm thủ tục lưu trữ và hàm.

Thủ tục lưu trữ có thuật ngữ Stored Procedure, là một đối tượng của CSDL tương tự như khung nhìn, thủ tục lưu trữ có thể tạo ra từ công cụ và câu lệnh. Thủ tục được thực hiện như câu lệnh (có thể thực hiện từ SQL Query analyzer, các vị trí gọi câu lệnh T-SQL).

Thủ tục lưu trữ được kết cấu từ một kịch bản câu lệnh T-SQL, thủ tục có những đặc điểm cơ bản sau:

- + Truyền tham số.
- + Gọi thủ tục khác.
- + Trả về các giá trị tham số, chuyển giá trị tham số cho các thủ tục được gọi.
- + Trả về giá trị trạng thái thủ tục là thành công hay không thành công.



Hình 3.27 Cửa sổ tạo xem thủ tục

Thủ tục lưu trữ có nhiều ưu điểm so với thực hiện câu lệnh T-SQL từ các máy khách:

- + Lập trình theo module: Thủ tục được thiết lập trong từng CSDL một lần, có thể gọi thực hiện nhiều lần trong một ứng dụng, có thể gọi từ nhiều ứng dụng.
- + Thực hiện nhanh hơn: Khi cần thực hiện một lượng lớn câu lệnh T-SQL, thủ tục lưu trữ thực hiện nhanh hơn vì khi máy chủ nhận được nhiều câu lệnh cùng một lúc đều phải kiểm tra tính hợp lệ quyền của tài khoản từ máy khách và các tham số khác. Khi thủ tục cần gọi nhiều lần trên các máy khách thì thủ tục thực hiện một lần đầu tiên, những lần sau máy khách sẽ chạy thủ tục đã được biên dịch.
- + Làm giảm lưu lượng trên mạng: Thay cho vì máy khách phải gửi nhiều dòng lệnh từ các ứng dụng đến máy chủ, khi sử dụng thủ tục thì nó chỉ cần gửi một lệnh, từ đó dẫn đến lưu lượng thông tin lệnh truyền qua mạng giảm.
 - + An ninh bảo mật hơn: Khi không muốn cho một user trực tiếp khai thác một đối tượng hay bảng dữ liệu nào đó, mà cần cho user đó được khai thác thì thủ tục có thể giúp bạn gán quyền khai thác cho người đó. Việc gán quyền khai thác như nói trên sẽ giúp cho vấn đề an ninh bảo mật trong CSDL tốt hơn.

1.8.2 Phân loại thủ tục lưu trữ.

Thủ tục lưu trữ được phân thành 5 loại như sau:

System Stored Procedure.

Là thủ tục được lưu trữ tổng CSDL Master, thủ tục loại này được bắt đầu bằng chữ **sp_** thủ tục loại này thường được sử dụng trong quản trị CSDL và an ninh bảo mật.

Ví dụ: Muốn biết tất cả các tiến trình đang thực hiện bởi user nào:

```
sp_who @loginame='sa'
```

Kết quả:

spid	ecid	status	loginame	hostname	blk	dbname	cmd
1	0	background	sa		0	NULL	LAZY WRITER
2	0	sleeping	sa		0	NULL	LOG WRITER
3	0	background	sa		0	master	SIGNAL HANDLER
4	0	background	sa		0	NULL	LOCK MONITOR
5	0	background	sa		0	master	TASK MANAGER
6	0	background	sa		0	master	TASK MANAGER
7	0	sleeping	sa		0	NULL	CHECKPOINT SLEEP
8	0	background	sa		0	master	TASK MANAGER
9	0	background	sa		0	master	TASK MANAGER

Hình 3.28 Cửa sổ thực thi thủ tục

Local Stored Procedure.

Đây là loại thủ tục thường dùng nhất, nằm trong CSDL do người dùng tạo ra, thực hiện một công việc nào đó. Thủ tục loại này thường được tạo bởi DBA (Database Administrator) hoặc người lập trình.

Temporary Stored Procedure.

Có chức năng tương tự như Local Stored Procedure nhưng thủ tục loại này tự hủy khi kết nối tạo ra nó ngắt hoặc SQL Server ngưng hoạt động và nó được tạo ra trên CSDL TempDB.

Extended Stored Procedure.

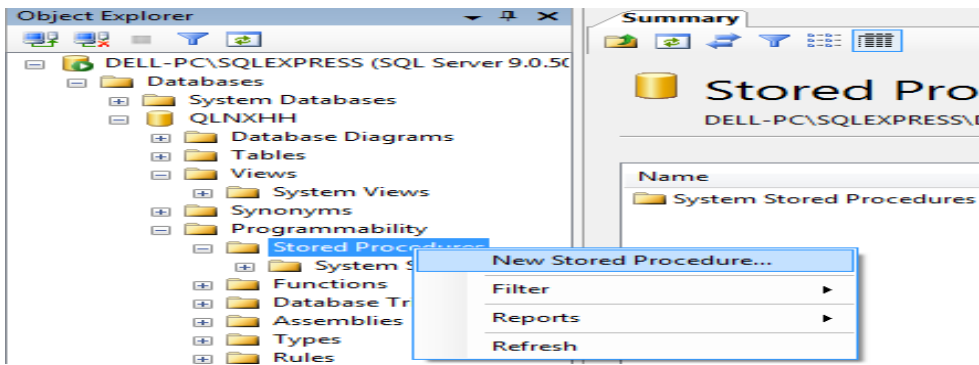
Đây là loại thủ tục sử dụng chương trình ngoại vi đã được biên dịch thành DLL. Tên thủ tục được bắt đầu bằng **xp_**. Ví dụ thủ tục xp_sendmail dùng gửi mail, thủ tục xp_cmdshell dùng thực hiện lệnh của DOS (xp_cmdshell 'dir c:\').

Remote Stored Procedure:

Là loại thủ tục sử dụng thủ tục của một server khác.

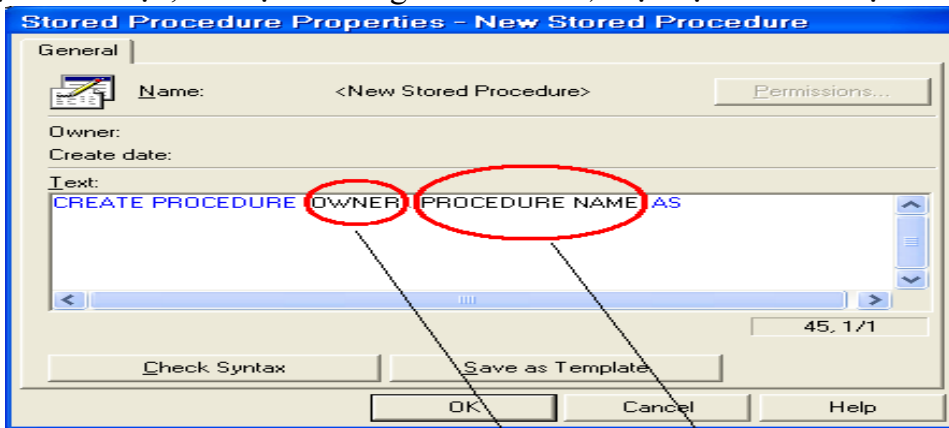
1.8.3 Thiết lập thủ tục lưu trữ.

- Chọn CSDL cần tạo thủ tục trong Enterprise Manager -> Stored Procedures
- Nhấn nút phải chuột -> New Stored Procedure...



Hình 3.29 Cửa sổ tạo thủ tục

- Đặt tên thủ tục, xác định role người khai thác, soạn kịch bản câu lệnh.



Tên thủ tục

Role xác định quyền người dùng

Hình 3.30 Cửa sổ role khai thác thủ tục

1.8.4 Sửa, xóa thủ tục

- Chọn thủ tục cần sửa, xóa -> thực hiện sửa nội dung hoặc chức năng xóa.

1.9 TRIGGER

1.9.1 Khái niệm trigger.

Trigger là một thủ tục đặc biệt mà việc thực thi của nó tự động khi có sự kiện xảy ra, các sự kiện gọi thủ tục đặc biệt này được định nghĩa trong câu lệnh, thông thường được thực hiện với các sự kiện liên quan đến Insert, Update, Delete dữ liệu.

Trigger được sử dụng trong việc bảo đảm toàn vẹn dữ liệu theo quy tắc xác định, được quản lý theo bảng dữ liệu hoặc khung nhìn.

1.9.2 Những trường hợp sử dụng Trigger.

- Sử dụng Trigger khi các biện pháp toàn vẹn dữ liệu như Constraint, rule,... không bảo đảm. Khác với các công cụ bảo đảm toàn vẹn dữ liệu đã nêu, các công cụ này sẽ thực hiện kiểm tra tính toán vẹn trước khi đưa dữ liệu vào CSDL (còn gọi là Declarative Data Integrity), còn Trigger thực hiện kiểm tra tính toán vẹn khi công việc đã thực hiện rồi (còn gọi là Procedural Data Integrity).

- Khi CSDL chưa được chuẩn hóa (Normalization) thì có thể xảy ra dữ liệu thừa, chứa ở nhiều vị trí trong CSDL thì yêu cầu đặt ra là dữ liệu cần cập nhật thống nhất trong mọi nơi. Trong trường hợp này ta phải sử dụng Trigger.

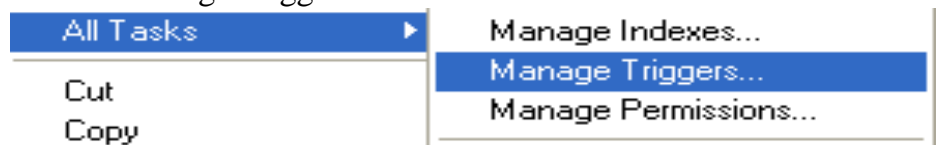
- Khi thay đổi day chuyên dữ liệu giữa các bảng với nhau (khi dữ liệu bảng này thay đổi thì dữ liệu trong bảng khác cũng được thay đổi theo).

1.9.3 Đặc điểm của Trigger.

- Một trigger có thể thực hiện nhiều công việc (theo kịch bản), có thể nhiều sự kiện kích hoạt thực thi trigger, có thể tách rời các sự kiện trong một trigger.
- Trigger không được tạo trên bảng template hay system.
- Trigger chỉ thực thi tự động thông qua các sự kiện mà không thực hiện bằng tay.
- Trigger sử dụng được với khung nhìn.
- Khi trigger thực thi theo các sự kiện Insert hoặc Delete thì dữ liệu khi thay đổi sẽ được chuyển sang các bảng Inserted Table, Deleted Table, là 2 bảng tạm thời chỉ chứa trong bộ nhớ, các bảng này chỉ được sử dụng với các lệnh trong trigger. Các bảng này thường được sử dụng để khôi phục lại phần dữ liệu đã thay đổi (roll back).
- Trigger chia thành 2 loại Instead of và After: Instead of là loại trigger mà hoạt động của sự kiện gọi nó sẽ bỏ qua và thay vào nó là các lệnh thực hiện trong trigger. After (tương đương với từ khóa For) đây là loại ngầm định, khác với loại Instead of thì loại trigger này sẽ thực hiện các lệnh trong nó sau khi đã thực hiện xong sự kiện gọi nó.

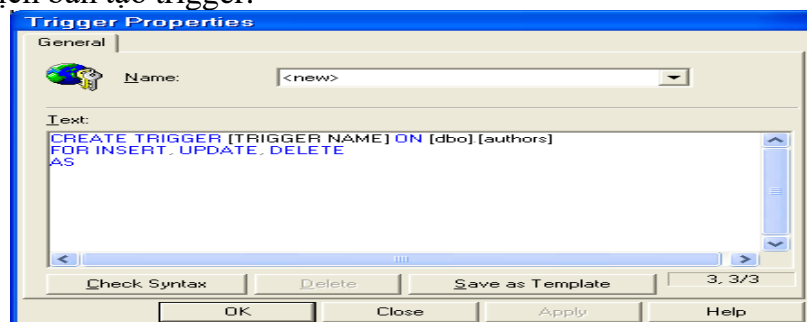
1.9.4 Tạo Trigger.

- Chọn bảng dữ liệu hoặc khung nhìn.
- Nhấn nút phải chuột.
- Chọn All tasks -> Manage Triggers...



Hình 3.31 Cửa sổ tạo Trigger

- Soạn kịch bản tạo trigger.



Hình 3.32 Cửa sổ code trigger

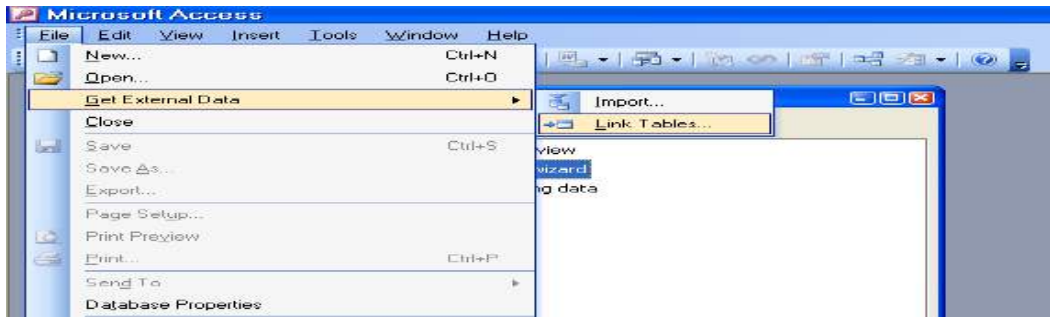
1.10. Tạo liên kết cơ sở dữ liệu dữ liệu

1.10.1 Tạo liên kết từ Access.

Từ hệ quản trị CSDL Access bạn có thể tạo liên kết đến các hệ quản trị CSDL khác (Access, Dbase,...), hoặc thông qua ODBC. Trong ví dụ minh họa sử dụng liên kết từ Access với ODBC (đối với SQL Server hoặc Oracle, My SQL thì Access phải liên kết thông qua ODBC vì các hệ quản trị CSDL này không thực hiện khai thác dữ liệu qua tập tin chỉ khai thác thông qua tên CSDL, mà Access chỉ thực hiện theo phương thức mở tập tin).

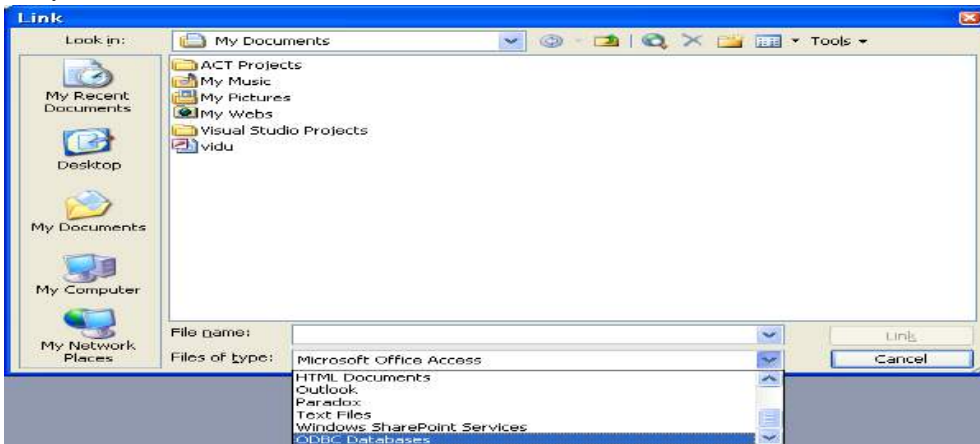
Các bước thực hiện như sau:

- Mở hệ quản trị CSDL Access.
- Mở hoặc tạo CSDL mới từ Access
- Chọn File -> Get External Data -> Link Tables.



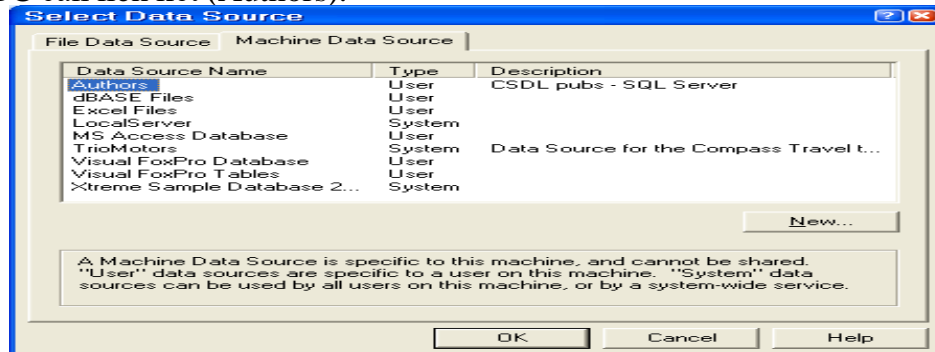
Hình 3.33 Cửa sổ tạo liên kết

- Chọn ODBC Databases.



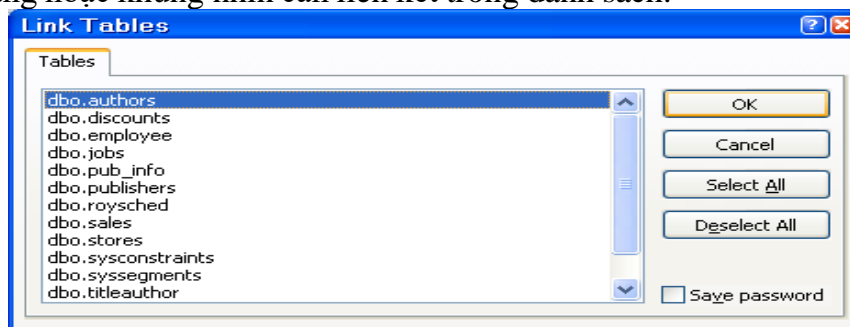
Hình 3.34 Cửa sổ chọn liên kết

- Chọn ODBC cần liên kết (Authors).



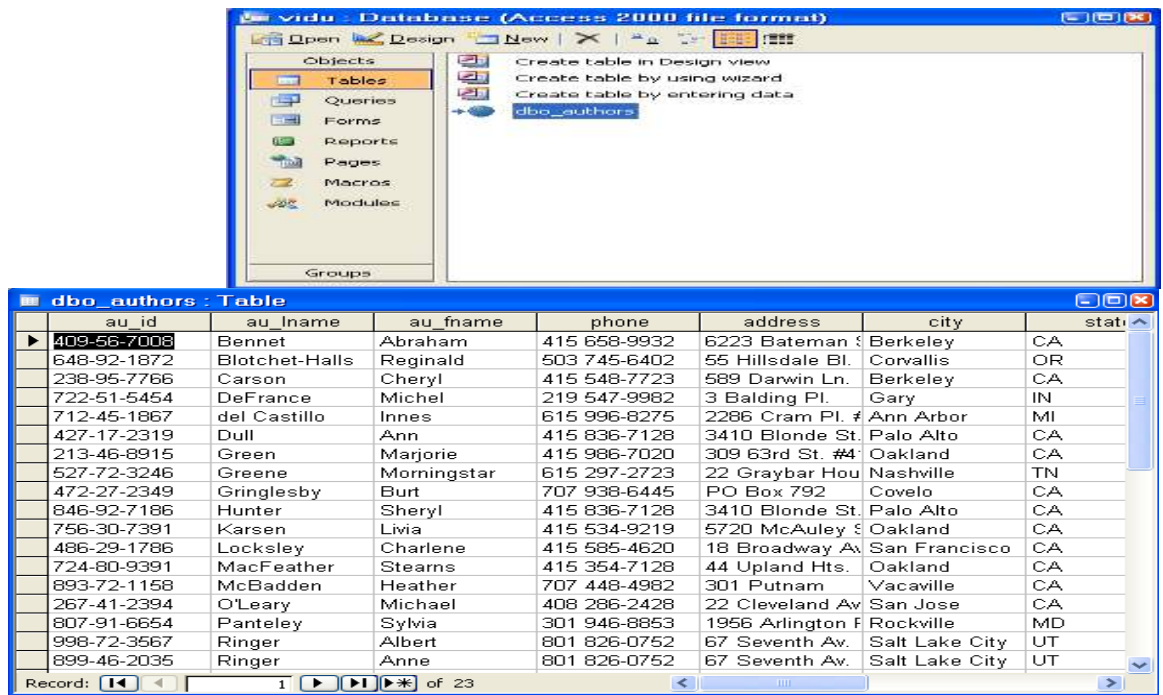
Hình 3.35 Cửa sổ Chọn ODBC liên kết

- Ok.
- Nhập Login ID và mật khẩu.
- Chọn bảng hoặc khung nhìn cần liên kết trong danh sách.



Hình 3.36 Cửa sổ chứng thực liên kết

- Nhấn Ok, danh sách các bảng trong Access được khai thác tương tự như các bảng khác.

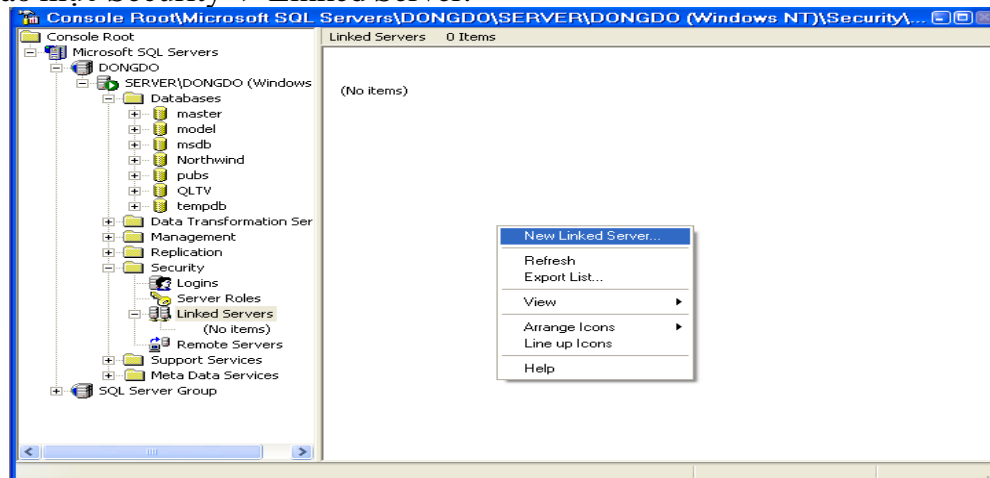


Hình 3.37 Cửa sổ dữ liệu liên kết

1.10.2 Tạo Server liên kết – Linked Server.

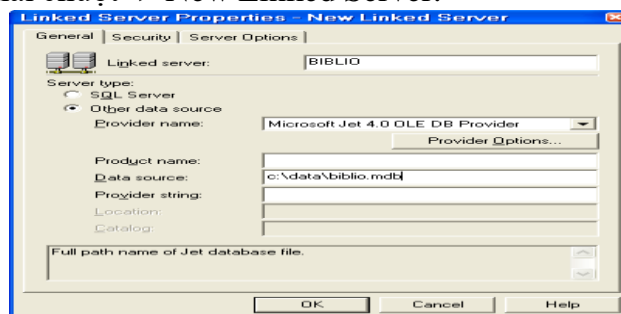
Từ SQL Server có thể tạo liên kết trực tiếp đến các hệ quản trị CSDL khác (Access, SQL Server, Oracle, My SQL,...) mà không cần thiết phải thông qua ODBC như Access đã xét trước.

- Vào mục Security -> Linked Server.



Hình 3.38 Cửa sổ Linked Server

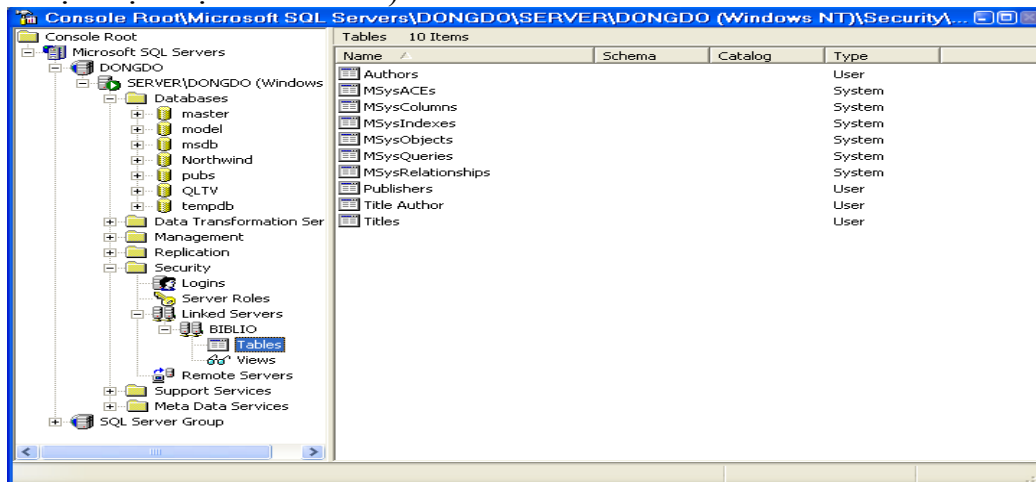
- Nhấn nút phải chuột -> New Linked Server.



Hình 3.39 Cửa sổ tạo mới

- Nhập các tham số:

- + Tên Server.
- + Provider (Driver của hệ quản trị CSDL cần thiết lập liên kết, trong ví dụ minh họa thực hiện với Access).



Hình 3.39 Cửa sổ tạo Provider

1.11 / Dettach và Attach cơ sở dữ liệu.

Mục này sẽ nói về kỹ thuật hủy và nối ghép tập tin CSDL với Server. Giả sử bạn đã có các tập tin của CSDL (gồm tập tin dữ liệu và nhật ký có thể được copy từ vị trí khác).

1.11.1 Copy tập tin của CSDL.

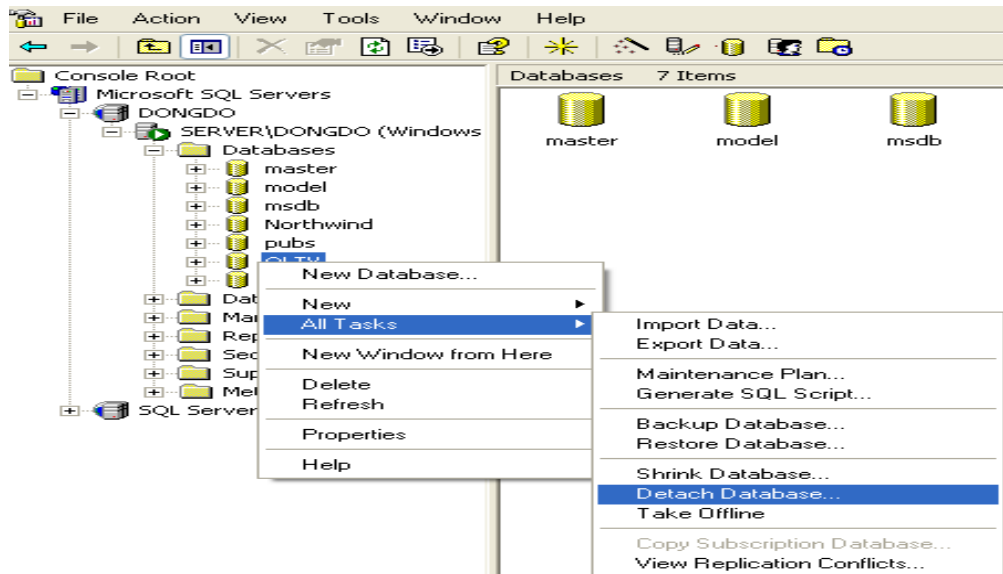
Trước tiên ta xem kỹ thuật copy các tập tin CSDL sang một vị trí khác (mà vẫn giữ vị trí), sau khi copy sang vị trí khác bạn có thể sử dụng sang Instance mới. Các bước thực hiện như sau:

- Stop dịch vụ SQL của Instance có CSDL.
- Copy các tập tin của CSDL sang vị trí cần thiết.
- Start dịch vụ SQL của Instance để tiếp tục làm việc.

1.11.2 Dettach cơ sở dữ liệu.

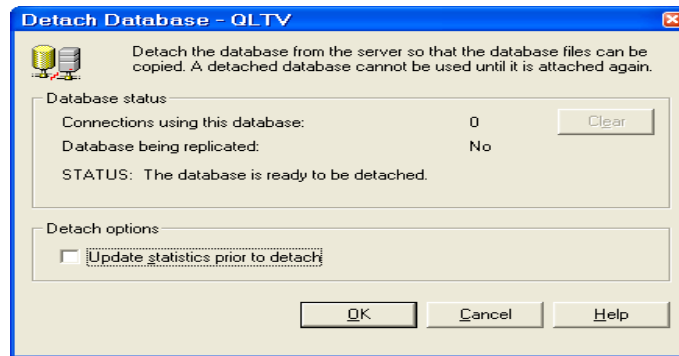
Là bước thực hiện tách CSDL khỏi Instance, Instance không quản lý CSDL nhưng khác với xóa CSDL là các tập tin chứa CSDL vẫn còn.

- Chọn CSDL cần dettach. -> All tasks -> Dettach Database



Hình 3.40 Cửa sổ tạo dettach CSDL

- Nhấn Ok.



Hình 3.41 Cửa sổ tạo hoàn tất Dettach CSDL

1.11.3 Attach tập tin CSDL vào Instance.

Mục này giới thiệu kỹ thuật ghép nối tập tin CSDL vào Instance, là bước tiếp theo của các bước Copy và Dettach. Các bước thực hiện như sau:

- Chọn Instance cần Attach CSDL -> Databases -> all tasks -> attach database...
- Chọn nút browse (...)



Hình 3.42 Cửa sổ Attach CSDL

- Chọn tập tin mdf của CSDL cần attach.
- Đặt tên CSDL.
- Xác định User owner.
- Ok.

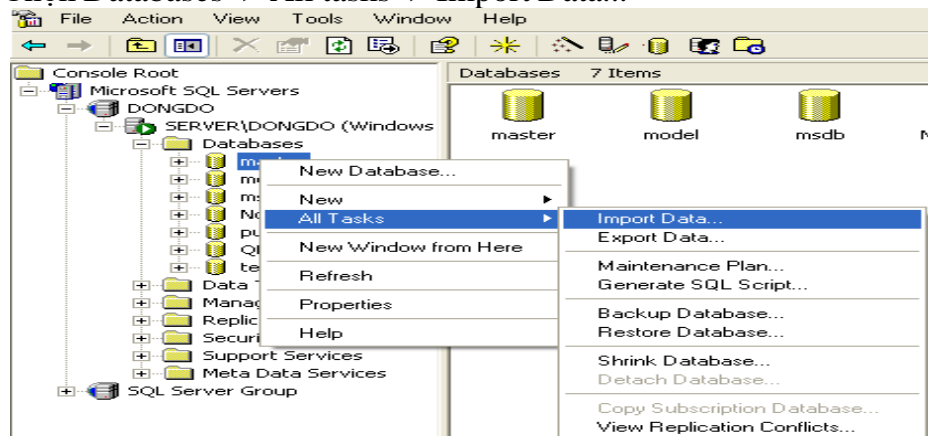
1.12 Import và Export cơ sở dữ liệu.

Phần này sẽ trình bày kỹ thuật nhập và xuất dữ liệu từ CSDL với các hệ quản trị CSDL khác hoặc Instance, CSDL khác của SQL Server.

1.12.1 Import – Nhập dữ liệu.

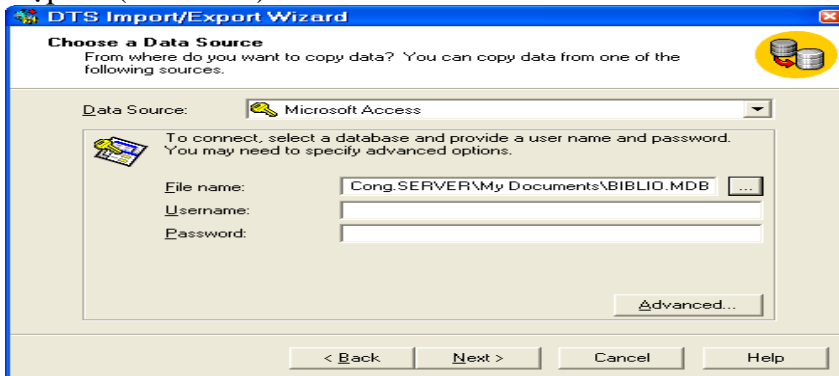
Dùng nhập dữ liệu từ ngoài vào CSDL từ hệ quản trị CSDL khác hoặc CSDL khác của SQL Server.

- Chọn Databases -> All tasks -> Import Data...



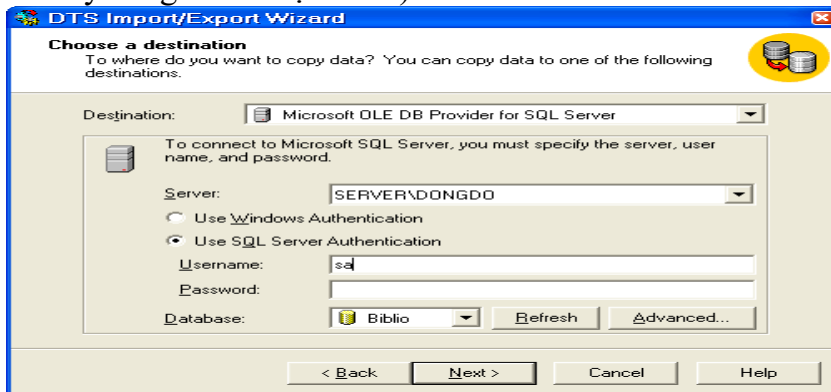
Hình 3.43 Cửa sổ Import Data

- Next -> Chọn Data Source (Có thể là SQL Server, Oracle, Access,...), trong ví dụ minh họa chọn Access.
- Chọn tập tin (file name) -> Next



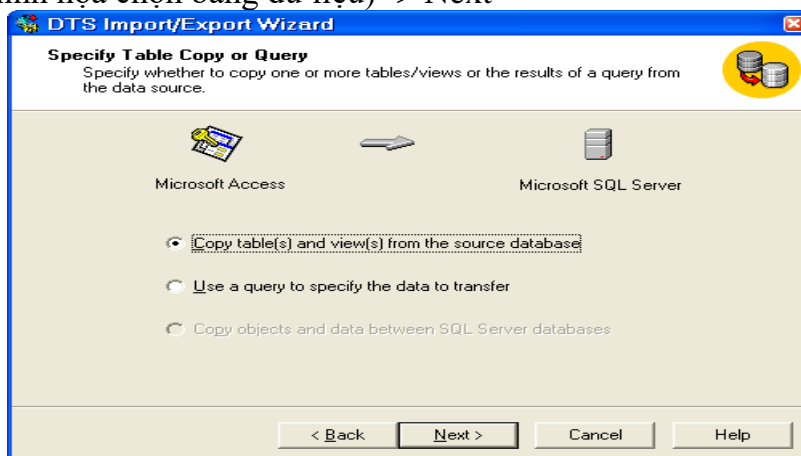
Hình 3.44 Cửa sổ chọn tập tin

- Chọn Instance cần chuyển dữ liệu vào, user name., tên CSDL (có trước hoặc tạo tại thời điểm này bằng cách chọn New) -> Next



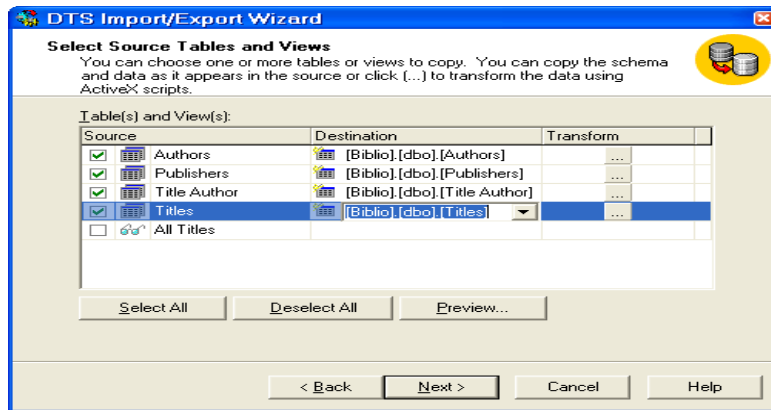
Hình 3.45 Cửa sổ chọn tập tin

- Chọn cách chuyển toàn bộ bảng dữ liệu hay thông qua câu lệnh truyền văn (trong ví dụ minh họa chọn bảng dữ liệu) -> Next



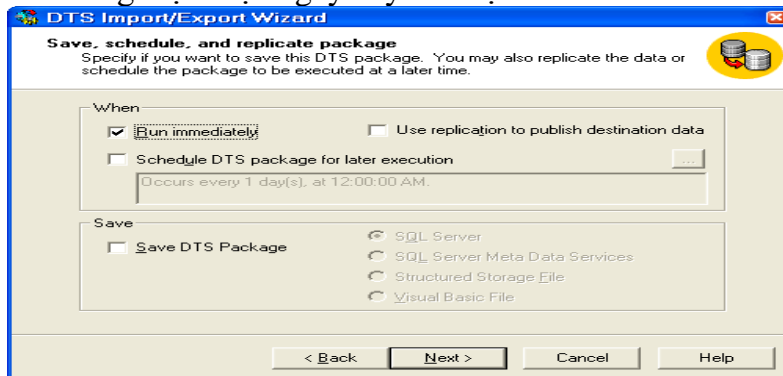
Hình 3.46 Cửa sổ chọn chuyển toàn bộ dữ liệu

- Chọn các bảng, khung nhìn cần Import (có thể lựa chọn một số chức năng khác cụ thể hơn, bạn đọc tự tìm hiểu), tên các bảng, khung nhìn của SQL Server nhận dữ liệu -> Next.



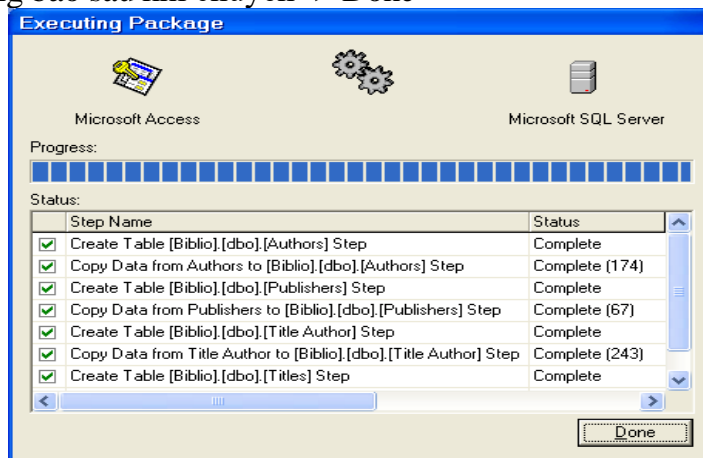
Hình 3.47 Cửa sổ bảng dữ liệu nguồn

- Chọn chức năng thực hiện ngay hay theo lịch -> Next -> Finish



Hình 3.48 Cửa sổ chọn chức năng thực lập lịch

- Xem thông báo sau khi chuyển -> Done



Hình 3.49 Cửa sổ hoàn thành

1.12.2 Export – xuất dữ liệu.

Phần này giới thiệu kỹ thuật xuất dữ liệu từ một CSDL của SQL Server ra một hệ quản trị CSDL khác hoặc một CSDL khác của SQL Server. Tương tự như Import nhưng Export thực hiện Data Source là SQL Server, còn Destination là hệ quản trị CSDL khác hoặc CSDL khác của SQL Server (phần này bạn đọc tự xem xét).



