



**Giáo trình hệ quản trị cơ  
sở dữ liệu MS SQL Server  
2000**

*Thành phố Hồ Chí Minh, tháng năm .....*

## MỤC LỤC

CHƯƠNG I.....	GIỚI THIỆU SQL SERVER 2000	
.....		8
<b>I.1</b>	<b>GIỚI THIỆU:</b>	<b>8</b>
<b>I.2</b>	<b>LƯU TRỮ DỮ LIỆU TRONG MS SQL SERVER 2000</b>	<b>8</b>
<b>I.3</b>	<b>CÁC CÔNG CỤ TƯƠNG TÁC VỚI SQL SERVER</b>	<b>9</b>
<b>I.3.1</b>	<b>Enterprise Manager (EM):</b>	<b>9</b>
<b>I.3.2</b>	<b>Query Analyzer (QA):</b>	<b>10</b>
CHƯƠNG II.....	THÀNH PHẦN CỦA SQL SERVER 2000	
.....		12
<b>II.1</b>	<b>CÁC THÀNH PHẦN CỦA SQL SERVER 2000</b>	<b>12</b>
<b>II.2</b>	<b>CƠ SỞ DỮ LIỆU (CSDL)</b>	<b>13</b>
<b>II.2.1</b>	<b>Master:</b>	<b>13</b>
<b>II.2.2</b>	<b>Model:</b>	<b>13</b>
<b>II.2.3</b>	<b>MSDB:</b>	<b>13</b>
<b>II.2.4</b>	<b>TempDB:</b>	<b>13</b>
<b>II.2.5</b>	<b>Pubs:</b>	<b>13</b>
<b>II.2.6</b>	<b>Northwind:</b>	<b>14</b>
<b>II.3</b>	<b>CÁC ĐỐI TƯỢNG BÊN TRONG MỘT CSDL</b>	<b>14</b>
<b>II.3.1</b>	<b>Bảng (Table):</b>	<b>14</b>
<b>II.3.2</b>	<b>Khung nhìn (View):</b>	<b>14</b>
<b>II.3.3</b>	<b>Lược đồ quan hệ (Diagram):</b>	<b>14</b>
<b>II.3.4</b>	<b>Chỉ mục (Index):</b>	<b>14</b>
<b>II.3.5</b>	<b>Trigger:</b>	<b>15</b>
<b>II.3.6</b>	<b>Ràng buộc (Constraint):</b>	<b>15</b>
<b>II.3.7</b>	<b>Thủ tục thường trú (Stored Procedure):</b>	<b>15</b>
<b>II.4</b>	<b>Kiểu dữ liệu (DATATYPE)</b>	<b>15</b>
CHƯƠNG III.....	THAO TÁC CĂN BẢN TRÊN SQL SERVER	
.....		17

<b>III.1 TẠO CSDL</b>	<b>17</b>
<b>III.1.1 Tạo bằng dòng lệnh với QA</b>	<b>17</b>
<b>III.1.2 Tạo CSDL bằng giao diện Enterprise Manager:</b>	<b>18</b>
<b>III.2 TẠO BẢNG (TABLE)</b>	<b>19</b>
<b>III.2.1 Tạo bằng dòng lệnh với QA</b>	<b>19</b>
<b>III.2.2 Tạo bằng giao diện EM:</b>	<b>20</b>
<b>III.3 THAY ĐỔI CẤU TRÚC BẢNG:</b>	<b>21</b>
<b>III.3.1 Thay đổi bằng dòng lệnh với QA</b>	<b>21</b>
<b>III.3.2 Thay đổi bằng giao diện EM:</b>	<b>21</b>
<b>III.4 XOÁ :</b>	<b>21</b>
<b>III.4.1 Xoá CSDL:</b>	<b>21</b>
<b>III.4.2 Xoá bảng:</b>	<b>21</b>
<b>III.5 TẠO SCRIPT:</b>	<b>21</b>
<b>CHƯƠNG IV.....</b>	<b>CHUẨN HOÁ VÀ QUAN HỆ</b>
.....	23
<b>IV.1 LƯỢC ĐỒ QUAN HỆ (DIAGRAM)</b>	<b>23</b>
<b>CHƯƠNG V.....</b>	<b>KHUNG NHÌN (VIEWS)</b>
.....	26
<b>V.1 KHÁI NIỆM CƠ BẢN</b>	<b>26</b>
<b>V.2 TẠO VIEW:</b>	<b>26</b>
<b>V.2.1 Tạo bằng dòng lệnh với QA</b>	<b>26</b>
<b>V.2.2 Tạo bằng giao diện EM:</b>	<b>26</b>
<b>CHƯƠNG VI.....</b>	<b>QUẢN TRỊ CSDL. CÁC THAO TÁC BACKUP, RESTORE, IMPORT, EXPORT.....</b>
.....	29
<b>VI.1 CĂN BẢN VỀ QUẢN TRỊ CSDL</b>	<b>29</b>
<b>VI.2 SAO LƯU (BACKUP) VÀ PHỤC HỒI (RESTORE)</b>	<b>29</b>
<b>VI.2.1 Backup</b>	<b>29</b>
<b>VI.2.2 Restore</b>	<b>31</b>
<b>VI.3 IMPORT – EXPORT</b>	<b>32</b>
<b>VI.3.1 IMPORT</b>	<b>32</b>
<b>VI.3.2 EXPORT</b>	<b>33</b>
<b>CHƯƠNG VII.....</b>	<b>SCRIPT VÀ BATCH</b>
.....	35

<b>VII.1</b>	<b>KHÁI NIỆM VỀ SCRIPT</b>	<b>35</b>
<b>VII.2</b>	<b>BATCH</b>	<b>35</b>
<b>VII.3</b>	<b>SCRIPT ĐƠN GIẢN</b>	<b>35</b>
<b>VII.3.1</b>	<b>USE:</b>	<b>36</b>
<b>VII.3.2</b>	<b>Khai báo biến trong SQL Server:</b>	<b>36</b>
<b>VII.3.3</b>	<b>OSQL:</b>	<b>36</b>
<b>VII.3.4</b>	<b>SQL động:</b>	<b>36</b>

**CHƯƠNG VIII. ....CÁC HÀM VÀ BIỂU THỨC HỖ TRỢ TRUY VẤN.**

.....37

<b>VIII.1</b>	<b>CÁC HÀM VÀ BIỂU THỨC HỖ TRỢ TRUY VẤN</b>	<b>37</b>
<b>VIII.1.1</b>	<b>AVG:</b>	<b>37</b>
<b>VIII.1.2</b>	<b>Min :</b>	<b>37</b>
<b>VIII.1.3</b>	<b>Max :</b>	<b>37</b>
<b>VIII.1.4</b>	<b>Count</b>	<b>37</b>
<b>VIII.1.5</b>	<b>Sum:</b>	<b>37</b>
<b>VIII.2</b>	<b>CÁC HÀM XỬ LÝ CHUỖI</b>	<b>37</b>
<b>VIII.2.1</b>	<b>ASCII:</b>	<b>37</b>
<b>VIII.2.2</b>	<b>Char:</b>	<b>38</b>
<b>VIII.2.3</b>	<b>Upper:</b>	<b>38</b>
<b>VIII.2.4</b>	<b>Lower:</b>	<b>38</b>
<b>VIII.2.5</b>	<b>Len:</b>	<b>38</b>
<b>VIII.2.6</b>	<b>Ltrim:</b>	<b>38</b>
<b>VIII.2.7</b>	<b>Rtrim:</b>	<b>38</b>
<b>VIII.2.8</b>	<b>Left:</b>	<b>38</b>
<b>VIII.2.9</b>	<b>Right:</b>	<b>38</b>
<b>VIII.2.10</b>	<b>CharIndex:</b>	<b>38</b>
<b>VIII.3</b>	<b>CÁC HÀM THỜI GIAN</b>	<b>38</b>
<b>VIII.3.1</b>	<b>GetDate():</b>	<b>38</b>
<b>VIII.3.2</b>	<b>DatePart():</b>	<b>39</b>
<b>VIII.3.3</b>	<b>DateDiff():</b>	<b>39</b>
<b>VIII.3.4</b>	<b>Day():</b>	<b>39</b>
<b>VIII.3.5</b>	<b>Month():</b>	<b>39</b>
<b>VIII.3.6</b>	<b>Year():</b>	<b>39</b>

<b>VIII.4</b>	<b>CÁC HÀM TOÁN HỌC</b>	<b>39</b>
<b>VIII.4.1</b>	<b>Square:</b>	<b>39</b>
<b>VIII.4.2</b>	<b>Sqrt:</b>	<b>39</b>
<b>VIII.4.3</b>	<b>Round:</b>	<b>39</b>
<b>VIII.5</b>	<b>CÁC HÀM CHUYỂN ĐỔI</b>	<b>39</b>
<b>VIII.5.1</b>	<b>Cast:</b>	<b>39</b>
<b>VIII.5.2</b>	<b>Convert:</b>	<b>39</b>
CHƯƠNG IX..... THỦ TỤC THƯỜNG TRÚ STORED PROCEDURE		
.....		40
<b>IX.1</b>	<b>GIỚI THIỆU CHUNG</b>	<b>40</b>
<b>IX.2</b>	<b>TẠO MỘT SP</b>	<b>40</b>
<b>IX.3</b>	<b>XOÁ MỘT SP</b>	<b>41</b>
<b>IX.4</b>	<b>THAM SỐ TRONG SP</b>	<b>42</b>
<b>IX.4.1</b>	<b>Khai báo tham số:</b>	<b>42</b>
<b>IX.4.2</b>	<b>Cung cấp giá trị mặc định:</b>	<b>42</b>
<b>IX.4.3</b>	<b>Tham số output:</b>	<b>42</b>
<b>IX.5</b>	<b>CÁC LỆNH ĐIỀU KHIỂN</b>	<b>43</b>
<b>IX.5.1</b>	<b>Lệnh điều khiển IF ... ELSE</b>	<b>43</b>
<b>IX.5.2</b>	<b>Lệnh điều khiển CASE</b>	<b>43</b>
<b>IX.5.3</b>	<b>Lệnh điều khiển WHILE</b>	<b>44</b>
<b>IX.5.4</b>	<b>Lệnh điều khiển WAITFOR:</b>	<b>44</b>
CHƯƠNG X..... HÀM NGƯỜI DÙNG VÀ HÀM HỆ THỐNG		
.....		45
<b>X.1</b>	<b>KHÁI NIỆM HÀM NGƯỜI DÙNG</b>	<b>45</b>
<b>X.2</b>	<b>TẠO VÀ XOÁ HÀM NGƯỜI DÙNG</b>	<b>45</b>
<b>X.2.1</b>	<b>Tạo hàm người dùng</b>	<b>45</b>
<b>X.2.2</b>	<b>Xoá hàm người dùng</b>	<b>45</b>
<b>X.3</b>	<b>HÀM HỆ THỐNG</b>	<b>45</b>
<b>X.3.1</b>	<b>Tạo hàm hệ thống từ hàm người dùng:</b>	<b>45</b>
CHƯƠNG XI..... TRIGGER		
.....		46
<b>XI.1</b>	<b>KHÁI NIỆM:</b>	<b>46</b>
<b>XI.2</b>	<b>PHÂN LOẠI</b>	<b>46</b>