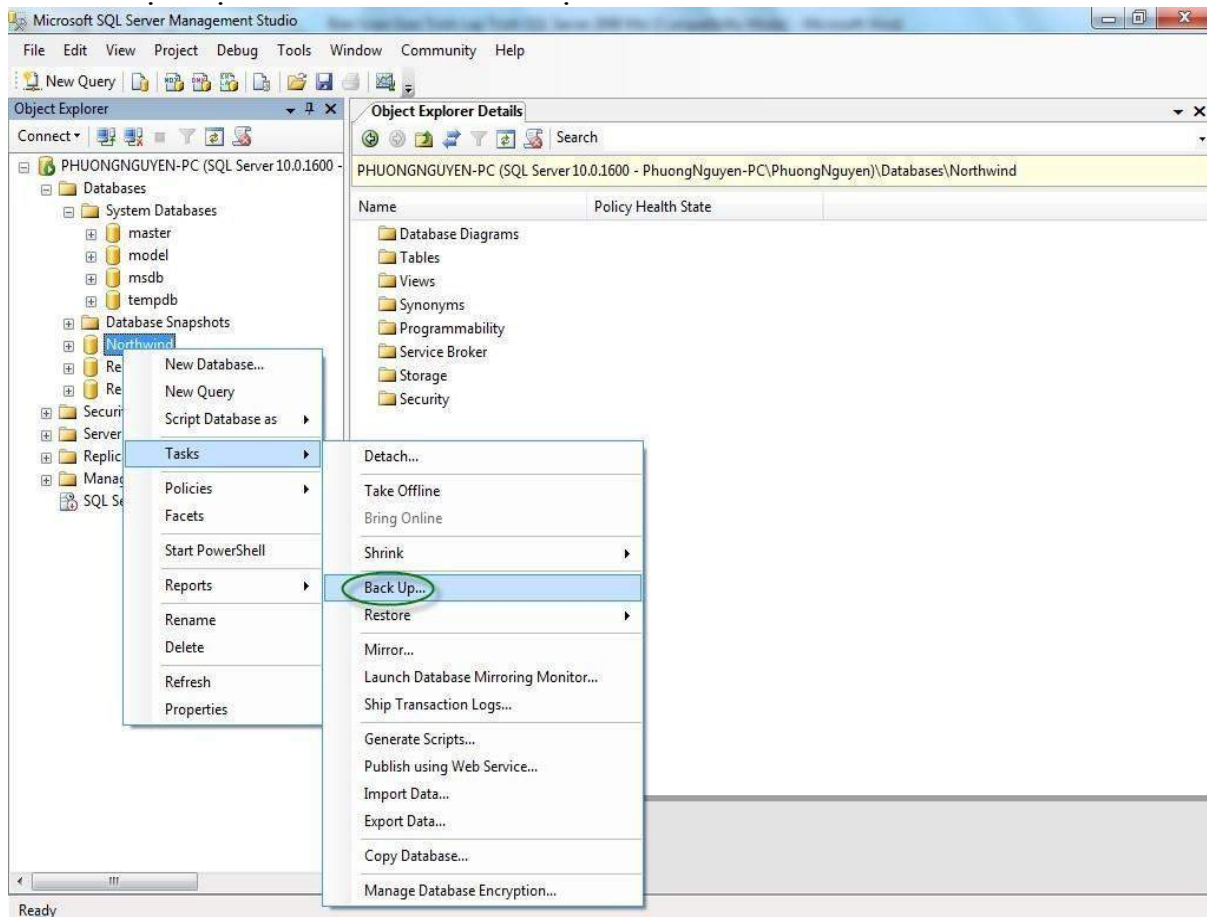
Hình 3.50 Cửa sổ xuất dữ liệu

1.13 Back up và Restore dữ liệu

1.13.1 Back up

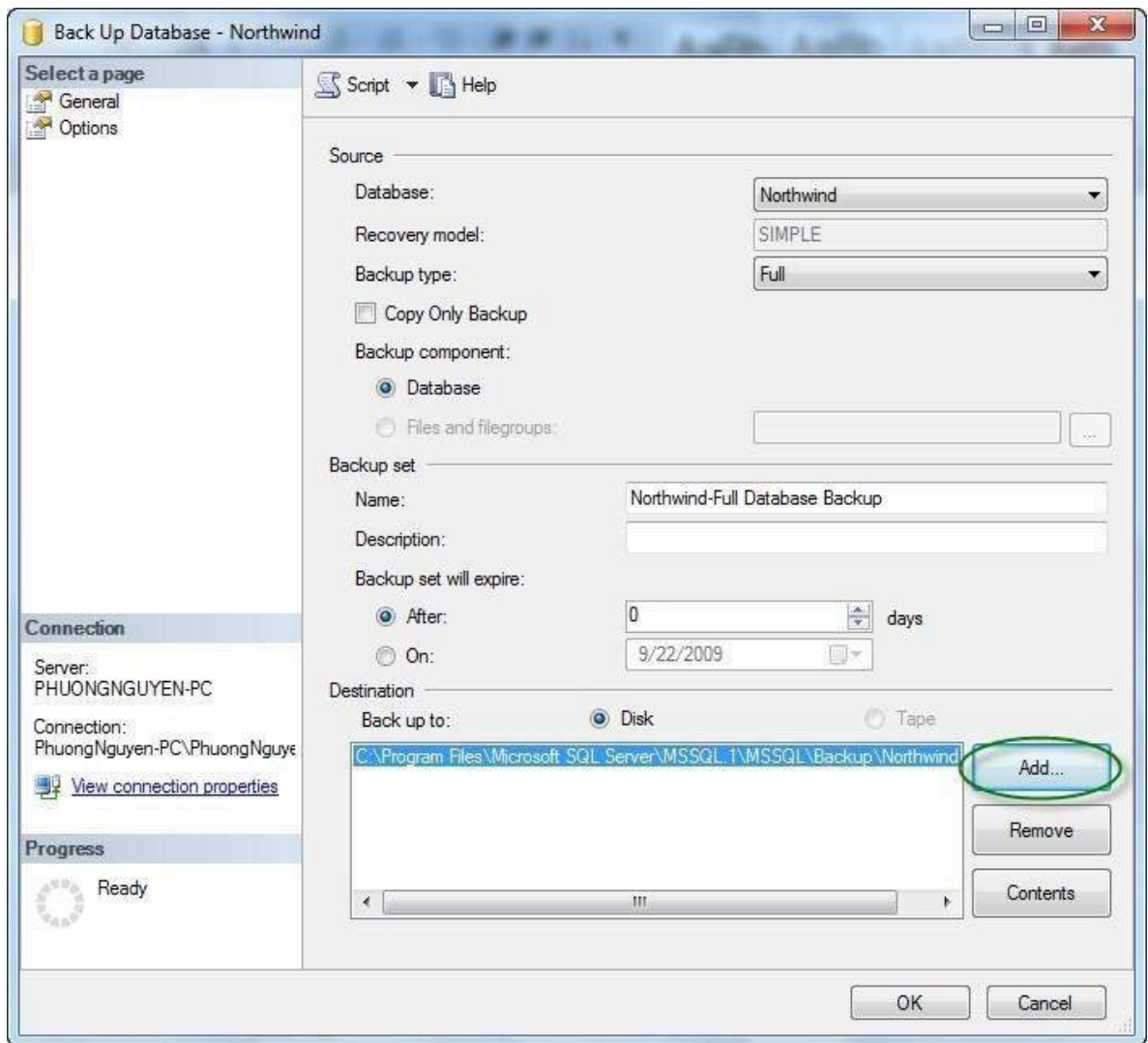
Click phải vào database cần back up -> Chọn Tasks -> Chọn Back up...

Thực hiện các thao tác theo thứ tự các hình bên dưới



Hình 3.51 Mở Back up

- Chọn CSDL để Backup



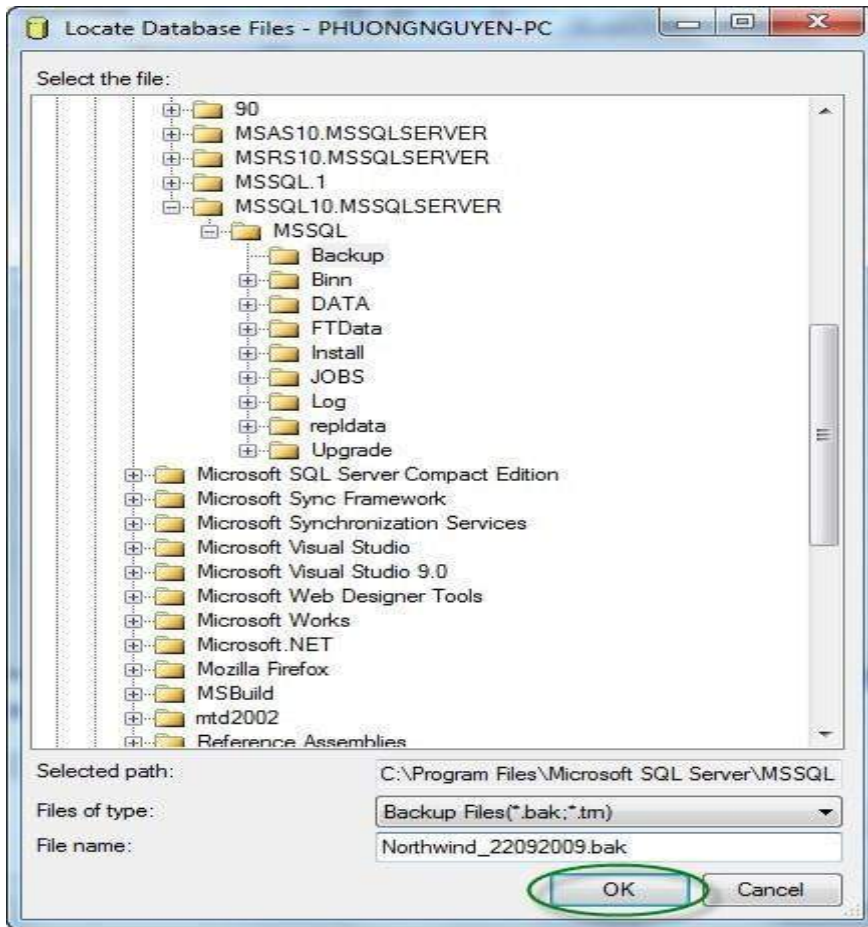
Hình 3.51 Chọn lịch để Back up

Chọn nơi chứa CSDL



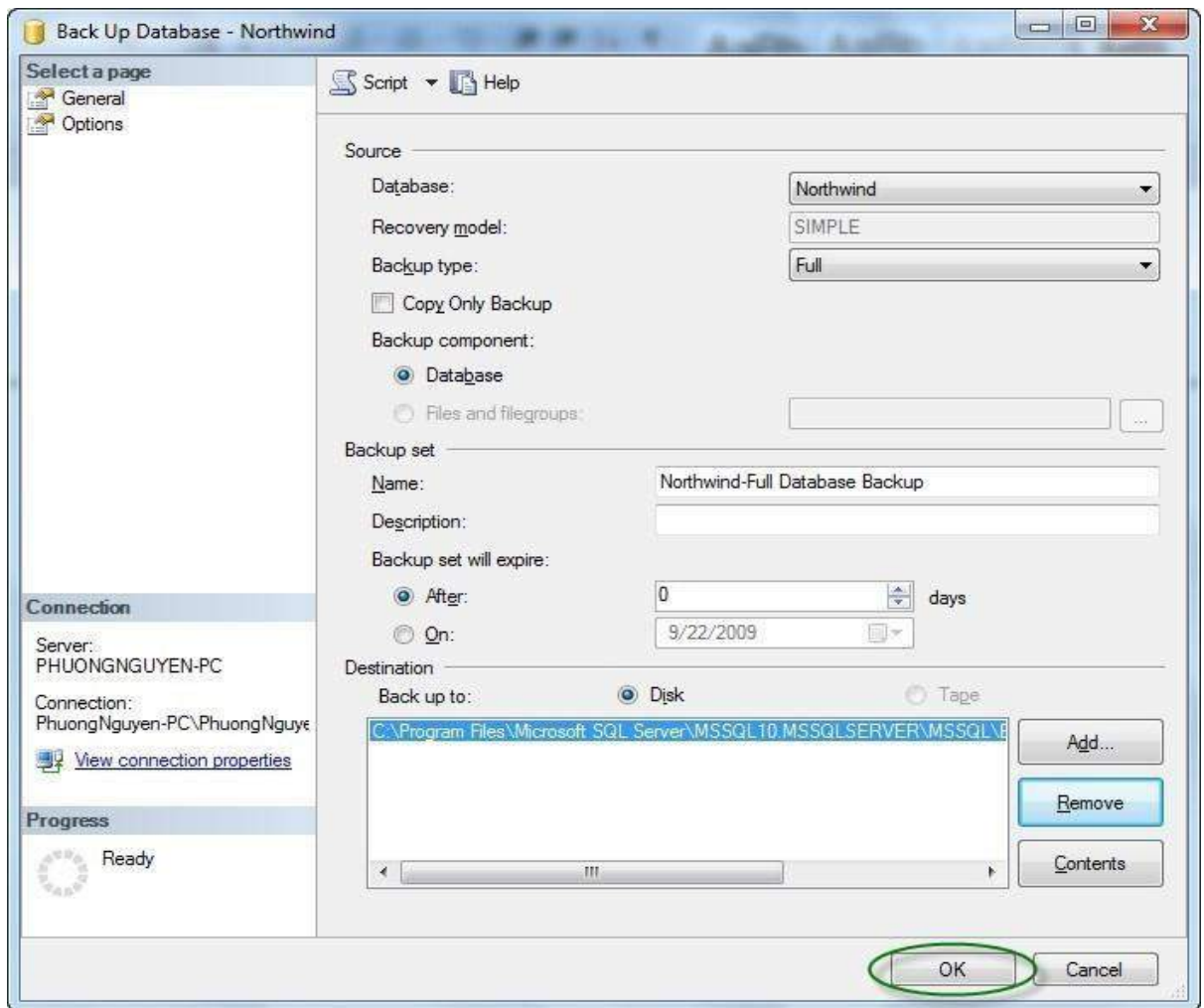
Hình 3.52 Chỉ nơi lưu CSDL để Back up

Chọn OK để chọn nơi lưu Backup



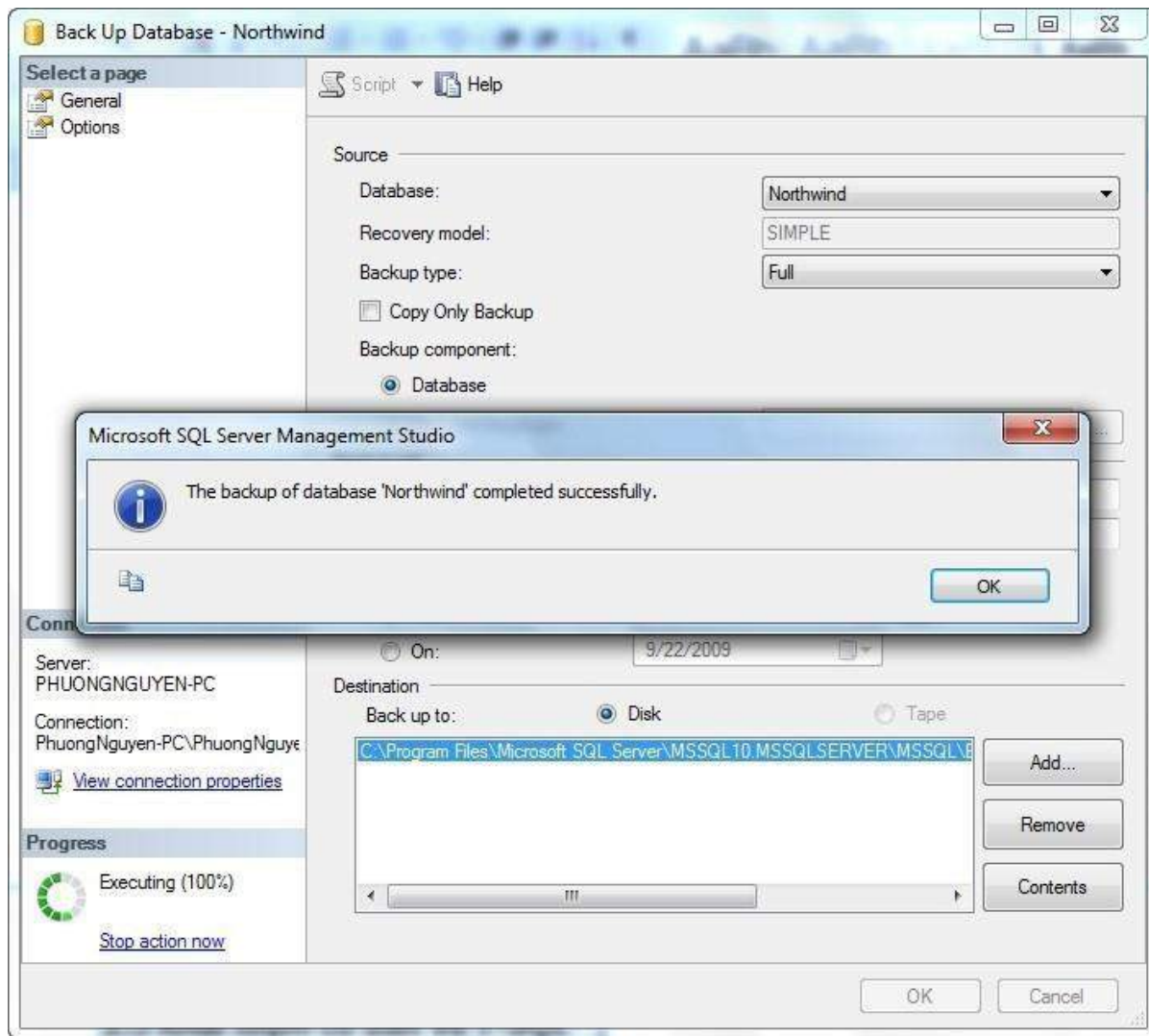
Hình 3.53 Xác định nơi lưu CSDL để Backup

Chọn OK đồng ý Backup



Hình 3.54 Chấp nhận để Back up

Hoàn thành việc Backup

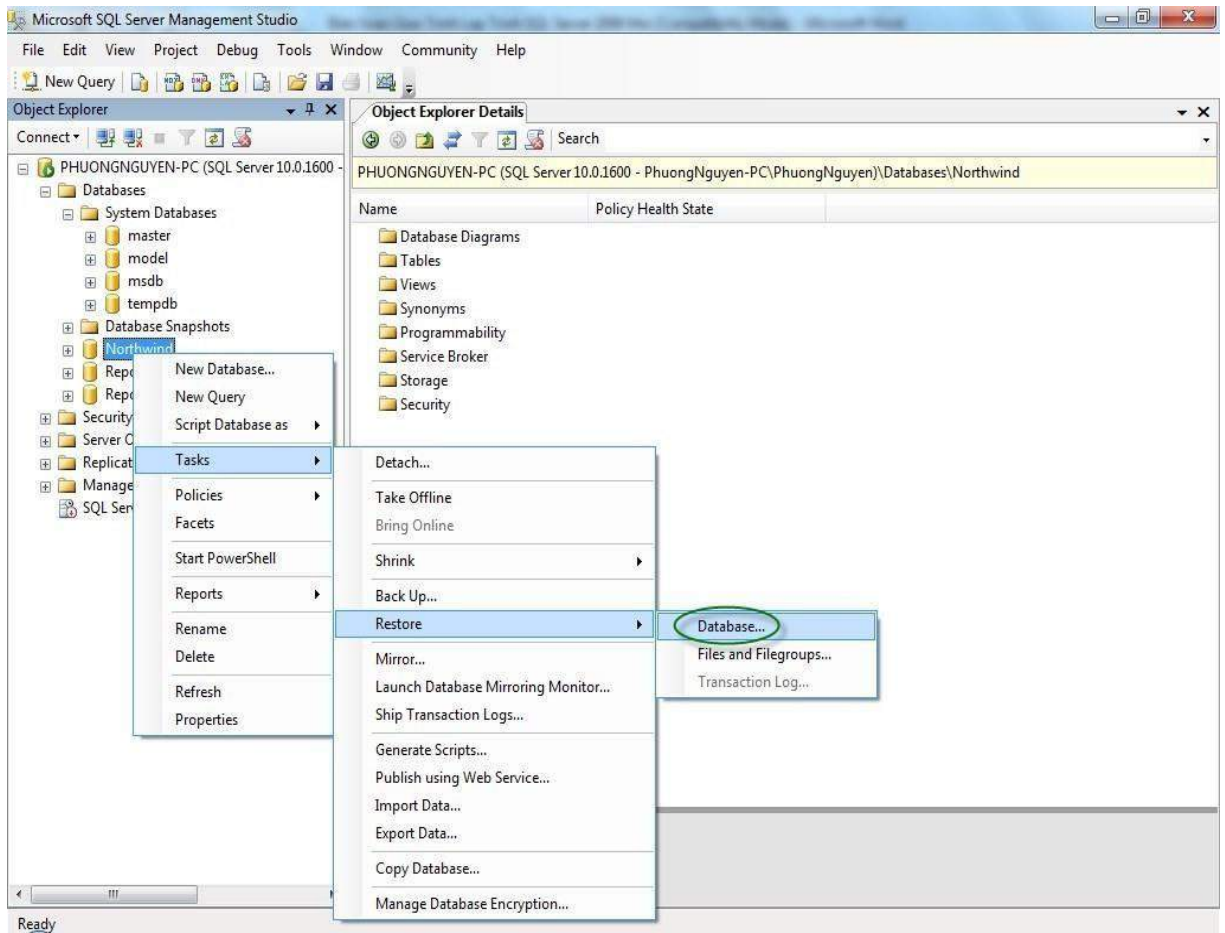


Hình 3.54 hoàn thành Back up

1.12.2. Restore

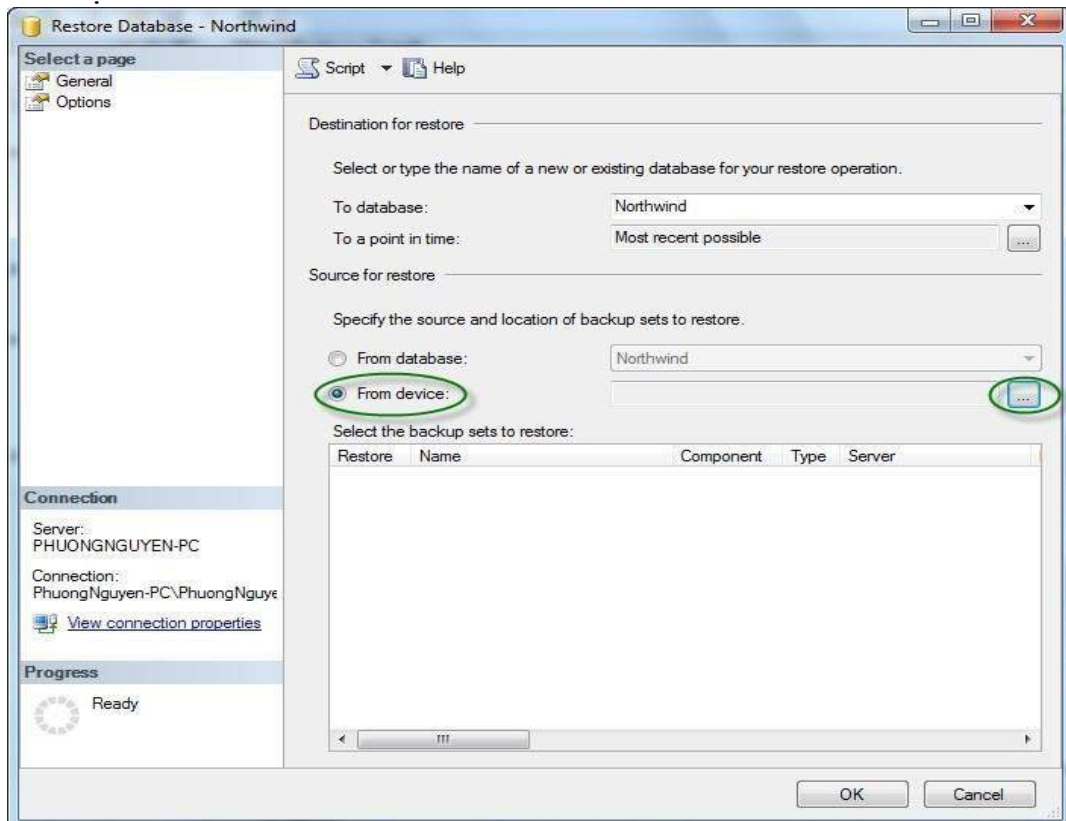
Click phải vào database cần Restore -> Chọn Tasks -> Chọn Restore -> Database...

Thực hiện các thao tác theo thứ tự các hình bên dưới



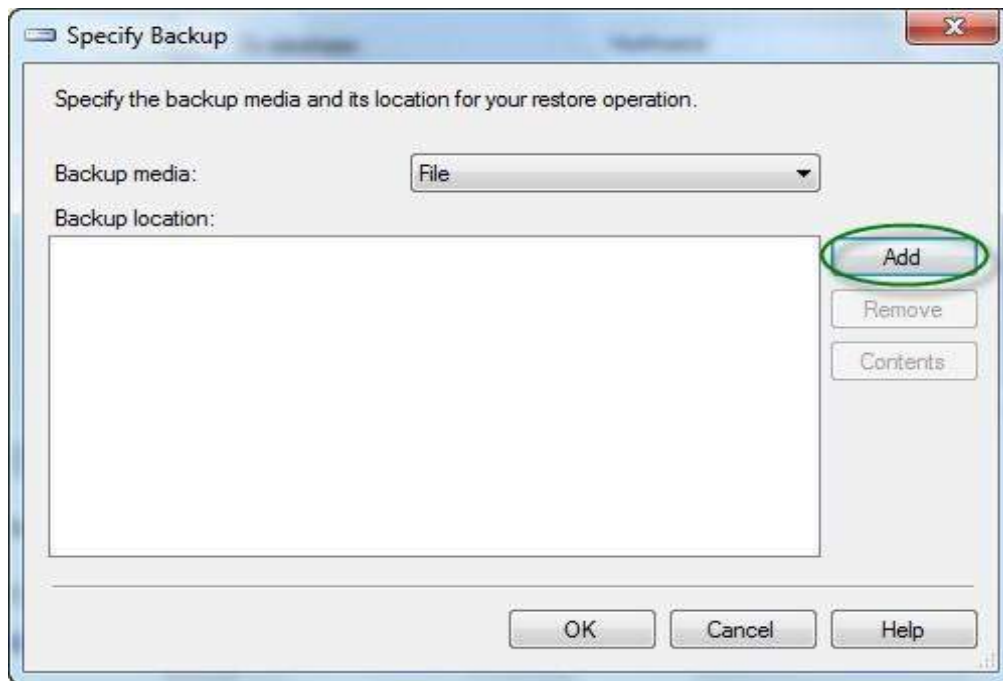
Hình 3.55 Mở cửa sổ Restore

Xác định nơi chứa CSDL



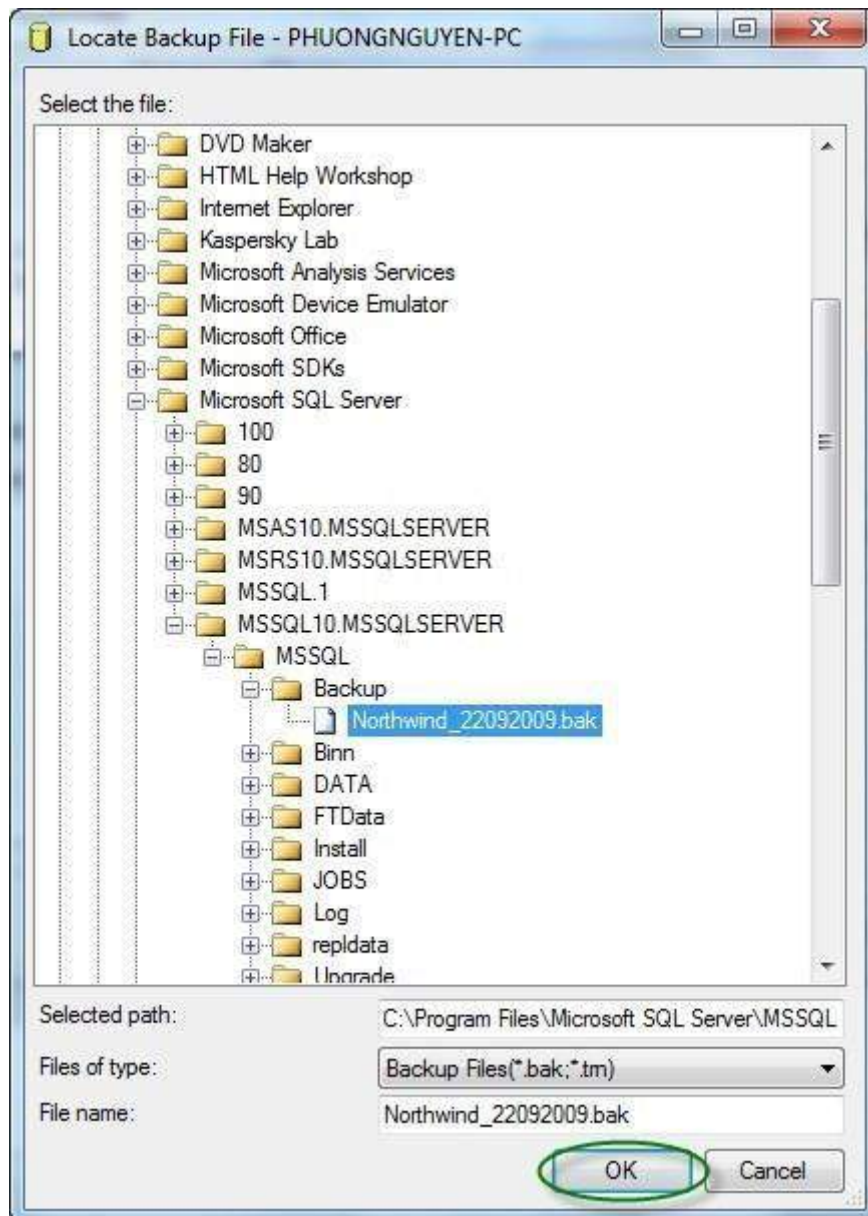
Hình 3.56 Chọn CSDL để số Restore

Chọn Add để mở file



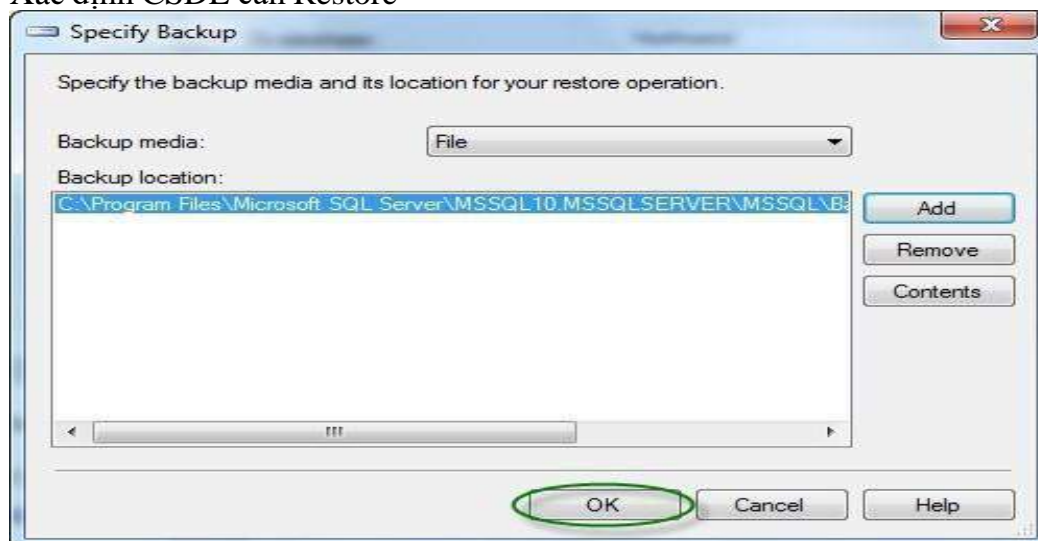
Hình 3.57 Cửa sổ Add file Restore

Xác định CSDL cần Restore



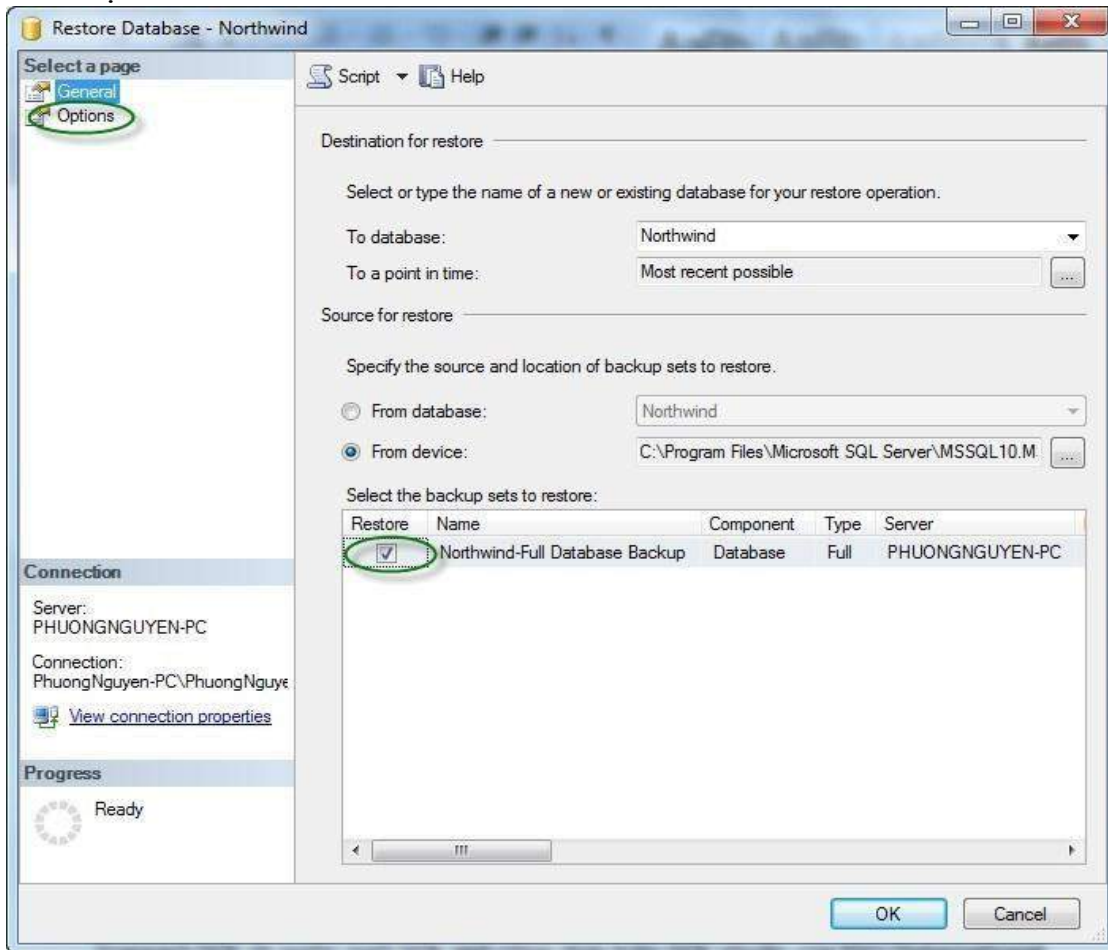
Hình 3.58 Chọn file để Restore

Xác định CSDL cần Restore



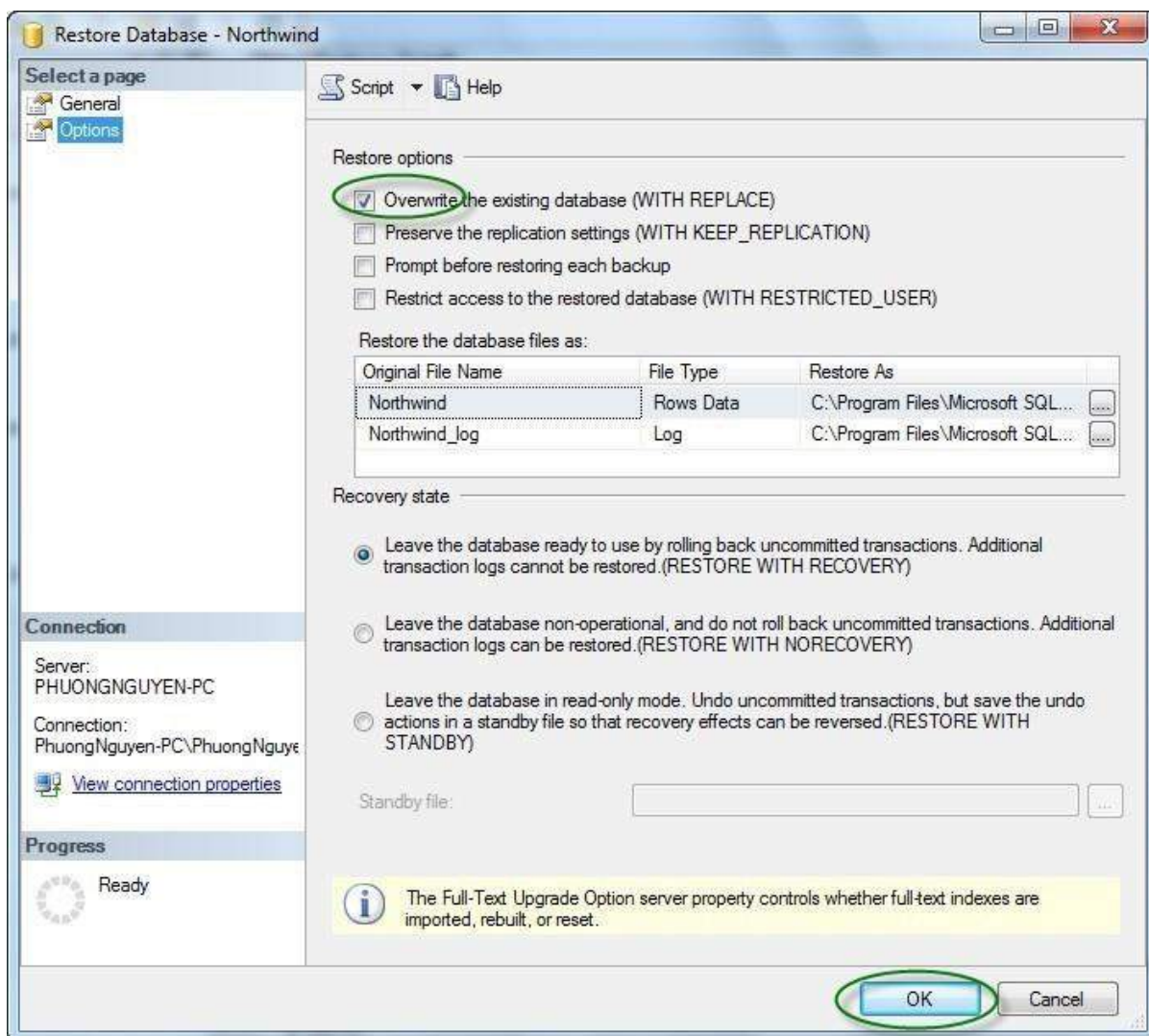
Hình 3.59 Chọn file để Restore

Xác nhận CSDL cần Restore



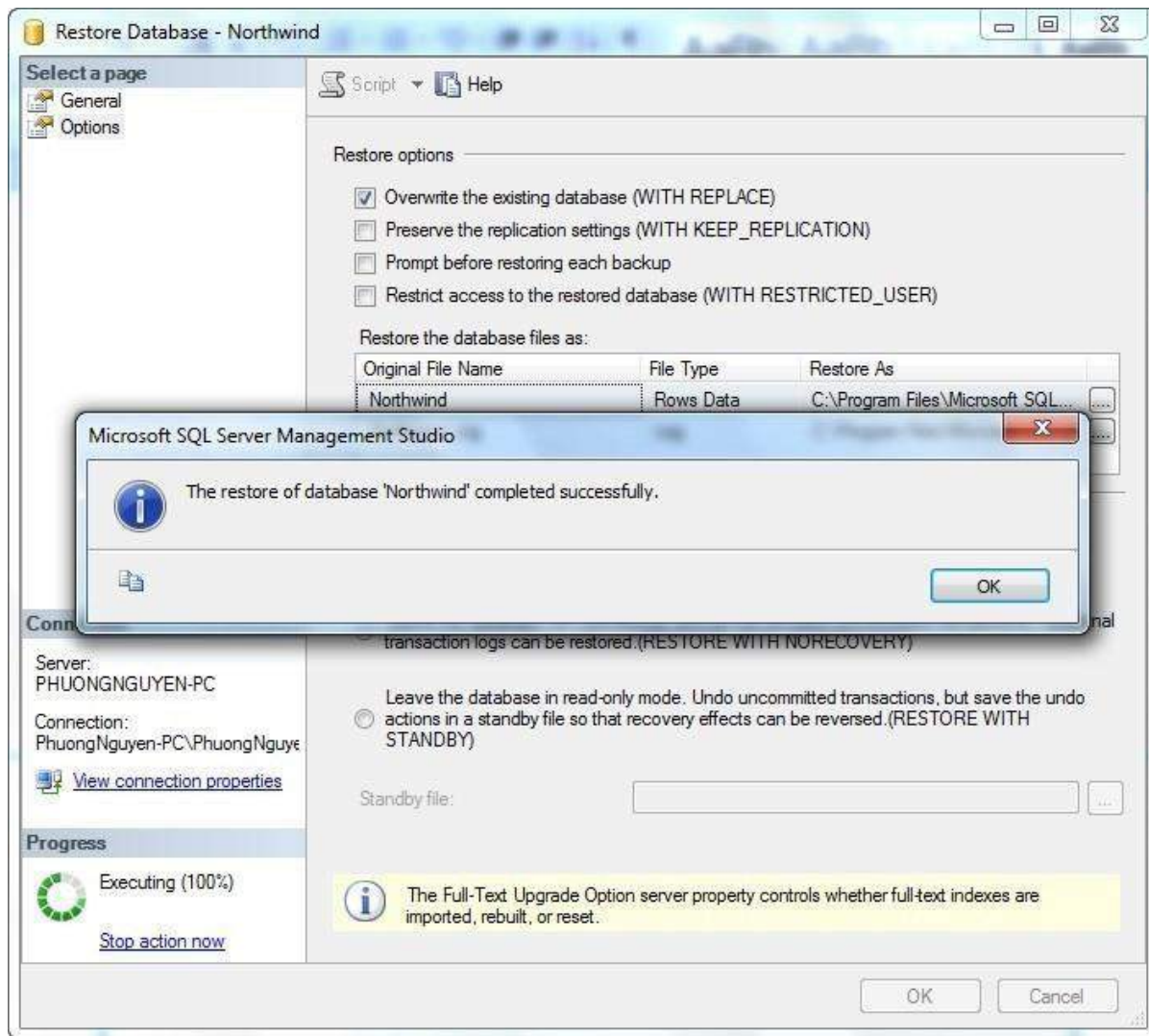
Hình 5.60 Xác nhận CSDL cần Restore

Lựa chọn cách thức Restore



Hình 5.61 Xác nhận CSDL cần Restore

Kết thúc việc Restore



Hình 5.63 Hoàn thành Restore

2. Cách dùng công cụ Query Analyzer để tạo ra các CSDL

2.1 Giới thiệu.

SQL Query Analyzer là giao diện người dùng đồ họa (Graphical User Interface – GUI) dành cho các nhà lập trình. Query Analyzer cho phép thực hiện:

- Tạo các truy vấn (query), bố lệnh (script) và thực thi (execute) chúng để tác động đến CSDL của SQL Server.
- Tạo các đối tượng của CSDL một cách nhanh chóng từ những script được định nghĩa trước.
- Sao chép nhanh chóng các đối tượng của CSDL
- Tạo và thực thi các thủ tục (Stored procedures), hàm người dùng (user-defined function)
- Tìm lỗi (Debug) các thủ tục.
- Tìm lỗi (Debug) các vấn đề hiệu năng của của truy vấn (Show Execution Plan, Show Server Trace, Show Client Statistics, Index Tuning Wizard).
- Định vị các đối tượng trong các CSDL, xem và làm việc với các đối tượng.
- Chèn, cập nhật, xóa các mẫu tin trong table một cách nhanh chóng.

2.2. Khởi động Query Analyzer

SQL Query Analyzer có thể được khởi động từ SQL Server Enterprise Manager hoặc từ Start Menu hoặc từ cửa sổ Command bằng cách thực thi tiện ích ISQLW.

Khi bạn khởi động SQL Query Analyzer, thì hộp thoại Connect to SQL Server xuất hiện. Khi đó bạn phải xác định chế độ chứng thực được dùng để kết nối tới SQL Server

Có thể mở cùng lúc nhiều cửa sổ Query Analyzer để cho phép bạn làm việc trong các CSDL khác nhau hoặc thực thi các script khác nhau trong cùng một thời điểm. Mỗi cửa sổ tạo kết nối riêng biệt đến server. Những kết nối duy trì các cài đặt khác nhau và mỗi cửa sổ có CSDL hiện hành.

2.3. Thành phần chính của Query Analyzer

SQL Query Analyzer bao gồm các cửa sổ, hộp thoại, hướng dẫn (wizard) giúp chúng ta thiết kế các tác vụ (Task) cần thiết để tạo các CSDL, lưu trữ, khai thác dữ liệu trong các CSDL đó.

Cửa sổ Query Analyzer

Thanh tiêu đề (Title bar): Hiển thị tên của Server, CSDL hiện hành, và tài khoản kết nối tới.

Công cụ Database trên thanh công cụ, cho biết và cho phép bạn thay đổi CSDL được kết nối hiện hành.

Editor pane: dùng để đưa vào các câu lệnh và thực thi các câu lệnh T-SQL.

Results pane: hiển thị kết quả của truy vấn được thực thi. Một hoặc nhiều hơn một trang được hiển thị trong results pane.

Trang Messages: Hiển thị thông báo và các lỗi gửi trả từ server.

Trang Results: Hiển thị những kết quả như văn bản tự do.

Trang Results Grid: Hiển thị các kết quả trong bảng kẻ lưới. Dữ liệu trong lưới chỉ để xem, không thể hiệu chỉnh được. Một vài truy vấn yêu cầu server trả về nhiều hơn một tập kết quả thì sẽ có nhiều hơn một trang Results Grid sẽ được hiển thị. Nếu lưới kết quả trống sau khi thi hành một truy vấn thì truy vấn không được trả về một bảng kết quả

Trang Execution Plan: Hiển thị một biểu đồ của kế hoạch thực thi của truy vấn hiện hành. Bật tắt Execution Plan ta chọn từ thực đơn Query.