<b>XI.3 TẠO TRIGGER</b>	<b>46</b>
<b>XI.3.1 Cú pháp</b>	<b>46</b>
<b>XI.3.2 Tạo Trigger bằng EM.</b>	<b>47</b>
<b>CHƯƠNG XII..... CHUYỂN TÁC (TRANSACTION) VÀ KHOÁ (LOCK)</b>	<b>49</b>
<b>XII.1 KHÁI NIỆM CHUYỂN TÁC</b>	<b>49</b>
<b>XII.2 KHOÁ (LOCK) VÀ ĐỒNG HÀNH (CONCURRENCY)</b>	<b>49</b>
<b>CHƯƠNG XIII. ....SỬ DỤNG NGÔN NGỮ LẬP TRÌNH TƯƠNG TÁC VỚI SQL SERVER.....</b>	<b>51</b>
<b>XIII.1 LẬP TRÌNH BẰNG ASP</b>	<b>51</b>
<b>XIII.1.1 Kết nối đến CSDL SQL Server:</b>	<b>51</b>
<b>XIII.1.2 Thực thi câu lệnh SQL:</b>	<b>51</b>
<b>XIII.1.3 Thao tác với các bản ghi của một bảng:</b>	<b>51</b>
<b>XIII.1.4 Cập nhật vào CSDL bằng Recordset:</b>	<b>52</b>
<b>XIII.1.5 Thực thi một SP của SQL Server</b>	<b>53</b>
<b>XIII.2 LẬP TRÌNH BẰNG VISUAL BASIC</b>	<b>53</b>
<b>XIII.2.1 Kết nối đến CSDL:</b>	<b>53</b>
<b>XIII.2.2 Thực hiện câu lệnh SQL:</b>	<b>54</b>
<b>XIII.2.3 Thực thi SP</b>	<b>54</b>
<b>XIII.2.4 Thao tác với bản ghi</b>	<b>54</b>
<b>CHƯƠNG XIV. ....MỘT SỐ VẤN ĐỀ NÂNG CAO.</b>	<b>56</b>
<b>XIV.1 SECURITY</b>	<b>56</b>
<b>XIV.2 CHUYỂN TÁC VÀ TRUY VẤN PHÂN TÁN</b>	<b>56</b>
<b>XIV.3 XML</b>	<b>56</b>
<b>XIV.4 TÌM KIẾM FULL TEXT SEARCH</b>	<b>56</b>
<b>XIV.5 DỊCH VỤ PHÂN TÍCH ANALYSIS SERVICES</b>	<b>56</b>
<b>XIV.6 DATABASE REPLICATE</b>	<b>56</b>
<b>XIV.7 ENGLISH QUERY</b>	<b>56</b>

## **LỜI GIỚI THIỆU**

**Giáo trình hệ quản trị cơ sở dữ liệu MS SQL Server 2000 cung cấp cho sinh viên các khái niệm lý thuyết căn bản và tương đối đầy đủ về SQL Server 2000.**

**Giáo trình có minh hoạ bằng ví dụ các kỹ thuật thao tác với SQL Server 2000 giúp cho sinh viên có thể làm việc thực tế trên SQL Server 2000.**

**Một số các khái niệm căn bản và quen thuộc về CSDL quan hệ mà sinh viên đã được trang bị ở các môn học khác như Access, ngôn ngữ truy vấn SQL.. được dùng trong giáo trình mà không giải thích lại.**

**Do chương trình giới hạn cho 60 tiết giảng dạy nên một vài nội dung nâng cao của SQL Server chỉ giới thiệu sơ lược.**

**Giáo trình được soạn để phục vụ kịp thời các lớp đào tạo CNTT của Viện nên không thể tránh được nhiều thiếu sót. Rất mong nhận được sự đóng góp của các thầy cô và các học viên các khoá để chúng tôi có thể bổ khuyết kịp thời.**

# CHƯƠNG I. GIỚI THIỆU SQL SERVER 2000

## 1.1 GIỚI THIỆU:

Đối với các ứng dụng lớn, một thành phần không thể thiếu là hệ thống lưu trữ dữ liệu, dù cho chúng ta đang xây dựng một website thương mại điện tử hay một ứng dụng Windows, tất cả đều cần đến một kiến trúc lưu trữ dữ liệu tin cậy, hiệu quả, đã được chứng minh. SQL Server 2000 là một công cụ đáp ứng được những nhu cầu đó. Thích hợp với quy mô lớn, độ tin cậy cao là những đặc trưng của SQL.

SQL Server 2000 là sản phẩm mới nhất trong dòng sản phẩm SQL Server, là một đại diện của thế hệ Server trong công nghệ DOT NET của Microsoft. Nâng cấp từ phiên bản 7.0, nó được tích hợp thêm nhiều tính năng nâng cao: hỗ trợ XML, OLAP, các khả năng khai phá dữ liệu, tích hợp quản trị cùng với Win2k Active Directory, các ưu điểm về hiệu năng, khả năng sử dụng, khả năng lập trình... Đây là một hệ quản trị cơ sở dữ liệu tương đối phức tạp dùng cho môi trường máy chủ, với các ứng dụng xí nghiệp lớn.

Để khai thác dữ liệu của một hệ Cơ sở dữ liệu (CSDL), người ta xây dựng các ứng dụng bằng những ngôn ngữ lập trình tương tác với dữ liệu lưu trữ trong CSDL. Bản thân SQL Server thuần túy là CSDL, không như Oracle có tích hợp Java, các công cụ thiết kế Form và Report, hay như Access có tích hợp khả năng thiết kế Form và Report... Vì vậy để cho phép người dùng thực hiện các chuyển tác với CSDL SQL, cần xây dựng ứng dụng phát triển bằng một số ngôn ngữ như Visual C++, Visual Basic, ASP...

## 1.2 LƯU TRỮ DỮ LIỆU TRONG MS SQL SERVER 2000

SQL Server có cơ chế quản lý file đối với CSDL tương đối phức tạp. Mỗi CSDL được lưu trữ vật lý bao gồm ít nhất 1 file dữ liệu chính với đuôi .mdf (Primary Data File), có thể có thêm 1 hoặc nhiều file dữ liệu phụ với đuôi .ndf (Secondary Data File), và file log đuôi .ldf (Transaction Log File) ghi lại nhật ký các chuyển tác.

Ngoài ra còn có khái niệm FileGroup, tức là nhóm các file của CSDL thành các group. Mỗi CSDL có ít nhất một FileGroup chính (Primary FileGroup), CSDL đó có thể có thêm các FileGroup khác nữa.



Việc phân chia nhiều file dữ liệu và FileGroup nhằm mục đích cải thiện hiệu năng xử lý dữ liệu đối với các CSDL, đặc biệt là CSDL lớn của SQL Server

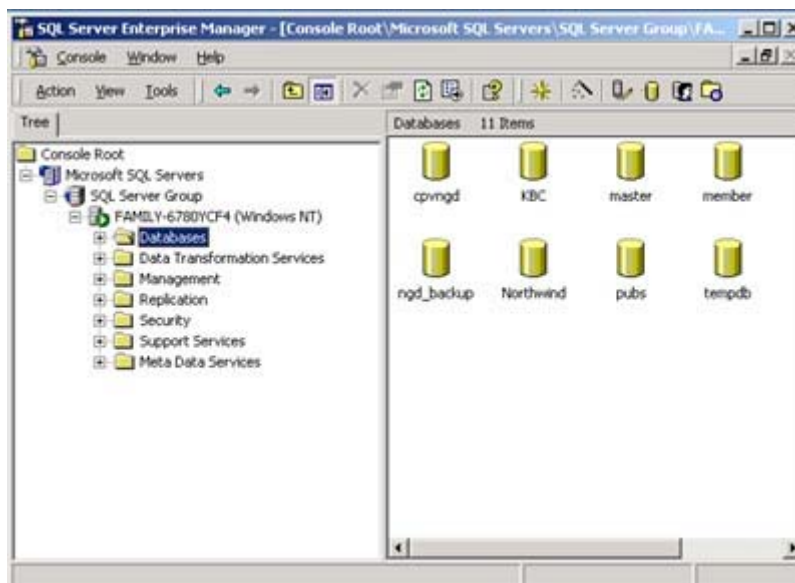
### 1.3 CÁC CÔNG CỤ TƯƠNG TÁC VỚI SQL SERVER

Hai công cụ chính để giúp nhà quản trị hay nhà phát triển giao tiếp với SQL Server (cấu hình, tạo đối tượng, chạy Script...) là Enterprise Manager và Query Analyzer

#### 1.3.1 Enterprise Manager (EM):

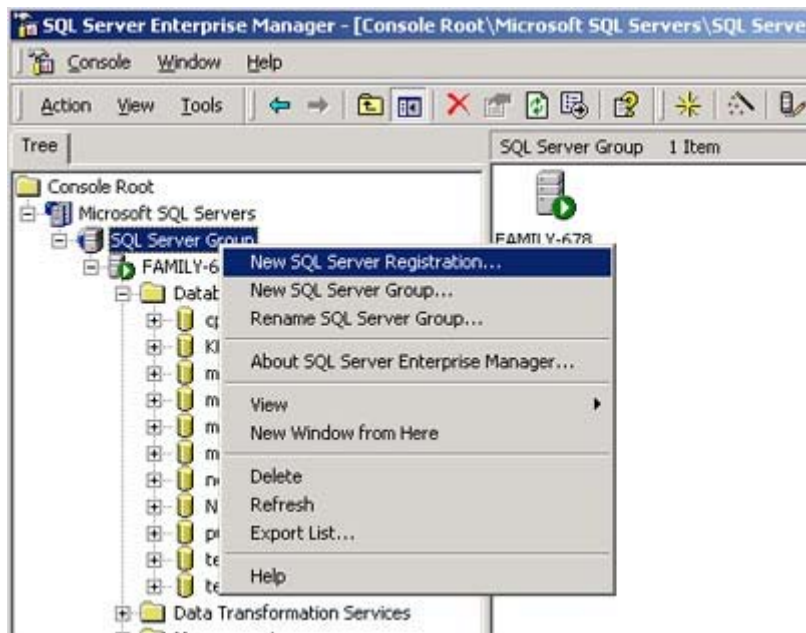
Là công cụ chính để tương tác với SQL Server trong vai trò một nhà quản trị Cơ sở dữ liệu (DBA – Database Administrator), nó cung cấp giao diện đồ họa giúp cấu hình, đặt các tùy chọn cho SQL Server một cách trực quan visual, hỗ trợ nhiều wizard.

Khởi động EM từ menu Start -> Programs-> Microsoft SQL Server -> Enterprise Manager.