3. Thiết lập một số cấu hình về Client network utility

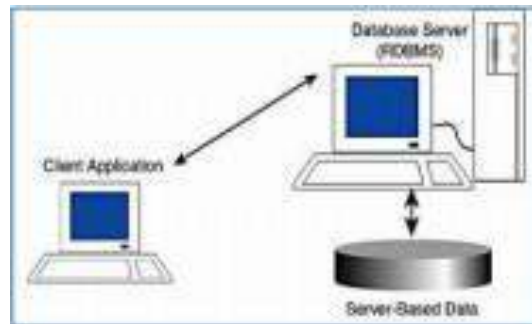
3.1 Giới thiệu về mô hình Client/Server và các hệ quản trị CSDL phục vụ cho mô hình Client/Server.

SQL 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
- Truy xuất và thao tác dữ liệu
- Điều khiển truy cập
- Đảm bảo toàn vẹn dữ liệu

3.2. 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 3.64: 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. 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)

3.3. Các thành phần quan trọng trong SQL Server

Mục tiêu: Trình bày được các thành phần quan trọng của SQL server

SQL Server (phiên bản 2000) được cấu tạo bởi nhiều thành phần như Relational Database Engine, Analysis Service và English Query.... Các thành phần này khi phối hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc lưu trữ và phân tích dữ liệu một cách dễ dàng.

3.3.1 Relational Database Engine - Cái lõi của SQL Server

Đây là một engine có khả năng chứa data ở các quy mô khác nhau dưới dạng table và support tất cả các kiểu kết nối (data connection) thông dụng của Microsoft như ActiveX Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC). Ngoài ra nó còn có khả năng tự điều chỉnh (tune up) ví dụ như sử dụng thêm các tài nguyên (resource) của máy khi cần và trả lại tài nguyên cho hệ điều hành khi một user log off.

3.3.2 Replication - Cơ chế tạo bản sao

Giả sử có một database dùng để chứa dữ liệu được các ứng dụng thường xuyên cập nhật. Muốn có một cái database giống y hệt như thế trên một server khác để chạy báo cáo (report database) (cách làm này thường dùng để tránh ảnh hưởng đến performance của server chính). Vấn đề là report server cũng cần phải được cập nhật thường xuyên để đảm bảo tính chính xác của các báo cáo. Không thể dùng cơ chế back up and restore trong trường hợp này. Thế thì phải làm sao? Lúc đó cơ chế replication của SQL Server sẽ được sử dụng để bảo đảm cho dữ liệu ở 2 database được đồng bộ (synchronized).

Cơ chế nhân bản này cho phép nhiều thể hiện của SQL Server ở vị trí từ xa có cùng dữ liệu. Vị trí từ xa có thể bao gồm các người dùng di động hoặc các site kết nối thông qua internet, dial-up hay intranet. Sự phân chia vật lý của dữ liệu cải thiện hiệu

năng của tổ chức khi dữ liệu cần được xử lý tại các nơi khác để trình bày ở site khác để tham chiếu.

3.3.3 Data Transformation Service (DTS)

Một dịch vụ chuyển dịch data vô cùng hiệu quả. Nếu làm việc trong một công ty lớn trong đó data được chứa trong nhiều nơi khác nhau và ở các dạng khác nhau cụ thể như chứa trong Oracle, DB2 (của IBM), SQL Server, Microsoft Access. chắc chắn sẽ có nhu cầu di chuyển data giữa các server này (migrate hay transfer) và không chỉ di chuyển mà còn muốn định dạng (format) nó trước khi lưu vào database khác, khi đó sẽ thấy DTS giúp bạn giải quyết công việc trên dễ dàng như thế nào. DTS sẽ được học trong bài 8.

3.3.4 Analysis Service

Một dịch vụ phân tích dữ liệu rất hay của Microsoft, Dữ liệu (Data) chứa trong database sẽ chẳng có ý nghĩa gì nhiều nếu như bạn không thể lấy được những thông tin (Information) bổ ích từ đó. Do đó Microsoft cung cấp cho bạn một công cụ rất mạnh giúp cho việc phân tích dữ liệu trở nên dễ dàng và hiệu quả bằng cách dùng khái niệm hình khối nhiều chiều (multi-dimension cubes) và kỹ thuật "đào mỏ dữ liệu" (data mining) sẽ được chúng tôi giới thiệu trong phần tiếp theo.

3.3.5. English Query - Một dịch vụ truy vấn

Đây là một dịch vụ giúp cho việc query data bằng tiếng Anh (English).

3.3.6. Meta Data Service

Dịch vụ này giúp cho việc chứa đựng và "xào nấu" Meta data dễ dàng hơn. Thế thì Meta Data là cái gì vậy? Meta data là những thông tin mô tả về cấu trúc của data trong database như data thuộc loại nào String hay Integer..., một cột nào đó có phải là Primary key hay không....Bởi vì những thông tin này cũng được chứa trong database nên cũng là một dạng data nhưng để phân biệt với data "chính thống" người ta gọi nó là Meta Data. Phần này chắc là bạn phải xem thêm trong một thành phần khác của SQL Server sắp giới thiệu sau đây là SQL Server Books Online vì không có bài nào trong loạt bài này nói rõ về dịch vụ này cả.

3.3.7. SQL Server Books Online

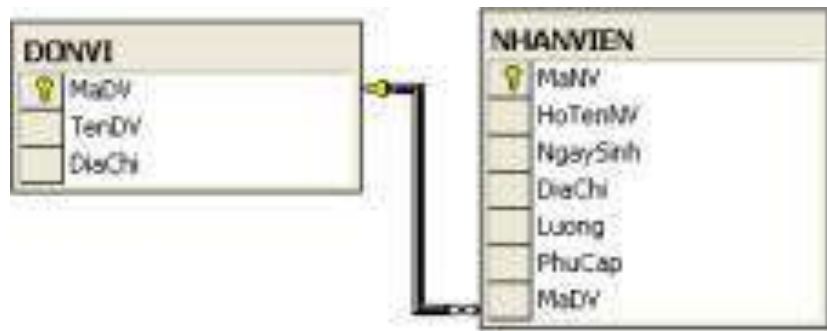
Cho dù bạn có đọc các sách khác nhau dạy về SQL server thì bạn cũng sẽ thấy books online này rất hữu dụng và không thể thiếu được (cho nên Microsoft mới hào phóng đính kèm theo SQL Server).

Người dùng ạn có thể tìm thông tin trong SQL Server Books Online bằng cách:

- Điều hướng thông qua bảng nội dung.
- Gõ một từ khóa trong index.
- Gõ một từ hoặc một cụm từ và thực hiện việc tìm kiếm.

Bài tập thực hành của học viên

1. Thiết lập một số cấu hình về Client network utility trên SQL Server
2. *Sử dụng Enterprise Manager của MS SQL thực hiện các yêu cầu sau*
 - o Tạo một CSDL tên QuanLyNhanVien.
 - Tạo và thiết lập quan hệ cho hai bảng như sơ đồ sau:



Hình 3.65 Sơ đồ quan hệ CSDL QuanLyNhanVien

- Nhập dữ liệu cho hai bảng, mỗi bảng ít nhất 10 bản ghi.
- Tạo khung nhìn xem dữ liệu của từng bảng
- Dettach và Attach cơ sở dữ liệu
- Back up và Restore dữ liệu

3. Sử dụng Enterprise Manager của MS SQL thực hiện các yêu cầu sau

- o Tạo CSDL có tên QLDeAn
 - Xây dựng các bảng và mối quan hệ
 - Nhập dữ liệu cho hai bảng, mỗi bảng ít nhất 10 bản ghi.
 - Tạo khung nhìn xem dữ liệu của từng bảng
 - Dettach và Attach cơ sở dữ liệu
 - Back up và Restore dữ liệu

Hướng dẫn thực hiện

1. Thiết lập một số cấu hình về Client network utility trên SQL Server, tham khảo mục 3 trong bài 3 trong giáo trình
2. **Sử dụng Enterprise Manager của MS SQL thực hiện các yêu cầu sau**
 - o Tạo một CSDL tên QuanLyNhanVien, tham khảo mục 1.3.2.1 trong bài 3 trong giáo trình
 - Tạo và thiết lập quan hệ cho hai bảng như sơ đồ, tham khảo mục 1.4, mục 1.4.2 và mục 1.6 trong bài 3 trong giáo trình
 - o Nhập dữ liệu cho hai bảng, mỗi bảng ít nhất 10 bản ghi, tham khảo mục 1.4.4 trong bài 3 trong giáo trình
 - o Tạo khung nhìn xem dữ liệu của từng bảng, tham khảo mục 1.7 trong bài 3 trong giáo trình
 - o Dettach và Attach cơ sở dữ liệu, tham khảo mục 1.11 trong bài 3 trong giáo trình
 - o Back up và Restore dữ liệu, tham khảo mục 1.13 trong bài 3 trong giáo trình
3. **Sử dụng Enterprise Manager của MS SQL thực hiện các yêu cầu sau**
 - o Tạo CSDL có tên QLDeAn, tham khảo mục 1.3.2.1 trong bài 3 trong giáo trình
 - Xây dựng các bảng và mối quan hệ, tham khảo mục 1.4, mục 1.4.2 và mục 1.6 trong bài 3 trong giáo trình
 - o Nhập dữ liệu cho hai bảng, mỗi bảng ít nhất 10 bản ghi, tham khảo mục 1.4.4 trong bài 3 trong giáo trình
 - o Tạo khung nhìn xem dữ liệu của từng bảng, tham khảo mục 1.7 trong bài 3 trong giáo trình
 - o Dettach và Attach cơ sở dữ liệu, tham khảo mục 1.11 trong bài 3 trong giáo trình
 - o Back up và Restore dữ liệu, tham khảo mục 1.13 trong bài 3 trong giáo trình

Những trọng tâm cần chú ý:

- Trình bày các thao tạo mới CSDL, chú ý các tham số đi kèm CSDL trên SQL Server
- Trình bày các thao tạo về CSDL như: Xóa, chỉnh sửa tham số, nơi lưu trữ, dung lượng ban đầu cho các dữ liệu và logic
- Trình bày các thao tác trên Table như: Tạo mới, thêm, sửa xóa, khóa và các ràng buộc
- Trình bày các thao tác trên khung nhìn như: Tạo mới, thêm, sửa xóa, khóa và truy vấn đơn giản
 - Trình bày các thao tác tạo liên kết cơ sở dữ liệu dữ liệu
 - Trình bày các thao tác Dettach và Attach cơ sở dữ liệu.
 - Trình bày các thao tác Import và Export cơ sở dữ liệu
 - Trình bày các thao tác Back up và Restore dữ liệu
 - Thực hiện đúng các thao tác trên công cụ Enterprise manager

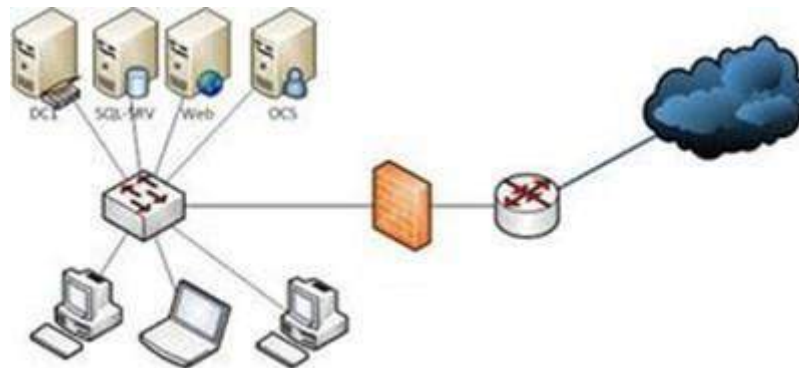
Bài mở rộng và nâng cao

- Tình huống

Công ty TNHH Thương mại Massan đang có một máy chủ MS SQL Server 2014, máy chủ này đang chứa cơ sở dữ liệu của tất cả phần mềm quản lý trong công ty để phục vụ cho các nhu cầu công việc của công ty.

- Phần mềm kế toán.
- Phần mềm quản lý nhân sự và tiền lương.
- Phần mềm quản lý sản xuất và thương mại.
- Phần mềm quản lý kinh doanh.

Mô hình



Hình 6.66 Sơ đồ tổng quát hệ thống mạng công ty Massan

Mô tả bài tập: Database của công ty được lưu trữ trên máy chủ SQL-SRV đang ngày càng nhiều, dữ liệu trong máy chủ là những tài liệu quan trọng của công ty. Cần phải có những kế hoạch giúp an toàn dữ liệu khi có những trục trặc xảy ra đối với máy chủ: hư ổ cứng, lỗi Mainboard, hoặc bị cháy máy...

Yêu cầu bài tập:

- Cấu hình chiến lược sao lưu thảm họa cho Database
- Tiến hành khôi phục dữ liệu cho Database
- Cấu hình các chế độ backup đối với các loại dữ liệu khác nhau.
- Thực hiện restore khi dữ liệu bị mất.

Yêu cầu đánh giá kết quả học tập

Nội dung

- Về kiến thức:
 - + Trình bày các thao tác trên công cụ Enterprise manager
 - + Trình bày các thao tác Back up và Restore dữ liệu

- Về kỹ năng:
 - + Thực hiện đúng việc cài đặt chiến lược Backup (Sao lưu)
 - + Thao tác đúng Full Backup CSDL AdventureWorks
 - + Thao tác đúng Log Backup CSDL AdventureWorks
 - +Thực hiện đúng, Differential Backup CSDL AdventureWorks và Log Backup CSDL AdventureWorks
 - + Thực hiện đúng các bước Restore CSDL
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

Phương pháp

- Về kiến thức: Đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp.
- Về kỹ năng: Đánh giá kỹ năng thực hành thực hiện các thao tác Back up và Restore.
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

BÀI 4: PHÁT BIỂU CƠ BẢN T-SQL

Mã bài: MD16-04

Mục tiêu:

- Trình bày cú pháp và công dụng của các phát biểu.
- Thực hiện được việc truy vấn dữ liệu trên câu lệnh T-SQL đúng yêu cầu.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

Đối với đa số người sử dụng, SQL được xem như là công cụ hữu hiệu để thực hiện các yêu cầu truy vấn và thao tác trên dữ liệu. Trong chương này, ta sẽ bàn luận đến nhóm các câu lệnh trong SQL được sử dụng cho mục đích này. Nhóm các câu lệnh này được gọi chung là ngôn ngữ thao tác dữ liệu (DML: Data Manipulation Language) bao gồm các câu lệnh sau:

- SELECT: Sử dụng để truy xuất dữ liệu từ một hoặc nhiều bảng.
- INSERT: Bổ sung dữ liệu.
- UPDATE: Cập nhật dữ liệu
- DELETE: Xoá dữ liệu

Trong số các câu lệnh này, có thể nói SELECT là câu lệnh tương đối phức tạp và được sử dụng nhiều trong cơ sở dữ liệu. Với câu lệnh này, ta không chỉ thực hiện các yêu cầu truy xuất dữ liệu đơn thuần mà còn có thể thực hiện được các yêu cầu thống kê dữ liệu phức tạp. Cũng chính vì vậy, phần đầu của chương này sẽ tập trung tương đối nhiều đến câu lệnh SELECT. Các câu lệnh INSERT, UPDATE và DELETE được bàn luận đến ở cuối chương.

1. Cú pháp các câu lệnh T-SQL

1.1 Identifiers (định danh)

Đây chính là tên của các database object. Nó dùng để xác định một object. (Chú ý khi nói đến Object trong SQL Server là chúng ta muốn đề cập đến table, view, stored procedure, index... Vì hầu như mọi thứ trong SQL Server đều được thiết kế theo kiểu hướng đối tượng (object-oriented)). Trong ví dụ sau TableX, KeyCol, Description là những identifiers

```
CREATE TABLE TableX
```

```
(KeyCol INT PRIMARY KEY, Description NVARCHAR(80))
```

Có hai loại Identifiers một loại thông thường (Regular Identifier) và một loại gọi là Delimited Identifier, loại này cần có dấu "" hay dấu [] để ngăn cách. Loại Delimited được dùng đối với các chữ trùng với từ khóa của SQL Server (reserved keyword) hay các chữ có khoảng trống.

Ví dụ:

```
SELECT * FROM [My Table]
```

```
WHERE [Order] = 10
```

Trong ví dụ trên chữ Order trùng với keyword Order nên cần đặt trong dấu ngoặc vuông [].

Đây chính là tên của các database object. Nó dùng để xác định một object. (Chú ý khi nói đến Object trong SQL Server là chúng ta muốn đề cập đến table, view, stored procedure, index... Vì hầu như mọi thứ trong SQL Server đều được thiết kế theo kiểu hướng đối tượng (object-oriented)). Trong ví dụ sau TableX, KeyCol, Description là những identifiers

```
CREATE TABLE TableX
```

```
(KeyCol INT PRIMARY KEY, Description NVARCHAR(80))
```


1.2. Biểu thức

Các biểu thức trong T-SQL thường có dạng: Identifier + Operators (như +, -, *, /, =...) + value

1.3. Comments (Chú thích)

T-SQL dùng -- ...-- để đánh dấu phần chú thích cho lệnh đơn và dùng /*...*/ để chú thích cho một nhóm lệnh.

1.4. Lệnh USE

Dùng thay đổi ngữ cảnh CSDL để chỉ định CSDL mà các câu lệnh tiếp theo sẽ có tác dụng trên đó (khai báo rằng ta sẽ dùng CSDL nào cho các câu lệnh phía sau nó)

USE <database_name>,

Trong đó <database_name>: Là tên của CSDL.

1.5. Một số hàm của T-SQL

1.5.1 Các hàm về xử lý ký tự

Tên hàm	Ví dụ	Miêu tả
ASCII	SELECT ASCII('ABC')	Trả về 65, là mã ASCII của 'A'
CHAR	SELECT CHAR(65)	Trả về 'A'.
CHARINDEX	SELECT CHARINDEX('E','HELLO')	Trả về 2
DIFFERENCE	SELECT DIFFERENCE('HELLO','HELL')	Trả về 4. Giá trị 4 là có độ tương tự cao nhất (0 -> 4)
LEFT	SELECT LEFT('RICHARD',4)	Trả lại 'RICH'
LEN	SELECT LEN('RICHARD')	Trả lại 7
LOWER	SELECT LOWER('RICHARD')	Trả lại 'richard', bao gồm toàn chữ thường
LTRIM	SELECT LTRIM(' RICHARD')	Trả lại 'RICHARD', bỏ đi các ký tự trống bên trái
PATINDEX	SELECT PATINDEX('%BOX%', 'ACTION BOX')	Trả lại 7, vị trí xuất hiện đầu tiên.
REVERSE	SELECT REVERSE('ACTION')	Trả lại 'NOTICA', đảo ngược của 'ACTION'
RIGHT	SELECT RIGHT('RICHARD', 4)	Trả về 'HARD'
RTRIM	SELECT RTRIM('RICHARD ')	RTRIM trả về 'RICHARD' (sau khi đã xóa bỏ các ký tự trống bên phải)
SPACE	SELECT 'RICHARD' + SPACE(2) + 'HILL'	Trả về 'RICHARD HILL'
SUBSTRING	SELECT SUBSTRING('weather',2,2)	Trả về 'EA'
UPPER	SELECT UPPER('Richard')	Trả về RICHARD, mọi ký tự ở dạng chữ hoa
STUFF	SELECT STUFF('Weather',2,3,'i')	Xóa 3 ký tự từ vị trí thứ 2, sau đó thêm vào ký tự i tại vị trí xóa. Kết quả thu được là 'Wiher'

1.5.2 Các hàm ngày tháng

Hàm	Tham số	Miêu tả
DATEADD	(datepart,number,date)	Cộng thêm datepart vào date
DATEDIFF	(datepart,date1,date2)	Tính toán độ sai lệch về datepart giữa date1 và date2
DATENAME	(datepart,date)	Trả lại tên datepart, ví dụ October
DATEPART	(datepart,date)	Trả lại datepart dưới giá trị số
GETDATE	()	Trả lại ngày giờ hiện tại

Trong đó

Datepart	Abbreviation
Year	yy, yyyy
Quarter	qq, q
Month	mm, m
dayofyear	dy, y
Day	dd, d
Week	wk, ww
weekday	dw
Hour	hh
Minute	mi, n
Second	ss, s
millisecond	ms

Ta có các ví dụ sau:

```
SELECT GETDATE()
```

Trả lại ngày giờ hiện tại

```
SELECT DATEDIF(yy,Ord_date, GETDATE())
FROM Sales:
```

Tính độ lệch về năm giữa Ord_date và ngày hiện tại của table Sales.

1.5.3 Các hàm số học

Tên hàm	Miêu tả
ABS	Giá trị tuyệt đối
ACOS, ASIN, ATAN	Trả lại giá trị radian của các giá trị cos, sin, tan tương ứng

COS, SIN, COT, TAN	Tính các giá trị cos, sin, tan, cotan tương ứng
DEGREES	Trả về số nguyên nhỏ nhất lớn hơn hoặc bằng
EXP	Hàm mũ cơ số e
FLOOR	Trả về số nguyên lớn nhất nhỏ hơn hoặc bằng
LOG	Hàm logarit cơ số e.
LOG10	Logarit cơ số 10
PI	Trả về số PI (3.141592653)
POWER	Tính hàm mũ dạng a^b với a,b là tham số đưa vào
RADIANS	Chuyển đổi từ độ sang radian
RAND	Trả về một số thực trong khoảng [0,1]
ROUND	Hàm làm tròn, sẽ được nói rõ sau
SIGN	Trả về dấu, có thể là dương, âm hoặc bằng không
SQRT	Căn bậc 2

Ví dụ về hàm ROUND(làm tròn)

```
SELECT Score = ROUND(79.9,0)
```

Kết quả trả về là 80.

Chi tiết về hàm ROUND như sau:

Hàm	Kết quả
ROUND(1234.567,2)	1234.570
ROUND(1234.567,1)	1234.600
ROUND(1234.567,0)	1235.000
ROUND(1234.567,-1)	1230.000
ROUND(1234.567,-2)	1200.000
ROUND(1234.567,32)	1000.000

2 Data Definition Language (DDL):

Đây là những lệnh dùng để quản lý các thuộc tính của một database như định nghĩa các hàng hoặc cột của một table, hay vị trí data file của một database...thường có dạng sau:

- **Create** *object_Name*
- **Alter** *object_Name*
- **Drop** *object_Name*

Trong đó *object_Name* có thể là một table, view, stored procedure, indexes...

2.1 Lệnh tạo cơ sở dữ liệu – Create database

Khi tạo một cơ sở dữ liệu, một không gian cơ sở dữ liệu cũng tự động được tạo ra và các bảng dữ liệu được đặt trong không gian cơ sở dữ liệu này.

2.1.1 Cách 1: Dùng Query Analyzer

```
CREATE DATABASE database_name
```

```
ON [ PRIMARY ]
```

```
(NAME = <logical_file_name>,
```

```
FILENAME = <'os_file_name'>
```

```

[, SIZE = <size> ]
[, MAXSIZE = <max_size> ]
[, FILEGROWTH = <growth_increment> ] ) [,...n ]
LOG ON
( NAME = <logical_file_name>,
  FILENAME = <'os_file_name'>
  [, SIZE = <size> ]
  [, MAXSIZE = <max_size >]
  [, FILEGROWTH = <growth_increment> ] ) [,...n ]

```

Trong đó:

Database_name: tên CSDL

Logical_file_name: Là tên file logic mà các câu lệnh T-SQL khi được thực hiện tham chiếu đến.

Os_file_name: đường dẫn thực đến file CSDL

Size: kích thước ban đầu của file CSDL

Max_size: kích thước tối đa của file CSDL

Growth_incremen: mức độ tăng trưởng dung lượng file dữ liệu mỗi khi cần mở rộng file CSDL (ngầm định là MB)

Chú ý: Có thể tạo CSDL bằng lệnh Create Database sau: (khi đó các thông số sẽ được chọn ngầm định)

```
Create Database <database_name>
```

2.2.2 Cách 2: Dùng Enterprise Manager

Nháy chuột phải vào Database/New Database

Đặt tên cho cơ sở dữ liệu

Ví dụ:

Lệnh **Create** sau sẽ tạo ra một table tên Importers với 3 cột

CompanyID, CompanyName, Contact

USE Northwind

```

CREATE TABLE Importers(
    CompanyID int NOT NULL,
    CompanyName varchar(40) NOT NULL,
    Contact varchar(40) NOT NULL
)

```

2.2. Lệnh tạo bảng – CREATE TABLE

2.2.1 Cú pháp

```

CREATE TABLE <Tên_bảng>
( <Tên_cột_1> <Kiểu_dữ_liệu> [ràng_buộc_cột],
  <Tên_cột_2> <Kiểu_dữ_liệu> [ràng_buộc_cột],
  .....
, <Tên_cột_n> <Kiểu_dữ_liệu> [ràng_buộc_cột]
[ <Ràng_buộc_bảng > ] )

```

Trong đó:

+ Tên_bảng, Tên_cột: Do người sử dụng tự định đặt (phải tuân theo quy tắc đặt tên)

+ Kiểu_dữ_liệu: Chọn một kiểu dữ liệu sao cho phù hợp nhất với dữ liệu người dùng sẽ nhập vào

+ ràng_buộc_cột: Là một số các ràng buộc về khóa, ràng buộc mặc định, hoặc các quy định về khuôn dạng dữ liệu

- Null: Dữ liệu giá trị tại cột này có thể nhận giá trị rỗng

- Not Null: Bắt buộc phải nhập giá trị cho cột này
- Unique: giá trị nhập vào cột phải duy nhất
- Ràng buộc khóa chính: Primary key
- Ràng buộc khóa ngoài: Foreign Key References <Tên_bảng_tham_chiếu>(Tên_cột_tham_chiếu).
- Ràng buộc mặc định: Default <giá trị mặc định>, dùng để quy định giá trị mặc định cho một cột. Giá trị này sẽ tự động được gán cho cột khi người sử dụng bổ sung một bản ghi mà không chỉ định giá trị cho cột (trên mỗi cột chỉ có tối đa một ràng buộc mặc định)
- Ràng buộc kiểm tra: CHECK (*Biểu thức logic*)

+ Ràng_buộc_bảng

- Ràng buộc khóa chính: Giá trị dùng để xác định duy nhất một đối tượng nên giá trị của chúng phải duy nhất, không thể nhận giá trị Null. Để khai báo ràng buộc khóa chính ta sử dụng cú pháp:

CONSTRAINT <Tên_ràng_buộc> PRIMARY KEY (<Ds_thuộc_tính_khóa >)

- Ràng buộc khóa ngoài: dùng để kiểm tra giá trị của cột có phù hợp với cột trong bảng khác quan hệ với bảng hiện có cột ràng buộc hay không. Để khai báo một ràng buộc khóa ngoài ta dùng cú pháp sau:

CONSTRAINT <Tên_ràng_buộc> FOREIGN KEY (Tên_thuộc_tính_khóa_ngoài) REFERENCES <Tên_bảng_liên_kết> (Tên_thuộc_tính_liên_kết).

- Ràng buộc Check: Dùng để chỉ định các giá trị hay khuôn dạng dữ liệu có thể được chấp nhận đối với một cột. Trên một cột có thể sử dụng nhiều ràng buộc CHECK. Để khai báo một ràng buộc CHECK đối với một cột nào đó ta sử dụng cú pháp sau:

CONSTRAINT <Tên_ràng_buộc> CHECK (*Biểu thức logic*), trong đó *Biểu thức logic* quy định giá trị hay khuôn dạng của dữ liệu. Khi đó, chỉ những giá trị dữ liệu nào làm cho biểu thức logic nhận giá trị đúng mới được chấp nhận.

2.2.2 Identity: Khi bạn muốn một cột có giá trị tăng tự động như AutoNumber nên định nghĩa cột đó như IDENTITY [(*seed, increment*)].

Trong đó Seed là giá trị khởi đầu khi QSL tự động tăng giá trị. Increment là bước tăng cho biết mỗi lần tăng cần bao nhiêu giá trị

Ví dụ: Create table MonHoc(MaM tinyint Identity(1,1) Primary key, TenMon VarChar(30))

Chú ý: Identity chỉ áp dụng với kiểu số nguyên.

2.2.3 Cột có khả năng tính toán - Cột ảo (Trường ảo):

Khi khai báo cột có thuộc tính Computed ta không cần không gian đĩa để lưu trữ dữ liệu cho cột đó, giá trị của chúng chính là sự kết hợp dữ liệu từ nhiều cột khác

Cú pháp: <Tên_cột> AS <Biểu_thức>

2.3 Lệnh ALTER

Lệnh Alter sau đây cho phép ta thay đổi một phần cấu trúc của các đối tượng table, view,

Cú pháp: Alter <Object type> <Object namre>

2.3.1 ALTER DATABASE:

Dùng để chỉnh sửa các thông số của cơ sở dữ liệu

Cú pháp:

ALTER DATABASE database

ADD FILE <filespec > [,...n]

| ADD LOG FILE <filespec > [,...n]
 | REMOVE FILE logical_file_name
 | MODIFY FILE <filespec >
 | MODIFY NAME = new_dbname

Filespec::=(NAME = logical_file_name
 [, NEWNAME = new_logical_name]
 [, FILENAME = 'os_file_name']
 [, SIZE = size]
 [, MAXSIZE = max_size]
 [, FILEGROWTH = growth_increment])

2.3.2. Sửa đổi cấu trúc bảng – ALTER TABLE

Sau khi đã tạo bảng, ta có thể tiến hành sửa đổi cấu trúc hay thuộc tính của bảng như bổ sung cột, bổ sung khóa, thay đổi các ràng buộc,... Để có thể sửa đổi cấu trúc bảng ta sử dụng lệnh ALTER với cú pháp sau:

+ Thêm cột mới

```
ALTER TABLE <table_name>
ADD <column_name> <data_type> [column_constraint], [... n]
```

+ Thêm ràng buộc bảng

```
ALTER TABLE <table_name>
ADD <table_constraint>
```

+ Xóa cột

```
ALTER TABLE <table_name>
DROP <column_name>[,...n]
```

+ Xóa ràng buộc bảng

```
ALTER TABLE <table_name>
DROP CONSTRAINT<Constraint_name>[...n]
```

2.4. Lệnh xóa – DROP

Lệnh Drop dùng để xóa đối tượng của cơ sở dữ liệu

```
DROP <Object_type > <Object_name>[,...n]
```

Object_type bao gồm các đối tượng: Database, table, view, trigger,...

3. Manipulation Language (DML):

Đây là những lệnh phổ biến dùng để xử lý data như Select, Update, Insert, Delete

3.1. Lệnh truy vấn dữ liệu - SELECT

Việc truy cập và lấy các thông tin từ database được SQL cho phép thực hiện qua câu lệnh SELECT. Câu lệnh SELECT có phạm vi ứng dụng rất rộng, có thể truy cập dữ liệu từ một table, hay từ nhiều table

Các từ khóa SELECT, FROM, WHERE được sử dụng để tạo nên một câu lệnh SELECT đơn giản nhất

Cú pháp tổng quát có dạng sau:

```
SELECT [ ALL | DISTINCT ] <danh_sách_chọn>
FROM <danh_sách_bảng>
[ WHERE <điều_kiện > ]
[ GROUP BY <danh_sách_cột > ]
[ HAVING <điều_kiện > ]
[ ORDER BY <tên_cột > [ ASC | DESC ] [,...n]]
```

a) Mệnh đề FROM

Mệnh đề FROM trong câu SELECT được sử dụng nhằm chỉ định các bảng cần truy xuất dữ liệu. Sau FROM là danh sách tên của các bảng và khung nhìn tham gia vào truy vấn. Tên của các bảng và các khung nhìn được phân cách nhau bởi dấu phẩy. Ví dụ: Câu lệnh dưới đây hiển thị danh sách các công ty.

```
SELECT MaCongTy, TenCongTy, DiaChi
FROM CongTy
```

Chú ý: Ta có thể sử dụng các bí danh cho các bảng hay khung nhìn trong câu lệnh SELECT. Bí danh được gán trong mệnh đề FROM bằng cách chỉ định bí danh ngay sau tên bảng.

Ví dụ: Câu lệnh sau gán bí danh là CC cho bảng CongTy

```
SELECT * FROM CongTy CT
```

b) Mệnh đề SELECT

Danh sách chọn trong câu lệnh SELECT được sử dụng để chỉ định các trường các biểu thức cần hiển thị trong các cột của kết quả truy vấn. Các trường các biểu thức được chỉ định ngay sau từ khoá SELECT và phân cách nhau bởi dấu phẩy. Sử dụng danh sách chọn trong câu lệnh SELECT bao gồm các trường hợp sau:

b.1 Chọn tất cả các cột trong bảng

Khi cần hiển thị tất cả các trường trong bảng, sử dụng ký tự * trong danh sách chọn thay vì phải liệt kê danh sách tất cả các cột. Trong trường hợp này, các cột được hiển thị trong kết quả truy vấn sẽ tuân thủ theo thứ tự mà chúng đã được tạo ra khi bảng được định nghĩa.

Ví dụ: Câu lệnh dùng để liệt kê danh sách các mặt hàng

```
SELECT * FROM HangHoa
```

b.2 Liệt kê tên cột trong danh sách chọn

Trong trường hợp cần chỉ định cụ thể các cột cần hiển thị trong kết quả truy vấn, ta chỉ định danh sách các tên cột trong danh sách chọn. Thứ tự của các cột trong kết quả truy vấn tuân theo thứ tự của các trường trong danh sách chọn.

Ví dụ: Liệt kê danh sách các mặt hàng gồm các thuộc tính sau: MaHang, TenHang, DonViTinh

```
SELECT MaHang, TenHang, DonViTinh
FROM HangHoa
```

Chú ý: Nếu truy vấn được thực hiện trên nhiều bảng và các bảng có các trường trùng tên thì tên của những trường này nếu xuất hiện trong danh sách chọn phải được viết dưới dạng:

```
<Tên_bảng>.<Tên_trường>
```

Ví dụ: Liệt kê danh sách các công ty đã cung cấp hàng hoá

```
SELECT CC.MaCT, TenCongTy, DiaChi
FROM CongTy CT, CungCap CC
WHERE CT.MaCT=CC.MaCT
```

b.3 Thay đổi tiêu đề các cột

Trong kết quả truy vấn, tiêu đề của các cột mặc định sẽ là tên của các trường tương ứng trong bảng. Tuy nhiên, để tiêu đề trở nên thân thiện hơn, ta có thể đổi lại tên tiêu đề của các cột. Để đặt tiêu đề cho một cột nào đó, ta sử dụng cách viết:

```
<tên_trường > AS <tiêu_đề_cột>
```

Ví dụ: Cho biết mã và tên của các công ty.

```
SELECT MaCongTy, TenCongTy AS Tên_Công_Ty
FROM CongTy
```

b.4 Hằng và biểu thức trong danh sách chọn

Ngoài danh sách trường, trong danh sách chọn của câu lệnh SELECT còn có thể sử dụng các biểu thức. Mỗi biểu thức trong danh sách chọn trở thành một cột trong kết quả truy vấn.

Ví dụ: Câu lệnh dưới đây cho biết mã công ty, mã hàng, số lượng, đơn giá và thành tiền của các mặt hàng đã được các công ty cung cấp.

```
SELECT MaCongTy, MaHang, SoLuong, DonGia, SoLuong*DonGia AS ThanhTien
FROM CungCap
```

b.4 Loại bỏ các bản ghi trùng nhau trong kết quả truy vấn

Trong kết quả của truy vấn có thể xuất hiện các dòng dữ liệu trùng nhau. Để loại bỏ các dòng này, ta chỉ định thêm từ khoá DISTINCT ngay sau từ khoá SELECT.

Ví dụ: Cho biết thông tin về mã của từng loại mặt hàng đã được cung cấp.

```
SELECT DISTINCT MaHang FROM CungCap
```

b.5 Giới hạn số lượng bản ghi trong kết quả truy vấn

Kết quả của truy vấn được hiển thị thường sẽ là tất cả các dòng dữ liệu truy vấn được. Trong trường hợp cần hạn chế số lượng các dòng

c) Mệnh đề điều kiện WHERE

Mệnh đề WHERE trong câu lệnh SELECT được sử dụng nhằm xác định các điều kiện đối với việc truy xuất dữ liệu. Sau mệnh đề WHERE là một biểu thức logic và chỉ những dòng dữ liệu nào thoả mãn điều kiện được chỉ định mới được hiển thị trong kết quả truy vấn.

Ví dụ: Câu lệnh dưới đây hiển thị mã số của các nhà cung cấp đã cung cấp mặt hàng có mã là H2

```
SELECT MaCongTy
FROM CungCap
WHERE MaHang='H2'
```

Trong mệnh đề WHERE thường sử dụng

Các toán tử kết hợp điều kiện (AND, OR)

Các toán tử so sánh.

Toán tử phạm vi và toán tử tập hợp

Các giá trị NULL

c.1 Các toán tử so sánh

Toán tử	Ý nghĩa
<u>= (Equals)</u>	Ngang bằng
<u>> (Greater Than)</u>	Lớn hơn
<u>< (Less Than)</u>	Nhỏ hơn
<u>>= (Greater Than or Equal To)</u>	Lớn hơn hoặc bằng
<u><= (Less Than or Equal To)</u>	Nhỏ hơn hoặc bằng
<u><> (Not Equal To)</u>	Không bằng
<u>!= (Not Equal To)</u>	Không bằng
<u>!< (Not Less Than)</u>	Không nhỏ hơn

[!> \(Not Greater Than\)](#)

Không lớn hơn

c.2 Toán tử phạm vi (Range Operator): [NOT] BETWEEN a AND b

Để kiểm tra xem giá trị dữ liệu nằm trong (ngoài) một khoảng nào đó, ta sử dụng toán tử [NOT] BETWEEN như sau:

Cách sử dụng	Ý nghĩa
giá_trị BETWEEN a AND b	$a \leq \text{giá_trị} \leq b$
giá_trị NOT BETWEEN a AND b	$(\text{giá_trị} < a) \text{ AND } (\text{giá_trị} > b)$

Ví dụ: Cho cơ sở dữ liệu quản lý điểm gồm các bảng sau:

- + LopHoc(MaLop, TenLop)
- + SinhVien(MaSV, HoDem, Ten, NgaySinh, GioiTinh, DiaChi, MaLop)
- + MonHoc(MaMon, TenMon, SoDvht)
- + Diem(MaSV, MaMon, DiemThi)

Câu lệnh dưới đây cho biết danh sách các môn học có số đơn vị học trình nằm trong khoảng từ 4 đến 5

```
SELECT *  
FROM MonHoc  
WHERE SoDvht Between 4 And 5
```

Câu lệnh dưới đây cho biết danh sách các môn học có số đơn vị học trình không nằm trong khoảng từ 4 đến 5

```
SELECT *  
FROM MonHoc  
WHERE SoDvht NOT Between 4 And 5
```

c.3 Toán tử tập hợp (IN và NOT IN)

Toán tử IN được sử dụng khi ta cần chỉ định điều kiện tìm kiếm dữ liệu cho câu SELECT là một danh sách các giá trị. Sau IN (hoặc NOT IN) có thể là một danh sách các giá trị hoặc là một câu lệnh SELECT khác.

Ví dụ: Để biết danh sách các môn học có số đơn vị học trình là 3, 4 hoặc 5 thay vì sử dụng câu lệnh

```
SELECT * FROM MonHoc  
WHERE SoDvht =3 OR SoDvht = 4 OR SoDvht = 5
```

Ta có thể sử dụng câu lệnh

```
SELECT * FROM MonHoc  
WHERE SoDvht IN (3,4,5)
```

c.4 Toán tử LIKE và các ký tự đại diện

Toán tử LIKE (hoặc NOT LIKE) sử dụng trong câu lệnh SELECT nhằm mô tả khuôn dạng của cần tìm kiếm. Chúng thường kết hợp với các ký tự đại diện sau đây:

Dấu phân trăm (%): Chỉ một chuỗi các ký tự bất kỳ.

Dấu gạch dưới (_): Chỉ một ký tự đơn bất kỳ

Ví dụ: Cho biết họ và tên của các sinh viên có họ đệm là Dương

```
SELECT HoDem, Ten FROM SinhVien  
WHERE HoDem = 'Dương%'
```

c.5 Giá trị NULL

Trong mệnh đề WHERE, để kiểm tra giá trị của một cột có giá trị NULL hay không ta sử dụng cách viết:

WHERE tên_cột IS NULL hoặc WHERE tên_cột IS NOT NULL

c.6 Toán tử SOME, ANY, ALL

+ Toán tử SOME, ANY: Chúng được sử dụng kết hợp thêm với các phép toán so sánh căn bản như: =, >, >=, <, <=, !=, !>, !<... Trả về TRUE nếu một vài biểu thức thỏa mãn

+ Toán tử ALL: Chúng được sử dụng kết hợp thêm với các phép toán so sánh căn bản như: =, >, >=, <, <=, !=, !>, !<... Trả về TRUE nếu tất cả các biểu thức đều thỏa mãn

d) Sắp xếp kết quả truy vấn

Mặc định các dòng dữ liệu trong kết quả của câu truy vấn tuân theo thứ tự của chúng trong bảng dữ liệu hoặc được sắp xếp theo chỉ mục (nếu trên bảng có chỉ mục). Trong trường hợp muốn dữ liệu được sắp xếp theo chiều tăng hoặc giảm của giá trị của một hoặc nhiều trường, ta sử dụng thêm mệnh đề ORDER BY trong câu lệnh SELECT.

Sau ORDER BY là danh sách các cột cần sắp xếp (tối đa là 16 cột). Dữ liệu được sắp xếp có thể theo chiều tăng (ASC) hoặc giảm (DESC), mặc định là sắp xếp theo chiều tăng.

Ví dụ: Câu lệnh dưới đây hiển thị danh sách các môn học và sắp xếp theo chiều giảm dần của số đơn vị học trình

```
SELECT * FROM MonHoc
ORDER BY SoDvht DESC
```

Chú ý: Nếu sau ORDER BY có nhiều cột thì việc sắp xếp dữ liệu sẽ được ưu tiên theo chiều từ trái qua phải.

Ví dụ: Liệt kê danh sách sinh viên và sắp xếp theo tên sinh viên theo Alphabet, nếu trùng tên thì sắp theo giới tính

```
SELECT * FROM SinhVien
ORDER BY Ten, GioiTinh
```

e) Phép kết nối

Khi cần thực hiện một yêu cầu truy vấn dữ liệu từ hai hay nhiều bảng, ta phải sử dụng đến phép kết nối

Để thực hiện được một phép nối, cần phải xác định được những yếu tố sau:

Những cột nào cần hiển thị trong kết quả truy vấn.

Những bảng nào có tham gia vào truy vấn.

Điều kiện để thực hiện phép nối giữa các bảng dữ liệu là gì?