Hình : Giao diện của Enterprise Manager

Có thể quản lý nhiều Server trên cùng một EM, giúp cho việc quản trị được dễ dàng đối với các Server ở xa thông qua môi trường mạng. Để thêm một Server vào EM ta phải tiến hành thao tác New SQL Server Registration và tuân thủ các bước theo Wizard. Riêng Server cài đặt trên máy đang sử dụng EM thì được Add vào một cách tự động, nghĩa là khi khởi động EM chúng ta đương nhiên đã có sẵn một Server Localhost để thực hiện thao tác quản trị trên đó.

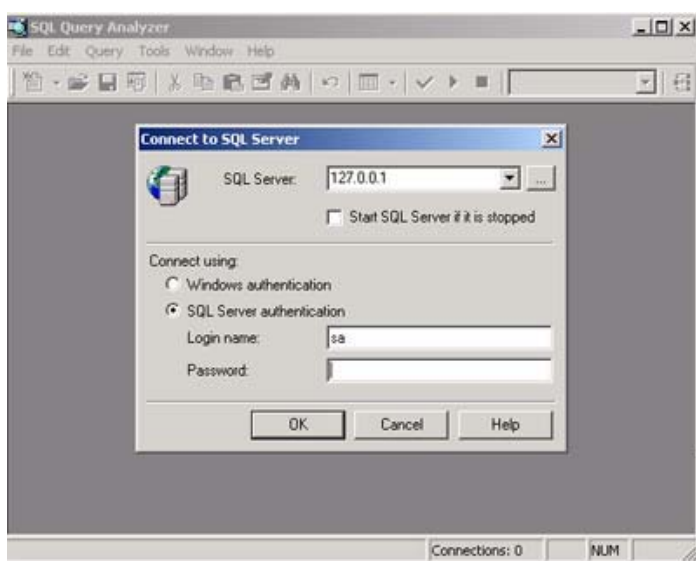


Hình. Thêm một Server vào EM

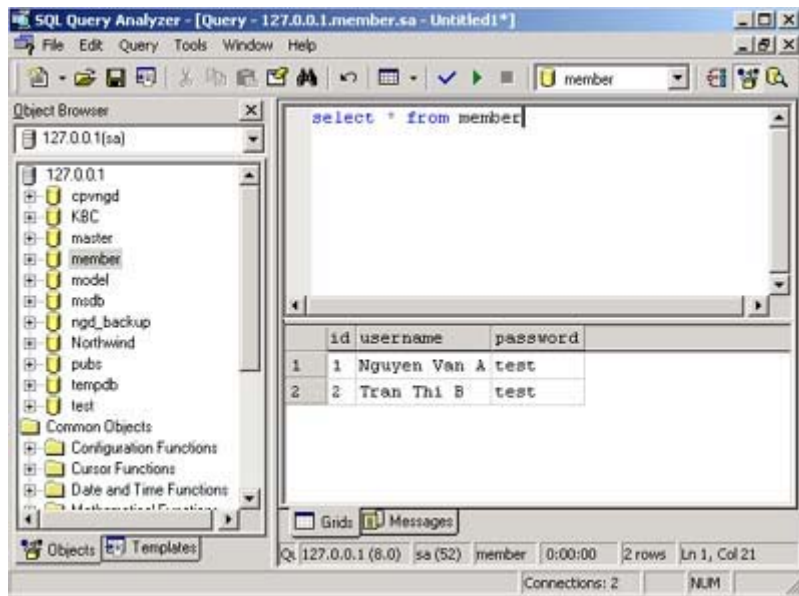
### 1.3.2 Query Analyzer (QA):

Một công cụ khác cũng không kém quan trọng để tương tác với SQL server là công cụ Query Analyzer. Nó cho phép tiếp cận SQL Server dưới vai trò một nhà phát triển (Developer). Nếu như dùng EM ta có thể nhìn và quản lý toàn bộ các đối tượng của SQL Server thì dùng QA chúng ta tiếp cận với SQL chủ yếu bằng các mã lập trình, các query.

Khởi động QA từ menu Start -> Programs-> Microsoft SQL Server -> Query Analyzer. Hệ thống sẽ hỏi SQL Server nào cần kết nối, tài khoản kết nối đến SQL Server đó.



Sau khi vượt qua giai đoạn đăng nhập, ta sẽ làm việc với màn hình chính của QA



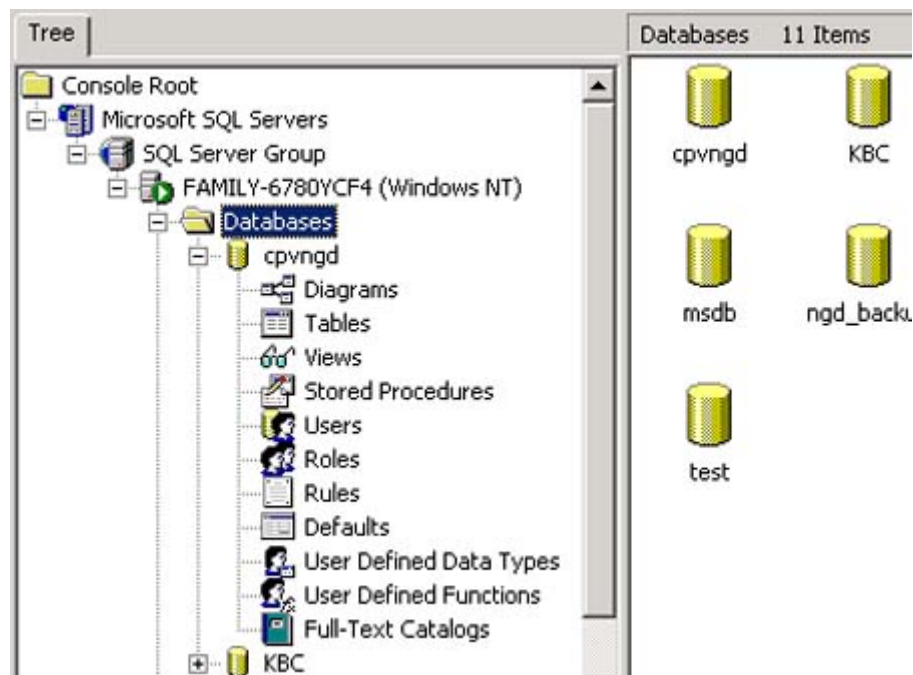
**Hình. Giao diện chính của QA**

**Trong giáo trình này, mọi thao tác về mã lập trình như Query, Script, Stored Procedure, hàm ...được ngầm định hiểu là thực hiện bằng QA. Ví dụ thao tác sử dụng Query ‘Select \* from member’ được thực thi và trả về kết quả trong QA như trong hình vẽ.**

## CHƯƠNG II. THÀNH PHẦN CỦA SQL SERVER 2000

### II.1 CÁC THÀNH PHẦN CỦA SQL SERVER 2000

SQL Server 2000 là một hệ quản trị CSDL quan hệ, vì vậy nó có những đặc trưng chung của loại CSDL quan hệ như database, bảng, ràng buộc...và những thành phần riêng của SQL Server. Lướt qua giao diện trên EM của một SQL Server ta thấy những thành phần căn bản sau:



**Database:** CSDL là đối tượng chính và chứa tất cả những đối tượng còn lại của CSDL như bảng, khung nhìn...

**Table:** Bảng trong CSDL. Bảng đại diện cho một thực thể (entity)

**Diagram:** sơ đồ quan hệ (relationship)

**View:** Khung nhìn hay Virtual Table (bảng ảo)

**Stored Procedure:** thủ tục thường trú.

**User Defined Function:** Hàm do người dùng định nghĩa

**Role:** các quy định vai trò và chức năng trong hệ thống

**Rule:** các quy tắc

**Default:** Các giá trị mặc định

**User Defined Data Type:** kiểu dữ liệu do người dùng tự định nghĩa.

**Full Text Catalog:** tập phân loại dữ liệu Text.

## *II.2 CƠ SỞ DỮ LIỆU (CSDL)*

CSDL trong SQL Server là loại đa người dùng (multi user). Chỉ có thể truy cập vào CSDL dựa vào những tài khoản người dùng (user + password) riêng biệt tương ứng với những quyền truy cập nhất định.

Khi cài đặt, SQL có sẵn một số CSDL mặc định sau:

**Master**

**Model**

**MSDB**

**TempDB**

**Pubs**

**Northwind**

### *II.2.1 Master:*

Là CSDL chứa cấu hình chính của SQL Server. Nó chứa thông tin về các CSDL khác tồn tại trên Server, chứa các cài đặt cấu hình của SQL Server, thông tin về tài khoản truy nhập... Chính vì sự quan trọng của CSDL này mà SQL Server có sẵn công cụ để phục hồi lại CSDL này trong trường hợp bị hỏng.

### *II.2.2 Model:*

Chứa tất cả các Template dùng làm mẫu để SQL Server tạo một CSDL mới. Khi ta tạo một CSDL mới thì SQL Server copy tất cả các mẫu bao gồm bảng, khung nhìn ... từ CSDL Model.

### *II.2.3 MSDB:*

Dùng cho một vài Component của SQL Server như dịch vụ SQL Server Agent, ngoài ra còn có các thông tin khác như replication,....

### *II.2.4 TempDB:*

SQL Server dùng CSDL này để làm không gian làm việc cho các bảng, Stored Procedure tạm thời

### *II.2.5 Pubs:*

Một trong 2 CSDL mẫu cho người sử dụng tham khảo, mô phỏng thông tin về xuất bản sách.

## II.2.6 Northwind:

CSDL mẫu thứ 2 của SQL Server cho người dùng tham khảo.

## II.3 CÁC ĐỐI TƯỢNG BÊN TRONG MỘT CSDL

### II.3.1 Bảng (Table):

Phần chính của một CSDL, là nơi lưu trữ dữ liệu thực sự. Mỗi bảng gồm nhiều trường (column), mỗi trường chứa một kiểu dữ liệu tương ứng. Khi tạo bảng, cần quan tâm một số yếu tố:

**Name:** tên trường

**Key:** trường đó có phải là khóa không

**Datatype:** kiểu dữ liệu của trường.

**Size:** kích thước dữ liệu (nếu có)

**Allow Null:** cho phép giá trị rỗng (null) không

**Default:** giá trị mặc định cho trường

**Identity:** trường có thể tự động tăng, tương tự như kiểu AutoNumber trong Access, nếu thuộc tính Identity là Yes .

**Identity Seed:** giá trị khởi đầu của trường có Identity là Yes.

**Identity Increment:** bước nhảy cho mỗi lần tăng đối với trường có Identity là Yes.

### II.3.2 Khung nhìn (View):

Là một bảng ảo được tạo thành từ một bảng, hoặc một vài bảng (nếu sử dụng truy vấn Join). Về bản chất View là kết quả của một truy vấn (query) trên bảng.

Ví dụ: `select * From Nha where Diadiem ='So 14 Lang Ha'`

### II.3.3 Lược đồ quan hệ (Diagram):

Quan hệ giữa các thực thể (entity, ở đây là bảng) được quyết định dựa trên việc phân tích thiết kế hệ thống, tương tự như Relationship trong Access.

### II.3.4 Chỉ mục (Index):

Tồn tại đối với Bảng hoặc Khung nhìn, giúp tăng tốc độ truy cập dữ liệu, đặc biệt là đối với những trường nào cần tìm kiếm thông tin.

Lập chỉ mục cho một trường cũng có tác dụng tương tự như lập mục lục cho một cuốn sách, rất cần thiết cho việc tìm kiếm thông tin.

### II.3.5 Trigger:

Tồn tại trong bảng, nó là một đoạn mã tự động thực thi khi có một hành động Insert, Update, Delete xảy ra trong bảng.

### II.3.6 Ràng buộc (Constraint):

Tồn tại đối với bảng, cho phép ràng buộc dữ liệu trong một bảng hoặc các bảng, buộc dữ liệu phải tuân theo một quy tắc nào đó.

### II.3.7 Thủ tục thường trú (Stored Procedure):

Đoạn mã lập trình (Script) được tạo ra, quản lý và thực thi trong SQL Server, cũng tương tự như các hàm lập trình của các ngôn ngữ lập trình.

## II.4 KIỂU DỮ LIỆU (DATATYPE)

Bất kỳ một trường nào trong bảng cũng phải có một kiểu dữ liệu xác định. Một số kiểu cho phép định nghĩa độ lớn của kiểu, một số khác thì không. Một số kiểu dữ liệu phổ biến :

Kiểu dữ liệu	Loại	Số Bytes	Mô tả
Bit	Integer	1	Có giá trị 0 (true) hoặc 1 (false)
BigInt, Int, SmallInt, TinyInt	Integer	8,4,2,1	Các kiểu số nguyên có khoảng giá trị từ lớn đến nhỏ
Decimal, Numeric	Decimal/Numeric		
Money, SmallMoney	Money	4	Dùng cho dữ liệu kiểu tiền tệ
Float			kiểu số thực dấu phẩy động
Datetime	Date/Time	8	kiểu dữ liệu ngày tháng
Char, varchar, nchar, nvarchar, text, ntext			kiểu dữ liệu ký tự thích hợp cho lưu trữ các ký tự hoặc chuỗi
Binary, VarBinary	Binary		Dữ liệu kiểu nhị phân
Image	Binary		Dữ liệu dạng nhị phân

			<b>dùng lưu ảnh</b>
<b>SQL_Variant</b>			<b>Tương tự kiểu variant trong Visual Basic</b>



## CHƯƠNG III. THAO TÁC CĂN BẢN TRÊN SQL SERVER

Trong hầu hết các trường hợp, chúng ta có 2 cách để thao tác với SQL Server: hoặc thao tác với giao diện trực quan, hỗ trợ Wizard của EM, hoặc thao tác bằng mã lệnh (Script) với QA. Thông thường dùng EM dễ dàng hơn vì không cần viết Script, nhưng cũng có nhiều trường hợp không thể dùng EM hoặc dùng QA tốt hơn. Đặc biệt có thể dịch các thao tác với EM thành các Script để dùng với QA trong những trường hợp cần thiết, chẳng hạn như cần đóng gói thao tác thành một file Script để mang cài đặt cho một Server khác.

### III.1 TẠO CSDL

#### III.1.1 Tạo bằng dòng lệnh với QA

Cú pháp: **Create Database <Name>**

**[on**

**()**

**|**

**[log on**

**()**

**|**

**VD: Create Database Quanlynha**

**Hay đầy đủ hơn là:**

**CREATE DATABASE [Quanlynha]**

**ON (NAME = N'Quanlynha', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL\data\Quanlynha.mdf' ,**

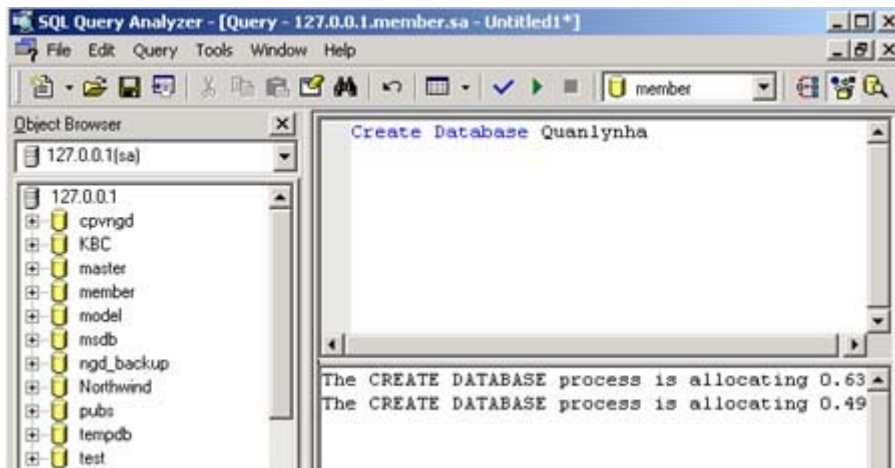
**SIZE = 1, FILEGROWTH = 10%)**

**LOG ON**

**(NAME = N'Quanlynha\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL\data\Quanlynha\_log.LDF' ,**

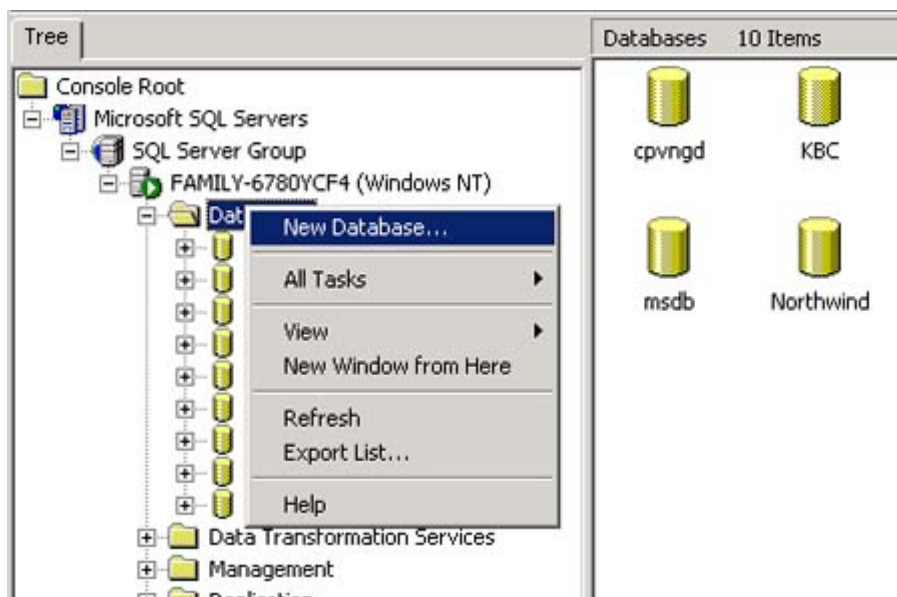
**FILEGROWTH = 10%)**

**GO**

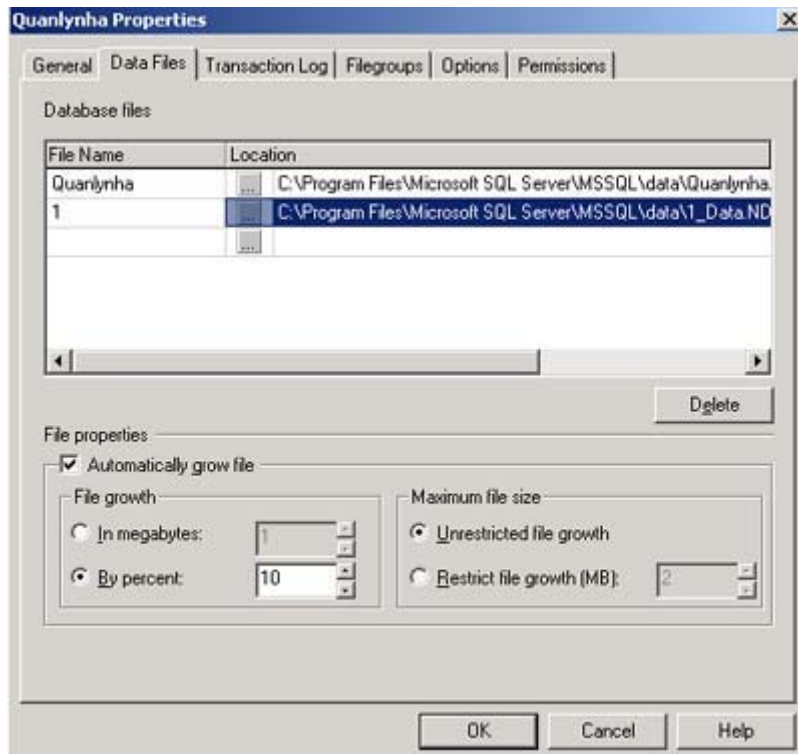


### III.1.2 Tạo CSDL bằng giao diện Enterprise Manager:

**Chọn tính năng New Database rồi đặt tên cho Database là Quanlynha trong Wizard đó.**



**Chúng ta có thể cấu hình các thuộc tính (properties) nâng cao cho một CSDL như tên Data Files, Transaction Log, FileGroups, Options, Permissions bằng cách nhấn vào Properties của CSDL đó.**



### III.2 TẠO BẢNG (TABLE)

Giả sử chúng ta xây dựng một CSDL Quanlynha gồm có 3 bảng

**Nha (NhaID, Diadiem, Dientich):** chứa thông tin về nhà

**Coquan (CoquanID, Tên):** chứa thông tin về cơ quan

**Hopdong (HopdongID, NhaID, CoquanID, NgayBatdau, NgayKetthuc, Giathue):**

chứa thông tin về hợp đồng thuê nhà.

với mục đích thực hiện lưu trữ dữ liệu về quản lý việc cho các cơ quan thuê nhà .