Trong các yếu tố kể trên, việc xác định chính xác điều kiện để thực hiện phép nối giữa các bảng đóng vai trò quan trọng nhất. Trong đa số các trường hợp, điều kiện của phép nối được xác định nhờ vào mối quan hệ giữa các bảng cần phải truy xuất dữ liệu. Thông thường, đó là điều kiện bằng nhau giữa khoá chính và khoá ngoài của hai bảng có quan hệ với nhau.

Ví dụ: Câu lệnh dưới đây hiển thị danh sách các sinh viên với các thông tin: Mã sinh viên, họ đệm và tên, mã lớp và tên lớp

```
SELECT MaSV, HoDem, Ten, Lop.MaLop, TenLop
FROM SinhVien, Lop
WHERE SinhVien.MaLop = Lop.MaLop
```

Trong câu lệnh trên, các bảng tham gia vào truy vấn bao gồm: SinhVien và Lop.

Điều kiện để thực hiện phép kết nối giữa hai bảng là điều kiện sau:

SinhVien.MaLop = Lop.MaLop

Chú ý:

Tên của một số cột nào đó trong các bảng có tham gia vào truy vấn. Nếu tên cột trong các bảng trùng tên nhau thì tên cột phải được viết dưới dạng: **Tên_bảng.tên_cột**

Dấu sao (*) được sử dụng trong danh sách chọn khi cần hiển thị tất cả các cột của các bảng tham gia truy vấn.

Trong trường hợp cần hiển thị tất cả các cột của một bảng nào đó, ta sử dụng cách viết: **tên_bảng.***

Ví dụ: Liệt kê danh sách các sinh viên tham gia học môn 'Cơ sở dữ liệu'
SELECT SinhVien.MaSV, HoDem, Ten, MaLop, NgaySinh, GioiTinh, DiaChi
FROM SinhVien, MonHoc, Diem
WHERE SinhVien.MaMon = Diem.MaMon AND MonHoc.MaMon=Diem.MaMon
AND TenMon='Cơ sở dữ liệu'

hoặc viết dưới dạng ngắn gọn hơn:

SELECT SinhVien.*
FROM SinhVien, MonHoc, Diem
WHERE SinhVien.MaMon = Diem.MaMon AND MonHoc.MaMon=Diem.MaMon
AND TenMon='Cơ sở dữ liệu'

f) Thống kê dữ liệu với GROUP BY

Mệnh đề GROUP BY sử dụng trong câu lệnh SELECT nhằm phân hoạch các dòng dữ liệu trong bảng thành các nhóm dữ liệu và trên mỗi nhóm dữ liệu thực hiện tính toán các giá trị thống kê như tính tổng, tính giá trị trung bình...

Các hàm nhóm được sử dụng để tính giá trị thống kê cho toàn bảng hoặc trên mỗi nhóm dữ liệu. Chúng có thể được sử dụng như là các cột trong danh sách chọn của câu lệnh SELECT hoặc xuất hiện trong mệnh đề HAVING, nhưng không được phép xuất hiện trong mệnh đề WHERE

SQL cung cấp các hàm nhóm dưới đây:

Hàm nhóm	Chức năng
SUM(biểu_thức)	Tính tổng các giá trị
AVG(biểu_thức)	Tính trung bình của các giá trị
COUNT(biểu_thức)	Đếm số các giá trị trong biểu thức
COUNT(*)	Đếm số các dòng được chọn
MAX(biểu_thức)	Tính giá trị lớn nhất
MIN(biểu_thức)	Tính giá trị nhỏ nhất

Trong đó:

Hàm SUM, AVG chỉ làm việc với các biểu thức số

Hàm SUM, AVG, COUNT, MIN và MAX bỏ qua các giá trị NULL khi tính toán.

Hàm COUNT(*) không bỏ qua các giá trị NULL

f.1 Thống kê trên toàn bộ dữ liệu

Khi cần tính toán giá trị thống kê trên toàn bộ dữ liệu, ta sử dụng các hàm nhóm trong danh sách chọn của câu lệnh SELECT. Trong trường hợp này, trong danh sách chọn không được sử dụng bất kỳ một tên cột hay biểu thức nào ngoài các hàm gộp.

Ví dụ: Để thống kê trung bình điểm thi của tất cả các môn học ta sử dụng câu lệnh như sau:

```
SELECT AVG(DiemThi) FROM Diem
```

f.2 Thống kê dữ liệu trên các nhóm

Trong trường hợp cần thực hiện tính toán các giá trị thống kê trên các nhóm dữ liệu, ta sử dụng mệnh đề GROUP BY để phân hoạch dữ liệu vào trong các nhóm. Các hàm nhóm được sử dụng sẽ thực hiện thao tác tính toán trên mỗi nhóm và cho biết giá trị thống kê theo các nhóm dữ liệu.

Ví dụ: Câu lệnh dưới đây cho biết số sinh viên của mỗi lớp

```
SELECT Lop.MaLop, TenLop, COUNT(MaSV) AS SiSo
```

```
FROM Lop, SinhVien
WHERE Lop.MaLop = SinhVien. MaLop
GROUP BY Lop.MaLop, TenLop
```

Chú ý:

Biểu thức nào điều khiển việc phân nhóm dữ liệu thì các biểu thức đó phải được liệt kê sau mệnh đề GROUP BY.

Trong trường hợp danh sách chọn của câu lệnh SELECT có các hàm nhóm và những biểu thức không phải là đối số của các hàm nhóm thì những biểu thức này phải được liệt kê đầy đủ trong mệnh đề GROUP BY, nếu không câu lệnh sẽ không hợp lệ

Ví dụ: Dưới đây là một câu lệnh sai do thiếu trường TenLop sau mệnh đề GROUP BY

```
SELECT Lop.MaLop, TenLop, COUNT(MaSV) AS SiSo
FROM Lop, SinhVien
WHERE Lop.MaLop = SinhVien. MaLop
GROUP BY Lop.MaLop
```

g) Mệnh đề điều kiện đối với hàm nhóm - HAVING

Mệnh đề HAVING là mệnh đề đặt điều kiện lên các nhóm dữ liệu. Mệnh đề HAVING thường không thực sự có nghĩa nếu như không sử dụng kết hợp với mệnh đề GROUP BY. Một điểm khác biệt giữa HAVING và WHERE là trong điều kiện của WHERE không được có các hàm nhóm trong khi HAVING lại cho phép sử dụng các hàm nhóm trong điều kiện của mình.

Ví dụ: Đưa ra danh sách sinh viên có trung bình điểm thi lớn hơn hoặc bằng 5.

```
SELECT SinhVien.MaSV, HoDem, Ten, SUM(DiemThi*SoDvht)/SUM(SoDvht)
FROM SinhVien, MonHoc, Diem
WHERE SinhVien.MaMon = Diem.MaMon AND
      MonHoc.MaMon=Diem.MaMon
GROUP BY SinhVien.MaSV, HoDem, Ten
HAVING SUM(DiemThi*SoDvht)/SUM(SoDvht) >=5
```

h) Truy vấn con (Subquery)

Truy vấn con là một câu lệnh SELECT được lồng vào bên trong một câu lệnh SELECT, INSERT, UPDATE, DELETE hoặc bên trong một truy vấn con khác. Loại truy vấn này được sử dụng để biểu diễn cho những truy vấn trong đó điều kiện truy vấn dữ liệu cần phải sử dụng đến kết quả của một truy vấn khác.

+ Tìm tên các công ty đã cung cấp mặt hàng H2

```
SELECT TenCongTy
FROM CongTy, CungCap
WHERE (CongTy.MaCongTy = CungCap.MaCongTy)
AND (MaHang = 'H2')
```

Hoặc có thể viết như sau:

```
SELECT TenCongTy
FROM CongTy
WHERE MaCongTy IN (SELECT MaCongTy
                   FROM CungCap
                   WHERE MaHang = 'H2')
```

Ví dụ: Câu lệnh sau liệt kê tên các công ty chưa từng cung cấp mặt hàng nào

```
SELECT TenCongTy
FROM CongTy
WHERE MaCongTy NOT IN (SELECT MaCongTy
```

FROM CungCap)

Hoặc có thể viết như sau:

```
SELECT *
FROM CongTy
WHERE MaCongTy <> ALL (SELECT MaCongTy FROM CungCap)
+ Tìm những công ty cung cấp ít nhất một mặt hàng màu đỏ
SELECT TenCongTy
FROM CongTy
WHERE MaCongTy IN (SELECT MaCongTy
FROM CungCap
WHERE MaHang IN (SELECT MaHang
FROM HangHoa
```

WHERE màu = “đỏ”))

+ Câu lệnh sau liệt kê tên các công ty chưa từng cấp mặt hàng có mã là H1.

```
SELECT TenCongTy
FROM CongTy
WHERE EXISTS (SELECT MaCongTy
FROM CungCap CC
WHERE CC.MaCongTy=CongTy.MaCongTy
AND MaHang <> 'H1')
```

3.2 Truy vấn sử dụng phép nối bảng

Trong hầu hết các câu lệnh truy vấn, kết quả mà ta mong muốn đều có liên quan đến một hoặc nhiều bảng (VIEW) khác nhau. Trong trường hợp như vậy ta cần sử dụng mệnh đề JOIN để kết hợp dữ liệu trên hai hay nhiều bảng dữ liệu lại với nhau.

Có 2 kiểu kết nối sau: Kết nối theo chiều ngang và kết nối theo chiều dọc

Kết nối theo chiều ngang: INNER JOIN (Kết nối tự nhiên), LEFT JOIN (Kết nối trái), RIGHT JOIN (Kết nối phải), FULL JOIN, CROSS JOIN (Phép Tích đề các)

Kết nối theo chiều dọc: Sử dụng phép toán UNION

3.2.1 Kết nối theo chiều ngang

```
SELECT [ ALL | DISTINCT ] < danh_sách_chọn >
FROM < Tên_bảng_thứ_nhất >
INNER JOIN | LEFT JOIN | RIGHT JOIN | FULL JOIN
< Tên_bảng_thứ_hai >
ON < Biểu_thức_điều_kiện_kết_nối_bảng >
[ INNER JOIN | LEFT JOIN | RIGHT JOIN | FULL JOIN
< Tên_bảng_thứ_3 >
ON < Biểu_thức_điều_kiện_kết_nối_bảng > ] [ ...n ]
[ WHERE < Biểu_thức_điều_kiện_chọn > ]
[ GROUP BY < danh_sách_cột > ]
[ HAVING < điều_kiện > ]
[ ORDER BY < tên_cột > [ ASC | DESC ] [ ,...n ] ]
```

Giải thích:

+ Nếu lệnh truy vấn sử dụng mệnh đề INNER JOIN (kết nối tự nhiên) để kết hợp hai hay nhiều bảng dữ liệu lại với nhau thì những bản ghi nào được đưa ra kết quả của câu truy vấn nếu bộ giá trị của nó thỏa mãn biểu thức điều kiện nối bảng.

+ Nếu lệnh truy vấn sử dụng mệnh đề LEFT JOIN (kết nối trái) để kết hợp hai hay nhiều bảng dữ liệu lại với nhau ta được kết quả sau: Những bản ghi của bảng thứ nhất (bảng

bên trái) so sánh được với các bản ghi của bảng thứ hai (dựa vào biểu thức điều kiện nối bảng) và những bản ghi của bảng thứ nhất không so sánh được với các bản ghi của bảng thứ hai (bảng bên phải) khi đó phần giá trị của các thuộc tính có trong bảng thứ 2 sẽ nhận giá trị NULL.

+ Nếu lệnh truy vấn sử dụng mệnh đề RIGHT JOIN (kết nối phải) để kết hợp hai hay nhiều bảng dữ liệu lại với nhau ta được kết quả sau: Những bản ghi của bảng thứ hai (bảng bên phải) so sánh được với các bản ghi của bảng thứ nhất (dựa vào biểu thức điều kiện nối bảng) và những bản ghi của bảng thứ hai không so sánh được với các bản ghi của bảng thứ nhất (bảng bên trái) khi đó phần giá trị của các thuộc tính có trong bảng thứ nhất sẽ nhận giá trị NULL.

+ Khi không cần quan tâm đến dữ liệu hiện ra cho dù dữ liệu bảng bên trái có tồn tại tương ứng dữ liệu bảng bên phải hay không (hoặc ngược lại) ta dùng FULL JOIN.

Ví dụ: Cho 2 bảng dữ liệu sau:

+ Bảng **LopHoc**

MaLop	TenLop
1	K2A
2	K2B
3	K2C

+ Bảng **SinhVien** để lưu danh sách sinh viên

MaSV	HoTen	GioiTinh	MaLop
SV01	Nguyễn Thị Hồng	0	1
SV02	Trần Hùng	1	2
SV03	Phạm Văn Tân	1	1

+ **Sử dụng phép kết nối tự nhiên**

SELECT SinhVien.*

FROM LopHoc

INNER JOIN SinhVien

ON LopHoc.MaLop=SinhVien.MaLop

Kết quả ta được bảng dữ liệu sau:

MaSV	HoTen	GioiTinh	MaLop
SV01	Nguyễn Thị Hồng	0	1
SV03	Phạm Văn Tân	1	1
SV02	Trần Hùng	1	2

+ **Sử dụng phép kết nối trái**

SELECT LopHoc.*, MaSV, HoTen

FROM LopHoc

LEFT JOIN SinhVien

ON LopHoc.MaLop=SinhVien.MaLop

Kết quả ta được bảng dữ liệu sau:

MaLop	TenLop	MaSV	HoTen
1	K2A	SV01	Nguyễn Thị Hồng
1	K2A	SV03	Phạm Văn Tân
2	K2B	SV02	Trần Hùng

3	K2C	NULL	NULL
---	-----	------	------

+ Sử dụng phép kết nối phải

```
SELECT LopHoc.*, MaSV, HoTen
FROM LopHoc
RIGHT JOIN SinhVien
ON LopHoc.MaLop=SinhVien.MaLop
```

+ Sử dụng phép kết nối FULL JOIN

```
SELECT LopHoc.*, MaSV, HoTen
FROM LopHoc
FULL JOIN SinhVien
ON LopHoc.MaLop=SinhVien.MaLop
```

+ Sử dụng phép Tích Đề Các - CROSS JOIN

```
SELECT LopHoc.*,MaSV, HoTen, GioiTinh
FROM LopHoc
CROSS JOIN SinhVien
```

Kết quả: Bảng kết quả trả về 9 bản ghi sau

MaLop	TenLop	MaSV	HoTen	GioiTinh
1	K2A	SV01	Nguyễn Thị Hồng	0
1	K2A	SV02	Trần Hùng	1
1	K2A	SV03	Phạm Văn Tân	1
2	K2B	SV01	Nguyễn Thị Hồng	0
2	K2B	SV02	Trần Hùng	1
2	K2B	SV03	Phạm Văn Tân	1
3	K2C	SV01	Nguyễn Thị Hồng	0
3	K2C	SV02	Trần Hùng	1
3	K2C	SV03	Phạm Văn Tân	1

3.3.2 Kết nối theo chiều dọc – Dùng phép hợp (UNION)

UNION là phép toán dùng để nối hai hay nhiều câu truy vấn lại với nhau

Ví dụ

```
SELECT *
FROM SinhVien
WHERE MaLop=1
UNION
SELECT *
FROM SinhVien
WHERE MaLop=2
```

Một số chú ý khi sử dụng UNION

Tất cả những truy vấn trong UNION phải cùng số cột hay trường. Nếu truy vấn thứ nhất có hai cột thì truy vấn thứ hai được UNION cũng có hai cột tương tự.

Tiêu đề cột trong kết quả của phép UNION sẽ là tiêu đề cột được chỉ định trong truy vấn đầu tiên.

Kiểu dữ liệu của các cột của truy vấn 2 phải tương thích với kiểu dữ liệu các cột tương ứng trong truy vấn thứ nhất.

Khi các kiểu dữ liệu khác nhau được kết hợp trong câu lệnh UNION, chúng sẽ được chuyển sang kiểu dữ liệu cao hơn (nếu có thể).

Phép toán UNION không thể sử dụng trong câu lệnh CREATE VIEW

3.3 Bổ sung dữ liệu.

Dữ liệu trong các bảng được thể hiện dưới các dòng (bản ghi). Để bổ sung thêm các dòng dữ liệu vào một bảng, ta sử dụng câu lệnh INSERT. SQL cung cấp các cách dưới đây để thực hiện thao tác bổ sung dữ liệu cho bảng:

Bổ sung từng dòng dữ liệu với mỗi câu lệnh INSERT.

Bổ sung nhiều dòng dữ liệu bằng cách truy xuất dữ liệu từ các bảng dữ liệu khác.

3.3.1 Bổ sung từng dòng dữ liệu với lệnh INSERT

```
INSERT INTO <tên_bảng>[(<danh_sách_tên_cột>)]  
VALUES (<danh_sách_các_giá_trị>)
```

Trong câu lệnh INSERT, danh_sách_tên_cột ngay sau tên bảng không cần thiết phải chỉ định nếu giá trị các trường của bản ghi mới được chỉ định đầy đủ trong danh sách các giá trị. Trong trường hợp này, thứ tự các giá trị trong danh sách trị phải bằng số lượng các trường của bảng cần bổ sung dữ liệu cũng như phải tuân theo đúng thứ tự của các trường như khi bảng được định nghĩa.

Ví dụ: Câu lệnh dưới đây sẽ bổ sung thêm một công ty vào quan hệ congty

```
INSERT INTO CongTy  
VALUES (C12, 'Hoà bình', 600000000, 'Thái nguyên', '0280754379')
```

Trong trường hợp chỉ nhập giá trị cho một số cột trong bảng, ta phải chỉ định danh sách các cột cần nhập dữ liệu ngay sau tên bảng. Khi đó các cột không được nhập dữ liệu sẽ nhận giá trị mặc định (nếu có) hoặc nhận giá trị NULL (nếu cột cho phép nhận giá trị NULL). Nếu một cột không có giá trị mặc định và không chấp nhận giá trị NULL mà không được nhập dữ liệu, câu lệnh sẽ bị lỗi.

Ví dụ: Câu lệnh sau sẽ bổ sung một bản ghi mới cho bảng CongTy

```
INSERT INTO CongTy ( MaCongTy, TenCongTy, DiaChi, NganSach)  
VALUES (C13, 'Trần Anh', 'Hà nội', 500000000).
```

Khi đó trường SoDT sẽ nhận giá trị NULL.

Câu lệnh trên còn có thể viết như sau:

```
INSERT INTO CongTy  
VALUES (C13, 'Trần Anh', 500000000, 'Hà nội', NULL).
```

3.3.2 Bổ sung nhiều dòng dữ liệu từ bảng khác

Một cách sử dụng khác của câu lệnh INSERT được sử dụng để bổ sung nhiều dòng dữ liệu vào một bảng, các dòng dữ liệu này được lấy từ một bảng khác thông qua câu lệnh SELECT. Ở cách này, các giá trị dữ liệu được bổ sung vào bảng không được chỉ định tường minh mà thay vào đó là một câu lệnh SELECT truy vấn dữ liệu từ bảng khác.

Cú pháp của câu lệnh có dạng như sau:

```
INSERT INTO <tên_bảng>[(<danh_sách_tên_cột>)]  
<câu_lệnh_SELECT>
```

Ví dụ: Giả sử ta có bảng LuuCongTy bao gồm các trường MaCongTy, TenCongTy, NganSach, DiaChi. Câu lệnh dưới đây sẽ bổ sung vào bảng LuuCongTy các dòng dữ liệu có được từ câu truy vấn SELECT.

```
INSERT INTO LuuCongTy  
SELECT MaCongTy, TenCongTy, NganSach, DiaChi  
FROM CongTy WHERE DiaChi = 'Hà nội'
```

Ghi chú: Kết quả của câu lệnh SELECT phải có số cột bằng với số cột được chỉ định trong bảng đích và phải tương thích về cả kiểu dữ liệu.

3.4 Cập nhật dữ liệu

Câu lệnh UPDATE trong SQL được sử dụng để cập nhật dữ liệu trong các bảng. Câu lệnh này có cú pháp như sau:

```
UPDATE <tên_bảng>  
SET <tên_cột_1>=<biểu_thức_1> [, ..., <tên_cột_k> = <biểu_thức_k>]  
[WHERE <điều_kiện>]
```

Sau UPDATE là tên của bảng cần cập nhật dữ liệu. Một câu lệnh UPDATE có thể cập nhật dữ liệu cho nhiều cột bằng cách chỉ định danh sách tên cột và biểu thức tương ứng sau từ khoá SET. Mệnh đề WHERE trong câu lệnh UPDATE thường được sử dụng để chỉ định các dòng dữ liệu chịu tác động của câu lệnh (nếu không chỉ định, phạm vi tác động của câu lệnh là toàn bộ các dòng trong bảng)

Ví dụ: Cập nhật lại địa chỉ của công ty có mã là CT1

```
UPDATE TABLE CongTy  
SET DiaChi = 'Thái nguyên' WHERE MaCongTy ='CT1'
```

3.5 Xóa dữ liệu

Để xóa dữ liệu trong một bảng, ta sử dụng câu lệnh DELETE. Cú pháp của câu lệnh như sau:

```
DELETE FROM <tên_bảng>  
[WHERE <điều_kiện>]
```

Trong đó, tên của bảng cần xóa dữ liệu được chỉ định sau DELETE FROM. Mệnh đề WHERE trong câu lệnh được sử dụng để chỉ định điều kiện đối với các dòng dữ liệu cần xóa. Nếu câu lệnh DELETE không có mệnh đề WHERE thì toàn bộ các dòng trong bảng đều bị xóa. Mệnh đề FROM chỉ định danh sách các bảng có dữ liệu liên quan đến việc xóa dữ liệu.

+ Ví dụ: Câu lệnh sau sẽ xóa khỏi bảng CONGTY những công ty có địa chỉ ở Hà nội

```
DELETE FROM CongTy  
WHERE DiaChi = 'Hà nội'
```

Ví dụ: Xóa khỏi bảng CongTy danh sách các công ty đã cung cấp mặt có mã là H1

```
DELETE FROM CongTy  
WHERE MaCongTy In (SELECT DISTINCT MaCongTy  
FROM CungCap  
WHERE MaHang='H1')
```

+ Xóa khỏi bảng CongTy nhưng công ty chưa từng cung cấp mặt hàng nào.

```
DELETE FROM CongTy  
WHERE MaCongTy NOT IN (SELECT DISTINCT MaCongTy  
FROM CungCap)
```

+ Xóa tất cả các thông tin trong bảng CungCap: DELETE FROM CungCap

4 Variables (Biến) trong SQL Server

Biến trong T-SQL cũng có chức năng tương tự như trong các ngôn ngữ lập trình khác nghĩa là cần khai báo trước loại dữ liệu trước khi sử dụng. Biến được bắt đầu bằng dấu @ (Đối với các global variable thì có hai dấu @@)

4.1 Khai báo biến:

```
DECLARE @<Tên_biến> <Kiểu_dữ_liệu>[,  
@<Tên_biến> <Kiểu_dữ_liệu>[,  
@<Tên_biến> <Kiểu_dữ_liệu>]]
```

Ghi chú:

Khi khai báo nhiều biến, bạn không cần dùng lại từ khóa mà chỉ cần dùng dấu (,) để phân biệt.

Khi khai báo một biến, biến có giá trị ban đầu là NULL cho đến khi biến đó được gán giá trị khác.

4.2 Gán giá trị cho biến

SET @<Tên_biến>=<Giá_trị>|<biểu_thức>| (Câu lệnh SELECT)

Ghi chú:

Giá trị gán vào biến phải có kiểu dữ liệu tương tự với kiểu dữ liệu đã khai báo cho biến.

Không nhất thiết gán biến với một giá trị, đôi khi giá trị đó là kết quả của câu lệnh truy vấn SELECT.

Ví dụ: Gán biến bằng giá trị cụ thể

```
Declare @x int, @y int
Set @y = 5
Set @x = @y + 3
```

Ví dụ: Gán biến bằng giá trị của SELECT

```
Declare @x int
Set @x=(Select count(*) From SinhVien)
Print 'Tổng số sinh viên là' + Str(@x)
```

5. Phát biểu điều khiển

5.1. Phát biểu IF...ELSE

Cú pháp:

```
IF <biểu_thức_điều_kiện>
    <Lệnh_đơn>|<Khối_lệnh>
ELSE
    <Lệnh_đơn>|<Khối_lệnh>
```

Ghi chú:

Khối lệnh là một tập các câu lệnh của T-SQL, khối lệnh phải được bao trong cặp từ khóa BEGIN END

Khối lệnh là một tập các câu lệnh của T-SQL

VD: Giả sử có 2 bảng LOP và LOP1 có cấu trúc giống hệt nhau. Ta viết SP nhập dữ liệu dùng chung cho cả 2 bảng nhưng cần phân biệt bởi một biến cờ bằng cách sử dụng tham số @flag. Nếu @flag=1 thì thêm vào bảng LOP ngược lại thì thêm vào bảng LOP1

```
CREATE STORED PROC NHAP
@flag bit, @malop char(3), @tenlop nvarchar(20)
AS
    IF (@flag=1)
        INSERT INTO LOP
        VALUES (@malop,@tenlop)
    ELSE
        INSERT INTO LOP
        VALUES (@malop,@tenlop)
```

5.2 Phát biểu CASE

Cú pháp:

```
CASE <biểu_thức_vào>
WHEN <gia_tri_1> THEN <biểu_thức_1>
[...n]
[ELSE <biểu_thức_n+1>]
```

Ví: Sử dụng phát biểu CASE để tính ngày nhập hàng là ngày thứ mấy trong tuần.

```

SELECT SHDN, MAKHACH, NGAYNHAP, NGAYTHU=
CASE DATEPART(W,NGAYNHAP)
    WHEN 1 THEN 'CN'
    WHEN 2 THEN 'THU 2'
    WHEN 3 THEN 'THU 3'
    WHEN 4 THEN 'THU 4'
    WHEN 5 THEN 'THU 5'
    WHEN 6 THEN 'THU 6'
[ELSE 'THU 7']
    WHEN 7 THEN 'THU 7'
END
FROM NHAP

```

+ Cú pháp 2:

```

CASE
    WHEN <biểu thức điều kiện 1> THEN <biểu thức kết quả 1>
    WHEN <biểu thức điều kiện 2> THEN <biểu thức kết quả 2>
    .....
    WHEN <biểu thức điều kiện n> THEN <biểu thức kết quả n>
[ELSE <biểu thức kết quả_ELSE]

```

Ví dụ: Cho bảng Diem(SBD, HoTen,NS,GT,TongDiem) để lưu tổng hợp điểm thi của các thí sinh trong một đợt tuyển sinh. Hãy dựa vào bảng trên để đưa ra màn hình danh sách gồm các thuộc tính (SBD, HoTen,NS,GT,TongDiem, Do/Truot). Với Do/Truot (Đỗ/Trượt) được tính theo nguyên tắc nếu sinh viên nào có tổng điểm < 20 thì ghi 'Truot' ngược lại ghi 'Đo'

```

Select Diem.*, Do_Trut=CASE
    WHEN TongDiem <20 THEN 'Truot'
    ELSE 'Đo'

```

From Diem

5.3 Phát biểu WHILE

Phát biểu WHILE là phát biểu điều khiển lặp. Vòng lặp sẽ được thực hiện cho đến khi biểu thức logic trong While nhận giá trị False.

Cú pháp:

```

WHILE <Biểu thức logic>
    <Các phát biểu SQL>
[Begin
    <Các phát biểu SQL>
    [BREAK]
    <Các phát biểu SQL>
    [CONTINUE]
End]

```

Trong đó:

- Phát biểu BREAK dùng để thoát khỏi vòng lặp. Khi gặp BREAK thì quá trình xử lý sẽ bỏ qua các phát biểu còn lại chưa xử lý.

- Phát biểu CONTINUE đối ngược với phát biểu BREAK, nếu gặp phát biểu này thì quá trình xử lý sẽ quay lại đầu vòng lặp.

VD:

```

DECLARE @DEM INT

```

```

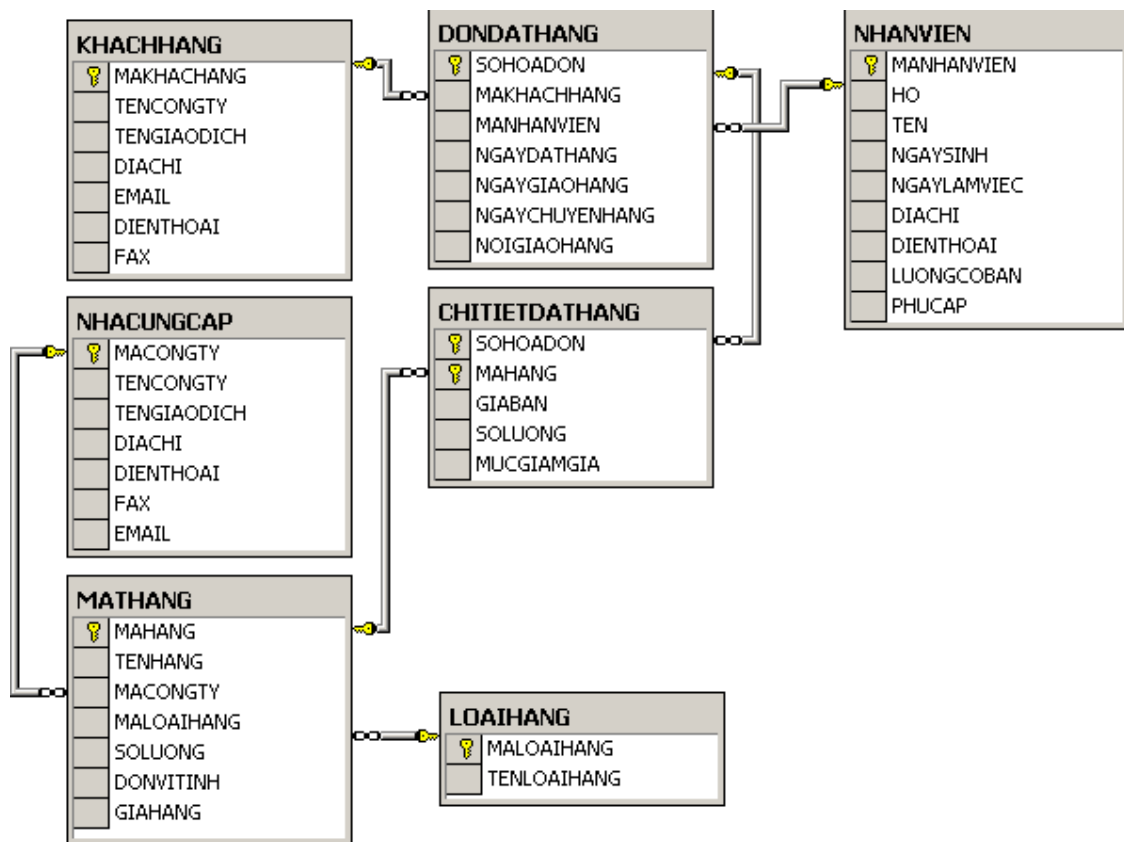
SET @DEM=1
WHILE @DEM<5
    BEGIN
        INSERT INTO THU1
        VALUES(@DEM,'THEM DONG MOI')
        SET @DEM=@DEM+1
    END

```

Bài tập thực hành của học viên

1. Tạo các CSDL sau:

Cơ sở dữ liệu dưới đây được sử dụng để quản lý công tác giao hàng trong một công ty kinh doanh. Các bảng trong cơ sở dữ liệu này được biểu diễn trong sơ đồ dưới đây:



Hình 4.1: Database Relationships CSDL QLNXHH

Trong đó:

- Bảng NHACUNGCAP lưu trữ dữ liệu về các đối tác cung cấp hàng cho công ty.
- Bảng MATHANG lưu trữ dữ liệu về các mặt hàng hiện có trong công ty.
- Bảng LOAIHANG phân loại các mặt hàng hiện có.
- Bảng NHANVIEN có dữ liệu là thông tin về các nhân viên làm việc trong công ty.
- Bảng KHACHHANG được sử dụng để lưu giữ thông tin về các khách hàng của công ty.
- Khách hàng đặt hàng cho công ty thông qua các đơn đặt hàng. Thông tin chung về các đơn đặt hàng được lưu trữ trong bảng DONDATHANG (Mỗi một đơn đặt hàng

phải do một nhân viên của công ty lập và do đó bảng này có quan hệ với bảng NHANVIEN)

- Thông tin chi tiết của các đơn đặt hàng (đặt mua mặt hàng gì, số lượng, giá cả,...) được lưu trữ trong bảng CHITIETDATHANG. Bảng này có quan hệ với hai bảng DONDATHANG và MATHANG.

1.1 Tạo các bảng và RBTV (ràng buộc toàn vẹn) cho CSDL Quản lý Nhập Xuất Hàng Hóa (QLNXHH) như sau:

a. Bảng Loại hàng:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
Maloai	Varchar	3	PK	
Tenloai	Varchar	30		

b. Bảng Hàng hóa:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
Mahang	Varchar	6	PK	
Maloai	Varchar	3	FK	
Tenhang	Varchar	50		

c. Bảng Nhà cung cấp:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
MaNCC	Varchar	10	PK	
TenNCC	Varchar	30		
Diachi	Varchar	100		
Dienthoai	Varchar	15		

d. Bảng Nhân viên:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
MaNV	Varchar	10	PK	
TenNV	Varchar	30		
UserName	Varchar	20		
Password	Varchar	20		

e. Bảng Hóa đơn nhập:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
MaHD	Varchar	10	PK	
Ngay	Datetime	8		
MaNCC	Varchar	10	FK	
MaNV	Varchar	10	FK	
Diengiai	Varchar	100		

f. Bảng Chi tiết hóa đơn nhập:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
MaHD	Varchar	10	PK,FK	
Mahang	Varchar	6	PK,FK	
Soluong	SmallInt	2		

Dongia	Float	8		
--------	-------	---	--	--

g. Bảng Hóa đơn xuất:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
MaHD	Varchar	10	PK	
Ngày	Datetime			
MaNV	Varchar	10	FK	
Diengiai	Varchar	100		

h. Bảng Chi tiết hóa đơn xuất:

Tên trường	Kiểu	Độ rộng	Khoá	Ghi chú
MaHD	Varchar	10	PK,FK	
Mahang	Varchar	6	PK,FK	
Soluong	SmallInt	2		
Dongia	Float	8		

1.2 Tạo các bảng và RBTV cho CSDL Quản lý Công Tác (QLCongTac) như sau:

Chucvu (macv char(5), tencv char(25))

Thanhpho (matp char(5), tentp char(25))

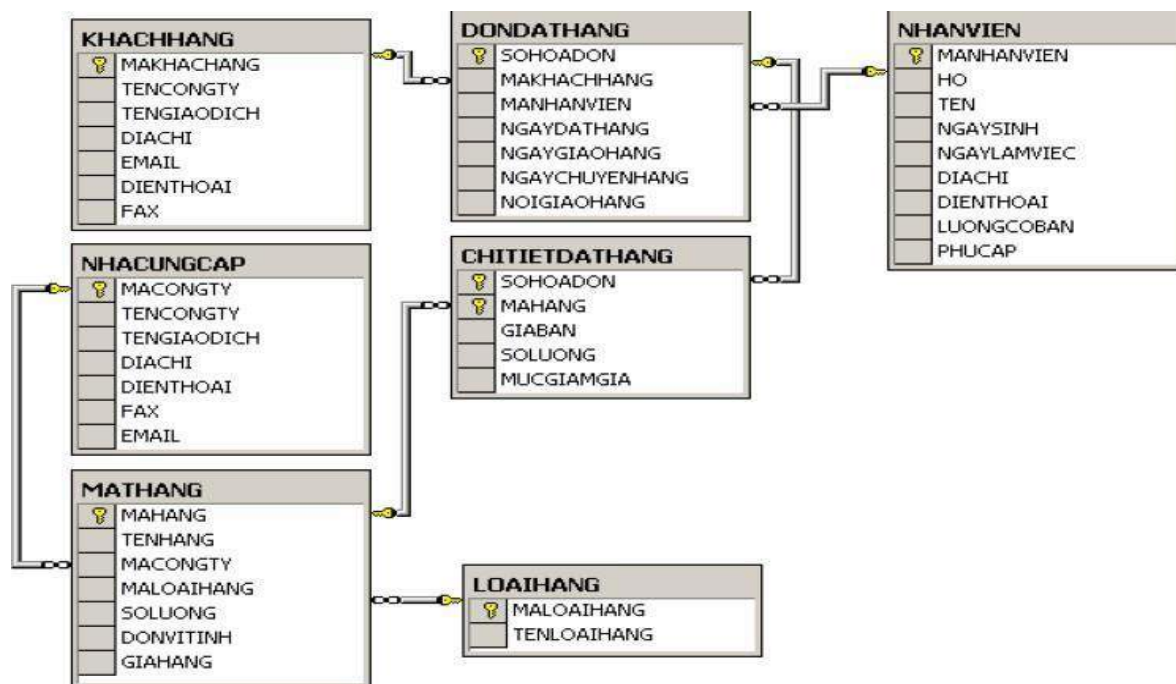
Phicongtac (macv char(5), matp char(5), phi_1_ngay int)

Nhanvien (manv char(5), tennv char(25), macv char(5))

Dotcongtac (madot int, tp_den char(5), songay int, ngaydi datetime)

Thamgiacongtac (madot int, manv char(5))

1.3/ Tạo các bảng và RBTV cho CSDL Quản lý Công Tác (QLCongTac) như sau:



Hình 4.2 Quan hệ CSDL QLCongTac

1.4/ Tạo các bảng và RBTV cho CSDL Quản lý Nước (QLNUOC) như sau:

- Tập tin dữ liệu: tên tập tin luận lý là qlnuoc_data1, tên tập tin vật lý là qlnuoc_data1.mdf, dung lượng ban đầu là 3MB, dung lượng tối đa là 10MB, lượng tăng trưởng là 1MB.
- Tập tin nhật ký: tên tập tin luận lý là qlnuoc_log1, tên tập tin vật lý là qlnuoc_log1.ldf, dung lượng ban đầu là 2MB, dung lượng tối đa là 5MB, lượng tăng trưởng là 1MB.

HO(Maho, Chuho, Diachi, Dienthoai)

GIANUOC(Ky, Gia)

CHISONUOC(Maho, Ky, Chiso)

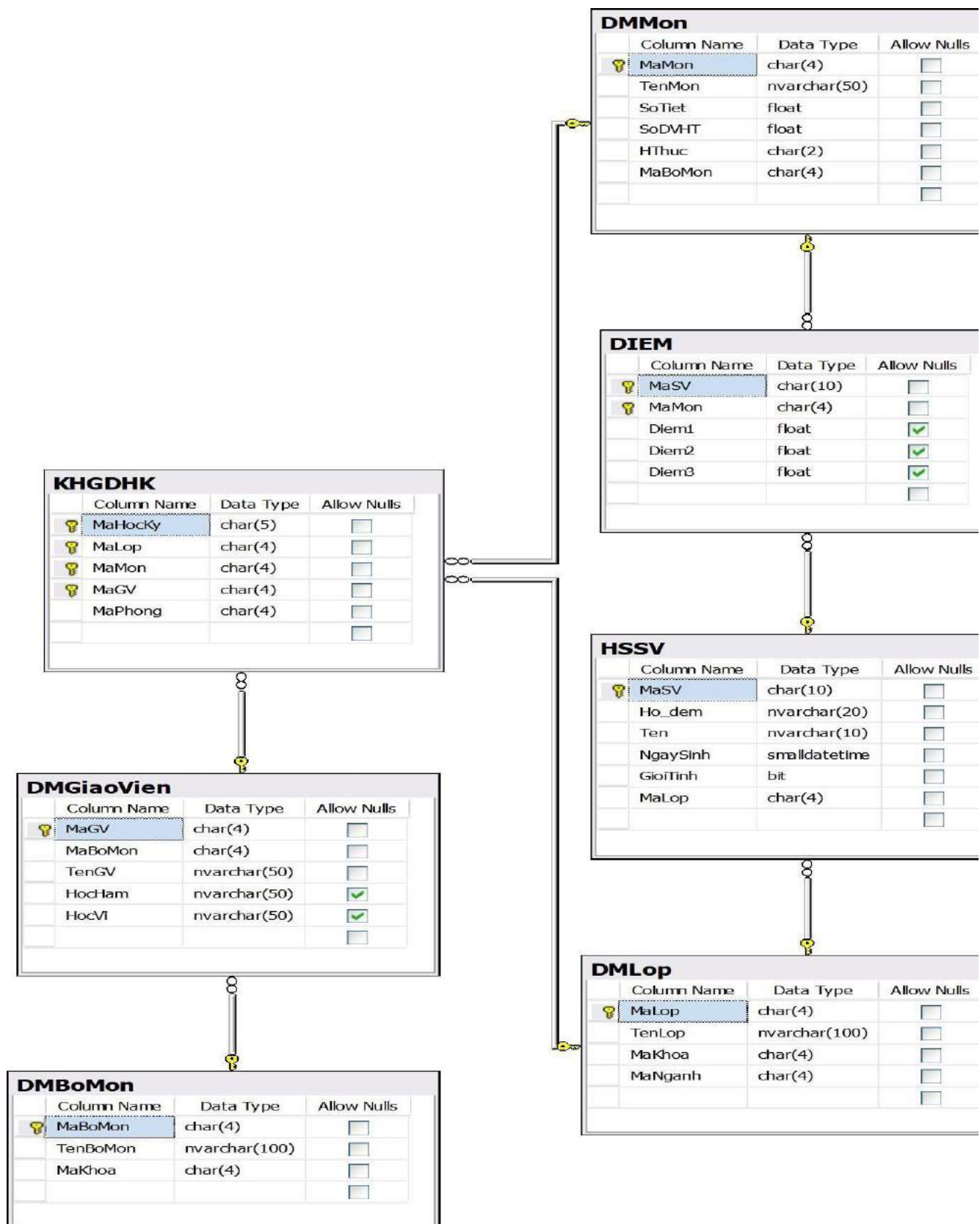
Với: Maho: 4 ký tự số

Ky: bigint. Dienthoai là không trùng nhau (duy nhất).

Chuho: not null. Chiso >= 0

Liên kết giữa các bảng phải có tính on Update, delete cascade

1.5/ Tạo các bảng và RBTV cho CSDL Quản lý Sinh viên (QLSV) như sau:



Hình 4.3 Quan hệ CSDL SINHVIEN

2. Sử dụng câu lệnh SELECT để viết các yêu cầu truy vấn dữ liệu sau đây:

- 2.1. Cho biết danh sách các đối tác cung cấp hàng cho công ty.
- 2.2. Mã hàng, tên hàng và số lượng của các mặt hàng hiện có trong công ty.
- 2.3. Họ tên và địa chỉ và năm bắt đầu làm việc của các nhân viên trong công ty.
- 2.4. Địa chỉ và điện thoại của nhà cung cấp có tên giao dịch VINAMILK là gì?