#### III.2.1 Tạo bằng dòng lệnh với QA

**Cú pháp:**

**Create Table [name]**

**(<column Name> <datatype>)**

**VD: CREATE TABLE Nha (**

**NhaId int IDENTITY (1, 1) NOT NULL ,**

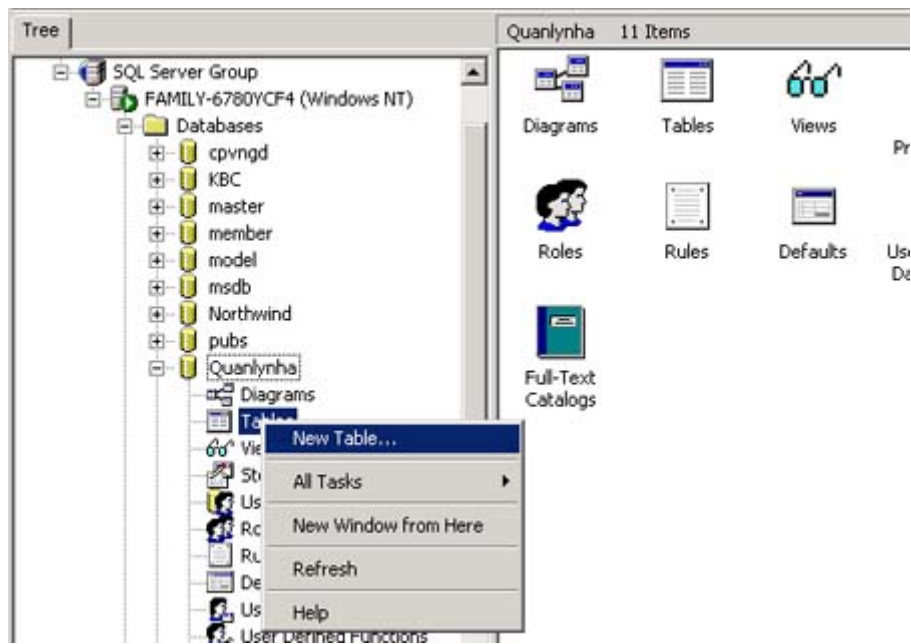
**DiaDiem nvarchar (50) NOT NULL ,**

**DienTich int NULL**

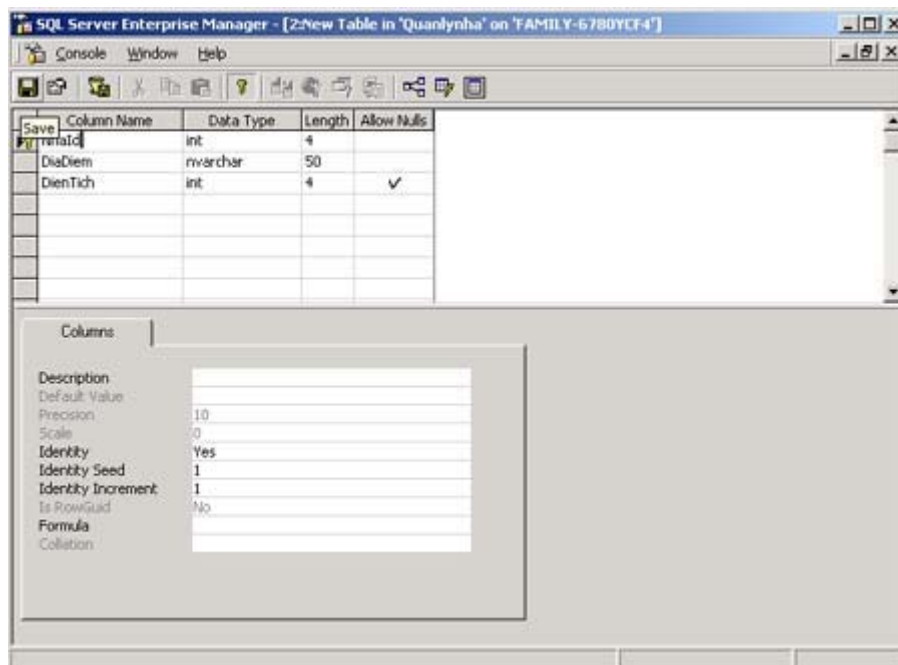
**) ON [PRIMARY]**

### III.2.2 Tạo bằng giao diện EM:

Từ đối tượng Table của CSDL Quanlynha chọn lệnh New Table



Tiếp theo EM sẽ cung cấp một bảng mới ở chế độ Design để ta thiết kế các trường dữ liệu, sau khi kết thúc nhấn Save và đặt tên cho bảng là Nha



### *III.3 THAY ĐỔI CẤU TRÚC BẢNG:*

#### *III.3.1 Thay đổi bằng dòng lệnh với QA*

**Cú pháp:**

**Alter Table [name]**

**(<column Name> <datatype>)**

#### *III.3.2 Thay đổi bằng giao diện EM:*

Từ đối tượng Table chọn bảng cần thay đổi cấu trúc rồi chọn Design. Ta sẽ gặp lại chế độ Design bảng giống như lúc tạo bảng. Sau khi kết thúc sửa bảng chọn Save

### *III.4 XOÁ :*

#### *III.4.1 Xoá CSDL:*

**DROP Database Quanlynha**

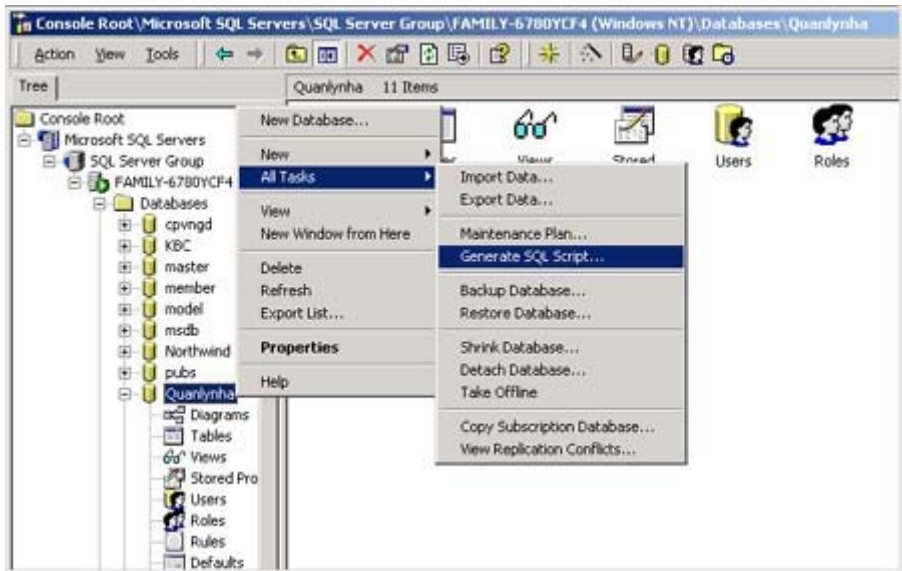
#### *III.4.2 Xoá bảng:*

**DROP Table Nha,Coquan**

### *III.5 TẠO SCRIPT:*

Trong trường hợp chúng ta muốn nhận được toàn bộ Script cho một thao tác trên EM, chẳng hạn ta thực hiện thủ tục tạo một bảng bằng EM và muốn nhận được Script tương ứng với toàn bộ thủ tục đó để có thể dùng trong QA (điều này đặc biệt tiện dùng khi ta cần cài đặt một ứng dụng phát triển ở chỗ này sang một Server ở chỗ khác)

Để làm như vậy ta sử dụng chức năng Generate Script của EM



## CHƯƠNG IV. CHUẨN HOÁ VÀ QUAN HỆ

Quan hệ giữa các bảng được xây dựng dựa vào kết quả của quá trình chuẩn hoá trong giai đoạn phân tích và thiết kế hệ thống. Có 3 loại quan hệ:

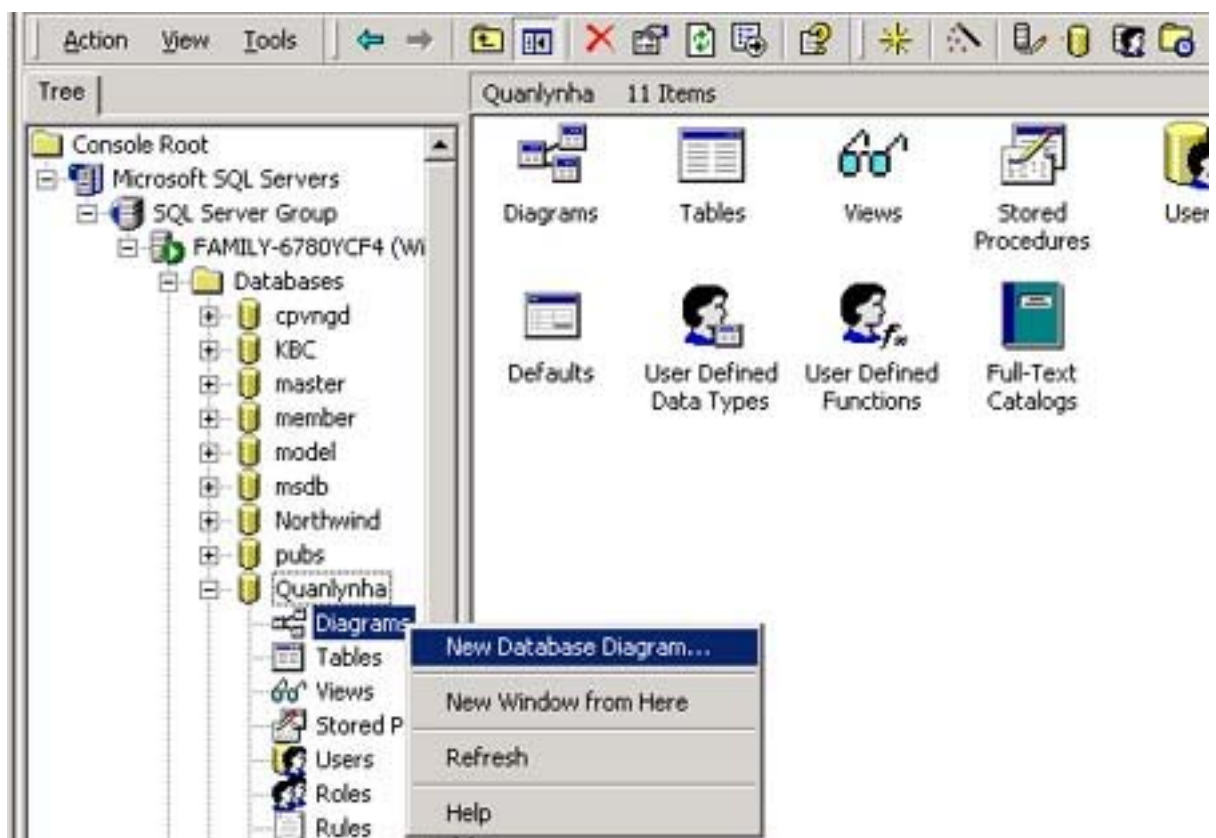
**Quan hệ 1-1:** ứng với mỗi bản ghi ở bảng thứ nhất có quan hệ duy nhất một bản ghi ở bảng thứ 2.

**Quan hệ 1 - nhiều:** một bản ghi ở bảng thứ nhất ứng với một hay nhiều bản ghi ở bảng thứ 2. Đây là quan hệ thông dụng nhất.

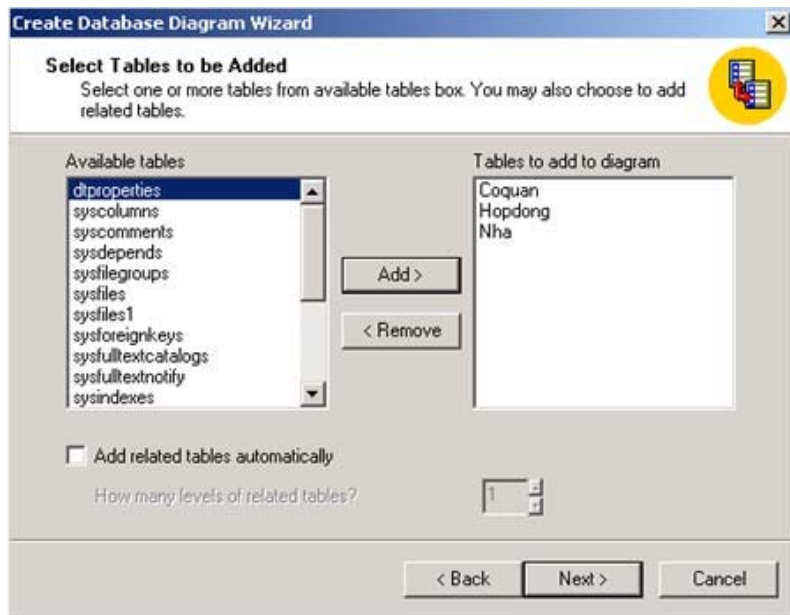
**Quan hệ nhiều - nhiều:** một bản ghi ở bảng thứ nhất ứng với một hay nhiều bản ghi ở bảng thứ 2 và ngược lại.

### IV.1 LƯỢC ĐỒ QUAN HỆ (DIAGRAM)

Dùng chức năng New Database Diagram của EM:



Sau đó dùng wizard chọn các bảng muốn tạo lược đồ quan hệ.



dùng chuột kéo và thả các trường của các bảng cần tạo quan hệ tương ứng. Khi chọn Option ta có thể đánh dấu chọn vào các mục Cascade Update Related Fields và Cascade Delete Related Records để xóa hay cập nhật các bảng có quan hệ khi dữ liệu một bảng thay đổi.



Cuối cùng ta được một lược đồ quan hệ như sau:





## CHƯƠNG V. KHUNG NHÌN (VIEWS)

### V.1 KHÁI NIỆM CƠ BẢN

View cũng tương tự như Query trong Access. Trong nhiều trường hợp chúng ta cần xây dựng kết quả dữ liệu từ nhiều bảng khác nhau, hoặc chỉ chọn lọc thông tin nhất định trong một bảng, đó chính là View. View có những ưu điểm sau:

- Hạn chế tính phức tạp của dữ liệu đến người dùng, tăng tốc độ đối với ứng dụng lớn.
- Kết nối dữ liệu từ nhiều bảng lại với nhau
- Sử dụng tài nguyên Server để truy vấn
- Tạo ra bảng ảo có dữ liệu như yêu cầu.
- Có thể kết hợp một số hàm và phương thức để tạo ra một cột (trường) mới

### V.2 TẠO VIEW:

#### V.2.1 Tạo bằng dòng lệnh với QA

Cú pháp:

```
Create View <name>
```

```
As
```

```
...
```

VD: Ta cần tạo một View có tên là ViewHopdonglon nhằm mục đích lưu giữ thông tin về các hợp đồng có giá tiền thuê một tháng từ 10.000.000 trở lên.

```
CREATE VIEW VIEWHopdonglon
```

```
AS
```

```
SELECT    Coquan.TenCoquan, Nha.DiaDiem, Hopdong.Giatien
```

```
FROM      Hopdong INNER JOIN
```

```
          Coquan ON Hopdong.CoquanID = Coquan.CoquanID
```

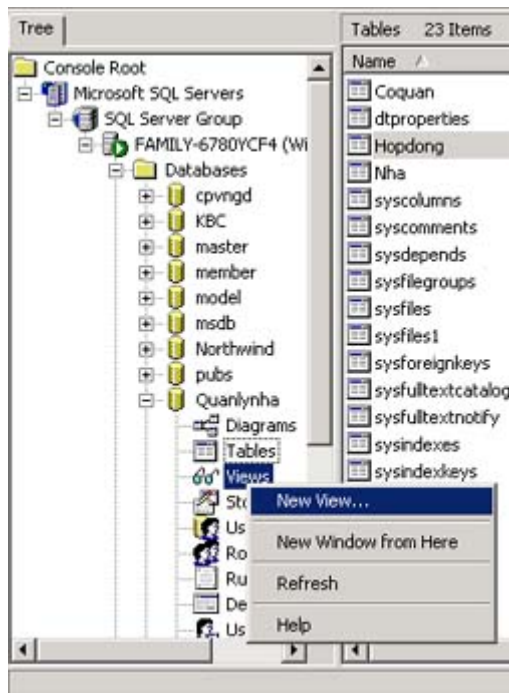
```
INNER JOIN
```

```
          Nha ON Hopdong.NhaID = Nha.NhaId
```

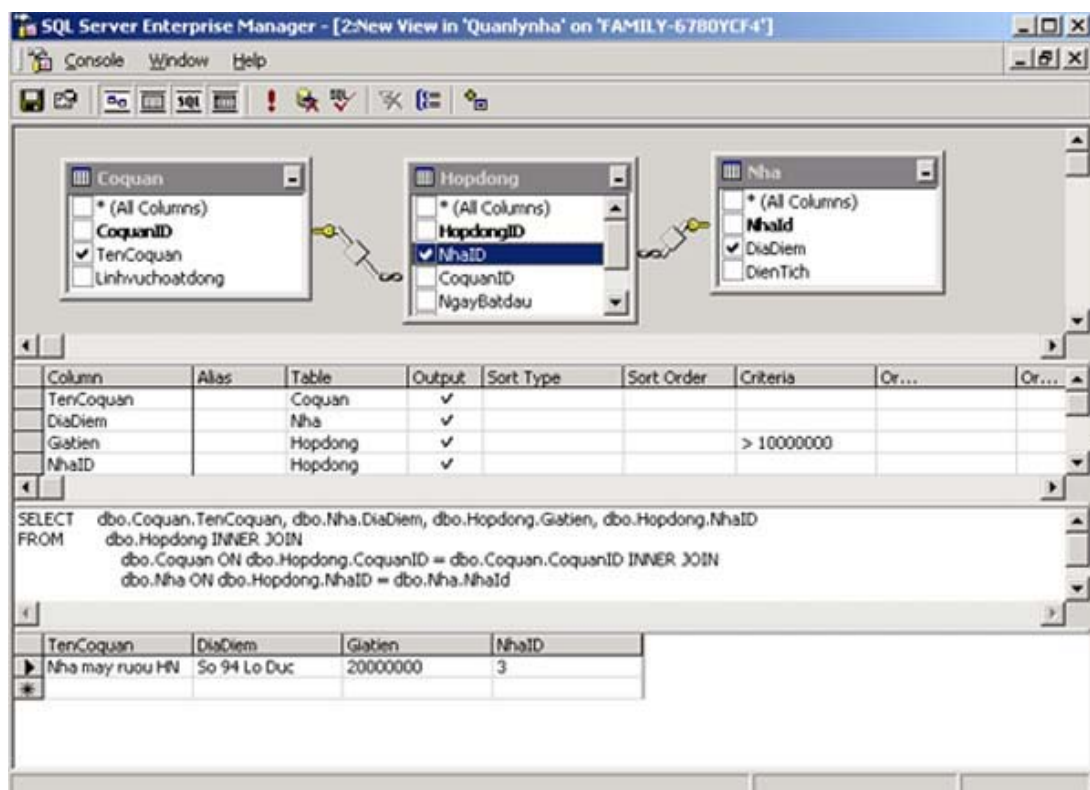
```
WHERE     (Hopdong.Giatien > 10000000)
```

#### V.2.2 Tạo bằng giao diện EM:

Từ đối tượng View của CSDL Quanlynha chọn lệnh New View



Tiếp theo EM sẽ cung cấp một cửa sổ ở chế độ Design để ta thiết kế view, hoàn thành thiết kế ta chọn Run để xem kết quả của View, Save và đặt tên cho view là ViewHopdonglon



Qua đây chúng ta thấy:

- View có tác dụng như một bộ lọc dữ liệu của bảng: khi chúng ta dùng mệnh đề **where** trong câu lệnh **view**. (chỉ xây dựng những hợp đồng có giá tiền >10.000.000).
- View có thể có cấu trúc phức tạp, được hợp thành từ nhiều bảng: dùng mệnh đề **inner join**.
- Sử dụng View để thay đổi dữ liệu: vì view là một bảng ảo nên không giống như bảng, ta chỉ có thể thay đổi dữ liệu của bảng bằng view trong một số trường hợp nhất định.

## **CHƯƠNG VI. QUẢN TRỊ CSDL. CÁC THAO TÁC BACKUP, RESTORE, IMPORT, EXPORT**

### *VI.1 CĂN BẢN VỀ QUẢN TRỊ CSDL*

Còn gọi là DBA (Database Administration). Thông thường đối với một ứng dụng CSDL, ngoài việc phát triển ứng dụng thì quản trị CSDL là phần không thể thiếu được trong quá trình ứng dụng vận hành.

Để quản trị và bảo trì CSDL đang vận hành, dữ liệu thay đổi theo thời gian và không gian, người quản trị cần quan tâm đến tất cả các yếu tố có thể xảy ra đối với CSDL.

Muốn trở thành một nhà quản trị CSDL, cần quan tâm tới những việc sau:

- Sao lưu (backup) và phục hồi (restore) dữ liệu.
- Import và Export dữ liệu.
- Quản lý tài khoản người dùng CSDL.
- Và nhiều chức năng khác.

### *VI.2 SAO LƯU (BACKUP) VÀ PHỤC HỒI (RESTORE)*

Đối với những hệ CSDL lớn như SQL Server, sao lưu dữ liệu là cần thiết bởi dữ liệu luôn luôn được thay đổi bởi nhiều người dùng, khả năng sai sót có thể xảy ra.

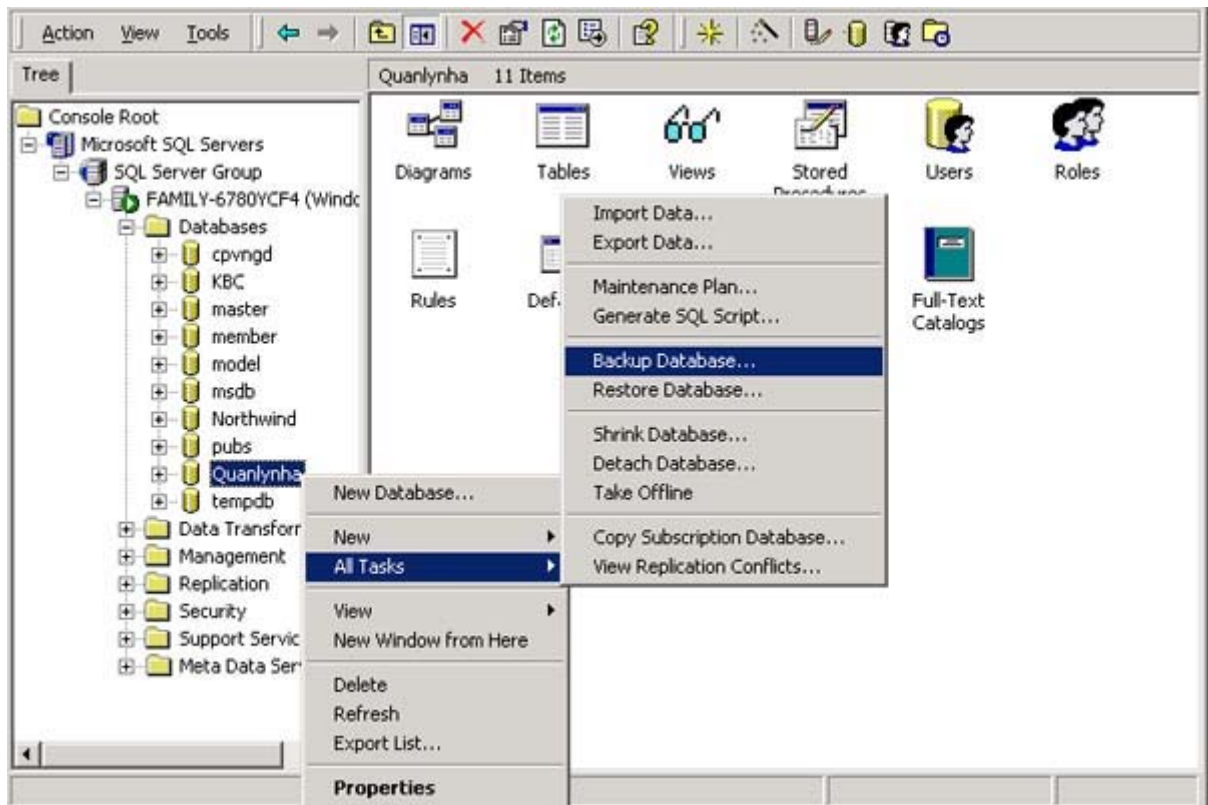
Mặt khác trong trường hợp xảy ra sự cố hệ thống, dữ liệu có thể bị hỏng.