- 2.5. Cho biết mã và tên của các mặt hàng có giá lớn hơn 100000 và số lượng hiện có ít hơn 50.
- 2.6. Cho biết mỗi mặt hàng trong công ty do ai cung cấp.
- 2.7. Công ty Việt Tiến đã cung cấp những mặt hàng nào?
- 2.8. Loại hàng thực phẩm do những công ty nào cung cấp và địa chỉ của các công ty đó là gì?
- 2.9. Những khách hàng nào (tên giao dịch) đã đặt mua mặt hàng Sữa hộp XYZ của công ty?
- 2.10. Đơn đặt hàng số 1 do ai đặt và do nhân viên nào lập, thời gian và địa điểm giao hàng là ở đâu?
- 2.11. Hãy cho biết số tiền lương mà công ty phải trả cho mỗi nhân viên là bao nhiêu (lương = lương cơ bản + phụ cấp).
- 2.12. Trong đơn đặt hàng số 3 đặt mua những mặt hàng nào và số tiền mà khách hàng phải trả cho mỗi mặt hàng là bao nhiêu (số tiền phải trả được tính theo công thức $SOLUONG \times GIABAN - SOLUONG \times GIABAN \times MUCGIAMGIA/100$)
- 2.13. Hãy cho biết có những khách hàng nào lại chính là đối tác cung cấp hàng của công ty (tức là có cùng tên giao dịch).
- 2.14. Trong công ty có những nhân viên nào có cùng ngày sinh?
- 2.15. Những đơn đặt hàng nào yêu cầu giao hàng ngay tại công ty đặt hàng và những đơn đó là của công ty nào?
- 2.16. Cho biết tên công ty, tên giao dịch, địa chỉ và điện thoại của các khách hàng và các nhà cung cấp hàng cho công ty.
- 2.17. Những mặt hàng nào chưa từng được khách hàng đặt mua?
- 2.18. Những nhân viên nào của công ty chưa từng lập bất kỳ một hoá đơn đặt hàng nào?
- 2.19. Những nhân viên nào của công ty có lương cơ bản cao nhất?
- 2.20. Tổng số tiền mà khách hàng phải trả cho mỗi đơn đặt hàng là bao nhiêu?
- 2.21. Trong năm 2003, những mặt hàng nào chỉ được đặt mua đúng một lần.
- 2.22. Hãy cho biết mỗi một khách hàng đã phải bỏ ra bao nhiêu tiền để đặt mua hàng của công ty? đã phải bỏ ra bao nhiêu tiền để đặt mua hàng của công ty?
- 2.23. Mỗi một nhân viên của công ty đã lập bao nhiêu đơn đặt hàng (nếu nhân viên chưa hề lập một hoá đơn nào thì cho kết quả là 0)
- 2.24. Cho biết tổng số tiền hàng mà cửa hàng thu được trong mỗi tháng của năm 2003 (thời được gian tính theo ngày đặt hàng).
- 2.25. Hãy cho biết tổng số tiền lời mà công ty thu được từ mỗi mặt hàng trong năm 2003.
- 2.26. Hãy cho biết tổng số lượng hàng của mỗi mặt hàng mà công ty đã có (tổng số lượng hàng hiện có và đã bán).
- 2.27. Nhân viên nào của công ty bán được số lượng hàng nhiều nhất và số lượng hàng bán được của những nhân viên này là bao nhiêu?
- 2.28. Đơn đặt hàng nào có số lượng hàng được đặt mua ít nhất?
- 2.29. Số tiền nhiều nhất mà mỗi khách hàng đã từng bỏ ra để đặt hàng trong các đơn đặt hàng là bao nhiêu?
- 2.30. Mỗi một đơn đặt hàng đặt mua những mặt hàng nào và tổng số tiền mà mỗi đơn đặt hàng phải trả là bao nhiêu?
- 2.31. Hãy cho biết mỗi một loại hàng bao gồm những mặt hàng nào, tổng số lượng hàng của mỗi loại và tổng số lượng của tất cả các mặt hàng hiện có trong công ty là bao nhiêu?

2.32. Thống kê xem trong năm 2003, mỗi một mặt hàng trong mỗi tháng và trong cả năm bán được với số lượng bao nhiêu

Yêu cầu: Kết quả được hiển thị dưới dạng bảng, hai cột đầu là mã hàng và tên hàng, các cột còn lại tương ứng với các tháng từ 1 đến 12 và cả năm. Như vậy mỗi dòng trong kết quả cho biết số lượng hàng bán được mỗi tháng và trong cả năm của mỗi mặt hàng.

3. Sử dụng câu lệnh truy vấn hành động để thực hiện các yêu cầu sau:

- 3.1 Cập nhật lại giá trị trường NGAYCHUYENHANG của những bản ghi có NGAYCHUYENHANG chưa xác định (NULL) trong bảng DONDATHANG bằng với giá trị của trường NGAYDATHANG.
- 3.2 Tăng số lượng hàng của những mặt hàng do công ty VINAMILK cung cấp lên gấp đôi.
- 3.3 Cập nhật giá trị của trường NOIGIAOHANG trong bảng DONDATHANG bằng địa chỉ của khách hàng đối với những đơn đặt hàng chưa xác định được nơi giao hàng (giá trị trường NOIGIAOHANG bằng NULL).
- 3.4 Cập nhật lại dữ liệu trong bảng KHACHHANG sao cho nếu tên công ty và tên giao dịch của khách hàng trùng với tên công ty và tên giao dịch của một nhà cung cấp nào đó thì địa chỉ, điện thoại, fax và e-mail phải giống nhau.
- 3.5 Tăng lương lên gấp rưỡi cho những nhân viên bán được số lượng hàng nhiều hơn 100 trong năm 2003.
- 3.6 Tăng phụ cấp lên bằng 50% lương cho những nhân viên bán được hàng nhiều nhất.
- 3.7 Giảm 25% lương của những nhân viên trong năm 2003 không lập được bất kỳ đơn đặt hàng nào.
- 3.8 Giả sử trong bảng DONDATHANG có thêm trường SOTIEN cho biết số tiền mà khách hàng phải trả trong mỗi đơn đặt hàng. Hãy tính giá trị cho trường này.
- 3.9 Thực hiện các yêu cầu dưới đây bằng câu lệnh DELETE.
- 3.10 Xoá khỏi bảng NHANVIEN những nhân viên đã làm việc trong công ty quá 40 năm.
- 3.11 Xoá những đơn đặt hàng trước năm 2012 ra khỏi cơ sở dữ liệu.
- 3.12 Xoá khỏi bảng LOAIHANG những loại hàng hiện không có mặt hàng.
- 3.13 Xoá khỏi bảng KHACHHANG những khách hàng hiện không có bất kỳ đơn đặt hàng nào cho công ty.
- 3.14 Xoá khỏi bảng MATHANG những mặt hàng có số lượng bằng 0 và không được đặt mua trong bất kỳ đơn đặt hàng nào.

Hướng dẫn thực hiện

1. Tạo các CSDL sau:

Thực hiện tạo CSDL, tạo bảng, tạo khóa, tạo các ràng buộc, tạo quan cho các bảng, nhập dữ liệu: Tham khảo mục 2 ở bài 4 của giáo trình.

2. Sử dụng câu lệnh SELECT để viết các yêu cầu truy vấn dữ liệu sau đây:

- 2.1. SELECT macongty,tencongty,tengiaodich
FROM nhacungcap
- 2.2. SELECT mahang,tenhang,soluong
FROM mathang
- 2.3. SELECT ho,ten,year(ngaylamviec) AS namlamviec
FROM nhanvien

- 2.4. SELECT diachi,dienthoai
FROM nhacungcap
WHERE tengiaodich='VINAMILK'
- 2.5. SELECT mahang,tenhang
FROM mathang
WHERE giahang>100000 AND soluong<50
- 2.6. SELECT mahang,tenhang, nhacungcap.macongty,tencongty,tengiaodich
FROM mathang INNER JOIN nhacungcap
ON mathang.macongty=nhacungcap.macongty
- 2.7. SELECT mahang,tenhang
FROM mathang INNER JOIN nhacungcap
ON mathang.macongty=nhacungcap.macongty
WHERE tencongty='Việt Tiến'
- 2.8. SELECT DISTINCT nhacungcap.macongty,tencongty,diachi
FROM (loaihang INNER JOIN mathang
ON loaihang.maloaihang=mathang.maloaihang)
INNER JOIN nhacungcap
ON mathang.macongty=nhacungcap.macongty
WHERE tenloaihang='Thực phẩm'
- 2.9. SELECT DISTINCT tengiaodich
FROM ((mathang INNER JOIN chitietdathang
ON mathang.mahang=chitietdathang.mahang)
INNER JOIN dondathang
ON chitietdathang.sohoadon=dondathang.sohoadon)
INNER JOIN khachhang
ON dondathang.makhachhang=khachhang.makhachhang
WHERE tenhang='Sữa hộp'
- 2.10. SELECT dondathang.manhanvien,ho,ten, ngaygiaohang,noigiaohang
FROM nhanvien INNER JOIN dondathang
ON nhanvien.manhanvien=dondathang.manhanvien
WHERE sohoadon=1

Các câu còn lại làm tương tự: tham khảo mục 3 ở bài 4 của giáo trình.

3. Sử dụng câu lệnh truy vấn hành động để thực hiện các yêu cầu sau

- 3.1 UPDATE dondathang
SET ngaychuyenhang = ngaydathang
WHERE ngaychuyenhang IS NULL
- 3.2 UPDATE mathang
SET soluong=soluong*2
FROM nhacungcap
WHERE nhacungcap.macongty=mathang.macongty AND
tencongty='VINAMILK'
- 3.3 UPDATE dondathang
SET noigiaohang=diachi
FROM khachhang
WHERE dondathang.makhachhang=khachhang.makhachhang AND
noigiaohang IS NULL

Các câu còn lại làm tương tự: tham khảo mục 3 ở bài 4 của giáo trình.

Những trọng tâm cần chú ý:

- Trình bày cú pháp tạo mới CSDL, chú ý các tham số đi kèm CSDL
- Trình bày cú pháp về CSDL như: Xóa, chỉnh sửa tham số, nơi lưu trữ, dung lượng ban đầu cho các dữ liệu và logic
- Trình bày cú pháp trên Table như: Tạo mới, thêm, sửa xóa, khóa và các ràng buộc với lệnh ALTER
- Trình bày cú pháp truy vấn dữ liệu - SELECT
- Trình bày cú pháp truy vấn dữ liệu – hành động.
- Thực hiện đúng các thao tác trên lệnh T-SQL

Bài mở rộng và nâng cao

1. Phần mô tả đề bài

1.1 Cơ sở dữ liệu quản lý bán hàng (QLBH) gồm có các quan hệ sau

KHACHHANG (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANHSO, NGDK)

Tân từ: Quan hệ khách hàng sẽ lưu trữ thông tin của khách hàng thành viên gồm có các thuộc tính: mã khách hàng, họ tên, địa chỉ, số điện thoại, ngày sinh, ngày đăng ký và doanh số (tổng trị giá các hóa đơn của khách hàng thành viên này).

NHANVIEN (MANV, HOTEN, NGVL, SODT)

Tân từ: Mỗi nhân viên bán hàng cần ghi nhận họ tên, ngày vào làm, điện thoại liên lạc, mỗi nhân viên phân biệt với nhau bằng mã nhân viên.

SANPHAM (MASP, TENS, DVT, NUOCSX, GIA)

Tân từ: Mỗi sản phẩm có một mã số, một tên gọi, đơn vị tính, nước sản xuất và một giá bán.

HOADON (SOHD, NGHD, MAKH, MANV, TRIGIA)

Tân từ: Khi mua hàng, mỗi khách hàng sẽ nhận một hóa đơn tính tiền, trong đó sẽ có số hóa đơn, ngày mua, nhân viên nào bán hàng, trị giá của hóa đơn là bao nhiêu và mã số của khách hàng nếu là khách hàng thành viên.

CTHD (SOHD, MASP, SL)

Tân từ: Diễn giải chi tiết trong mỗi hóa đơn gồm có những sản phẩm gì với số lượng là bao nhiêu. (sơ đồ thể hiện mối quan hệ giữa các bảng)

1.2 Cấu trúc bảng như sau:

NHANVIEN

MANV	HOTEN	DTHOAI	NGVL
NV01	Nguyen Nhu Nhut	0927345678	13/4/2006
NV02	Le Thi Phi Yen	0987567390	21/4/2006
NV03	Nguyen Van B	0997047382	27/4/2006
NV04	Ngo Thanh Tuan	0913758498	24/6/2006
NV05	Nguyen Thi Truc Thanh	0918590387	20/7/2006

KHACHHANG

MAKH	HOTEN	DCHI	SODT	NGSINH	DOANHSO	NGDK
KH01	Nguyen Van A	731 Tran Hung Dao, Q5, TpHCM	08823451	22/10/1960	13,060,000	22/07/2006
KH02	Tran Ngoc Han	23/5 Nguyen Trai, Q5, TpHCM	0908256478	3/4/1974	280,000	30/07/2006
KH03	Tran Ngoc Linh	45 Nguyen Canh Chan, Q1, TpHCM	0938776266	12/6/1980	3,860,000	05/08/2006
KH04	Tran Minh Long	50/34 Le Dai Hanh, Q10, TpHCM	0917325476	9/3/1965	250,000	02/10/2006
KH05	Le Nhat Minh	34 Truong Dinh, Q3, TpHCM	08246108	10/3/1950	21,000	28/10/2006
KH06	Le Hoai Thuong	227 Nguyen Van Cu, Q5, TpHCM	08631738	31/12/1981	915,000	24/11/2006
KH07	Nguyen Van Tam	32/3 Tran Binh Trong, Q5, TpHCM	0916783565	6/4/1971	12,500	01/12/2006
KH08	Phan Thi Thanh	45/2 An Duong Vuong, Q5, TpHCM	0938435756	10/1/1971	365,000	13/12/2006
KH09	Le Ha Vinh	873 Le Hong Phong, Q5, TpHCM	08654763	3/9/1979	70,000	14/01/2007
KH10	Ha Duy Lap	34/34B Nguyen Trai, Q1, TpHCM	08768904	2/5/1983	67,500	16/01/2007

SANPHAM

MASP	TENSP	DVT	NUOCSX	GIA
BC01	But chi	cay	Singapore	3,000
BC02	But chi	cay	Singapore	5,000
BC03	But chi	cay	Viet Nam	3,500
BC04	But chi	hop	Viet Nam	30,000
BB01	But bi	cay	Viet Nam	5,000
BB02	But bi	cay	Trung Quoc	7,000
BB03	But bi	hop	Thai Lan	100,000
TV01	Tap 100 giay mong	quyen	Trung Quoc	2,500
TV02	Tap 200 giay mong	quyen	Trung Quoc	4,500
TV03	Tap 100 giay tot	quyen	Viet Nam	3,000
TV04	Tap 200 giay tot	quyen	Viet Nam	5,500
TV05	Tap 100 trang	chuc	Viet Nam	23,000
TV06	Tap 200 trang	chuc	Viet Nam	53,000
TV07	Tap 100 trang	chuc	Trung Quoc	34,000
ST01	So tay 500 trang	quyen	Trung Quoc	40,000
ST02	So tay loại 1	quyen	Viet Nam	55,000
ST03	So tay loại 2	quyen	Viet Nam	51,000
ST04	So tay	quyen	Thai Lan	55,000
ST05	So tay mong	quyen	Thai Lan	20,000
ST06	Phan viet bang	hop	Viet Nam	5,000
ST07	Phan khong bui	hop	Viet Nam	7,000
ST08	Bong bang	cai	Viet Nam	1,000
ST09	But long	cay	Viet Nam	5,000
ST10	But long	cay	Trung Quoc	7,000

HOADON

SOHD	NGHD	MAKH	MANV	TRIGIA
1001	23/07/2006	KH01	NV01	320,000
1002	12/08/2006	KH01	NV02	840,000
1003	23/08/2006	KH02	NV01	100,000
1004	01/09/2006	KH02	NV01	180,000
1005	20/10/2006	KH01	NV02	3,800,000
1006	16/10/2006	KH01	NV03	2,430,000
1007	28/10/2006	KH03	NV03	510,000
1008	28/10/2006	KH01	NV03	440,000
1009	28/10/2006	KH03	NV04	200,000
1010	01/11/2006	KH01	NV01	5,200,000
1011	04/11/2006	KH04	NV03	250,000
1012	30/11/2006	KH05	NV03	21,000
1013	12/12/2006	KH06	NV01	5,000
1014	31/12/2006	KH03	NV02	3,150,000
1015	01/01/2007	KH06	NV01	910,000
1016	01/01/2007	KH07	NV02	12,500
1017	02/01/2007	KH08	NV03	35,000
1018	13/01/2007	KH08	NV03	330,000
1019	13/01/2007	KH01	NV03	30,000
1020	14/01/2007	KH09	NV04	70,000
1021	16/01/2007	KH10	NV03	67,500
1022	16/01/2007	Null	NV03	7,000
1023	17/01/2007	Null	NV01	330,000

CTHD			CTHD			CTHD		
SOHD	MASP	SL	SOHD	MASP	SL	SOHD	MASP	SL
1001	TV02	10	1006	ST01	30	1014	BB01	50
1001	ST01	5	1006	ST02	10	1015	BB02	30
1001	BC01	5	1007	ST03	10	1015	BB03	7
1001	BC02	10	1008	ST04	8	1016	TV01	5
1001	ST08	10	1009	ST05	10	1017	TV02	1
1002	BC04	20	1010	TV07	50	1017	TV03	1
1002	BB01	20	1010	ST07	50	1017	TV04	5
1002	BB02	20	1010	ST08	100	1018	ST04	6
1003	BB03	10	1010	ST04	50	1019	ST05	1
1004	TV01	20	1010	TV03	100	1019	ST06	2
1004	TV02	10	1011	ST06	50	1020	ST07	10
1004	TV03	10	1012	ST07	3	1021	ST08	5
1004	TV04	10	1013	ST08	5	1021	TV01	7
1005	TV05	50	1014	BC02	80	1021	TV02	10
1005	TV06	50	1014	BB02	100	1022	ST07	1
1006	TV07	20	1014	BC04	60	1023	ST04	6

BẢNG THUỘC TÍNH

Quan hệ	Thuộc tính	Diễn giải	Kiểu dữ liệu
KHACHHANG	MAKH	Mã khách hàng	char(4)
	HOTEN	Họ tên	varchar(40)
	DCHI	Địa chỉ	varchar(50)
	SODT	Số điện thoại	varchar(20)
	NGSINH	Ngày sinh	smalldatetime
	NGDK	Ngày đăng ký thành viên	smalldatetime
	DOANHSO	Tổng trị giá các hóa đơn khách hàng đã mua	money
NHANVIEN	MANV	Mã nhân viên	char(4)
	HOTEN	Họ tên	varchar(40)
	SODT	Số điện thoại	varchar(20)
	NGVL	Ngày vào làm	smalldatetime
SANPHAM	MASP	Mã sản phẩm	char(4)
	TENSP	Tên sản phẩm	varchar(40)
	DVT	Đơn vị tính	varchar(20)
	NUOCSX	Nước sản xuất	varchar(40)
	GIA	Giá bán	money
HOADON	SOHD	Số hóa đơn	int
	NGHD	Ngày mua hàng	smalldatetime
	MAKH	Mã khách hàng nào mua	char(4)
	MANV	Nhân viên bán hàng	char(4)
	TRIGIA	Trị giá hóa đơn	money
CTHD	SOHD	Số hóa đơn	int
	MASP	Mã sản phẩm	char(4)
	SL	Số lượng	int

Hình 4.4 Mô tả thông tin CSDL QL BH

2. Phần mô tả câu hỏi

2.1. Các yêu cầu tạo dữ liệu

1. Tạo các quan hệ và khai báo các khóa chính, khóa ngoại của quan hệ.
2. Thêm vào thuộc tính GHICHU có kiểu dữ liệu *varchar(20)* cho quan hệ SANPHAM.

3. Thêm vào thuộc tính LOAIKH có kiểu dữ liệu là *tinyint* cho quan hệ KHACHHANG.
4. Sửa kiểu dữ liệu của thuộc tính GHICHU trong quan hệ SANPHAM thành *varchar(100)*.
5. Xóa thuộc tính GHICHU trong quan hệ SANPHAM.
6. Làm thế nào để thuộc tính LOAIKH trong quan hệ KHACHHANG có thể lưu các giá trị là: “Vang lai”, “Thuong xuyen”, “Vip”, ...
7. Đơn vị tính của sản phẩm chỉ có thể là (“cay”, ”hop”, ”cai”, ”quyen”, ”chuc”)
8. Giá bán của sản phẩm từ 500 đồng trở lên.
9. Mỗi lần mua hàng, khách hàng phải mua ít nhất 1 sản phẩm.
10. Ngày khách hàng đăng ký là khách hàng thành viên phải lớn hơn ngày sinh của người đó.
11. Ngày mua hàng (NGHD) của một khách hàng thành viên sẽ lớn hơn hoặc bằng ngày khách hàng đó đăng ký thành viên (NGDK).
12. Ngày bán hàng (NGHD) của một nhân viên phải lớn hơn hoặc bằng ngày nhân viên đó vào làm.
13. Mỗi một hóa đơn phải có ít nhất một chi tiết hóa đơn.
14. Trị giá của một hóa đơn là tổng thành tiền (số lượng*đơn giá) của các chi tiết thuộc hóa đơn đó.
15. Doanh số của một khách hàng là tổng trị giá các hóa đơn mà khách hàng thành viên đó đã mua.

2.2. Các yêu cầu truy vấn dữ liệu

1. In ra danh sách các sản phẩm (MASP,TENSP) do “Trung Quoc” sản xuất.
2. In ra danh sách các sản phẩm (MASP, TENSP) có đơn vị tính là “cay”, ”quyen”.
3. In ra danh sách các sản phẩm (MASP,TENSP) có mã sản phẩm bắt đầu là “B” và kết thúc là “01”.
4. In ra danh sách các sản phẩm (MASP,TENSP) do “Trung Quốc” sản xuất có giá từ 30.000 đến 40.000.
5. In ra danh sách các sản phẩm (MASP,TENSP) do “Trung Quoc” hoặc “Thai Lan” sản xuất có giá từ 30.000 đến 40.000.
6. In ra các số hóa đơn, trị giá hóa đơn bán ra trong ngày 1/1/2007 và ngày 2/1/2007.
7. In ra các số hóa đơn, trị giá hóa đơn trong tháng 1/2007, sắp xếp theo ngày (tăng dần) và trị giá của hóa đơn (giảm dần).
8. In ra danh sách các khách hàng (MAKH, HOTEN) đã mua hàng trong ngày 1/1/2007.
9. In ra số hóa đơn, trị giá các hóa đơn do nhân viên có tên “Nguyen Van B” lập trong ngày 28/10/2006.
10. In ra danh sách các sản phẩm (MASP,TENSP) được khách hàng có tên “Nguyen Van A” mua trong tháng 10/2006.
11. Tìm các số hóa đơn đã mua sản phẩm có mã số “BB01” hoặc “BB02”.
12. Tìm các số hóa đơn đã mua sản phẩm có mã số “BB01” hoặc “BB02”, mỗi sản phẩm mua với số lượng từ 10 đến 20.
13. Tìm các số hóa đơn mua cùng lúc 2 sản phẩm có mã số “BB01” và “BB02”, mỗi sản phẩm mua với số lượng từ 10 đến 20.
14. In ra danh sách các sản phẩm (MASP,TENSP) do “Trung Quoc” sản xuất hoặc các sản phẩm được bán ra trong ngày 1/1/2007.

15. In ra danh sách các sản phẩm (MASP,TENSP) không bán được.
16. In ra danh sách các sản phẩm (MASP,TENSP) không bán được trong năm 2006.
17. In ra danh sách các sản phẩm (MASP,TENSP) do “Trung Quoc” sản xuất không bán được trong năm 2006.
18. Tìm số hóa đơn đã mua tất cả các sản phẩm do Singapore sản xuất.
19. Có bao nhiêu hóa đơn không phải của khách hàng đăng ký thành viên mua?
20. Có bao nhiêu sản phẩm khác nhau được bán ra trong năm 2006.
21. Cho biết trị giá hóa đơn cao nhất, thấp nhất là bao nhiêu ?
22. Trị giá trung bình của tất cả các hóa đơn được bán ra trong năm 2006 là bao nhiêu?
23. Tính doanh thu bán hàng trong năm 2006.
24. Tìm số hóa đơn có trị giá cao nhất trong năm 2006.
25. Tìm họ tên khách hàng đã mua hóa đơn có trị giá cao nhất trong năm 2006.
26. In ra danh sách 3 khách hàng (MAKH, HOTEN) có doanh số cao nhất.
27. In ra danh sách các sản phẩm (MASP, TENSP) có giá bán bằng 1 trong 3 mức giá cao nhất.
28. In ra danh sách các sản phẩm (MASP, TENSP) do “Thai Lan” sản xuất có giá bằng 1 trong 3 mức giá cao nhất (của tất cả các sản phẩm).
29. In ra danh sách các sản phẩm (MASP, TENSP) do “Trung Quoc” sản xuất có giá bằng 1 trong 3 mức giá cao nhất (của sản phẩm do “Trung Quoc” sản xuất).
30. * In ra danh sách 3 khách hàng có doanh số cao nhất (sắp xếp theo kiểu xếp hạng).
31. Tính tổng số sản phẩm do “Trung Quoc” sản xuất.
32. Tính tổng số sản phẩm của từng nước sản xuất.
33. Với từng nước sản xuất, tìm giá bán cao nhất, thấp nhất, trung bình của các sản phẩm.
34. Tính doanh thu bán hàng mỗi ngày.
35. Tính tổng số lượng của từng sản phẩm bán ra trong tháng 10/2006.
36. Tính doanh thu bán hàng của từng tháng trong năm 2006.
37. Tìm hóa đơn có mua ít nhất 4 sản phẩm khác nhau.
38. Tìm hóa đơn có mua 3 sản phẩm do “Viet Nam” sản xuất (3 sản phẩm khác nhau).
39. Tìm khách hàng (MAKH, HOTEN) có số lần mua hàng nhiều nhất.
40. Tháng mấy trong năm 2006, doanh số bán hàng cao nhất ?
41. Tìm sản phẩm (MASP, TENSP) có tổng số lượng bán ra thấp nhất trong năm 2006.
42. *Mỗi nước sản xuất, tìm sản phẩm (MASP,TENSP) có giá bán cao nhất.
43. Tìm nước sản xuất sản xuất ít nhất 3 sản phẩm có giá bán khác nhau.
44. *Trong 10 khách hàng có doanh số cao nhất, tìm khách hàng có số lần mua hàng nhiều nhất.

Yêu cầu đánh giá kết quả học tập

Nội dung

– Về kiến thức:

- + Trình bày cú pháp trên các câu lệnh T-SQL
- + Trình bày cú pháp các lệnh về Data Definition Language (DDL)

- + Trình bày cú pháp các lệnh về Manipulation Language (DML)
- Về kỹ năng:
 - + Thực hiện đúng cú pháp trên các câu lệnh T-SQL
 - + Thao tác đúng cú pháp các lệnh về Data Definition Language (DDL)
 - + Thao tác đúng cú pháp các lệnh về Manipulation Language (DML)
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

Phương pháp

- Về kiến thức: Đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp.
- Về kỹ năng: Đánh giá kỹ năng đúng cú pháp trên các câu lệnh T-SQL, Data Definition Language (DDL) và Manipulation Language (DML).
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

BÀI 5: THÀNH PHẦN KHÁC TRONG SQL SERVER

Mã bài: MD16-05

Mục tiêu:

- Hiểu được thế nào là view, sự giống nhau giữa table và view;
- Dùng view để lọc dữ liệu;
- Các phép tạo, cập nhật, thêm dữ liệu vào view.
- Mô tả được khái niệm và sử dụng được Trigger để ràng buộc dữ liệu
- Mô tả được khái niệm và sử dụng được Store Procedure
- Trình bày khái niệm cơ bản và sử dụng Function và User-Defined Function
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Giới thiệu về View

1.1 Khái niệm về View

Mục tiêu: Trình bày được các khái niệm về bảng ảo và các áp dụng của nó trên thực tế.

View là một bảng ảo có nội dung được xác định từ một truy vấn. Tuy nhiên khung nhìn không tồn tại như là một cấu trúc lưu trữ dữ liệu trong cơ sở dữ liệu. Dữ liệu bên trong khung nhìn thực chất là dữ liệu được xác định từ một hay nhiều bảng cơ sở đó đó nó phụ thuộc vào các bảng cơ sở.

View có thể được dùng như một kỹ thuật bảo mật. Nó đảm bảo rằng người sử dụng chỉ có thể truy nhập và sửa đổi dữ liệu mà view cung cấp. Người sử dụng không thể truy nhập hay sửa chữa phần dữ liệu còn lại trong các bảng cơ sở.

Việc sử dụng các truy vấn phức tạp cũng có thể được đơn giản hoá nhờ sử dụng view. Các truy vấn phức tạp đó có thể được lưu dưới dạng view và dữ liệu từ view có thể được lấy ra nhờ sử dụng các truy vấn đơn giản hơn.

Sau khi định nghĩa, view được sử dụng tương đương như bất kỳ bảng nào khác trong CSDL.

Ví dụ:

```
Create View OrderReport
As
Select OrderID,
(case when [Name] is null then 'New Customer'
else [Name]
end )As CustomerName,
ProductName,
DateProcessed
From Customers Right Outer Join Orders on
Customers.CustomerID=Orders.CustomerID
```

Trong ví dụ trên ta chủ yếu trả về data từ Orders table trong PracticeDB nhưng thay vì display CustomerID vốn không có ý nghĩa đối với user ta sẽ display tên của customer bằng cách join với Customer table. Nếu Customer Name là Null nghĩa là tên của customer đã đặt order không tồn tại trong system. Thay vì để Null ta sẽ display "New Customer" để dễ nhìn hơn cho user.

Nói chung câu lệnh SQL trong View có thể từ rất đơn giản như select toàn bộ data từ một table cho đến rất phức tạp với nhiều tính năng programming của T-SQL.

View Thường Được Dùng Vào Việc Gì?

View thường được sử dụng vào một số công việc sau:

Tập trung vào một số data nhất định: ta thường dùng view để select một số data mà user quan tâm hay chịu trách nhiệm và loại bỏ những data không cần thiết.

Ví dụ: Giả sử trong table ta có column "Deleted" với giá trị là True hay False để đánh dấu một record bị delete hay không. Việc này đôi khi được dùng cho việc Audit. Nghĩa là trong một ứng dụng nào đó khi user delete một record nào đó, thay vì ta physically delete record ta chỉ logically delete bằng cách đánh dấu record là đã được "Deleted" để đề phòng user yêu cầu roll back. Như vậy chủ yếu ta chỉ quan tâm đến data chưa delete còn data đã được đánh dấu deleted chỉ được để ý khi nào cần roll back hay audit mà thôi. Trong trường hợp này ta có thể tạo ra một view select data mà Deleted=False và làm việc chủ yếu trên view thay vì toàn bộ table.

Đơn giản hóa việc xử lý data: Đôi khi ta có những query rất phức tạp và sử dụng thường xuyên ta có thể chuyển nó thành View và đối xử nó như một table, như vậy sẽ làm cho việc xử lý data dễ dàng hơn.

Customize data: Ta có thể dùng view để làm cho users thấy data từ những góc độ khác nhau mặc dù họ đang dùng một nguồn data giống nhau. Ví dụ: Ta có thể tạo ra views trong đó những thông tin về customer được thể hiện khác nhau tùy login ID là normal user hay manager.

Export và Import data: Đôi khi ta muốn export data từ SQL Server sang các ứng dụng khác như Excel chẳng hạn ta có thể dùng view để join nhiều table và export dùng bcp.

Khi sử dụng view ta có thể select, insert, update, delete data bình thường như với một table.

Ví dụ:

```
Select * From OrderReport  
Where DateProcessed <'2003-01-01'
```

Lưu ý: Trong Enterprise Edition (và Developer Edition) ta có thể tạo Index cho View như cho table. Index sẽ được bàn đến trong các bài sau.

1.2. Ưu điểm của View

Cung cấp dữ liệu thích hợp cho người dùng (đầy đủ nhưng cũng chỉ vừa đủ): Nhìn chung, View thường được dùng để tập trung, một cách đơn giản, và tùy biến dữ liệu tùy theo nhu cầu và quyền hạn mỗi người. Nó có thể được sử dụng làm môi trường bảo mật bằng cách cho phép người sử dụng truy cập và xử lý dữ liệu qua view. Những dữ liệu không cần thiết, bí mật, hay không phù hợp có thể bị loại bỏ khỏi view.

Giấu độ phức tạp của CSDL: View giấu đi sự phức tạp của thiết kế CSDL đối với người sử dụng. Điều này cho phép người phát triển thay đổi thiết kế CSDL mà không làm ảnh hưởng đến sự tương tác giữa người sử dụng và CSDL. Ngoài ra, người sử dụng có thể xem dữ liệu bằng cách sử dụng các tên dễ hiểu hơn so với các tên mã hoá thường được dùng trong CSDL.

Trong các câu truy vấn phức tạp, bao gồm các câu truy vấn từ những dữ liệu không đồng bộ, các dữ liệu chi tiết được che dấu qua view (để hiểu rõ hơn về dữ liệu không đồng bộ, xem phần Tổ chức dữ liệu từ các nguồn không đồng nhất" dưới đây). Người sử dụng truy vấn view thay vì viết truy vấn để lấy dữ liệu hay thực hiện một script.

Tổ chức dữ liệu từ các nguồn không đồng nhất: Bạn có thể tạo view dựa vào những truy vấn phức tạp kết hợp hai hay nhiều bảng (table). View, đối với người sử dụng, giống như một bảng đơn được gọi là view phân vùng. Chẳng hạn, nếu một bảng chứa thông tin chi tiết về lương cho nhân viên ở Mỹ, và bảng khác chứa thông tin chi tiết về lương cho nhân viên ở Anh, một view có thể được tạo từ sự kết hợp (UNION) của những table này. View sẽ thể hiện thông tin chi tiết về lương cho các nhân viên trong cả 2 nước.

Các view phân vùng có thể dựa vào dữ liệu từ các nguồn không đồng nhất, như máy chủ từ xa (remote server), chứ không phải chỉ từ các bảng trong cùng CSDL. Ví dụ, để kết hợp dữ liệu từ các máy chủ từ xa (remote server), trong đó mỗi một máy chủ lưu dữ liệu cho một khu vực khác nhau của một tổ chức, bạn có thể tạo các truy vấn phân tán (những truy vấn truy xuất dữ liệu từ các nguồn dữ liệu không đồng nhất được lưu trên chính máy đó hoặc trên một máy khác) lấy dữ liệu từ mỗi nguồn dữ liệu. Do đó, bạn có thể tạo một view dựa vào các truy vấn phân tán (distributed queries) đó. Truy vấn chỉ đọc dữ liệu được yêu cầu từ một bảng trên máy chủ từ xa có chứa dữ liệu mà truy vấn đó yêu cầu. Các server khác được tham chiếu bởi các truy vấn phân tán trong view đó không truy cập được.

Hạn chế kích cỡ đối tượng: Các view không chứa dữ liệu. SQL server chỉ lưu định nghĩa của view trong CSDL.

1.3 Cách dùng view để lọc dữ liệu

Mục tiêu: Trình bày và tạo được một khung nhìn đơn giản.

1.3.1 Cách dùng view

Các bảng trong cơ sở dữ liệu đóng vai trò là các đối tượng tổ chức và lưu trữ dữ liệu. Như vậy, ta có thể quan sát được dữ liệu trong cơ sở dữ liệu bằng cách thực hiện các truy vấn trên bảng dữ liệu. Ngoài ra, SQL còn cho phép chúng ta quan sát được dữ liệu thông qua việc định nghĩa các khung nhìn.

Một khung nhìn (view) có thể được xem như là một bảng “ảo” trong cơ sở dữ liệu có nội dung được định nghĩa thông qua một truy vấn (câu lệnh SELECT). Như vậy, một khung nhìn trông giống như một bảng với một tên khung nhìn và là một tập bao gồm các dòng và các cột. Điểm khác biệt giữa khung nhìn và bảng là khung nhìn không được xem là một cấu trúc lưu trữ dữ liệu tồn tại trong cơ sở dữ liệu. Thực chất dữ liệu quan sát được trong khung nhìn được lấy từ các bảng thông qua câu lệnh truy vấn dữ liệu.

```
SELECT masv, hodem, ten,  
       DATEDIFF(YY, ngaysinh, GETDATE()) AS tuoi, tenlop  
FROM sinhvien, lop  
WHERE sinhvien.malop = lop.malop
```

Khi khung nhìn DSSV đã được định nghĩa, ta có thể sử dụng câu lệnh SELECT để truy vấn dữ liệu từ khung nhìn như đối với các bảng. Khi trong câu truy vấn xuất hiện khung nhìn, hệ quản trị cơ sở dữ liệu sẽ dựa vào định nghĩa của khung nhìn để chuyển yêu cầu truy vấn dữ liệu liên quan đến khung nhìn thành yêu cầu tương tự trên các bảng cơ sở và việc truy vấn dữ liệu được thực hiện bởi yêu cầu tương đương trên các bảng.

Việc sử dụng khung nhìn trong cơ sở dữ liệu đem lại các lợi ích sau đây:

+ Bảo mật dữ liệu: Người sử dụng được cấp phát quyền trên các khung nhìn với những phần dữ liệu mà người sử dụng được phép. Điều này hạn chế được phần nào việc người sử dụng truy cập trực tiếp dữ liệu.

+ Đơn giản hoá các thao tác truy vấn dữ liệu: Một khung nhìn đóng vai trò như là một đối tượng tập hợp dữ liệu từ nhiều bảng khác nhau vào trong một “bảng”. Nhờ vào đó, người sử dụng có thể thực hiện các yêu cầu truy vấn dữ liệu một cách đơn giản từ khung nhìn thay vì phải đưa ra những câu truy vấn phức tạp.

Tập trung và đơn giản hoá dữ liệu: Thông qua khung nhìn ta có thể cung cấp cho người sử dụng những cấu trúc đơn giản, dễ hiểu hơn về dữ liệu trong cơ sở dữ liệu đồng thời giúp cho người sử dụng tập trung hơn trên những phần dữ liệu cần thiết.

Độc lập dữ liệu: Một khung nhìn có thể cho phép người sử dụng có được cái nhìn về dữ liệu độc lập với cấu trúc của các bảng trong cơ sở dữ liệu cho dù các bảng cơ sở có bị thay đổi phần nào về cấu trúc.

Tuy nhiên, việc sử dụng khung nhìn cũng tồn tại một số nhược điểm sau:

+ Do hệ quản trị cơ sở dữ liệu thực hiện việc chuyển đổi các truy vấn trên khung nhìn thành những truy vấn trên các bảng cơ sở nên nếu một khung nhìn được định nghĩa bởi một truy vấn phức tạp thì sẽ dẫn đến chi phí về mặt thời gian khi thực hiện truy vấn liên quan đến khung nhìn sẽ lớn.

+ Mặc dù thông qua khung nhìn có thể thực hiện được thao tác bổ sung và cập nhật dữ liệu cho bảng cơ sở nhưng chỉ hạn chế đối với những khung nhìn đơn giản. Đối với những khung nhìn phức tạp thì thường không thực hiện được; hay nói cách khác là dữ liệu trong khung nhìn là chỉ đọc.

1.3.2 Tạo khung nhìn - Khung nhìn như bộ lọc

Mục tiêu: Tạo được khung nhìn

Sử dụng câu lệnh Create View để tạo View.

Cú pháp

```
CREATE VIEW view_name
[(column_name[,column_name]...)]
[WITH ENCRYPTION |SCHEMABINDING ]
AS
<select_Statement >
[WITH CHECK OPTION]
```

Trong đó

+ **view_name:** xác định tên của view và cần phải tuân theo các qui tắc đặt tên.

+ **column_name:** xác định tên của các cột sẽ xuất hiện trong view. Nếu không chỉ ra **column_name** thì view được tạo có tên của các cột cũng như kiểu dữ liệu của chúng sẽ tương ứng với các cột trong danh sách chọn của câu lệnh SELECT.

+ **WITH ENCRYPTION:** Nếu có mệnh đề này SQL Server sẽ mã hoá đoạn mã lệnh dùng để định nghĩa view, phục vụ cho mục đích bảo mật.

WITH SCHEMABINDING: Tất cả các Table liên quan đến việc tạo View không thể xoá được khi View vẫn tồn tại.

+ **select_statement:** xác định khai báo câu lệnh SELECT định nghĩa view. View có thể sử dụng dữ liệu chứa trong các view và bảng khác.

+ **WITH CHECK OPTION:** buộc các lệnh sửa đổi dữ liệu phải thoã mãn tiêu chí mà câu lệnh SELECT xác định view đưa ra.

Ghi chú:

+ Lệnh Create View phải được viết đầu tiên trong một Batch

+ Phải chỉ định phần **column_name** trong câu lệnh CREATE VIEW trong các trường hợp sau:

Trong View có cột được pháp sinh từ một biểu thức số học.

Hai hay nhiều cột của khung nhìn có trùng tên.

Bạn muốn thay đổi tên cột trong khung nhìn khác với tên cột của bảng cơ sở

Ví dụ: Cho cơ sở dữ liệu gồm các bảng dữ liệu sau:

Bảng DonVi(MaDV, TenDV)

Bảng NhanVien(MaNv, HoTen, NgaySinh, GioTinh, MaDV)

+ Tạo view để chứa các nhân viên có giới tính bằng 1

```
Create View vwgioitinh
```

```
As
```

```
Select MaNV, HoTen, NgaySinh, GioiTinh
```

```
From NhanVien
```

+ Câu lệnh dưới đây sẽ bị lỗi do tên của cột thứ 3 chưa được xác định

```
Create View vwnhanvien
```

```
As
```

```
Select MaNV, HoTen, Datediff(year,NgaySinh,Getdate()),TenDV
```

```
From NhanVien, DonVi
```

```
Where NhanVien.MaDV=DonVi.MaDV
```

+ Để câu lệnh trên có thể thực hiện được, bạn phải đặt tên cho các cột trong khung nhìn như sau:

```
Create View vwnhanvien(MaNV, HoTen, Tuoi, DonVi)
```

```
As
```

```
Select MaNV, HoTen, Datediff(year,NgaySinh,Getdate()),TenDV
```

```
From NhanVien, DonVi
```

```
Where NhanVien.MaDV=DonVi.MaDV
```

Một số ví dụ tổng hợp:

Ví dụ1:hiển thị thông tin trong bảng khoa

```
create view V-khoa
```

```
as select makhoa,tenkhoa,nam_tl (*)
```

```
from khoa
```

hoặc

```
create view V-khoa
```

```
as select *
```

```
from khoa
```

Ví dụ2: Hiển thị thông tin NV chưa từng bán mặt hàng nào

```
use QLHH
```

```
go
```

```
create view cau4b
```

```
as select a.makh as N'Mã KH',a.tenk as N'Tên KH',a.diachi as N'Địa Chỉ',a.dienthoai
```

```
as N'Điện Thoại'
```

```
from dmkh a
```

```
where (NOT EXISTS
```

```
(select b.makh
```

```
from dmhd b
```

```
where a.makh=b.makh))
```

1.4. Các hạn chế của view

- Tên của view phải tuân theo qui tắc đặt tên, phải duy nhất và không được giống tên của bảng cơ sở.
- View chỉ được tạo nếu có quyền SELECT trên bảng cơ sở.
- View không thể lấy dữ liệu từ các table tạm.
- Định nghĩa View không thể bao gồm các mệnh đề ORDER BY, COMPUTER COMPUTER BY hoặc từ khoá INTO
- Không thể sử dụng ORDER BY cùng câu lệnh SELECT trong view

1.5. Cách cập nhật dữ liệu vào view

Mục tiêu: Trình bày được các cách để cập nhật, bổ sung và xóa dữ liệu thông qua khung nhìn. Thực hiện được việc cập nhật, xóa dữ liệu.