Để bảo vệ CSDL được an toàn và có thể phục hồi khi cần, chúng ta phải có kế hoạch sao lưu dữ liệu một cách thường xuyên và theo định kỳ.

Một ứng dụng khác của 2 quá trình backup và restore là khi ta muốn copy một CSDL từ Server này sang Server khác. Khi đó ta có thể backup ở Server 1 và mang sang restore ở Server 2

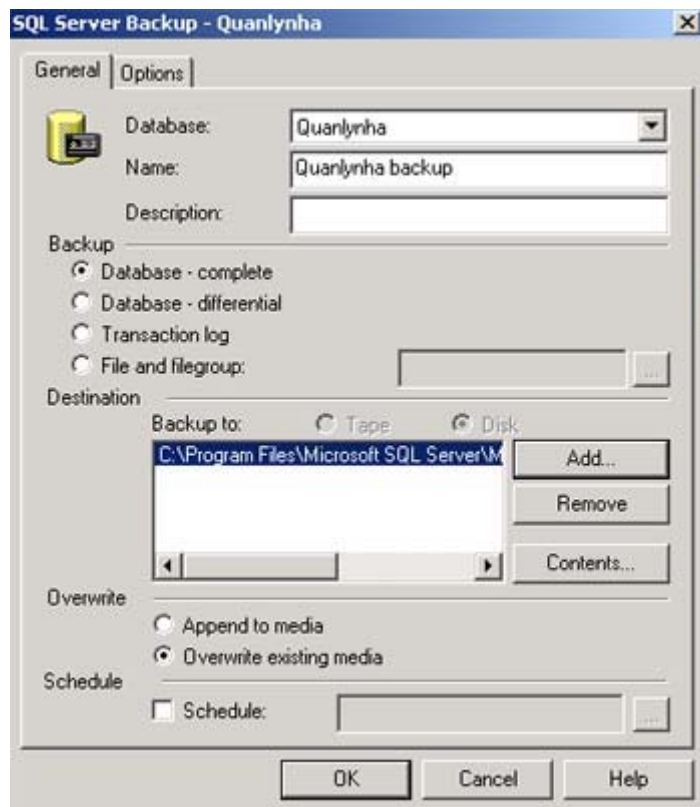
#### *VI.2.1 Backup*

Có nhiều thiết bị để backup dữ liệu, thông thường chúng ta backup dữ liệu ra đĩa cứng. Cũng như các thao tác khác, ta có thể backup thông qua giao diện EM hoặc bằng script thông qua QA. Ở đây trình bày quá trình backup bằng EM. chọn chức năng Backup Database như hình vẽ:



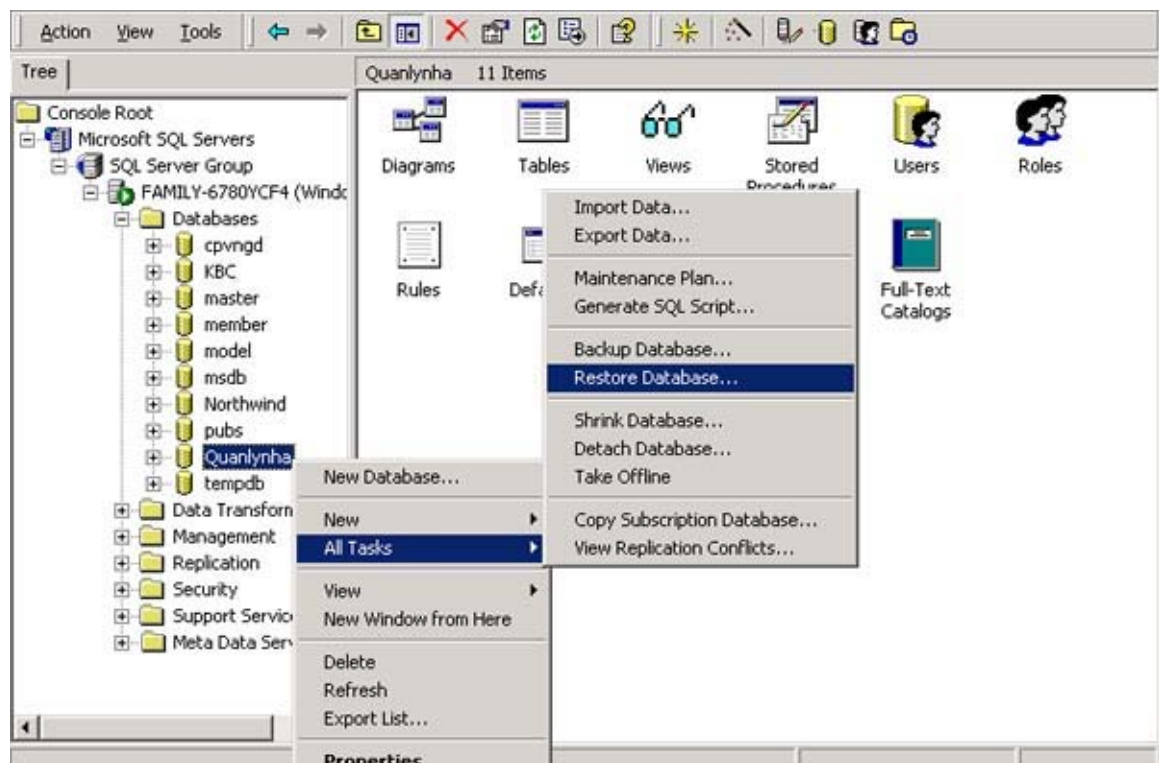
**Đặt tên file Backup, và xác nhận để SQL thực hiện quá trình Backup CSDL.**

**Toàn bộ CSDL được Backup thành một file, ta có thể lưu file này để phục hồi lại CSDL khi cần thiết. Cũng có thể chọn một số tùy chọn để Backup như Backup toàn bộ CSDL, Backup những dữ liệu mới cập nhật, Backup log, Backup đè lên file Backup cũ hay chỉ bổ sung vào file Backup cũ, và có thể đặt lịch (scheduled) cho quá trình Backup diễn ra tự động theo định kỳ...**

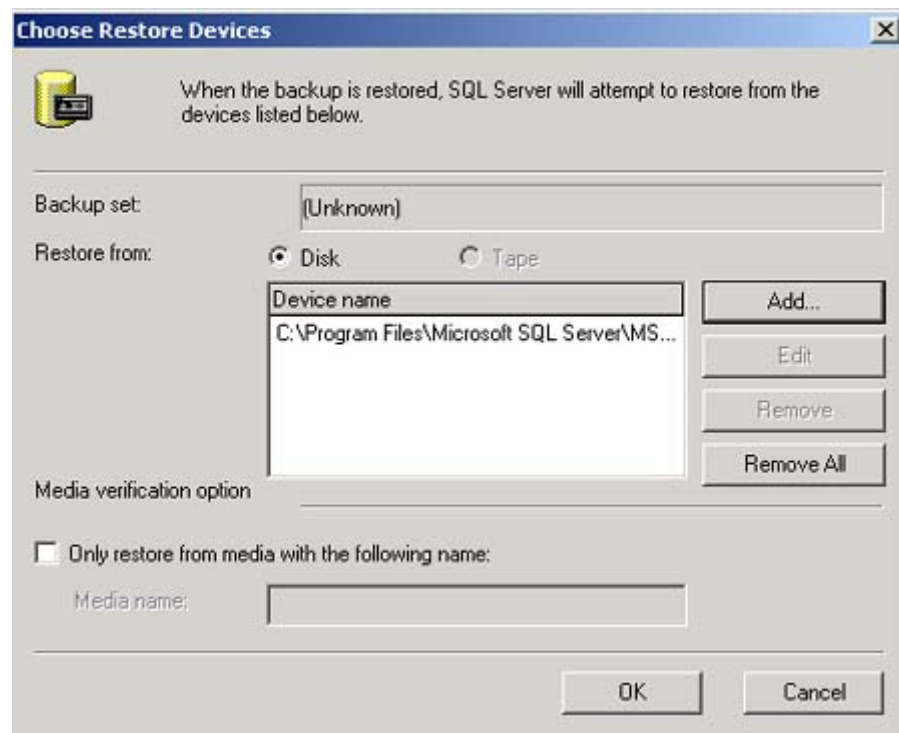


### VI.2.2 Restore

Là quá trình ngược lại với Backup, khi muốn phục hồi lại dữ liệu chúng ta chọn **Restore Database** như hình vẽ:



**chọn file chứa backup để restore rồi nhấn ok**



### *VI.3 IMPORT – EXPORT*

**Chúng ta đang có một CSDL và muốn nhận dữ liệu hoặc các đối tượng từ một CSDL khác, có thể trong cùng một SQL Server hoặc ở một SQL Server khác, hoặc thậm chí là CSDL của một hệ quản trị CSDL khác như Access, Oracle...**