1.5.1 cập nhật dữ liệu vào view

Đối với một số khung nhìn, ta có thể tiến hành thực hiện các thao tác cập nhật, bổ sung và xoá dữ liệu. Thực chất, những thao tác này sẽ được chuyển thành những thao tác tương tự trên các bảng cơ sở và có tác động đến những bảng cơ sở.

Về mặt lý thuyết, để có thể thực hiện thao tác bổ sung, cập nhật và xoá, một khung nhìn trước tiên phải thoả mãn các điều kiện sau đây:

- Trong câu lệnh SELECT định nghĩa khung nhìn không được sử dụng từ khoá DISTINCT, TOP, GROUP BY và UNION.

- Các thành phần xuất hiện trong danh sách chọn của câu lệnh SELECT phải là các cột trong các bảng cơ sở. Trong danh sách chọn không được chứa các biểu thức tính toán, các hàm gộp.

Ngoài những điều kiện trên, các thao tác thay đổi đến dữ liệu thông qua khung nhìn còn phải đảm bảo thoả mãn các ràng buộc trên các bảng cơ sở, tức là vẫn đảm bảo tính toàn vẹn dữ liệu. Ví dụ dưới đây sẽ minh hoạ cho ta thấy việc thực hiện các thao tác bổ sung, cập nhật và xoá dữ liệu thông qua khung nhìn.

Ví dụ: Xét định nghĩa hai bảng DONVI và NHANVIEN như sau:

```
CREATE TABLE donvi
(
    madv          INT          PRIMARY KEY,
    tendv NVARCHAR(30) NOT NULL,
    dienthoai     NVARCHAR(10) NULL,
)
CREATE TABLE nhanvien
(
    manv          NVARCHAR(10) PRIMARY KEY,
    hoten NVARCHAR(30) NOT NULL,
    ngaysinh     DATETIME     NULL,
    diachi NVARCHAR(50) NULL,
    madv          INT          FOREIGN KEY
                                REFERENCES donvi(madv)
                                ON DELETE CASCADE
                                ON UPDATE CASCADE
)
```

Câu lệnh dưới đây định nghĩa khung nhìn NV1 cung cấp các thông tin về mã nhân viên, họ tên và mã đơn vị nhân viên làm việc:

```
CREATE VIEW nv1
AS
SELECT manv,hoten,madv FROM nhanvien
```

Nếu ta thực hiện câu lệnh

```
INSERT INTO nv1 VALUES('NV04','Le Thi D',1)
```

Một bản ghi mới sẽ được bổ sung vào bảng NHANVIEN

Thông qua khung nhìn này, ta cũng có thể thực hiện thao tác cập nhật và xoá dữ liệu. Chẳng hạn, nếu ta thực hiện câu lệnh:

```
DELETE FROM nv1 WHERE manv='NV04'
```

Thì bản ghi tương ứng với nhân viên có mã NV04 sẽ bị xoá khỏi bảng NHANVIEN

Nếu trong danh sách chọn của câu lệnh SELECT có sự xuất hiện của biểu thức tính toán đơn giản, thao tác bổ sung dữ liệu thông qua khung nhìn không thể thực hiện được. Tuy nhiên, trong trường hợp này thao tác cập nhật và xoá dữ liệu vẫn có thể có

khả năng thực hiện được (hiển nhiên không thể cập nhật dữ liệu đối với một cột có được từ một biểu thức tính toán).

Ví dụ: Xét khung nhìn NV2 được định nghĩa như sau:

```
CREATE VIEW nv2
AS
    SELECT manv,hoten,YEAR(ngaysinh) AS namsinh,madv
    FROM nhanvien
```

Đối với khung nhìn NV2, ta không thể thực hiện thao tác bổ sung dữ liệu nhưng có thể cập nhật hoặc xoá dữ liệu trên bảng thông qua khung nhìn này. Câu lệnh dưới đây là không thể thực hiện được trên khung nhìn NV2

```
INSERT INTO nv2(manv,hoten,madv)
VALUES('NV05','Le Van E',1)
```

Nhưng câu lệnh:

```
UPDATE nv2 SET hoten='Le Thi X' WHERE manv='NV04'
```

hoặc câu lệnh

```
DELETE FROM nv2 WHERE manv='NV04'
```

lại có thể thực hiện được và có tác động đối với dữ liệu trong bảng NHANVIEN.

Trong trường hợp khung nhìn được tạo ra từ một phép nối (trong hoặc ngoài) trên nhiều bảng, ta có thể thực hiện được thao tác bổ sung hoặc cập nhật dữ liệu nếu thao tác này chỉ có tác động đến đúng một bảng cơ sở (câu lệnh DELETE không thể thực hiện được trong trường hợp này).

Ví dụ: Với khung nhìn được định nghĩa như sau:

```
CREATE VIEW nv3
AS
    SELECT manv,hoten,ngaysinh,
           diachi,nhanvien.madv AS noilamviec,
           donvi.madv,tendv,dienthoai
    FROM nhanvien FULL OUTER JOIN donvi
    ON nhanvien.madv=donvi.madv
```

Câu lệnh:

```
INSERT INTO nv3(manv,hoten,noilamviec)
VALUES('NV05','Le Van E',1)
```

sẽ bổ sung thêm vào bảng NHANVIEN một bản ghi mới. Hoặc câu lệnh:

```
INSERT INTO nv3(madv,tendv) VALUES(3,'P. Ke toan')
```

bổ sung thêm vào bảng DONVI một bản ghi do cả hai câu lệnh này chỉ có tác động đến đúng một bảng cơ sở.

Câu lệnh dưới đây không thể thực hiện được do có tác động một lúc đến hai bảng cơ sở.

```
INSERT INTO nv3(manv,hoten,noilamviec,madv,tendv)
VALUES('NV05','Le Van E',1,3,'P. Ke toan')
```

1.5.2. Sửa đổi khung nhìn

Bạn có thể sửa lại view mà không cần phải hủy nó. Như thế sẽ đảm bảo rằng các quyền trên view không bị mất đi khi hủy view. Bạn cũng có thể sửa đổi view mà không làm ảnh hưởng tới các đối tượng phụ thuộc của view.

Bạn có thể sử dụng câu lệnh ALTER VIEW để sửa view.

Cú pháp

```
ALTER VIEW view_name [(column_name)]
[WITH ENCRYPTION]
```



```
AS select_statement
[WITH CHECK OPTION]
```

+ view_name: tên của view sẽ được thay đổi.

+ column_name xác định tên của (các) cột được sử dụng trong view.

+ Nếu column_name không được xác định, view được tạo sẽ có cùng các cột như đã xác định trong select_statement.

+ WITH ENCRYPTION mã hoá định dạng text cho the view trong bảng syscomments.

+ select_statement chỉ ra câu lệnh SELECT xác định view. View có thể sử dụng các view hoặc các bảng khác.

+ WITH CHECK OPTION buộc các lệnh sửa đổi dữ liệu phải thoã mãn tiêu chí mà câu lệnh SELECT xác định view đưa ra.

Ví dụ: Ta định nghĩa khung nhìn như sau:

```
CREATE VIEW viewlop
AS
    SELECT malop,tenlop,tenkhoa
    FROM lop INNER JOIN khoa ON lop.makhoa=khoa.makhoa
    WHERE tenkhoa='Khoa Vật lý'
và có thể định nghĩa lại khung nhìn trên bằng câu lệnh:
ALTER VIEW view_lop
AS
    SELECT malop,tenlop,hedaotao
    FROM lop INNER JOIN khoa ON lop.makhoa=khoa.makhoa
    WHERE tenkhoa='Khoa Công nghệ thông tin'
```

1.5.3. Xoá khung nhìn

Bạn có thể sử dụng câu lệnh DROP_VIEW để hủy một view trong CSDL.

Việc hủy view không gây ảnh hưởng gì đến các bảng cơ sở của view.

Việc hủy view chỉ hủy các định nghĩa của view và các quyền được phân trên view.

nếu bạn truy vấn bất kể một view nào tham chiếu đến một bảng đã bị xóa, bạn sẽ nhận được thông báo lỗi.

Cú pháp: DROP VIEW view_name

Trong đó

+ view_name là tên của view cần xóa.

+ Bạn có thể xóa nhiều view bằng một câu lệnh DROP VIEW. Tên của các view cần xóa được phân cách bởi dấu “;”

Nếu một khung nhìn bị xoá, toàn bộ những quyền đã cấp phát cho người sử dụng trên khung nhìn cũng đồng thời bị xoá. Do đó, nếu ta tạo lại khung nhìn thì phải tiến hành cấp phát lại quyền cho người sử dụng.

Ví dụ: Câu lệnh dưới đây xoá khung nhìn VIEW_LOP ra khỏi cơ sở dữ liệu

```
DROP VIEW view_lop
```

1.5.4 Đổi tên View

Bạn có thể đổi tên một view mà không cần xóa nó. Điều này đảm bảo cho các quyền được phân trên view không bị mất. Có thể được đổi tên view bằng cách sử dụng stored procedure hệ thống sp_rename.

Cú pháp: sp_rename old_viewname, new_viewname

Trong đó

old_viewname là tên view cần đổi

new_viewname là tên mới của view

Chú ý: Khi đổi tên view, bạn phải chắc chắn rằng:

- View phải nằm trong database hiện tại
- Tên mới phải được đặt đúng theo quy tắc đặt tên
- Chỉ người tạo view mới có quyền đổi tên
- Người tạo database cũng có quyền đổi tên view
- Định nghĩa View tồn tại ngay cả khi bảng tham gia đã bị xoá

Ví dụ

```
Đổi tên khung nhìn view_lop thành tên mới là view_class trong CSDL
QLSINHVIEN
Use QLSINHVIEN
sp_rename view_lop, view_class
```

2.STORE PROCEDURE

2.1 Thủ tục lưu trữ (Stored procedure)

Là một nhóm các câu lệnh T-SQL đã được biên dịch từ trước(pre-compiled). Một Store Procedure có thể không chứa hoặc chứa nhiều tham số truyền vào; đồng thời có thể trả về một giá trị, một bảng hoặc không trả về 1 giá trị nào đó. Sử dụng Store Procedure độ thực thi nhanh hơn, dễ thay đổi, nâng cấp và bảo mật,....

Có 2 dạng Store Procedure:

- System Store Procedure
- User-defined Store Procedure.

Với thủ tục lưu trữ, một phần nào đó khả năng của ngôn ngữ lập trình được đưa vào trong ngôn ngữ SQL. Thủ tục lưu trữ có thể có các thành phần sau:

Các cấu trúc điều khiển (IF, WHILE, FOR) có thể được sử dụng trong thủ tục.

Bên trong thủ tục lưu trữ có thể sử dụng các biến như trong ngôn ngữ lập trình nhằm lưu giữ các giá trị tính toán được, các giá trị được truy xuất được từ cơ sở dữ liệu.

Lợi ích của việc sử dụng thủ tục lưu trữ:

SQL Server chỉ biên dịch các thủ tục lưu trữ một lần và sử dụng lại kết quả biên dịch này trong các lần tiếp theo trừ khi người dùng có những thiết lập khác. Việc sử dụng lại kết quả biên dịch không làm ảnh hưởng đến hiệu suất hệ thống khi thủ tục lưu trữ được gọi liên tục nhiều lần.

Thủ tục lưu trữ được phân tích, tối ưu khi tạo ra nên việc thực thi chúng nhanh hơn nhiều so với việc phải thực hiện một tập rời rạc các câu lệnh SQL tương đương theo cách thông thường.

Thủ tục lưu trữ cho phép chúng ta thực hiện cùng một yêu cầu bằng một câu lệnh đơn giản thay vì phải sử dụng nhiều dòng lệnh SQL. Điều này sẽ làm giảm thiểu sự lưu thông trên mạng.

Thay vì cấp phát quyền trực tiếp cho người sử dụng trên các câu lệnh SQL và trên các đối tượng cơ sở dữ liệu, ta có thể cấp phát quyền cho người sử dụng thông qua các thủ tục lưu trữ, nhờ đó tăng khả năng bảo mật đối với hệ thống.

Các thủ tục lưu trữ trả về kết quả theo 4 cách:

Sử dụng các tham số output

Sử dụng các lệnh trả về giá trị, các lệnh này luôn trả về giá trị số nguyên.

Tập các giá trị trả về của mỗi câu lệnh SELECT có trong thủ tục lưu trữ hoặc của quá trình gọi một thủ tục lưu trữ khác trong một thủ tục lưu trữ.

Một biến con trỏ toàn cục có thể tham chiếu từ bên ngoài thủ tục.

2.1.1 Tạo thủ tục lưu trữ

Thủ tục lưu trữ được tạo thông qua câu lệnh CREATE PROCEDURE.

```
CREATE PROCEDURE tên_thủ_tục [(danh_sách_tham_số)]  
[WITH RECOMPILE|ENCRYPTION|RECOMPILE,ENCRYPTION] AS  
Các_câu_lệnh_của_thủ_tục
```

Trong đó:

WITH RECOMPILE: yêu cầu SQL Server biên dịch lại thủ tục lưu trữ mỗi khi được gọi. WITH ENCRYPTION: yêu cầu SQL Server mã hóa thủ tục lưu trữ.
Các_câu_lệnh_của_thủ_tục: Các lệnh T-SQL. Các lệnh này có thể nằm trong cặp BEGIN...END hoặc không.

Ví dụ: Giả sử cần thực hiện các công việc theo thứ tự như sau:

Nhập một đơn đặt hàng mới của khách hàng có mã khách hàng là 3

Nhập các chi tiết đơn đặt hàng cho đơn đặt hàng trên.

Để thực hiện các công việc trên chúng ta cần các câu lệnh như sau: Trước tiên nhập đơn đặt hàng cho khách hàng có mã khách hàng là 3 *insert into orders values(3, '7/22/2019')*

Tiếp theo thêm các chi tiết đơn đặt hàng cho hóa đơn này. Giả sử rằng đơn đặt hàng có mã là 4 và khách hàng đặt một mặt hàng có mã là 1.

```
insert into orderdetail values(4, 1, 10)
```

Cách viết như trên có hạn chế là: trong quá trình làm việc sẽ có rất nhiều đơn đặt hàng mới, do đó người dùng sẽ phải viết đi viết lại những câu lệnh tương tự nhau cho các khách hàng khác nhau. Một cách giải quyết vấn đề này là dùng thủ tục lưu trữ và dùng tham số để nhận các thông tin thay đổi.

```
create procedure sp_InsertOrderAndOrderDetail  
@customerid int,  
@orderdate datetime,  
@orderid int,  
@itemid int,
```

```
@quantity decimal, as  
begin
```

```
insert into orders values(@customerid, @orderdate)
```

```
insert into orderdetail  
values(@orderid, @itemid, @quantity)  
end
```

Thực hiện thủ tục lưu trữ này như sau:

```
sp_InsertOrderAndOrderDetail '3', '22/7/2019', '4', '1', '10')
```

2.1.2 Lời gọi thủ tục

Thủ tục lưu trữ được gọi theo cấu trúc

```
Tên_thủ_tục_lưu_trữ [danh_sách_tham_số]
```

Cần Lưu ý là danh sách tham số truyền vào trong lời gọi phải theo đúng thứ tự khai báo các tham số trong thủ tục lưu trữ.

Nếu thủ tục được gọi từ một thủ tục khác, thực hiện bên trong một trigger hay phối hợp với câu lệnh SELECT, cấu trúc như sau;

```
Exec Tên_thủ_tục_lưu_trữ [danh_sách_tham_số]
```

2.1.3 Biến trong thủ tục lưu trữ

Trong thủ tục lưu trữ có thể có các biến nhằm lưu các kết quả tính toán hay truy xuất từ CSDL. Các biến trong thủ tục được khai báo bằng từ khóa DECLARE theo cấu trúc như sau:

```
DECLARE @tên_biến kiểu dữ liệu
```

Ví dụ:

```
create procedure sp_SelectCustomerWithMaxAge as
begin
declare @maxAge int
select @maxAge = max(year(getdate())-year(BIRTHDAY))
from customers
select CUSTOMERNAME, BIRTHDAY
from customers
where year(getdate())-year(BIRTHDAY)=@maxAge
end
```

2.1.4 Giá trị trả về trong thủ tục lưu trữ

Trong các Ví dụ trước, nếu đối số truyền cho thủ tục khi có lời gọi đến thủ tục là biến, những thay đổi giá trị của biến trong thủ tục sẽ không được giữ lại khi kết thúc quá trình thực hiện thủ tục.

Ví dụ: Có thủ tục lưu trữ như sau

```
create procedure sp_TestOutput
@a int,
@b int,
@c int as
select @c = @a + @b Thực thi thủ tục: Declare @tong int
set @tong = 0
sp_TestOutput 100, 200, @tong select @tong
Kết quả là 0.
```

Sử dụng tham số OUTPUT

Trong trường hợp cần phải giữ lại giá trị của đối số sau khi kết thúc thủ tục, ta phải khai báo tham số của thủ tục theo cú pháp như sau:

```
@tên_tham_số kiểu dữ liệu OUTPUT
```

Ví dụ trên được viết lại như sau:

```
create procedure sp_TestOutput
@a int,
@b int,
@c int output as
select @c = @a + @b Thực thi thủ tục: Declare @tong int
set @tong = 0
sp_TestOutput 100, 100, @tong output select @tong
Kết quả là 200.
```

Sử dụng lệnh RETURN

Trương tự như việc sử dụng tham số OUTPUT, câu lệnh RETURN trả về giá trị cho đối tượng thực thi stored procedure.

Ví dụ:

```
create procedure sp_TestReturn as
begin
```

```

declare @out int
select @out = count(*)
from customers return @out end

```

Thực thi thủ tục lưu trữ

```

declare @a int
exec @a = sp_TestReturn select @a

```

2.1.5 Tham số với giá trị mặc định

Các tham số được khai báo trong thủ tục có thể nhận các giá trị mặc định. Giá trị mặc định sẽ được gán cho tham số trong trường hợp không truyền đối số cho tham số khi có lời gọi đến thủ tục.

Tham số với giá trị mặc định được khai báo theo cú pháp như sau:

```
@tên_tham_số kiểu_dữ_liệu = giá_trị_mặc_định
```

Ví dụ:

```

create procedure sp_TestDefault
@customerid int = 3
as begin
select * from customers
where customerid = @customerid end

```

Thực thi thủ tục lưu trữ theo giá trị mặc định của tham số.

sp_TestDefault

CUSTOMERID	CUSTOMERNAME	BIRTHDAY	GENDER	ADDRESS
3	Tran Viet Cuong	1980-01-01 00:00:00.000	1	45/2B Da Tuong

Thực thi thủ tục và truyền giá trị cho tham số:

sp_TestDefault 4

CUSTOMERID	CUSTOMERNAME	BIRTHDAY	GENDER	ADDRESS
4	Nguyen Van Dai	1955-03-04 00:00:00.000	1	76 Tran Phu

2.1.6 Sửa đổi thủ tục

Khi một thủ tục đã được tạo ra, ta có thể tiến hành định nghĩa lại thủ tục đó bằng câu lệnh

ALTER PROCEDURE có cú pháp như sau:

```

ALTER PROCEDURE tên_thủ_tục [(danh_sách_tham_số)]
[WITH RECOMPILE/ENCRYPTION/RECOMPILE,ENCRYPTION] AS

```

Các_câu_lệnh_của_thủ_tục

Câu lệnh này sử dụng tương tự như câu lệnh CREATE PROCEDURE. Việc sửa đổi lại một thủ tục đã có không làm thay đổi đến các quyền đã cấp phát trên thủ tục cũng như không tác động đến các thủ tục khác hay trigger phụ thuộc vào thủ tục này.

2.1.7 Xóa thủ tục

Để xóa một thủ tục đã có, ta sử dụng câu lệnh DROP PROCEDURE với cú pháp như sau:

```
DROP PROCEDURE tên_thủ_tục
```

Khi xóa một thủ tục, tất cả các quyền đã cấp cho người sử dụng trên thủ tục đó cũng đồng thời bị xóa bỏ. Do đó, nếu tạo lại thủ tục, ta phải tiến hành cấp phát lại các quyền trên thủ tục đó.

2.2. Hàm do người dùng định nghĩa (User Defined Function-UDF)

Hàm do người dùng định nghĩa được chia làm 3 loại: (1) scalar (hàm vô hướng), (2) inline table-valued (hàm nội tuyến, giá trị trả về dạng bảng), (3) multi-statement table-valued (hàm bao gồm nhiều câu lệnh SQL bên trong, trả về giá trị dạng bảng)

Scalar UDF: được sử dụng để trả về một duy nhất một giá trị dựa trên một các tham số truyền vào. Ví dụ: ta có thể tạo ra một UDF vô hướng nhận Customerid là tham số và trả về CustomerName.

Inline table-valued: trả về một bảng dựa trên một câu lệnh SQL duy nhất định nghĩa các dòng và các cột trả về.

Multi-statement table-value: cũng trả về kết quả là một tập hợp nhưng có thể dựa trên nhiều câu lệnh SQL.

2.2.1 Hàm vô hướng - Scalar UDF

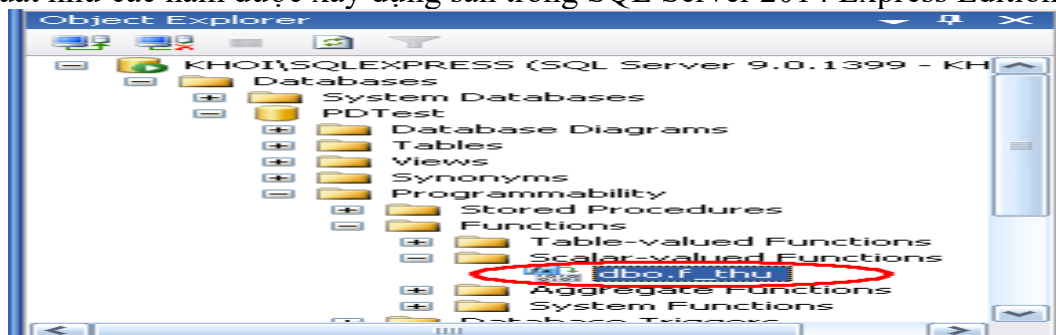
Scalar UDF được tạo ra bằng câu lệnh CREATE FUNCTION có cấu trúc như sau;
CREATE FUNCTION tên_hàm
([danh_sách_tham_số]) RETURNS (kiểu_trả_về_của_hàm) AS BEGIN
các_câu_lệnh_của_hàm
END

Ví dụ:

Câu lệnh dưới đây định nghĩa hàm tính ngày trong tuần (thứ trong tuần) của một giá trị kiểu ngày

```
create function f_thu(@ngay datetime)  
returns nvarchar(10)  
as begin  
declare @st nvarchar(10)  
select @st=case datepart(dw,@ngay)  
when 1 then N'chủ nhật' when 2 then N'thứ hai' when 3 then N'thứ ba' when 4 then  
N'thứ tư' when 5 then N'thứ năm' when 6 then N'thứ sáu' else N'thứ bảy'  
end  
return (@st) /* trả về của hàm */  
end
```

Sau khi chạy thành công, hàm trở thành một đối tượng trong CSDL và có thể được truy xuất như các hàm được xây dựng sẵn trong SQL Server 2014 Express Edition.



Ví dụ:

```
select CUSTOMERNAME, dbo.f_thu(BIRTHDAY)  
from customers
```

CUSTOMERNAME	(No column name)
Cao Van Trung	thứ sáu
Tran Van Phuc	thứ hai
Tran Viet Cuong	thứ ba
Nguyen Van Dai	thứ sáu
Le Thi Hoa	thứ hai
Nguyen Thanh Thai	thứ sáu
Cao Van Chung	thứ bảy
NGuyen Van An	thứ sáu
NGuyen Van An	thứ sáu

2.2.2 Hàm nội tuyến - Inline UDF

Hàm nội tuyến được định nghĩa bằng lệnh CREATE FUNCTION.

```
CREATE FUNCTION tên_hàm ([danh_sách_tham_số])
RETURN
S TABLE
AS
RETURN (câu_lệnh_select)
```

Cú pháp của hàm nội tuyến phải tuân theo các qui tắc sau:

Kiểu trả về của hàm phải được chỉ định bởi mệnh đề RETURNS TABLE.

Trong phần thân của hàm chỉ có duy nhất một câu lệnh RETURN xác định giá trị trả về của hàm thông qua duy nhất một câu lệnh SELECT. Ngoài ra, không sử dụng bất kỳ câu lệnh nào khác trong phần thân của hàm.

Ví dụ: Ví dụ dưới đây lấy ra các khách hàng tùy thuộc vào giá trị mã khách hàng truyền vào cho tham số.

```
create function f_SelectCustomer
(@customerid int)
returns table as
return (select * from customers
where customerid > @customerid)
```

Việc gọi các hàm nội tuyến cũng tương tự như việc gọi các hàm vô hướng.

Ví dụ:

```
select tmp.CUSTOMERNAME, o.ORDERDATE
from orders o inner join dbo.f_SelectCustomer(3) as tmp on o.customerid =
tmp.customerid
```

CUSTOMERNAME	ORDERDATE
Cao Van Trung	2007-12-06 00:00:00.000

2.2.3 Hàm bao gồm nhiều câu lệnh bên trong – Multi statement UDF

Hàm này cũng được định nghĩa bằng lệnh CREATE FUNCTION

```
CREATE FUNCTION tên_hàm
([danh_sách_tham_số])
RETURNS @biến_bảng TABLE định_nghĩa_bảng
AS
BEGIN các_câu_lệnh_trong_thân_hàm
RETURN END
```

Lưu ý: sau từ khóa RETURNS là một biến bảng được định nghĩa. Và sau từ khóa RETURN ở cuối hàm không có tham số nào đi kèm. Ví dụ:

```
create function f_SelectCustomer (@customerid int)
```

```

returns @myCustomers table
(
  customerid int,
  customername nvarchar(50), orderdate datetime
)
as begin
  if @customerid = 0
  insert into @myCustomers
  select c.customerid, c.customername, o.orderdate
  from customers c inner join orders o on o.customerid = c.customerid
  else
  return end

```

```

insert into @myCustomers
select c.customerid, c.customername, o.orderdate
from customers c inner join orders o on c.customerid = o.customerid
where c.customerid = @customerid

```

Việc gọi hàm multi statement UDF cũng tương tự các loại hàm khác

```
select * from f_SelectCustomer(0)
```

customerid	customername	orderdate
6	Cao Van Trung	2007-12-06 00:00:00.000
3	Tran Viet Cuong	2008-01-01 00:00:00.000
3	Tran Viet Cuong	2008-05-01 00:00:00.000

```
select * from f_SelectCustomer(3)
```

customerid	customername	orderdate
3	Tran Viet Cuong	2008-01-01 00:00:00.000
3	Tran Viet Cuong	2008-05-01 00:00:00.000

2.2.4 Thay đổi hàm

Dùng lệnh ALTER FUNCTION để thay đổi định nghĩa hàm. Cấu trúc của câu lệnh ALTER FUNCTION tương tự như CREATE FUNCTION

Ví dụ:

```

alter function f_SelectCustomer
(@customerid int)
returns table as
return (select * from customers
        where customerid > @customerid)

```

2.2.5 Xóa hàm

Dùng lệnh DROP FUNCTION để xóa hàm. Cấu trúc lệnh DROP FUNCTION như sau

```
DROP FUNCTION tên_hàm
```

Ví dụ:

```
drop function f_thu
```

Tương tự như thủ tục lưu trữ, khi hàm bị xóa các quyền cấp cho người dùng trên hàm đó cũng bị xóa. Do đó khi định nghĩa lại hàm này, ta phải cấp lại quyền cho các người dùng.

3 Trigger

Trigger là đối tượng gắn liền với một bảng, tự động thực hiện khi xảy ra sự thay đổi dữ liệu trong bảng như Update, Insert hay Delete. Trigger được dùng để đảm bảo Data Integrity (toàn vẹn dữ liệu) hay thực hiện các Business Rule (ràng buộc dữ liệu) nào đó.

Định nghĩa Trigger cần chú ý:

- Trigger được tạo trong bảng nào?
- Trigger được kích hoạt khi câu lệnh nào được thực thi (Insert, Update hay Delete)..

3.1 Các đặc điểm của trigger

Trigger chỉ thực thi tự động thông qua các sự kiện mà không thực hiện bằng tay. Trigger sử dụng được với khung nhìn.

Khi trigger thực thi theo các sự kiện Insert hoặc Delete thì dữ liệu khi thay đổi sẽ được chuyển sang các bảng INSERTED và DELETED, là 2 bảng tạm thời chỉ chứa trong bộ nhớ, các bảng này chỉ được sử dụng với các lệnh trong trigger. Các bảng này thường được sử dụng để khôi phục lại phần dữ liệu đã thay đổi (roll back).

Trigger chia thành 2 loại INSTEAD OF và AFTER: INSTEAD OF là loại trigger mà hoạt động của sự kiện gọi trigger sẽ bị bỏ qua và thay vào đó là các lệnh trong trigger được thực hiện. AFTER trigger là loại ngầm định, khác với loại INSTEAD OF thì loại trigger này sẽ thực hiện các lệnh bên trong sau khi đã thực hiện xong sự kiện kích hoạt trigger.

3.2 Các trường hợp sử dụng trigger

Sử dụng Trigger khi các biện pháp bảo đảm toàn vẹn dữ liệu khác không bảo đảm được. Các công cụ này sẽ thực hiện kiểm tra tính toàn vẹn trước khi đưa dữ liệu vào CSDL, còn Trigger thực hiện kiểm tra tính toàn vẹn khi công việc đã thực hiện

Khi CSDL chưa được chuẩn hóa (Normalization) thì có thể xảy ra dữ liệu thừa, chứa ở nhiều vị trí trong CSDL thì yêu cầu đặt ra là dữ liệu cần cập nhật thống nhất trong mọi nơi. Trong trường hợp này ta phải sử dụng Trigger.

Khi xảy ra thay đổi dây chuyền dữ liệu giữa các bảng với nhau (khi dữ liệu bảng này thay đổi thì dữ liệu trong bảng khác cũng được thay đổi theo).

3.3 Khả năng sau của trigger

Một trigger có thể nhận biết, ngăn chặn và huỷ bỏ được những thao tác làm thay đổi trái phép dữ liệu trong cơ sở dữ liệu.

Các thao tác trên dữ liệu (xoá, cập nhật và bổ sung) có thể được trigger phát hiện ra và tự động thực hiện một loạt các thao tác khác trên cơ sở dữ liệu nhằm đảm bảo tính hợp lệ của dữ liệu.

Thông qua trigger, ta có thể tạo và kiểm tra được những mối quan hệ phức tạp hơn giữa các bảng trong cơ sở dữ liệu mà bản thân các ràng buộc không thể thực hiện được.

3.4 Định nghĩa trigger

Câu lệnh CREATE TRIGGER được sử dụng để định nghĩa trigger và có cấu trúc như sau:

```
CREATE TRIGGER tên_trigger
ON tên_bảng
FOR {[INSERT][,][UPDATE][,][DELETE]} AS
[IF UPDATE(tên_cột)
[AND UPDATE(tên_cột)|OR UPDATE(tên_cột)]
...]
```

các_câu_lệnh_của_trigger

Lưu ý: Như đã nói ở trên, chuẩn SQL định nghĩa hai bảng logic INSERTED và DELETED để sử dụng trong các trigger. Cấu trúc của hai bảng này tương tự như cấu trúc của bảng mà trigger tác động. Dữ liệu trong hai bảng này tùy thuộc vào câu lệnh tác động lên bảng làm kích hoạt trigger; cụ thể trong các trường hợp sau:

Khi câu lệnh DELETE được thực thi trên bảng, các dòng dữ liệu bị xóa sẽ được sao chép vào trong bảng DELETED. Bảng INSERTED trong trường hợp này không có dữ liệu.

Dữ liệu trong bảng INSERTED sẽ là dòng dữ liệu được bổ sung vào bảng gây nên sự kích hoạt đối với trigger bằng câu lệnh INSERT. Bảng DELETED trong trường hợp này không có dữ liệu.

Khi câu lệnh UPDATE được thực thi trên bảng, các dòng dữ liệu cũ chịu sự tác động của câu lệnh sẽ được sao chép vào bảng DELETED, còn trong bảng INSERTED sẽ là các dòng sau khi đã được cập nhật.

Hoạt động	Bảng INSERTED	Bảng DELETED
INSERT	dữ liệu được insert	không có dữ liệu
DELETE	không có dữ liệu	dữ liệu bị xóa
UPDATE	dữ liệu được cập nhật	dữ liệu trước khi cập nhật

Ví dụ 1: Ví dụ dưới đây minh họa việc trigger được kích hoạt khi thêm dữ liệu vào bảng CUSTOMERS

```

if exists (select name from sysobjects
where name = 't_CheckCustomerName' and type = 'TR')
drop trigger t_CheckCustomerName go
create trigger t_CheckCustomerName on customers
for insert as
declare @lengthOfName int
select @lengthOfName = len(inserted.customername)
from inserted
if @lengthOfName <= 1 print N'Tên không hợp lệ' rollback tran
go
    
```

Thêm một khách hàng mới có tên là A

```

insert into customers
values('A', '5/5/1978', 'True', '35 Hung Vuong')
    
```

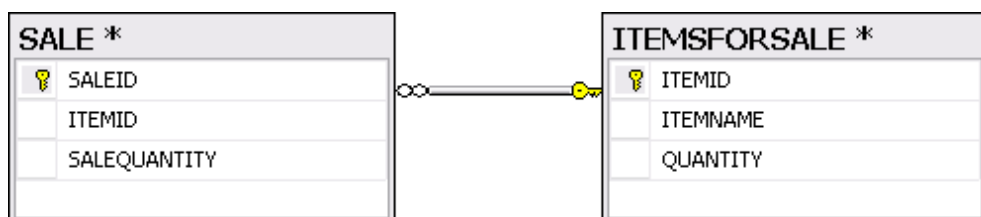
Tên không hợp lệ

Msg 3609, Level 16, State 1, Line 2

The transaction ended in the trigger. The batch has been aborted.

Ví dụ 2: Ví dụ dưới đây minh họa trigger được kích hoạt khi có sự thay đổi mang tính đây chuyển giữa các bảng.

Giả sử có CSDL như sau:



Với dữ liệu trong từng bảng là:

ITEMID	ITEMNAME	QUANTITY
1	LAPTOP	100.00
2	PPC	2000.00
3	IPOD	10.00

SALEID	ITEM...	SALEQUA
1	1	10.00
2	2	10.00

Giả sử có một khách hàng mua 10 đơn vị mặt hàng LAPTOP. Khi đó số lượng LAPTOP trong bảng ITEMFORSALE sẽ giảm xuống còn 90. Trigger dưới đây sẽ thực hiện công việc đó.

```

if exists (select name from sysobjects
where name = 't_DecreaseQuantityOfItemForSale')
drop trigger t_DecreaseQuantityOfItemForSale go
create trigger t_DecreaseQuantityOfItemForSale on SALE
for insert as
update ITEMSFORSALE
set itemsforsale.quantity = itemsforsale.quantity - inserted.salequantity from
itemsforsale inner join inserted
on itemsforsale.itemid = inserted.itemid go

```

Thực hiện thêm dòng vào bảng SALE

```
insert into sale values( 1, 10)
```

ITEMID	ITEMNAME	QUANTITY	SALEID	ITEM...	SALEQUANTITY
1	LAPTOP	90.00	1	1	10.00
2	PPC	100.00	2	2	10.00
3	IPOD	20.00	9	1	10.00

Ví dụ 3: Ví dụ này minh họa cũng minh họa trigger được kích hoạt khi có sự thay đổi mang tính dây chuyền giữa các bảng nhưng trong trường hợp này dữ liệu thay đổi liên quan đến nhiều dòng.

Giả sử người quản lý muốn thay đổi số lượng bán mặt hàng LAPTOP trong bảng SALE lên thêm 5 đơn vị. Như vậy từ kết quả Ví dụ 2, ta thấy cần phải giảm số lượng LAPTOP trong bảng ITEMSFORSALE xuống 10 đơn vị. Tuy nhiên, trong thực tế khi số lượng các dòng trong bảng SALE rất lớn, khi đó phải sử dụng trigger:

```

if exists (select name from sysobjects
where name = 't_DecreaseSumQuantityOfItemForSale')
drop trigger t_DecreaseSumQuantityOfItemForSale go
create trigger t_DecreaseSumQuantityOfItemForSale on SALE
for update as
if update(salequantity)
update ITEMSFORSALE
set itemsforsale.quantity = itemsforsale.quantity -
(select sum(inserted.salequantity - deleted.salequantity)
from deleted inner join inserted
on deleted.saleid = inserted.saleid
where inserted.itemid = itemsforsale.itemid)
where itemsforsale.itemid in (select inserted.itemid

```

Thực hiện cập nhật cho bảng SALE:

```
update sale
```

```

set salequantity = salequantity + 10
where itemid = 1
from inserted)

```

SALEID	ITEM...	SALEQUANTITY	ITEMID	ITEMNAME	QUANTITY
1	1	20.00	1	LAPTOP	70.00
2	2	10.00	2	PPC	100.00
9	1	20.00	3	IPOD	20.00

Ví dụ 4: Ví dụ này minh họa INSTEAD OF trigger. Trigger dưới đây sẽ không cho thực hiện thao tác xóa trên bảng CUSTOMERS.

```

create trigger t_RollbackDelete on customers
after delete as
rollback tran

```

3.5 Kích hoạt trigger dựa trên sự thay đổi dữ liệu trên cột

Thay vì chỉ định một trigger được kích hoạt trên một bảng, ta có thể chỉ định trigger được kích hoạt và thực hiện những thao tác cụ thể khi việc thay đổi dữ liệu chỉ liên quan đến một số cột nhất định nào đó của cột. Trong trường hợp này, ta sử dụng mệnh đề IF UPDATE trong trigger. IF UPDATE không sử dụng được đối với câu lệnh DELETE.

Trở lại Ví dụ 3 trong phần định nghĩa trigger:

```

if exists (select name from sysobjects
where name = 't_DecreaseSumQuantityOfItemForSale')
drop trigger t_DecreaseSumQuantityOfItemForSale go
create trigger t_DecreaseSumQuantityOfItemForSale on SALE
for update as
if update(salequantity)
update ITEMSFORSALE
set itemsforsale.quantity = itemsforsale.quantity -
(select sum(inserted.salequantity - deleted.salequantity)
from deleted inner join inserted
on deleted.saleid = inserted.saleid
where inserted.itemid = itemsforsale.itemid)
where itemsforsale.itemid in (select inserted.itemid
from inserted)

```

Trong Ví dụ này trigger sẽ được kích hoạt khi có sự thay đổi dữ liệu trong cột *salequantity* của bảng Sale. Nếu có sự thay đổi dữ liệu trên các cột khác thì trigger sẽ không được kích hoạt. Câu lệnh dưới đây không làm cho trigger kích hoạt.

```

update sale
set itemid = 3
where itemid = 2

```

Mệnh đề IF UPDATE có thể xuất hiện nhiều lần trong phần thân của trigger. Khi đó, mệnh đề IF UPDATE nào đúng thì phần câu lệnh của mệnh đề đó sẽ được thực thi khi trigger được kích hoạt.

3.6 Sử dụng trigger và Giao tác (TRANSACTION)

Khi một trigger được kích hoạt, SQL Server luôn tạo ra một giao tác theo dõi những thay đổi do câu lệnh kích hoạt trigger hoặc do bản thân trigger gây ra. Sự theo dõi này cho phép CSDL quay trở lại trạng thái trước đó.

Ví dụ: Ví dụ dưới đây xây dựng trigger không cho phép nhập vào một bản ghi trong bảng SALE khi số lượng hàng bán lớn hơn số lượng hàng thực tế còn lại trong bảng ITEMSFORSALE

```
if exists (select name from sysobjects
where name = 't_CheckQuantity' and type = 'TR')
drop trigger t_CheckQuantity go
```

```
create trigger t_CheckQuantity on sale
for insert as
declare @insertedQuantity decimal(18,2) declare @currentQuantity
decimal(18,2) declare @itemid int
```

```
select @itemid = itemid from inserted
select @insertedQuantity = salequantity from inserted select @currentQuantity
= quantity
from itemsforsale
where itemid = @itemid
if (@currentquantity < @insertedquantity)
print N'số lượng nhập vào lớn hơn số lượng hiện có'
rollback tran
```

Tiến hành thêm vào bảng SALE số liệu như sau:

```
insert into sale values(2, 1000)
```

```
{1 row(s) affected}
```

```
số lượng nhập vào lớn hơn số lượng hiện có
```

```
Msg 3609, Level 16, State 1, Line 2
```

```
The transaction ended in the trigger. The batch has been aborted.
```

3.7. DDL TRIGGER

Được giới thiệu trong SQL Server, khác với DML trigger được kích hoạt khi có sự thay đổi dữ liệu trên bảng, DDL trigger được thiết kế để đáp ứng lại các sự kiện diễn ra trên server hay trên CSDL. Một DDL trigger có thể được kích hoạt khi người dùng thực hiện các lệnh CREATE TABLE hay DROP TABLE. Ở cấp độ server, DDL trigger có thể được kích hoạt khi có một tài khoản mới được tạo ra

DDL trigger được lưu trữ trong CSDL mà DDL trigger được gắn vào. Với các Server DDL Trigger theo dõi các thay đổi ở cấp độ Server, được lưu trữ trong CSDL master.

DDL trigger được tạo ra cũng bằng câu lệnh CREATE TRIGGER với cấu trúc như sau:

```
CREATE TRIGGER tên_trigger
ON { ALL SERVER / DATABASE } FOR { loại_sự_kiện } [,...n ]
AS { các_câu_lệnh_SQL }
```

Trong đó:

- *ALL SERVER / DATABASE*: quy định trigger sẽ kích hoạt dựa trên các sự kiện diễn ra trên Server hay các sự kiện diễn ra trên CSDL.
- *loại_sự_kiện*: là một sự kiện đơn ở cấp độ Server hay cấp độ CSDL làm kích hoạt DDL trigger như: CREATE_TABLE, ALTER_TABLE, DROP_TABLE...
- Ví dụ 1: Câu lệnh dưới đây xây dựng một trigger được kích hoạt khi xảy ra các sự kiện ở cấp độ CSDL. Trigger này sẽ ngăn chặn các lệnh DROP TABLE và ALTER TABLE.

```

create trigger t_safety on database
for CREATE_TABLE, DROP_TABLE
as
print N'Phải xóa trigger t_safety trước khi ALTER hay DROP bảng'
rollback tran
Tiến hành xóa bảng ORDERDETAIL
drop table orderdetail

```

Ví dụ 2: Câu lệnh dưới đây xây dựng một trigger được kích hoạt khi xảy ra các sự kiện ở cấp độ Server. Trigger này sẽ ngăn chặn việc tạo ra một account login mới

```

IF EXISTS (SELECT * FROM sys.server_triggers WHERE name =
't_DoNotAllowCreateNewLogin') DROP TRIGGER
t_DoNotAllowCreateNewLogin
ON ALL SERVER GO
CREATE TRIGGER t_DoNotAllowCreateNewLogin
ON ALL SERVER
FOR CREATE_LOGIN AS
PRINT N'Phải DROP trigger t_DoNotAllowCreateNewLogin trước khi
tạo account'
rollback
GO

```

Tiến hành tạo một account login mới:

```
create login test with password = '123456'
```

Phải DROP trigger t_DoNotAllowCreateNewLogin trước khi tạo account

```
Msg 3609, Level 16, State 2, Line 1
```

```
The transaction ended in the trigger. The batch has been aborted.
```

3.8. Enable/ Disable TRIGGER

Trigger cần bị vô hiệu hóa trong một số trường hợp:

Trigger gây ra lỗi trong quá trình xử lý CSDL

Quá trình nhập hay khôi phục những dữ liệu không thỏa trigger.

Vô hiệu hóa trigger bằng lệnh DISABLE TRIGGER có cấu trúc như sau:

```
DISABLE TRIGGER tên_trigger
ON { tên_đối_tượng | DATABASE | SERVER }
```

Ví dụ 1: Ví dụ này sẽ vô hiệu hóa trigger t_DoNotAllowCreateNewLogin disable trigger t_DoNotAllowCreateNewLogin

```
on all server
```

Tiến hành tạo một account login mới:

```
create login newLogin with password = '12345'
```

Command(s) completed successfully

Ví dụ 2: Ví dụ này sẽ khôi phục lại trigger t_DoNotAllowCreateNewLogin enable trigger t_DoNotAllowCreateNewLogin

```
on all server
```

Tiến hành tạo một account login mới:

```
create login newLogin1 with password = '12345'
```

Phải DROP trigger t_DoNotAllowCreateNewLogin trước khi tạo account
Msg 3609, Level 16, State 2, Line 1
The transaction ended in the trigger. The batch has been aborted.

Bài tập thực hành của học viên

Sử dụng CSDL QLBanHang thực hiện các yêu cầu sau:

1/ Tạo các view sau

- 1.1 Tạo khung nhìn để hiển thị thông tin cho bảng DONDATHANG
- 1.2 Tạo khung nhìn để hiển thị thông tin trong bảng CHITIETDATHANG với ThanhTien= giaban*soluong
- 1.3 Tạo ra view NHANVIEN_VIEW với tùy chọn WITH CHECK OPTION

2/ Tạo các thủ tục lưu trữ sau

- 2.1 Tạo thủ tục lưu trữ để thông qua thủ tục này có thể bổ sung thêm một bản ghi mới cho bảng MATHANG (thủ tục phải thực hiện kiểm tra tính hợp lệ của dữ liệu cần bổ sung: không trùng khoá chính và đảm bảo toàn vẹn tham chiếu)
- 2.2 Tạo thủ tục lưu trữ có chức năng thống kê tổng số lượng hàng bán được của một mặt hàng có mã bất kỳ (mã mặt hàng cần thống kê là tham số của thủ tục).

3/ Tạo các trigger sau

- 3.1 Khi thay đổi cột SL trên bảng CTHD, chúng ta phải cập nhật lại dữ liệu cho cột SoTon trong bảng MATHANG
- 3.2 Khi xoá một dòng trong bảng CTHD, chúng ta phải tăng số lượng của bảng MATHANG
- 3.3 Kiểm tra ràng buộc một CTHD không chứa nhiều hơn 10 SoHD. Trong hình, chúng tôi đang chọn SoHD là 10249. Hiện tại trong bảng CTHD trước khi thêm đang có 2. Dòng màu vàng là dữ liệu chúng ta muốn thêm vào

Hướng dẫn thực hiện

1/ Tạo các view sau

Tạo khung nhìn để hiển thị thông tin cho bảng DONDATHANG

```
CREATE VIEW view_donhang  
AS
```

```
SELECT dondathang.sohoadon,makhachhang,manhanvien,  
ngaydathang,ngaygiaohang,ngaychuyenhang,  
noigiaohang,mahang,  
giaban,soluong,mucgiamgia  
FROM dondathang INNER JOIN chitietdathang  
ON dondathang.sohoadon = chitietdathang.sohoadon
```

*1.2 Tạo khung nhìn để hiển thị thông tin trong bảng CHITIETDATHANG với ThanhTien= giaban*soluong*

```
CREATE VIEW view_donhang  
AS
```

```
SELECT dondathang.sohoadon,makhachhang,manhanvien,  
ngaydathang,ngaygiaohang,ngaychuyenhang,  
noigiaohang,mahang,
```

```

giaban*soluong as thanhtien,
mucgiamgia
FROM dondathang INNER JOIN chitietdathang
ON dondathang.sohoadon = chitietdathang.sohoadon

```

1.3 Tạo ra view NHANVIEN_VIEW với tùy chọn WITH CHECK OPTION.

```

CREATE VIEW NHANVIEN_VIEW
AS
SELECT ten, tuoi
FROM NHANVIEN
WHERE tuoi IS NOT NULL
WITH CHECK OPTION;

```

1.4 Cập nhật *tuoi* cho nhân viên có tên là *Thanh*.

```

SQL > UPDATE NHANVIEN_VIEW
SET AGE = 35
WHERE ten = 'Thanh';

```

2/ Tạo các thủ tục lưu trữ sau

2.1 Tạo thủ tục lưu trữ để thông qua thủ tục này có thể bổ sung thêm một bản ghi mới cho bảng MATHANG (thủ tục phải thực hiện kiểm tra tính hợp lệ của dữ liệu cần bổ sung: không trùng khoá chính và đảm bảo toàn vẹn tham chiếu)

```

CREATE PROCEDURE sp_insert_mathang(

```

```

    @mahang    NVARCHAR(10),
    @tenhang   NVARCHAR(50),
    @macongty  NVARCHAR(10) = NULL,
    @maloihang INT = NULL,
    @soluong   INT = 0,
    @donvitinh NVARCHAR(20) = NULL,
    @giahang   money = 0)

```

AS

```

IF NOT EXISTS(SELECT mahang FROM mathang
WHERE mahang=@mahang)

```

```

IF (@macongty IS NULL OR EXISTS(SELECT macongty
FROM nhacungcap
WHERE macongty=@macongty))

```

AND

```

(@maloihang IS NULL OR
EXISTS(SELECT maloihang FROM loaihang
WHERE maloihang=@maloihang))

```

```

INSERT INTO mathang

```



```
VALUES(@mahang,@tenhang,
       @macongy,@maloaihang,
       @soluong,@donvitinh,@giahang)
```

2.2 Tạo thủ tục lưu trữ có chức năng thống kê tổng số lượng hàng bán được của một mặt hàng có mã bất kỳ (mã mặt hàng cần thống kê là tham số của thủ tục).
 Create PROCEDURE sp_thongkebanhang(@mahang NVARCHAR(10))

```
AS
```

```
SELECT mathang.mahang,tenhang,
       SUM(chitietdathang.soluong) AS 'Tong so luong'
FROM mathang LEFT JOIN chitietdathang
ON mathang.mahang=chitietdathang.mahang
WHERE mathang.mahang=@mahang
GROUP BY mathang.mahang,tenhang
//thực thi
exec sp_thongkebanhang '001'
```

3/ Tạo các trigger sau

3.1 Khi thay đổi cột SL trên bảng CTHD, chúng ta phải cập nhật lại dữ liệu cho cột SoTon trong bảng MATHANG

```
Create Trigger tr_SuaCTHD On CTHD
For Update As
Declare @D int
Select @D = Count(*)
From MatHang a, DELETED b, INSERTED c
Where a.MaMH = b.MaMH
      And a.MaMH = c.MaMH
      And SoTon + b.SL - c.SL <0
If (@D > 0 )
Begin
Print 'Không đủ hàng để bán'
RollBack Tran
Return
End
--Cập nhật số lượng tồn trong bảng MATHANG
Update MatHang Set SoTon = SoTon + b.SL - c.SL
From MatHang a, DELETED b, INSERTED c
Where a.MaMH = b.MaMH And a.MaMH = c.MaMH
```

3.2 Khi xóa một dòng trong bảng CTHD, chúng ta phải tăng số lượng của bảng MATHANG

```
Create Trigger tr_XoaCTHD On CTHD
```

```

For Delete
As
Update MATHANG Set SoTon = SoTon + SL From DELETED
Where MATHANG.MaMH = DELETED.MaMH

```

3.3 Kiểm tra ràng buộc một CTHD không chứa nhiều hơn 10 SoHD. Trong hình, chúng tôi đang chọn SoHD là 10249. Hiện tại trong bảng CTHD trước khi thêm đang có 2. Dòng màu vàng là dữ liệu chúng ta muốn thêm vào

```

Create Trigger tr_SoCTHD On CTHD For Insert
As
If (Select Count(a.SoHD)
From CTHD a Inner Join INSERTED b On a.SoHD = b.SoHD) > 10
Begin
Print 'So CTHD Khong the > 10'
RollBack Tran
End

```

Những trọng tâm cần chú ý:

- Trình bày các lệnh về khung nhìn, Stored procedure, Trigger
- Trình bày mở thành công CSDL để thực hiện các thao tác về khung nhìn, Stored procedur, Trigger
- Thực hiện đúng việc tạo khung nhìn theo yêu cầu
- Thực hiện đúng việc tạo Thủ tục lưu trữ theo yêu cầu
- Thực hiện đúng việc tạo Trigger theo yêu cầu
- Sử dụng lệnh sqlcmd để liệt kê danh sách CSDL trên server

Bài mở rộng và nâng cao

Sử dụng cơ sở dữ liệu QuanLyBanHang

1/ Tạo các view sau

Câu 1. Tạo view có tên **vw_CTPNHAP** bao gồm các thông tin sau: số phiếu nhập hàng, mã vật tư, số lượng nhập, đơn giá nhập, thành tiền nhập.

Câu 2. Tạo view có tên **vw_CTPNHAP_VT** bao gồm các thông tin sau: số phiếu nhập hàng, mã vật tư, tên vật tư, số lượng nhập, đơn giá nhập, thành tiền nhập.

Câu 3. Tạo view có tên **vw_CTPNHAP_VT_PN** bao gồm các thông tin sau: số phiếu nhập hàng, ngày nhập hàng, số đơn đặt hàng, mã vật tư, tên vật tư, số lượng nhập, đơn giá nhập, thành tiền nhập.

Câu 4. Tạo view có tên **vw_CTPNHAP_VT_PN_DH** bao gồm các thông tin sau: số phiếu nhập hàng, ngày nhập hàng, số đơn đặt hàng, mã nhà cung cấp, mã vật tư, tên vật tư, số lượng nhập, đơn giá nhập, thành tiền nhập.

Câu 5. Tạo view có tên **vw_CTPNHAP_loc** bao gồm các thông tin sau: số phiếu nhập hàng, mã vật tư, số lượng nhập, đơn giá nhập, thành tiền nhập. Và chỉ liệt kê các chi tiết nhập có số lượng nhập > 5.

Câu 6. Tạo view có tên **vw_CTPNHAP_VT_loc** bao gồm các thông tin sau: số phiếu nhập hàng, mã vật tư, tên vật tư, số lượng nhập, đơn giá nhập, thành tiền nhập. Và chỉ liệt kê các chi tiết nhập vật tư có đơn vị tính là Bộ.

Câu 7. Tạo view có tên **vw_CTPXUAT** bao gồm các thông tin sau: số phiếu xuất hàng, mã vật tư, số lượng xuất, đơn giá xuất, thành tiền xuất.

Câu 8. Tạo view có tên **vw_CTPXUAT_VT** bao gồm các thông tin sau: số phiếu xuất hàng, mã vật tư, tên vật tư, số lượng xuất, đơn giá xuất.

Câu 9. Tạo view có tên **vw_CTPXUAT_VT_PX** bao gồm các thông tin sau: số phiếu xuất hàng, tên khách hàng, mã vật tư, tên vật tư, số lượng xuất, đơn giá xuất.

2/ Tạo các procedure sau

Câu 1. Tạo Stored procedure (SP) cho biết tổng số lượng cuối của vật tư với mã vật tư là tham số vào.

Câu 2. Tạo SP cho biết tổng tiền xuất của vật tư với mã vật tư là tham số vào.

Câu 3. Tạo SP cho biết tổng số lượng đặt theo số đơn hàng với số đơn hàng là tham số vào.

Câu 4. Tạo SP dùng để thêm một đơn đặt hàng.

Câu 5. Tạo SP dùng để thêm một chi tiết đơn đặt hàng.

3/ Tạo các trigger sau

Câu 1. Tạo trigger khi update hoặc insert dữ liệu của bảng **TONKHO**, cột **SLCuoi** được tính theo công thức **SLCuoi = SLDau + TongSLN – TongSLX**

Câu 2. Tạo trigger khi update cột **SLNhap** của bảng **CTPNHAP** thì cột **TongSLN** của bảng **TONKHO** được cập nhật theo.

Câu 3. Tạo trigger khi update cột **SLXuat** của bảng **CTPXUAT** thì cột **TongSLX** của bảng **TONKHO** được cập nhật theo.

Câu 4. Tạo trigger khi update hoặc insert cột **SLNhap** của bảng **CTPNHAP** thì tổng số lượng nhập \leq tổng số lượng đặt của bảng **CTDONDH**.

Câu 5. Tạo trigger khi delete một **CTPNHAP** thì cột **TongSLN** của bảng **TONKHO** được cập nhật tương ứng.

Yêu cầu đánh giá kết quả học tập

Nội dung

- Về kiến thức:
 - + Trình bày các thao tác trên View
 - + Trình bày các thao tác về Stored procedure
 - + Trình bày các thao tác về Trigger
- Về kỹ năng:
 - + Thực hiện đúng các thao tác trên View
 - + Sử dụng đúng các lệnh thao tác về Stored procedure
 - + Thực hiện đúng các thao tác về Trigger
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

Phương pháp

- Về kiến thức: Đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp.
- Về kỹ năng: Đánh giá kỹ năng thực hành các thao tác trên View, Stored procedure, Trigger.
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

BÀI 6 QUẢN LÝ BẢO MẬT VÀ NGƯỜI DÙNG

Mã bài: MĐ16-06

Mục tiêu:

- Thiết lập và tạo được Login và user;
- Thiết lập bảo mật dữ liệu SQL Server;
- Bảo mật SQL Server.
- Thiết lập xác thực những đăng nhập với 2 cơ chế, đó là SQL Server authentication và Windows authentication.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Khái niệm cơ bản về bảo mật

Bảo mật là một trong những yếu tố đóng vai trò quan trọng đối với sự sống còn của cơ sở dữ liệu. Hầu hết các hệ quản trị cơ sở dữ liệu thương mại hiện nay đều cung cấp khả năng bảo mật cơ sở dữ liệu với những chức năng như:

Cấp phát quyền truy cập cơ sở dữ liệu cho người dùng và các nhóm người dùng

Phát hiện và ngăn chặn những thao tác trái phép của người sử dụng trên cơ sở dữ liệu.

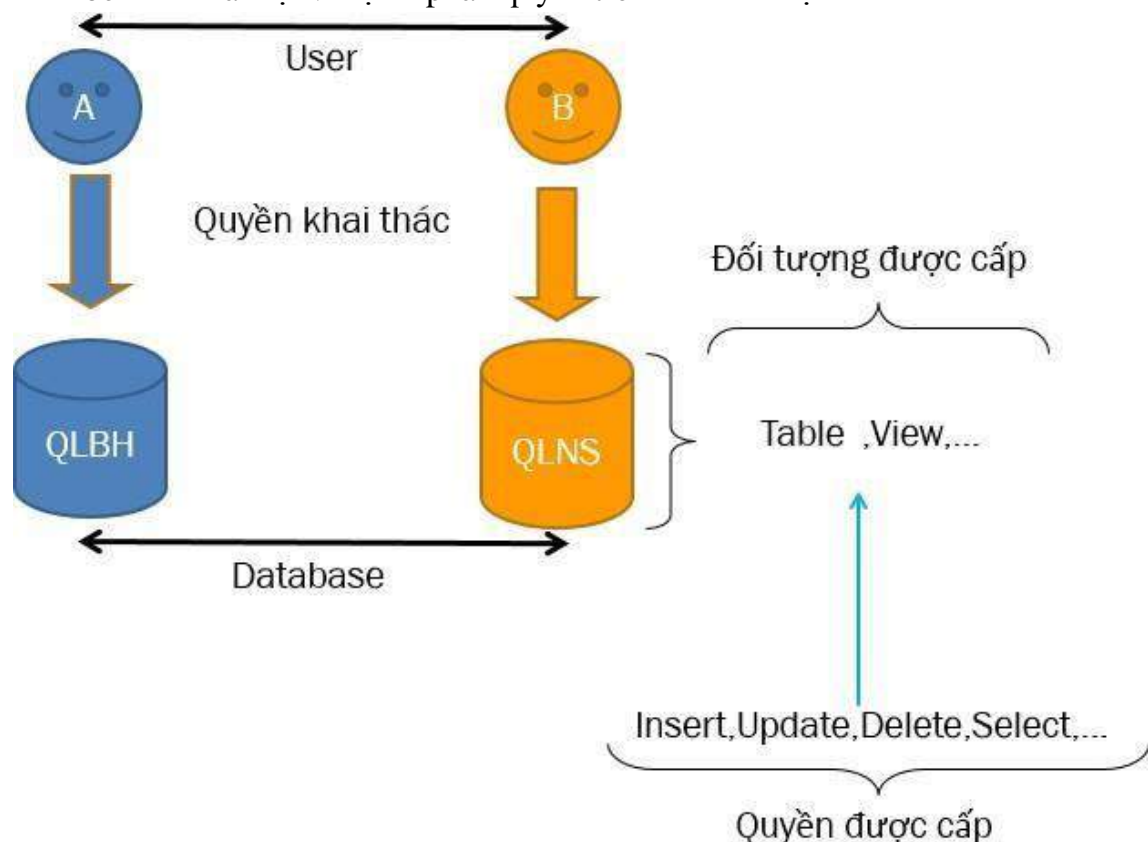
Cấp phát quyền sử dụng các câu lệnh, các đối tượng cơ sở dữ liệu đối với người dùng.

Thu hồi (huỷ bỏ) quyền của người dùng.

Bảo mật cơ sở dữ liệu sql server là giải pháp cho phép các quản trị viên cơ sở dữ liệu thiết lập quyền hạn cho người dùng hoặc nhóm người dùng khai thác cơ sở dữ liệu.

Người dùng hoặc nhóm người dùng sau khi được cấp quyền, có thể đăng nhập vào hệ thống và thực hiện các quyền hạn mà mình được cấp.

Bảo mật cơ sở dữ liệu sql server giúp phân quyền người dùng trên cơ sở dữ liệu. Hình bên dưới là một ví dụ về phân quyền trên cơ sở dữ liệu



Hình 6.1 Mô hình quản trị CSDL

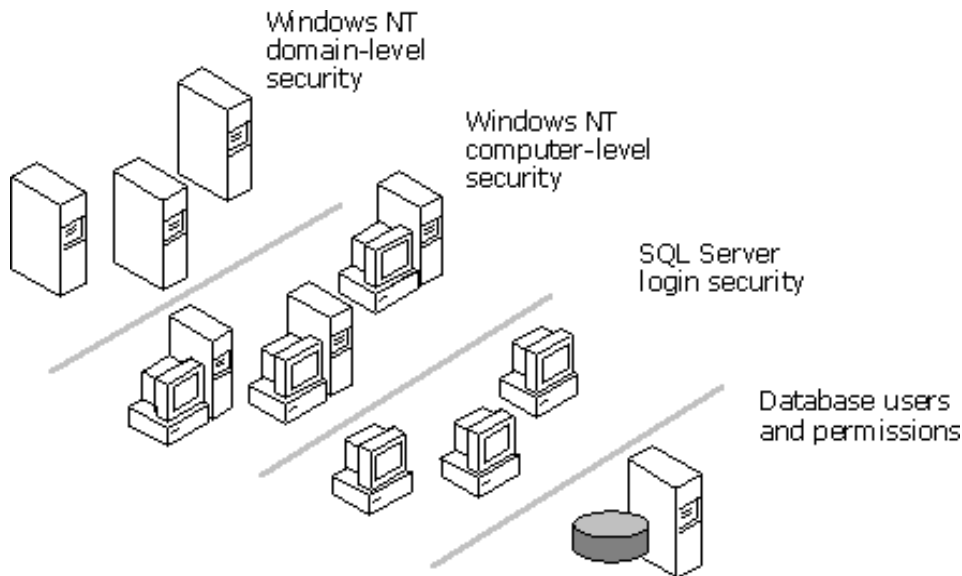
2 Các chế độ bảo mật

SQL Server có 2 chế độ bảo mật:

- + Windows Authentication Mode (Windows Authentication)
- + Mixed Mode (Windows Authentication and SQL Server Authentication)

2.1 Windows Authentication

Là chế độ bảo mật mà những User truy nhập SQL Server phải là những User của Windows. Khi Server đặt ở chế độ bảo mật này, những User phải là những User được Windows quản lý mới được truy nhập.



Hình 6.2 Mô hình Windows Authentication

Nhìn trên hình ta thấy khi thực hiện chế độ này người sử dụng muốn khai thác SQL Server phải thông qua 4 bước xác thực (1- Domain, 2- Computer, 3- SQL Server, 4- Database).

2.2 SQL Server Authentication

Khi thiết lập ở chế độ bảo mật này, những User được quyền khai thác phải là những User do quản trị SQL Server tạo ra, mà những user của Windows không được khai thác.

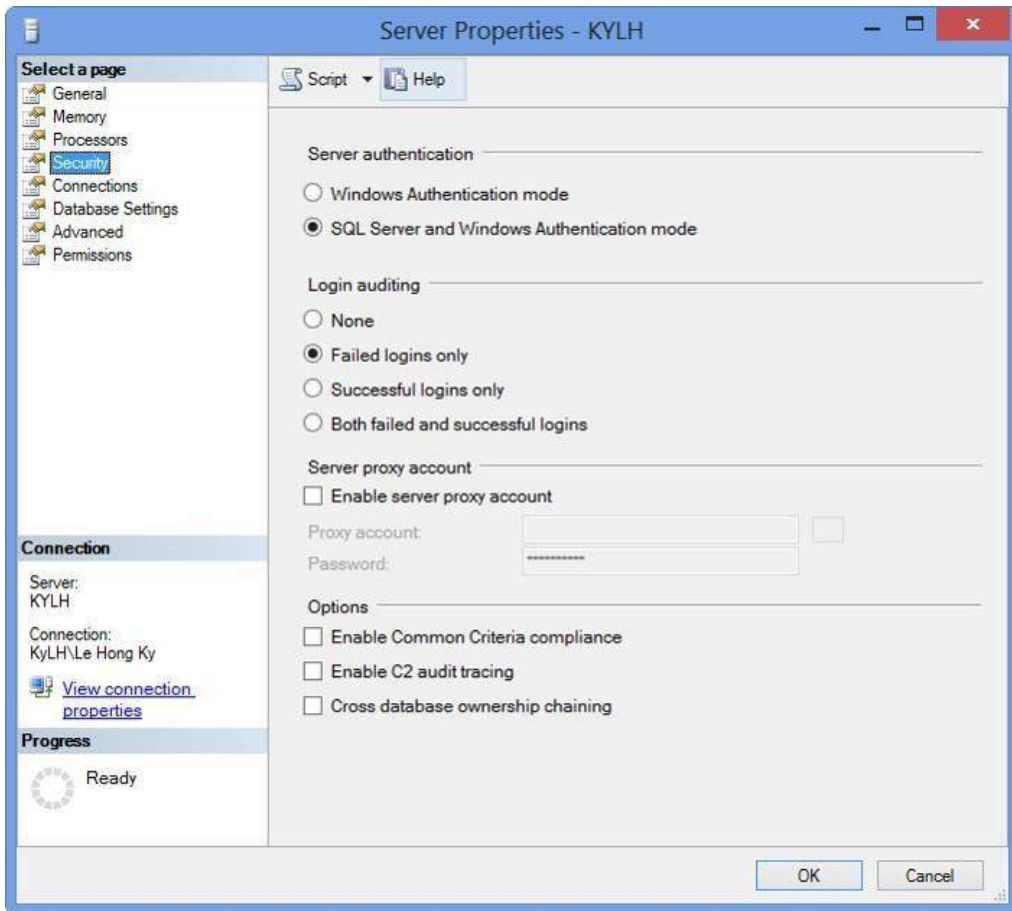
Tuy nhiên, SQL Server cho phép thiết lập hai chế độ Windows Authentication Mode (Windows Authentication) và Mixed Mode (Windows Authentication and SQL Server Authentication), chế độ Mixed Mode là sự kết hợp của Windows Authentication và SQL Server Authentication, ở chế độ này cả user của Windows và SQL Server đều có thể thiết lập để truy nhập SQL Server.

2.3 Thiết lập chế độ chứng thực Windows và SQL Server

Bước 1: Đăng nhập vào SQL Server bằng chứng thực Windows với quyền admin

Bước 2: Chuột phải lên trên cùng -> chọn Properties

Bước 3: Chọn SQL Server and Windows Authentication mode



Hình 6.3 Chứng thực SQL Server and Windows Authentication mode

Bước 4: Khởi động lại SQL Server



Hình 6.4 Khởi động lại SQL Server

3. Quản lý người dùng

Người dùng trong SQL Server được chia thành 2 mức: Người truy nhập vào SQL Server gọi là Login, người khai thác CSDL gọi là User.

Login là đối tượng được quyền truy nhập vào SQL Server, tùy theo chế độ bảo mật của SQL Server mà những login là account của Windows NT hay của SQL Server, login do Server quản lý trực tiếp.

User là đối tượng khai thác CSDL, nếu login chỉ xác định truy nhập vào SQL Server thì User là login ID tham gia khai thác CSDL, user do CSDL quản lý trực tiếp.

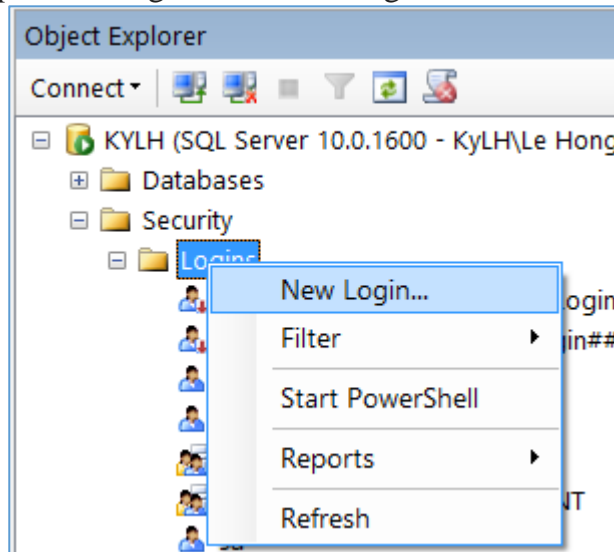
3.1 Tạo tài khoản đăng nhập cho người dùng (Login)

3.1.1 Thực hiện bằng giao diện

Đăng nhập vào hệ thống với quyền quản trị (Administrator) và thực hiện một trong hai cách sau:

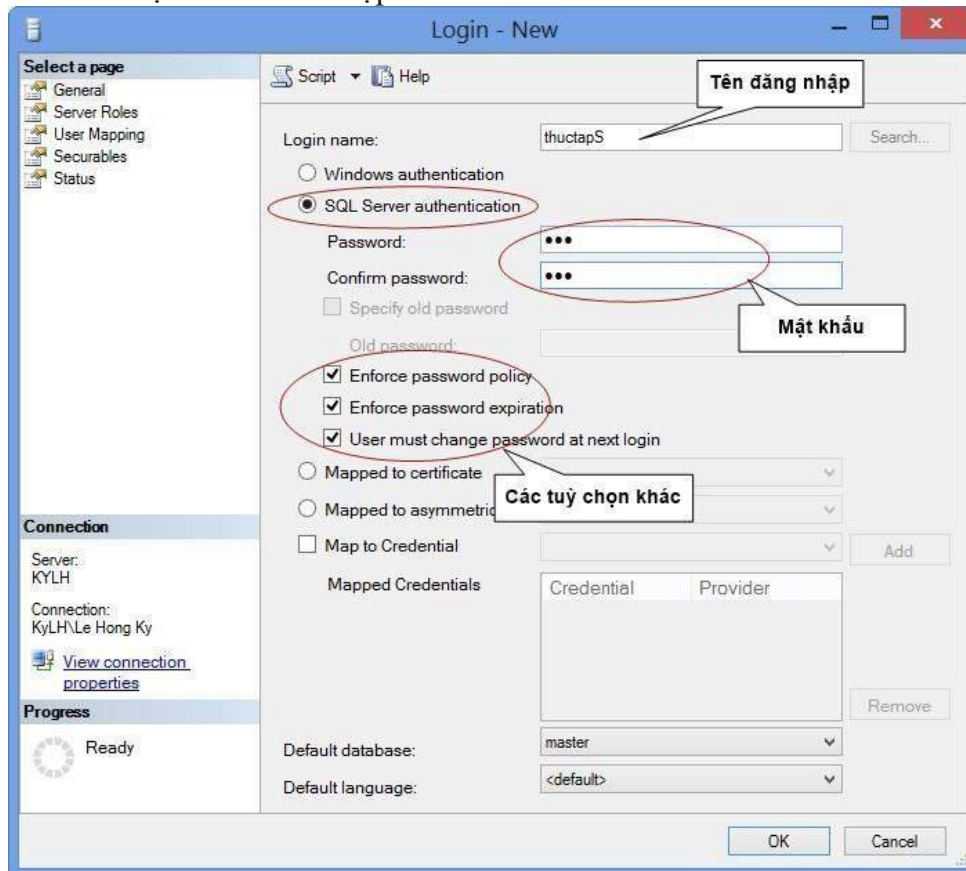
Bước 1: Kết nối vào SQL Server

Bước 2: Click chuột phải vào login, chọn New login



Hình 6.5 Tạo mới login

Bước 3: Tạo tên và thiết lập các tham số



Hình 6.6 Tạo và thiết lập login

3.1.2 Thực hiện bằng lệnh

Cách 1: Tạo tài khoản người dùng, ta dùng thủ tục với cú pháp sau

```
Sp_addlogin [@login=] <'Tên người dùng'>
```

```
[,[@Password=] <'Mật khẩu'>]
```

```
[,[@defdb=] <'Tên tệp CSDL'>]
```

Ví dụ 1: Tạo tài khoản người dùng có các tham số sau:

```
EXEC SP_ADDLOGIN 'HA', 'HA1', 'QL_BanHag'
```

Để thay đổi mật khẩu người dùng ta dùng thủ tục với cú pháp sau:

```
SP_Password[@old=] <'Mật khẩu cũ',>  
  [@New=] <'Mật khẩu mới'>  
  [, @login=] <'Tên người dùng'>
```

Ví dụ 2: Thay đổi mật khẩu người dùng
SP_Password 'HA1', 'HAI', 'HA'

Cách 2:

Tạo tài khoản đăng nhập sử dụng xác thực Windows Authentication

```
CREATE LOGIN [ten_mien\ten_dangnhap]  
FROM WINDOWS  
[ WITH DEFAULT_DATABASE = ten_cosodulieu  
| DEFAULT_LANGUAGE = ten_ngonngu];
```

Tạo tài khoản đăng nhập sử dụng xác thực SQL Server Authentication

```
CREATE LOGIN ten_dangnhap  
WITH PASSWORD = { 'matkhau' | matkhau_bam HASHED } [  
MUST_CHANGE]  
[, SID = giatri_duynhat  
| DEFAULT_DATABASE = ten_cosodulieu  
| DEFAULT_LANGUAGE = ten_ngonngu  
| CHECK_EXPIRATION = { ON | OFF }  
| CHECK_POLICY = { ON | OFF }  
| CREDENTIAL = ten_chungthuc];
```

Tạo tài khoản đăng nhập từ chứng nhận

```
CREATE LOGIN ten_dangnhap  
FROM CERTIFICATE ten_chungnhan;
```

Tạo tài khoản đăng nhập từ khóa bất đối xứng

```
CREATE LOGIN ten_dangnhap  
FROM ASYMMETRIC KEY ten_khoabatdoixung;
```

3.2 Cấp phát quyền truy cập vào CSDL

Để cấp phát quyền truy cập cho người dùng vào CSDL ta sử dụng thủ tục có cú pháp sau:

```
SP_grantdbaccess [ @login= ] <'Tên người dùng'>  
  [, @Name_in_db= ] <'Tên bí danh'>
```

Tham số @Name_in_db là bí danh tên của tài khoản người dùng thành một tên khác khi truy cập vào CSDL chỉ định.

Ví dụ: Cấp phát quyền cho người dùng

```
SP_grantdbaccess 'HA', 'HABD'
```

Loại bỏ quyền sử dụng

Cú pháp:

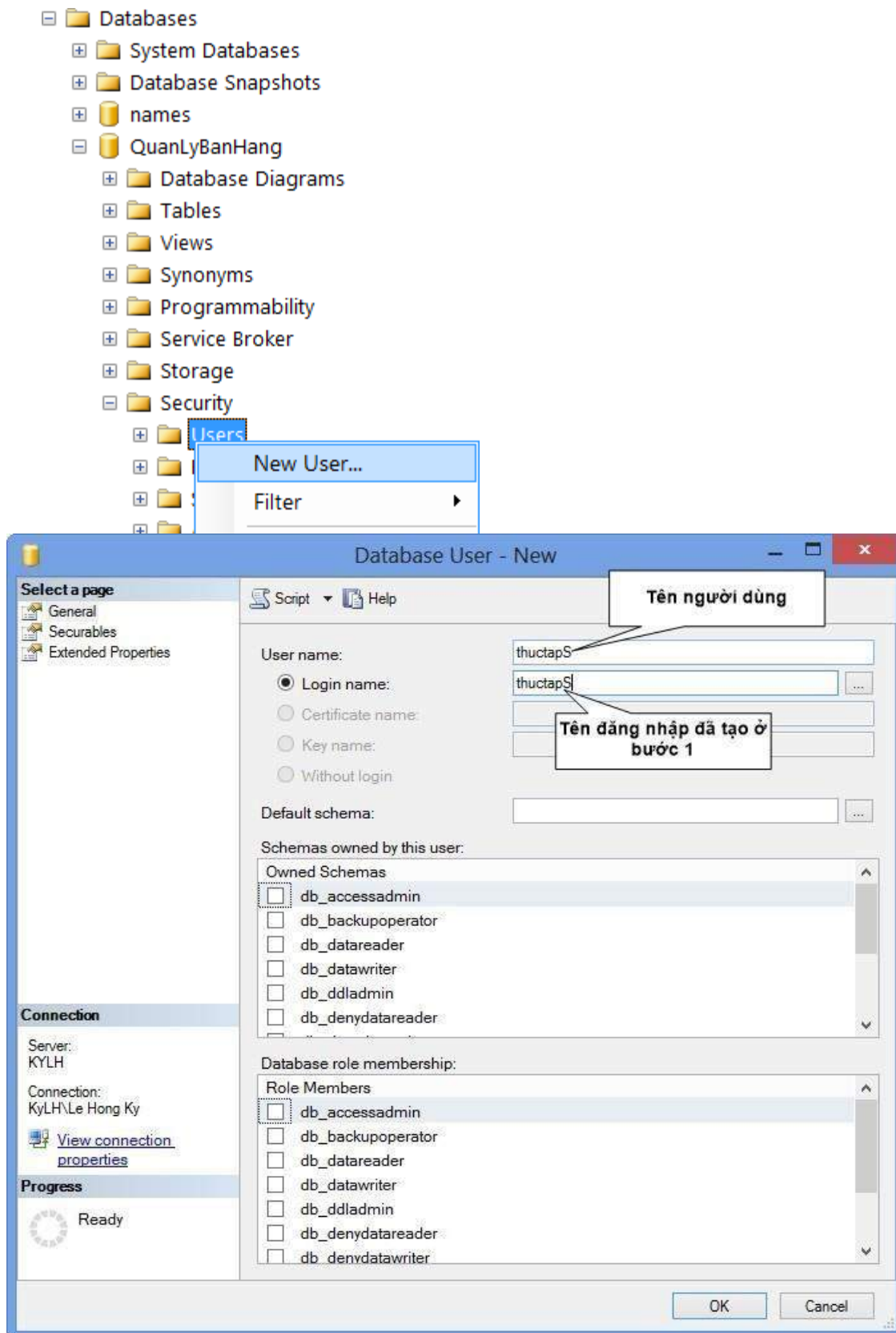
```
SP_revokedbaccess [ @Name_in_db = ] <'Tên bí danh'>
```

Ví dụ: Hủy bỏ quyền của người dùng

```
SP_revokedbaccess 'HABD'
```

4. Quyền người dùng

4.1 Tạo người sử dụng từ giao diện



Hình 6.7. Tạo và thiết lập User

4.2 Tạo người sử dụng từ lệnh

Cú pháp để tạo người dùng trong MS SQL Server.

`Create user <username> for login <loginname>`

Ví dụ:

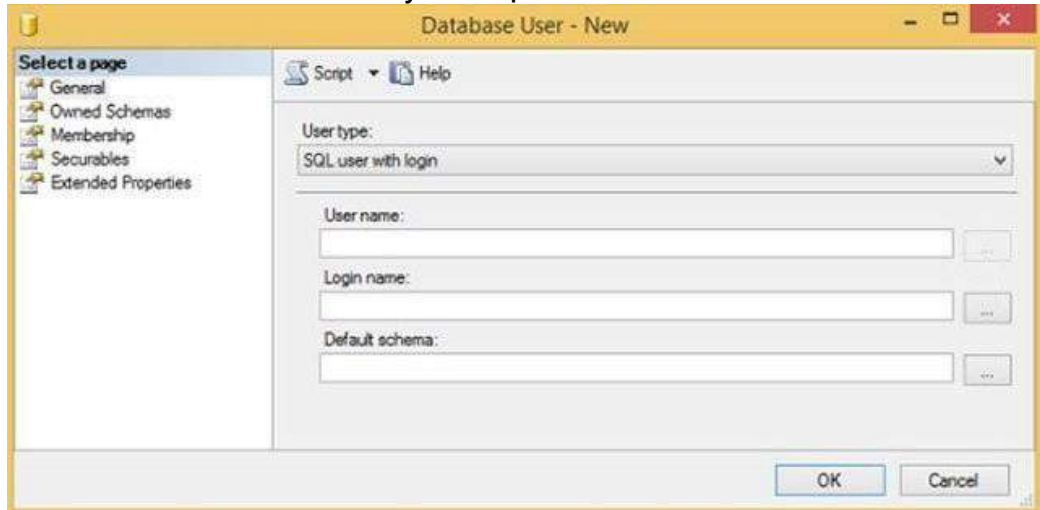
Để tạo người dùng có tên *TestUser* với tên đăng nhập là *TestLogin* trên CSDL *TestDB*, chạy truy vấn dưới đây.

```
create user TestUser for login TestLogin
```

Cách 2: Sử dụng SQL Server Management Studio SSMS

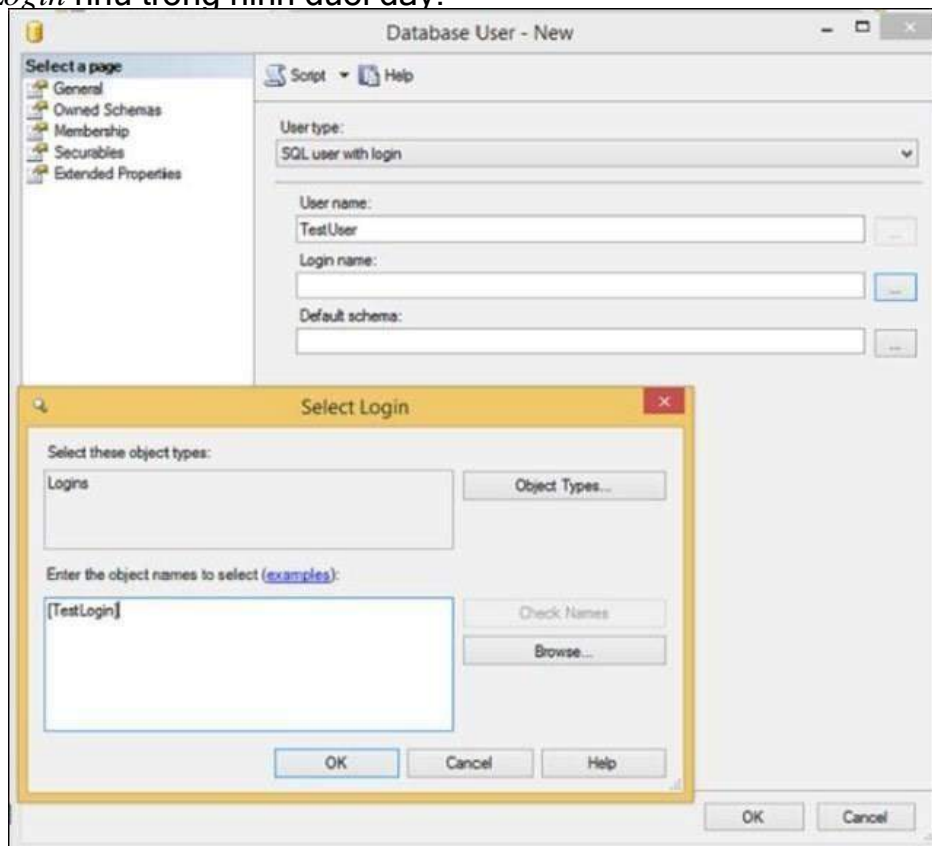
Lưu ý: Trước hết cần tạo Login bằng bất kì cái tên nào trước khi tạo tài khoản người dùng. Hãy dùng tên Login là **TestLogin**.

Bước 1: Kết nối tới SQL Server và mở rộng thư mục CSDL. Sau đó mở CSDL có tên *TestDB*, nơi ta sẽ tạo tài khoản người dùng, tìm thư mục **Security**. Tại đây tìm thư mục **Users** hoặc **Logins** (trên bản SQL Server mới) và chọn **New Login**. Khi đó màn hình dưới đây sẽ hiện ra.



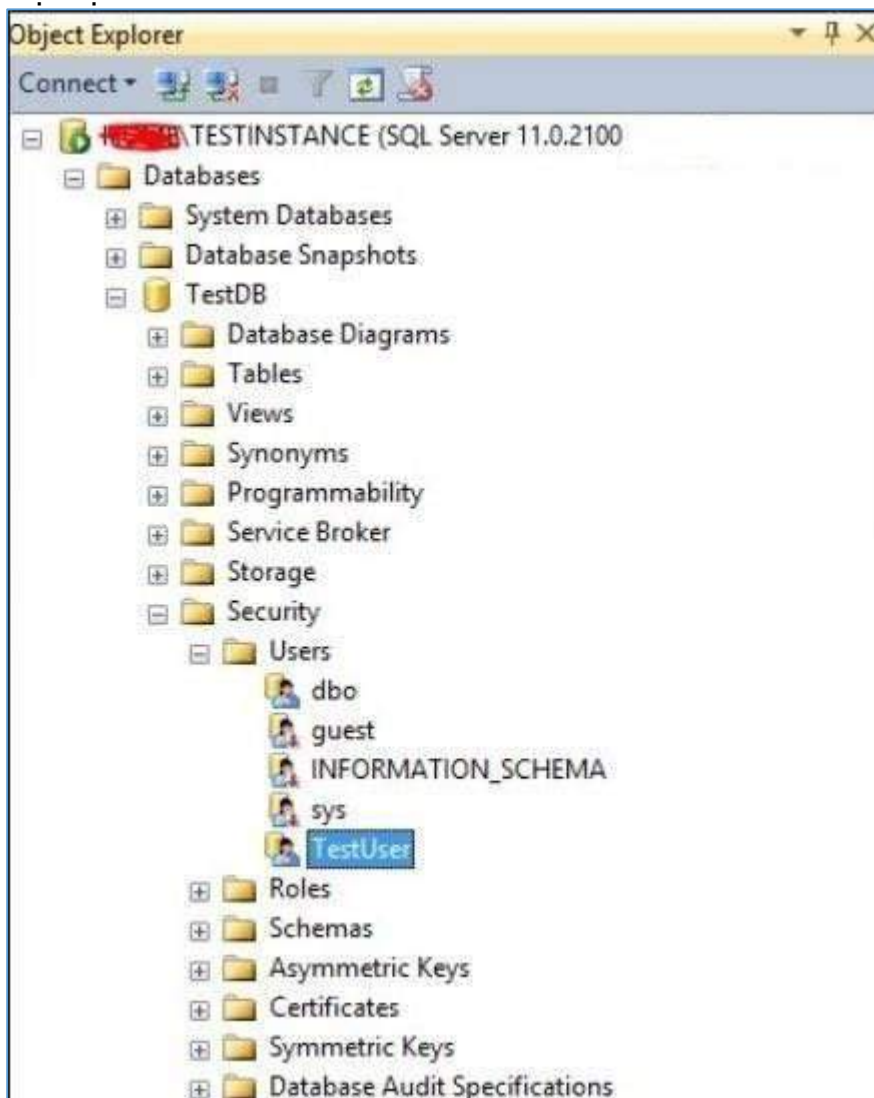
Hình 6.8 Màn hình tạo người dùng mới cho MS SQL Server

Bước 2: Điền tên *TestUser* cho tên người dùng và chọn tên Login có tên *TestLogin* như trong hình dưới đây.



hình 6.9 Tạo tài khoản người dùng mới

Bước 3: Click **OK** để hoàn tất. **Refresh** lại thư mục, bạn sẽ thấy tên người dùng mới được tạo.



Hình 6.10 Tạo thành công người dùng mới trên MS SQL Server

Bảng danh sách các quyền thao tác trên các đối tượng của CSDL

Quyền	Diễn giải
SELECT	Cho phép người dùng nhìn thấy dữ liệu, nếu người dùng có quyền này thì họ chỉ có thể thực thi những phát biểu Select để truy vấn dữ liệu trên các bảng hay 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.
UPDATE	Cho phép người sử dụng chỉnh sửa dữ liệu trong CSDL, với quyền Update người sử dụng có thể cập nhật dữ liệu bằng phát biểu Update
DELETE	Quyền này cho phép xóa dữ liệu trong CSDL với quyền Delete người sử dụng có thể xóa dữ liệu bằng phát biểu Delete.
REFERENCE	Cho phép người sử dụng thêm dữ liệu và bảng có khóa ngoài, nếu người dùng có quyền này họ có thể thực hiện phát biểu INSERT, trong trường hợp SQL Server thì quyền này không liên quan đến quyền truy vấn SELECT.

EXECUTE	Quyền này cho phép người sử dụng thực thi thủ tục trong CSDL, Với quyền này người sử dụng có thể xóa mẫu tin bằng phát biểu thực thi thủ tục
---------	--

4.1 Cấp phát quyền

Câu lệnh GRANT được sử dụng để cấp phát quyền cho người dùng hay nhóm người dùng trên các đối tượng cơ sở dữ liệu. Câu lệnh này thường được sử dụng trong các trường hợp sau:

- Người sở hữu đối tượng cơ sở dữ liệu muốn cho phép người dùng khác quyền sử dụng những đối tượng mà anh ta đang sở hữu.
- Người sở hữu cơ sở dữ liệu cấp phát quyền thực thi các câu lệnh (như CREATE TABLE, CREATE VIEW,...) cho những người dùng khác.

a) Cấp phát quyền cho người dùng trên các đối tượng cơ sở dữ liệu

Chỉ có người sở hữu cơ sở dữ liệu hoặc người sở hữu đối tượng cơ sở dữ liệu mới có thể cấp phát quyền cho người dùng trên các đối tượng cơ sở dữ liệu. Câu lệnh GRANT trong trường hợp này có cú pháp như sau:

```
GRANT ALL | các_quyền_cấp_phát
[(<danh_sách_cột>)] ON tên_bảng | tên_khung_nhìn
|ON tên_bảng | tên_khung_nhìn [(<danh_sách_cột>)]
|ON tên_thủ_tục
|ON tên_hàm
TO danh_sách_người_dùng | nhóm_người_dùng
[WITH GRANT OPTION]
```

Trong đó:

- ALL: Cấp phát tất cả các quyền cho người dùng trên đối tượng cơ sở dữ liệu được chỉ định. Các quyền có thể cấp phát cho người dùng bao gồm:
- Đối với bảng, khung nhìn, và hàm trả về dữ liệu kiểu bảng: SELECT, INSERT, DELETE, UPDATE và REFERENCES.
- Đối với cột trong bảng, khung nhìn: SELECT và UPDATE.
- Đối với thủ tục lưu trữ: EXECUTE.
- Trong các quyền được đề cập đến ở trên, quyền REFERENCES được sử dụng nhằm cho phép tạo khóa ngoài tham chiếu đến bảng cấp phát.

<i>các_quyền_cấp_phát</i>	Danh sách các quyền cần cấp phát cho người dùng trên đối tượng cơ sở dữ liệu được chỉ định. Các quyền được phân cách nhau bởi dấu phẩy
<i>tên_bảng</i> <i>tên_khung_nhìn</i>	Tên của bảng hoặc khung nhìn cần cấp phát quyền.
<i>danh_sách_cột</i>	Danh sách các cột của bảng hoặc khung nhìn cần cấp phát quyền.
<i>tên_thủ_tục</i>	Tên của thủ tục được cấp phát cho người dùng.
<i>tên_hàm</i>	Tên hàm (do người dùng định nghĩa) được cấp phát quyền.
<i>danh_sách_người_dùng</i>	Danh sách tên người dùng nhận quyền được cấp phát. Tên của các người dùng được phân cách nhau bởi dấu phẩy.

WITH GRANT OPTION Cho phép người dùng chuyển tiếp quyền cho người dùng khác.

Ví dụ: Cấp phát cho người dùng có tên Ha quyền thực thi các câu lệnh SELECT, INSERT và UPDATE trên bảng Hang.

```
GRANT SELECT, INSERT, UPDATE
ON Hang
TO thuchanh
```

Cho phép người dùng Ha quyền xem các thuộc tính (MaH,TenH,SoLuong)

```
GRANT SELECT
ON Hang(MaH,TenH,SoLuong) ON Hang) TO thuchanh
```

Với quyền được cấp phát như trên, người dùng **Ha** có thể thực hiện câu lệnh sau trên bảng **Hang**

```
SELECT MaH, TenH, SoLuong
FROM Hang
```

Nhưng câu lệnh dưới đây lại không thể thực hiện được

```
SELECT * FROM Hang
```

Ví dụ 1: Câu lệnh dưới đây cấp phát cho người dùng **Ha** các quyền SELECT, INSERT, UPDATE, DELETE VÀ REFERENCES trên bảng DIEMTHI

```
GRANT ALL ON DIEMTHI TO Ha
```

Khi ta cấp phát quyền nào đó cho một người dùng trên một đối tượng cơ sở dữ liệu, người dùng đó có thể thực thi câu lệnh được cho phép trên đối tượng đã cấp phát. Tuy nhiên, người dùng đó không có quyền cấp phát những quyền mà mình được phép cho những người sử dụng khác. Trong một số trường hợp, khi ta cấp phát quyền cho một người dùng nào đó, ta có thể cho phép người đó chuyển tiếp quyền cho người dùng khác bằng cách chỉ định tùy chọn WITH GRANT OPTION trong câu lệnh GRANT.

Ví dụ 2: Cho phép người dùng **Ha** quyền xem dữ liệu trên bảng **Hang** đồng thời có thể chuyển tiếp quyền này cho người dùng khác

```
GRANT SELECT On
Hang
WITH GRANT OPTION
```

b) Cấp phát quyền thực thi các câu lệnh

Ngoài chức năng cấp phát quyền cho người sử dụng trên các đối tượng cơ sở dữ liệu, câu lệnh GRANT còn có thể sử dụng để cấp phát cho người sử dụng một số quyền trên hệ quản trị cơ sở dữ liệu hoặc cơ sở dữ liệu. Những quyền có thể cấp phát trong trường hợp này bao gồm:

- Tạo cơ sở dữ liệu: CREATE DATABASE.
- Tạo bảng: CREATE TABLE
- Tạo khung nhìn: CREATE VIEW
- Tạo thủ tục lưu trữ: CREATE PROCEDURE
- Tạo hàm: CREATE FUNCTION
- Sao lưu cơ sở dữ liệu: BACKUP DATABASE

Câu lệnh GRANT sử dụng trong trường hợp này có cú pháp như sau:

```
GRANT ALL | danh_sách_câu_lệnh
TO danh_sách_người_dùng
```

Ví dụ: Để cấp phát quyền tạo bảng và khung nhìn cho người dùng có tên là **Ha** ta sử dụng câu lệnh như sau:

GRANT CREATE TABLE,CREATE VIEW TO Ha

Với câu lệnh GRANT, ta có thể cho phép người sử dụng tạo các đối tượng cơ sở dữ liệu trong cơ sở dữ liệu. Đối tượng cơ sở dữ liệu do người dùng nào tạo ra sẽ do người đó sở hữu và do đó người này có quyền cho người dùng khác sử dụng đối tượng và cũng có thể xóa bỏ (DROP) đối tượng do mình tạo ra.

Khác với trường hợp sử dụng câu lệnh GRANT để cấp phát quyền trên đối tượng cơ sở dữ liệu, câu lệnh GRANT trong trường hợp này không thể sử dụng tùy chọn WITH GRANT OPTION, tức là người dùng không thể chuyển tiếp được các quyền thực thi các câu lệnh đã được cấp phát.

4.2 Thu hồi quyền

Câu lệnh REVOKE được sử dụng để thu hồi quyền đã được cấp phát cho người dùng. Tương ứng với câu lệnh GRANT, câu lệnh REVOKE được sử dụng trong hai trường hợp:

- Thu hồi quyền đã cấp phát cho người dùng trên các đối tượng cơ sở dữ liệu.
- Thu hồi quyền thực thi các câu lệnh trên cơ sở dữ liệu đã cấp phát cho người dùng.

a) Thu hồi quyền trên đối tượng cơ sở dữ liệu

Cú pháp câu lệnh REVOKE sử dụng để thu hồi quyền đã cấp phát trên đối tượng cơ sở dữ liệu có cú pháp như sau:

```
REVOKE ALL| các_ quyền_ cần_ thu_ hồi  
[(danh_sách_cột)] ON tên_bảng | tên_khung_nhìn  
|ON tên_bảng | tên_khung_nhìn [(danh_sách_cột)]  
|ON tên_thủ_tục  
|ON tên_hàm  
FROM danh_sách_người_dùng  
[CASCADE]
```

Câu lệnh REVOKE có thể sử dụng để thu hồi một số quyền đã cấp phát cho người dùng hoặc là thu hồi tất cả các quyền (ALL).

Ví dụ: Thu hồi quyền thực thi lệnh INSERT trên bảng Hang đối với người dùng Ha.

```
REVOKE INSERT ON Hang  
FROM Ha
```

Giả sử người dùng Ha đã được cấp phát quyền xem dữ liệu trên các cột MaH, TenH, SoLuong của bảng Hang, câu lệnh dưới đây sẽ thu hồi quyền đã cấp phát trên cột SoLuong (chỉ cho phép xem dữ liệu trên cột MaH, TenH)

```
REVOKE SELECT  
ON sinhvien(SoLuong)  
FROM Ha
```

Khi ta sử dụng câu lệnh REVOKE để thu hồi quyền trên một đối tượng cơ sở dữ liệu từ một người dùng nào đó, chỉ những quyền mà ta đã cấp phát trước đó mới được thu hồi, những quyền mà người dùng này được cho phép bởi những người dùng khác vẫn còn có hiệu lực. Nói cách khác, nếu hai người dùng khác nhau cấp phát cùng các quyền trên cùng một đối tượng cơ sở dữ liệu cho một người dùng khác, sau đó người thứ nhất thu hồi lại quyền đã cấp phát thì những quyền mà người dùng thứ hai cấp phát vẫn có hiệu lực.

Ví dụ: Giả sử trong cơ sở dữ liệu ta có 3 người dùng là A, B và C. A và B đều có quyền sử dụng và cấp phát quyền trên bảng R. A thực hiện lệnh sau để cấp phát quyền xem dữ liệu trên bảng R cho C:

GRANT SELECT ON R TO C

và B cấp phát quyền xem và bổ sung dữ liệu trên bảng R cho C bằng câu lệnh:

```
GRANT SELECT,  
INSERT ON R TO C
```

Như vậy, C có quyền xem và bổ sung dữ liệu trên bảng R. Bây giờ, nếu B thực hiện lệnh:

```
REVOKE SELECT,  
INSERT ON R FROM C
```

Người dùng C sẽ không còn quyền bổ sung dữ liệu trên bảng R nhưng vẫn có thể xem được dữ liệu của bảng này (quyền này do A cấp cho C và vẫn còn hiệu lực).

Nếu ta đã cấp phát quyền cho người dùng nào đó bằng câu lệnh GRANT với tùy chọn WITH GRANT OPTION thì khi thu hồi quyền bằng câu lệnh REVOKE phải chỉ định tùy chọn CASCADE. Trong trường hợp này, các quyền được chuyển tiếp cho những người dùng khác cũng đồng thời được thu hồi.

Ví dụ: Ta cấp phát cho người dùng A trên bảng R với câu lệnh GRANT như sau:

```
GRANT SELECT ON R TO A  
WITH GRANT OPTION
```

sau đó người dùng A lại cấp phát cho người dùng B quyền xem dữ liệu trên R với câu lệnh:

```
GRANT SELECT ON R TO B
```

Nếu muốn thu hồi quyền đã cấp phát cho người dùng A, ta sử dụng câu lệnh REVOKE như sau:

```
REVOKE SELECT ON  
NHANVIEN FROM A  
CASCADE
```

Câu lệnh trên sẽ đồng thời thu hồi quyền mà A đã cấp cho B và như vậy cả A và B đều không thể xem được dữ liệu trên bảng R.

b) Thu hồi quyền thực thi các câu lệnh

Việc thu hồi quyền thực thi các câu lệnh trên cơ sở dữ liệu (CREATE DATABASE, CREATE TABLE, CREATE VIEW,...) được thực hiện đơn giản với câu lệnh REVOKE có cú pháp:

```
REVOKE ALL | các_câu_lệnh_cần_thu_hồi  
FROM danh_sách_người_dùng
```

Ví dụ: Để không cho phép người dùng Ha thực hiện lệnh CREATE TABLE trên cơ sở dữ liệu, ta sử dụng câu lệnh:

```
REVOKE CREATE TABLE FROM Ha
```

Bài tập thực hành của học viên

Bài tập 1: Thực hiện các yêu cầu sau

1. Cho phép người dùng test2 được phép xem tất cả dữ liệu của cơ sở dữ liệu QuanLyBanHang. Lưu ý: Kiểm tra quyền của người dùng này bằng cách sử dụng máy tính khác kết nối từ xa vào máy chủ (kết nối thông qua mạng LAN)
2. Cấp quyền xem và thêm dữ liệu trên bảng VATTU cho người dùng thanhv.
3. cấp quyền cho nhóm xemdulieu quyền xem dữ liệu cơ sở dữ liệu QuanLyBanHang. Trong nhóm quyền này có một user là DangLH

Bài tập 2 Thực hiện các yêu cầu sau

Câu 1: Tạo tài khoản

1.1 Tạo tài khoản đăng nhập SQLServer với tên đăng nhập là *nhanvien* và password là *123456*

1.2 Tạo tài khoản đăng nhập SQLServer với tên đăng nhập là *giamdoc* và password là *giamdoc*

1.3 Tạo login có tên là *thukho* với password là *thukho*

Câu 2: Tạo người dùng

2.1 Tạo người dùng có tên *nhanvien* với tài khoản đăng nhập là *nhanvien*

2.2 Tạo người dùng có tên *giamdoc* với tài khoản đăng nhập là *giamdoc*

2.3 Tạo người dùng tên *thukho* với tên login là *thukho*

Câu 3: Cấp quyền

3.1 Cấp quyền cho những người dùng có tên là: *nhanvien*, *giamdoc* quyền thực thi các câu lệnh SELECT, INSERT và UPDATE trên bảng VATTU.

3.2 Cho phép những người dùng *nhanvien*, *giamdoc* quyền xem: năm tháng, mã vật tư, số lượng đầu, số lượng cuối trên bảng TONKHO (cột NAMTHANG, MAVTU, SLDAU, SLCUOI của bảng TONKHO).

3.3 Cấp quyền cho những người dùng *nhanvien*, *giamdoc* quyền SELECT, INSERT, UPDATE, DELETE trên bảng NHACC (Nhà cung cấp).

3.4 Cho phép người dùng *thukho* quyền xem dữ liệu trên bảng TONKHO, đồng thời có thể cấp lại quyền này cho những người dùng khác

3.5 Cấp quyền tạo cơ sở dữ liệu (Create Database) và tạo bảng (Create Table) cho người dùng có tên là *giamdoc*.

Hướng dẫn thực hiện

Bài tập 1: Thực hiện các yêu cầu sau

1. Cho phép người dùng *test2* được phép xem tất cả dữ liệu của cơ sở dữ liệu *QuanLyBanHang*. Lưu ý: Kiểm tra quyền của người dùng này bằng cách sử dụng máy tính khác kết nối từ xa vào máy chủ (kết nối thông qua mạng LAN)

Bước 1: Tạo tài khoản đăng nhập

Cách 1:

```
create login tên_login with password = 'nhập_mật_khẩu'
```

Cách 2:

```
sp_addlogin 'tên_login','mật_khẩu'
```

Bước 2: Tạo người dùng

Cách 1:

```
create user tên_user for login tên_login
```

Cách 2:

```
sp_adduser 'tên_login','tên_user'
```

Bước 3: Cấp quyền cho người dùng

```
grant danh_sách_quyền on tên_bảng to tên_user [with grant option]
```

2. Cấp quyền xem và thêm dữ liệu trên bảng VATTU cho người dùng *thanhv*.

Bước 1: Tạo tài khoản đăng nhập

create login thanhnv with password = 'Abc12345'

Bước 2: Tạo người dùng

create user thanhnv for login thanhnv

Bước 3: Cấp quyền

grant select,insert on VATTU to thanhnv

3. Cấp quyền cho nhóm xemdulieu quyền xem dữ liệu cơ sở dữ liệu QuanLyBanHang. Trong nhóm quyền này có một user là DangLH

Bước 1: Tạo nhóm quyền

Use QuanLyBanHang --Chọn cơ sở dữ liệu

Create role xemdulieu

Bước 2: Cấp quyền cho nhóm

Grant select to xemdulieu

Bước 3: Tạo login tên DangLH, mật khẩu Abc12345

Create Login DangLH With Password = 'Abc12345'

Bước 4: Tạo user DangLH sử dụng login DangLH

Use QuanLyBanHang --Chọn cơ sở dữ liệu

Create User DangLH For Login DangLH

Bước 5: Thêm usser DangLH vào nhóm xemdulieu

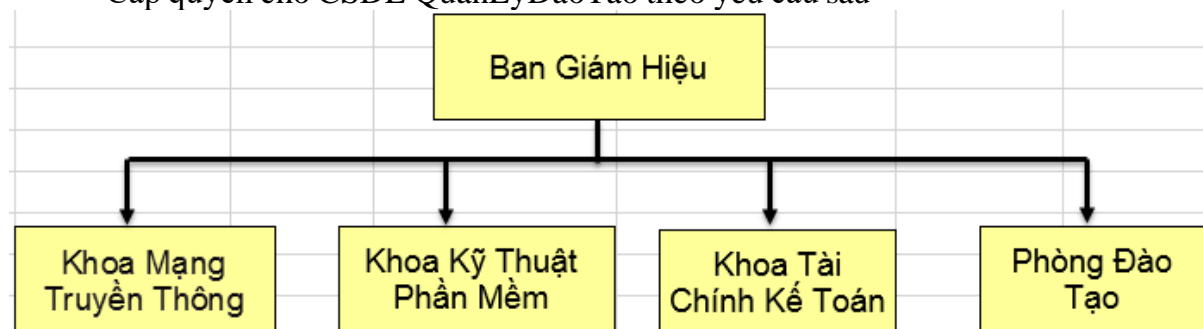
Sp_addRoleMember 'xemdulieu', 'DangLH'

Những trọng tâm cần chú ý:

- Trình bày các lệnh về tạo login, tạo user
- Trình bày cấp quyền cho user theo từng nhóm
- Thực hiện đúng việc tạo login, tạo user theo yêu cầu
- Thực hiện đúng việc tạo cấp quyền, thu hồi quyền yêu cầu
- Thực hiện đúng việc tạo quyền cho nhóm làm việc
- Sử dụng lệnh cấp quyền cho các cột, bảng và CSDL

Bài mở rộng và nâng cao

Cấp quyền cho CSDL QuanLyDaoTao theo yêu cầu sau



1 Tạo các nhóm quyền thuộc CSDL QuanLyDaoTao tương ứng với các phòng ban như mô tả trên

2 Cấp quyền cho các phòng ban trên CSDL QuanLyDaoTao

Ban Giám Hiệu được phép xem tất dữ liệu tất cả các bảng của database

Khoa Mạng Truyền Thông được phép xem trên bảng SINHVIEN, LOP

Khoa Kỹ thuật phần mềm được phép xem, thêm, xóa và cập nhật dữ liệu trên bảng SINHVIEN, LOP, MONHOC

Khoa Tài chính kế toán chỉ được phép xem dữ liệu trên bảng DiemTP

Phòng Đào tạo được phép xem, thêm, xóa và cập nhật dữ liệu trên tất cả các bảng của database

3 Tạo danh sách nhân sự cho các phòng ban (**Lưu ý tất cả tài khoản phải đổi mật khẩu trong lần đăng nhập đầu tiên**)

Ban Giám Hiệu

Tên	Mật khẩu
AnhNH	Abc12345
HoangNT	Abc12345

Khoa Mạng Truyền Thông

Tên	Mật khẩu
TrungDM	Abc12345
CongND	Abc12345
DangNS	Abc12345

Khoa Kỹ thuật phần mềm

Tên	Mật khẩu
ThuatDV	Abc12345
DatDT	Abc12345
NguyenTT	Abc12345

Khoa Tài chính kế toán

Tên	Mật khẩu
TuanTV	Abc12345
DieuNT	Abc12345
GiangNN	Abc12345

Phòng Đào tạo

Tên	Mật khẩu
TramNTH	Abc12345
ThuyLT	Abc12345

Yêu cầu đánh giá kết quả học tập

Nội dung

- Về kiến thức:
 - + Trình bày các thao tác trên Windows Authentication
 - + Trình bày các thao tác về SQL Server Authentication
 - + Trình bày các thao tác về quản lý người dùng
 - + Trình bày các thao tác về quyền người dùng
- Về kỹ năng:
 - + Thực hiện đúng các thao tác tạo các login và user kèm theo password
 - + Sử dụng đúng các lệnh thao tác về Stored procedure
 - + Thực hiện đúng các thao tác cấp quyền theo nhóm phòng ban.
- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

Phương pháp

- Về kiến thức: Đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp.
- Về kỹ năng: Đánh giá kỹ năng thực hành các thao tác về tạo, cấp quyền và thu hồi quyền.

- Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, linh hoạt và ngăn nắp trong công việc.

BÀI TẬP TỔNG HỢP

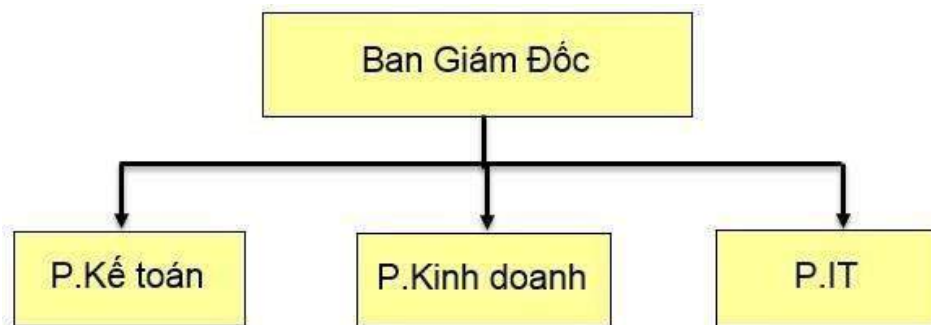
QUẢN TRỊ SQL SERVER 1

1. Chuẩn đầu ra

- Tạo được cơ sở dữ liệu, bảng
- Thiết lập được ràng buộc dữ liệu
- Cấp quyền được cho người dùng và nhóm người dùng trên cơ sở dữ liệu
- Sao lưu và phục hồi được cơ sở dữ liệu

2. Tình huống

Cho sơ đồ tổ chức của khách sạn ABC như sau:



3. Chuẩn bị

- Chi tiết các thành phần cần chuẩn bị: máy tính có cài phần mềm Microsoft SQL Server 2014 trở lên

4. Nội dung các bài tập

Bài tập 1:

Mô tả bài tập

- Tạo cơ sở dữ liệu tên QuanLyKhachSan với yêu cầu chi tiết như sau:

Loại file	Size	MaxSize	FileGrowth
Data	100MB	2000 MB	50MB
Log	30 MB	Không giới hạn	30 MB

Yêu cầu thực hiện:

1.1- Viết câu lệnh tạo cơ sở dữ liệu theo như mô tả ở trên

1.2- Thư mục lưu cơ sở dữ liệu là tùy ý

Hướng dẫn thực hiện:

- Tạo thư mục lưu cơ sở dữ liệu
- Sử dụng câu lệnh create database để tạo cơ sở dữ liệu

Bài tập 2:

Mô tả bài tập:

- Tạo các bảng và diagram cho cơ sở dữ liệu QuanLyKhachSan

Bảng

KhachHang

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaKh	varchar	20	Khóa chính
TenKh	nvarchar	100	
CMND	varchar	15	
DiaChi	nvarchar	100	

Bảng

LoaiPhong

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaLoaiPh	varchar	20	Khóa chính
TenLoaiPh	nvarchar	100	
DonGia	int		

Bảng Phong

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaPh	Varchar	20	Khóa chính
TenPh	Nvarchar	100	
TinhTrang	Nvarchar	50	
GhiChu	Nvarchar	100	
MaLoaiPh	Varchar	20	Khóa ngoại

Bảng

PhieuThuePhong

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaPhieu	varchar	20	Khóa chính
NgayBDau	datetime		Ngày bắt đầu
MaPh	varchar	20	Khóa ngoại

Bảng HoaDon

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaHd	varchar	20	Khóa chính
TriGia	int		
MaKh	varchar	20	Khóa ngoại

Yêu cầu thực hiện:

2.1 Viết câu lệnh tạo các bảng trên

Hướng dẫn thực hiện:

- Sử dụng lệnh use để chỉ định cơ sở dữ liệu cần sử dụng
- Sử dụng câu lệnh create table để tạo bảng

Bài tập 3:

Mô tả bài tập

- Thiết lập ràng buộc dữ liệu cho các trường của các table

Yêu cầu thực hiện:

- 3.1- Chứng minh nhân dân của khách hàng không được nhập trùng
- 3.2- Đơn giá phòng và trị giá không được nhỏ hơn 0
- 3.3- Ngày bắt đầu không được lớn hơn ngày hiện hành
- 3.4- Giá trị mặc định cho cột đơn giá và trị giá là 0

Hướng dẫn thực hiện:

- Sử dụng câu lệnh alter table

Bài tập 4:

Mô tả bài tập

- Phân quyền sử dụng cơ sở dữ liệu cho các phòng ban

Yêu cầu thực hiện:

- 4.1- Ban giám đốc được phép xem toàn bộ cơ sở dữ liệu
- 4.2- Phòng kế toán được phép thêm, cập nhật trên bảng phiếu thuê phòng và hóa đơn

Lưu ý: Đối với hóa đơn chỉ được phép cập nhật trị giá

- 4.3- Phòng kinh doanh được phép thêm, và cập nhật trên bảng khách hàng

- 4.4- Phòng IT được phép xem, thêm, xóa và cập nhật trên cơ sở dữ liệu

Lưu ý: chỉ có trưởng phòng mới được phép cấp lại quyền

Hướng dẫn thực hiện:

- Sử dụng câu lệnh create role để tạo các nhóm quyền tương ứng với các phòng ban
- Sử dụng câu lệnh sp_adduser để tạo danh sách nhân sự thuộc các phòng ban

Ban giám đốc

Username	Password
CuongLD	123

Phòng kế toán

Username	Password
MyLT	123
TrinhLT	123

Phòng kinh doanh

Username	Password
QuangNM	123
NguyetNTT	123

Phòng IT

Username	Password	Ghi chú
TungLD	123	Trưởng phòng
TuongLD	123	nhân viên

Lưu ý: Tất cả các login với thiết lập default database là QuanLyKhachSan

Bài tập 5:

Mô tả bài tập

- Thay đổi kích thước cơ sở dữ liệu

Yêu cầu thực hiện:

5.1- Thay đổi kích thước tối đa của data file thành không giới hạn

5.2- Thay đổi kích thước khởi tạo của log file thành 300 MB

Hướng dẫn thực hiện:

- Sử dụng câu lệnh alter database và dbcc shrinkfile

Bài tập 6:

Mô tả bài tập

- Sao lưu và phục hồi cơ sở dữ liệu

Yêu cầu thực hiện:

6.1- Tạo kịch bản sao lưu

Thêm dữ liệu cho cho bảng PhieuThuePhong như sau:

MaPhieu	NgayBDau	MaPh
P001	01/01/2012	A101

=> Thực hiện full backup

Thêm tiếp dữ liệu cho bảng PhieuThuePhong như sau:

MaPhieu	NgayBDau	MaPh
P002	01/01/2012	A102

=> Thực hiện differential backup

Thêm tiếp dữ liệu cho bảng PhieuThuePhong như sau:

MaPhieu	NgayBDau	MaPh
P003	01/01/2012	A201

=> Thực hiện log backup

Thêm tiếp dữ liệu cho bảng PhieuThuePhong như sau:

MaPhieu	NgayBDau	MaPh
P004	01/01/2012	A202

=> Thực hiện log backup, sử dụng cùng tên file với lần backup log trước

6.2- Tạo kịch bản phục hồi

Giả lập máy chủ bị sự cố bằng cách xoá cơ sở dữ liệu QuanLyKhachSan, sau đó phục hồi

Hướng dẫn thực hiện:

- Sử dụng các câu lệnh backup database, backup log để tạo kịch bản sao lưu
- Sử dụng câu lệnh drop database để xoá cơ sở dữ liệu và sử dụng các câu lệnh restore database, restore log để khôi phục cơ sở dữ liệu

QUẢN TRỊ SQL SERVER

1. Chuẩn đầu ra

Tạo được cơ sở dữ liệu, bảng

Thiết lập được ràng buộc dữ liệu

Cấp quyền được cho người dùng và nhóm người dùng trên cơ sở dữ liệu

Sao lưu và phục hồi được cơ sở dữ liệu

2. Tình huống

Trường Cao đẳng nghề Cần Thơ cần triển khai một cơ sở dữ liệu nhằm phục vụ công tác đào tạo. Trong cơ sở dữ liệu này gồm có các bảng. Để tiện cho việc quản lý, trường yêu cầu phải tạo những nhóm quyền và cấp những quyền tương ứng.

3. Chuẩn bị

Chi tiết các thành phần cần chuẩn bị: máy tính có cài phần mềm Microsoft SQL Server 2014 trở lên

4. Nội dung các bài tập

Bài tập 1:

Mô tả bài tập

- Tạo cơ sở dữ liệu tên QuanLyDaoTao với yêu cầu chi tiết như sau:

Loại file	Size	MaxSize	FileGrowth
Data	10MB	Không giới hạn	10MB
Log	10MB	Không giới hạn	30MB

Yêu cầu thực hiện:

1.1- Viết câu lệnh tạo cơ sở dữ liệu theo như mô tả ở trên

1.2- Thư mục lưu cơ sở dữ liệu là tùy ý

Hướng dẫn thực hiện:

- Tạo thư mục lưu cơ sở dữ liệu
- Sử dụng câu lệnh create database để tạo cơ sở dữ liệu

Bài tập 2:

Mô tả bài tập:

- Tạo các bảng thuộc cơ sở dữ liệu QuanLyDaoTao

Bảng Khoa

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaKhoa	varchar	20	Khóa chính
TenKhoa	nvarchar	100	

Bảng GiảngVien

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaGv	varchar	20	Khóa chính
TenGv	nvarchar	100	
ChuyenNganh	varchar	20	
MaKhoa	varchar	20	Khóa ngoại

Bảng Lop

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaLop	varchar	20	Khóa chính
TenLop	nvarchar	100	
MaKhoa	varchar	20	Khóa ngoại

Bảng MonHoc

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaMh	varchar	20	Khóa chính
TenMh	nvarchar	100	
SoGio	int		

Bảng SinhVien

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaHd	varchar	20	Khóa chính
TriGia	int		
MaKh	varchar	20	Khóa ngoại

Bảng SinhVien

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaSv	varchar	20	Khóa chính
TenSv	nvarchar	100	
NgaySinh	datetime		
MaLop	varchar	20	Khóa ngoại

Bảng DiemTk

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaSv	varchar	20	Khóa chính
MaMh	varchar	20	
Diem	int		

Yêu cầu thực hiện:

2.1 Viết câu lệnh tạo các bảng trên bao gồm việc tạo khoá chính và khoá ngoại

Hướng dẫn thực hiện:

- Sử dụng lệnh use để chỉ định cơ sở dữ liệu cần sử dụng
- Sử dụng câu lệnh create table để tạo bảng

Bài tập 3:

Mô tả bài tập

- Thiết lập ràng buộc dữ liệu cho các trường của các bảng

Yêu cầu thực hiện:

3.1- Điểm chỉ được nhập từ 0 đến 10

3.2- Tên khoa, tên lớp, tên môn học không được nhập trùng

3.3- Giá trị cho cột ngày sinh không được lớn hơn ngày hiện hành

3.4- Số giờ không được nhỏ hơn 0

Hướng dẫn thực hiện:

- Sử dụng câu lệnh alter table

Bài tập 4:

Mô tả bài tập

- Phân quyền sử dụng cơ sở dữ liệu

Yêu cầu thực hiện:

4.1- Tạo nhóm quyền

Tên nhóm
xemdulieu
capnhatudulieu
xoadulieu
themdulieu

4.2- Tạo tài khoản đăng nhập với chứng thực sql

Tên đăng nhập	Mật khẩu
LanNT	Abc12345
LanhNT	Abc12345
LyNT	Abc12345
LeNT	Abc12345
IT	IT

4.3- Tạo danh sách người dùng

Tên user	Tên đăng nhập	Thuộc nhóm quyền
LanNT	LanNT	xemdulieu
LanhNT	LanhNT	themdulieu
LyNT	LyNT	xoadulieu
LeNT	LeNT	capnhatudulieu
IT	IT	Không thuộc nhóm

4.4- Phân quyền

Tên user/Tên role	Tên bảng	Quyền được cấp
xemdulieu	Tất cả các bảng	Xem dữ liệu
themdulieu	Khoa,GiangVien,Lop	Thêm dữ liệu
xoadulieu	DiemTk	Xóa dữ liệu
capnhatudulieu	DiemTk	Chỉ được cập nhật dữ liệu cho cột điểm
IT	Tất cả các bảng	Xem, thêm,xóa, cập nhật và cấp lại quyền

Hướng dẫn thực hiện:

- Sử dụng câu lệnh create role để tạo các nhóm quyền tương ứng với các phòng ban
- Sử dụng câu lệnh create login để tạo tài khoản đăng nhập
- Sử dụng câu lệnh sp_adduser để tạo danh sách nhân sự thuộc các phòng ban
- Sử dụng câu lệnh grant để cấp quyền

Bài tập 5:

Mô tả bài tập

- Sao lưu và phục hồi cơ sở dữ liệu

Yêu cầu thực hiện:

5.1- Tạo kịch bản sao lưu

Thêm dữ liệu cho cho bảng Khoa như sau:

Thêm dữ liệu cho cho bảng Khoa như sau:

MaKhoa	TenKhoa
CNTT	Công nghệ thông tin

=> Thực hiện full backup

Thêm tiếp dữ liệu cho bảng Khoa như sau:

MaKhoa	TenKhoa
QTKD	Quản trị kinh doanh

=> Thực hiện differential backup

Thêm tiếp dữ liệu cho bảng Khoa như sau:

MaKhoa	TenKhoa
TCKT	Tài chính kế toán

=> Thực hiện log backup

Thêm tiếp dữ liệu cho bảng Khoa như sau:

MaKhoa	TenKhoa
KHCB	Khoa học cơ bản

=> Thực hiện log backup, sử dụng cùng tên file với lần backup log trước

Thêm tiếp dữ liệu cho bảng Khoa như sau:

MaKhoa	TenKhoa
NN	Khoa ngoại ngữ

=> Thực hiện log backup, sử dụng cùng tên file với lần backup log trước

5.2- Tạo kịch bản phục hồi

Giả lập máy chủ bị sự cố bằng cách xoá cơ sở dữ liệu QuanLyKhachSan, sau đó phục hồi

Hướng dẫn thực hiện:

- Sử dụng các câu lệnh backup database, backup log để tạo kịch bản sao lưu
- Sử dụng câu lệnh drop database để xoá cơ sở dữ liệu và sử dụng các câu lệnh restore database, restore log để khôi phục cơ sở dữ liệu

CÁC THUẬT NGỮ CHUYÊN MÔN

Thuật ngữ	Giải thích
Analysys	Phân tích
Backup	Sao lưu
Check	Kiểm tra
Columns	Cột
Constrain	Ràng buộc
Database	Cơ sở dữ liệu
Delete	Xóa
Foreign key	Khóa ngoài
Insert	Chèn
Instalation	Cài đặt
Join	Nối
Key	Khóa
Master	Cơ sở dữ liệu chủ
Online	Trực tuyến
Primary key	Khóa chính
Query	Truy vấn
Quy tắc	Role
Relational	Quan hệ
Restore	Phục hồi
Rollback	Quay lui
Row	Hàng
Select	Lựa chọn
Subquery	Truy vấn con
Table	Bảng biểu
Transaction	Giao dịch
Union	Hợp lại
Update	Cập nhật
User	Người dùng
Version	Phiên bản
View	Khung nhìn

Điều kiện để hoàn thành mô đun để được dự thi kết thúc mô đun:

+ Người học tham dự ít nhất 70% thời gian học lý thuyết và đầy đủ các bài học tích hợp, bài học thực hành, thực tập

+ Điểm trung bình chung các điểm kiểm tra đạt từ 5,0 điểm trở lên theo thang điểm 10;

+ Người học có giấy xác nhận khuyết tật theo quy định thì được hiệu trưởng xem xét, quyết định ưu tiên điều kiện dự thi trên cơ sở sinh viên đó phải bảo đảm điều kiện về điểm trung bình các điểm kiểm tra.

+ Số lần dự thi kết thúc mô đun theo quy định tại khoản 2 Điều 13 Thông tư 09/2017/TT-BLĐTBXH, ngày 13 tháng 3 năm 2017.

Điều kiện để được công nhận, cấp chứng nhận đạt mô đun đào tạo:

Người học được công nhận và cấp chứng nhận đạt mô đun này khi có điểm trung bình mô đun theo thang điểm 10 đạt từ 4,0 trở lên

Điều kiện để hoàn thành mô đun để được dự thi kết thúc mô đun:

+ Người học tham dự ít nhất 70% thời gian học lý thuyết và đầy đủ các bài học tích hợp, bài học thực hành, thực tập

+ Điểm trung bình chung các điểm kiểm tra đạt từ 5,0 điểm trở lên theo thang điểm 10;

+ Người học có giấy xác nhận khuyết tật theo quy định thì được hiệu trưởng xem xét, quyết định ưu tiên điều kiện dự thi trên cơ sở sinh viên đó phải bảo đảm điều kiện về điểm trung bình các điểm kiểm tra.

+ Số lần dự thi kết thúc mô đun theo quy định tại khoản 2 Điều 13 Thông tư 09/2017/TT-BLĐT BXH, ngày 13 tháng 3 năm 2017.

Điều kiện để được công nhận, cấp chứng nhận đạt mô đun đào tạo:

Người học được công nhận và cấp chứng nhận đạt mô đun này khi có điểm trung bình mô đun theo thang điểm 10 đạt từ 4,0 trở lên

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Phạm Hữu Khang, Lập trình ứng dụng chuyên nghiệp SQL Server, Nhà xuất bản Lao động xã hội, 2003.
- [2] Microsoft SQL Server 2014 Book Online.
- [3] Đỗ Phúc, Tài liệu môn học CSDL nâng cao, Đại học Công nghệ thông tin, Đại học Quốc gia Tp HCM, 2006.
- [4] Trần Đức Quang, Nguyên lý các hệ cơ sở dữ liệu phân, Nhà xuất bản Thống kê, tập 1 và 2, 1999.
- [5] Phạm Thế Quế, Giáo trình cơ sở dữ liệu phân tán, Học viên công nghệ bưu chính viễn thông, 2010.
- [6] Nguyễn Bá Tường, Nhập môn cơ sở dữ liệu phân tán, Nhà xuất bản Khoa học Kỹ thuật, 2014.
- [7] Trần Quốc Chiến, Giáo trình cơ sở dữ liệu nâng cao, Đại học Đà Nẵng, 2007.

Tiếng Anh

- [8] M.Tamer Ozsü and Patrick Vadurıes, Principles of Distributed Database System, Frenrice-Hall, 2006.

Trang Web

- [9] <http://www.lirc.udn.vn/>
- [10] <http://msdn.microsoft.com>