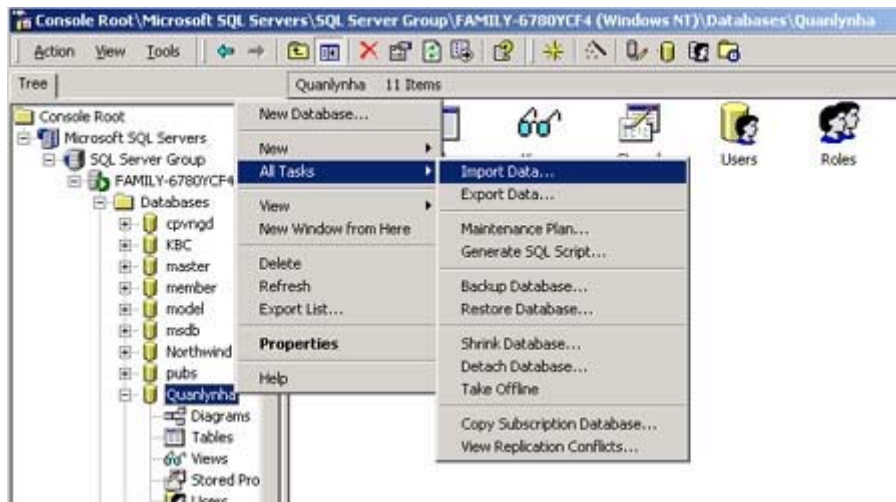
**Ta thực hiện việc này bằng IMPORT trong EM**

**Ngược lại với quá trình Import là Export, khi ta muốn xuất các đối tượng bên trong CSDL hiện hành cho CSDL khác.**

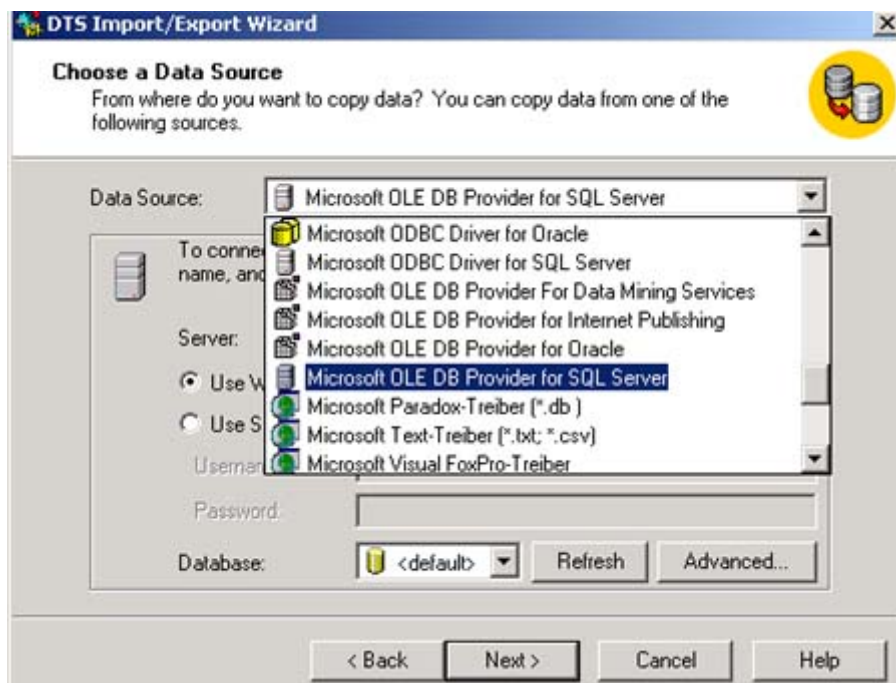
#### *VI.3.1 IMPORT*

**Từ CSDL hiện hành chọn chức năng Import Data. Để Import ta phải chọn CSDL nguồn (Source) và CSDL đích (Destination – chính là CSDL hiện hành).**



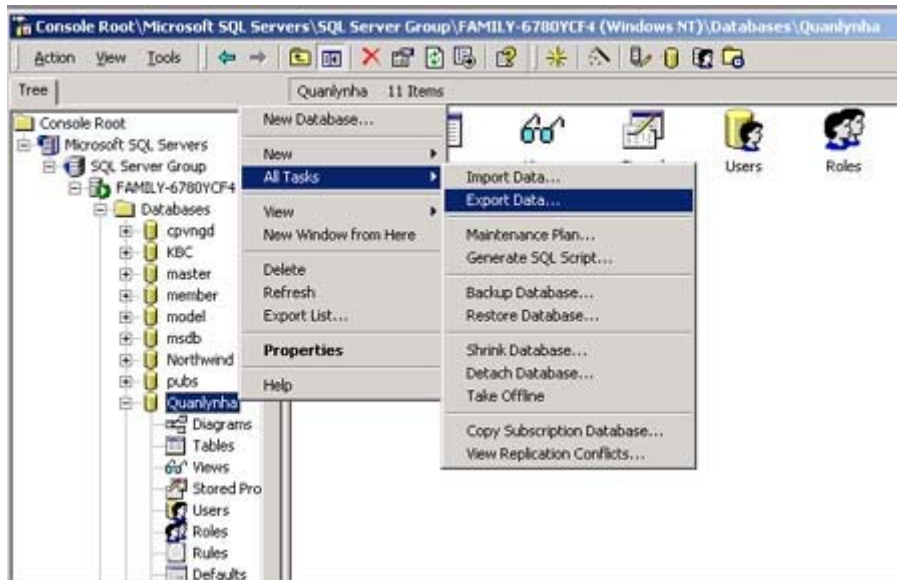


CSDL nguồn có thể cùng loại SQL server hoặc khác loại (Access, Oracle..)

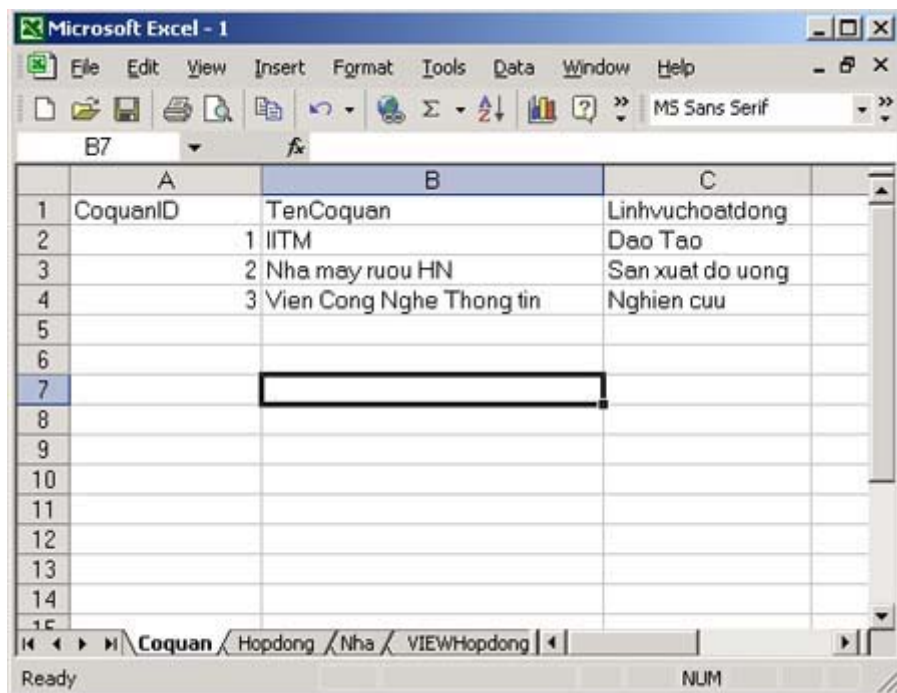


### VI.3.2 EXPORT

Từ CSDL hiện hành chọn chức năng Export Data



Chọn nguồn (Destination) là CSDL hiện hành, đích là CSDL muốn Export ra.  
 Sau đây là một ví dụ Export Data từ CSDL Quanlynha ra file Excel, kết quả ta  
 được 1 file Excel có 4 worksheet: 3 bảng và 1 view



## CHƯƠNG VII. SCRIPT VÀ BATCH

### VII.1 KHÁI NIỆM VỀ SCRIPT

Là tập hợp câu lệnh của SQL Server để thực hiện một thao tác. Chẳng hạn chuỗi lệnh tạo một bảng có tên là 'Hopdong' trong CSDL Quanlynha như sau được gọi là 1 Script:

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Hopdong]')
and OBJECTPROPERTY(id, N'IsUserTable') = 1)
```

```
drop table [dbo].[Hopdong]
```

```
GO
```

```
CREATE TABLE [dbo].[Hopdong] (
```

```
    [HopdongID] [int] IDENTITY (1, 1) NOT NULL ,
```

```
    [NhaID] [int] NOT NULL ,
```

```
    [CoquanID] [int] NOT NULL ,
```

```
    [NgayBatdau] [datetime] NULL ,
```

```
    [NgayKetthuc] [datetime] NULL ,
```

```
    [Giatien] [int] NULL
```

```
) ON [PRIMARY]
```

```
GO
```

### VII.2 BATCH

Một Script có thể chứa nhiều batch. Mỗi batch là một khối lệnh được kết thúc bởi lệnh GO. Ví dụ trên có 2 batch. Batch đầu tiên thực hiện việc kiểm tra xem bảng Hopdong đã tồn tại trong CSDL chưa, nếu có thì xoá bảng đó đi để chuẩn bị tạo một bảng mới. Batch thứ 2 thực hiện việc tạo bảng mới trong CSDL có tên là Hopdong. Sở dĩ người ta chia Script ra thành các Batch đánh dấu bởi lệnh kết thúc GO vì các Batch được xem như một khối lệnh độc lập. Nếu một Batch xảy ra lỗi thì các Batch khác vẫn có thể thực thi.

### VII.3 SCRIPT ĐƠN GIẢN

Script được ghi thành một tập tin thường có phần đuôi là .sql

Tham khảo ví dụ đơn giản sau:

```
Use account
```

```
Go
```

**Declare @tablename varchar(128)**

**Set @tablename='tblCustomer'**

**Exec ('select \* from '+@tablename+'where active=1')**

*VII.3.1 USE:*

**Khai báo Cơ sở dữ liệu hiện tại**

*VII.3.2 Khai báo biến trong SQL Server:*

**Declare @tenbien kiudulieu. VD: declare @amount float**

**Gán giá trị cho biến: Set @amount=1000**

**Set @amount = (select sum(amount) from test)**

*VII.3.3 OSQL:*

**Công cụ cho phép thực hiện một phát biểu SQL bằng dấu nhắc hay cửa sổ DOS.**

*VII.3.4 SQL động:*

**là một phát biểu SQL không định nghĩa trước, ví dụ câu lệnh select có tên bảng từ ngoài đưa vào hay tên bảng là kết quả của một phép toán.**

## **CHƯƠNG VIII. CÁC HÀM VÀ BIỂU THỨC HỖ TRỢ TRUY VẤN.**

Chúng ta đã biết để thao tác với data trong CSDL, người ta dùng cú pháp SQL với các loại câu lệnh căn bản như: **Select, Insert, Update, Delete.**

**VD: Select \* From Nha**

Những mệnh đề giúp cho việc thao tác với dữ liệu được linh hoạt hơn nữa như là **Where, Group By, Having, Order By, Join ...**

SQL Server sử dụng T-SQL (Transact SQL) để thao tác với dữ liệu, hay nói khác đi T-SQL chính là ngôn ngữ để viết Script trong SQL. Ngoài những lệnh SQL chuẩn như trên, T-SQL còn bao gồm các hàm để hỗ trợ xử lý dữ liệu ngay bên trong SQL Server thay vì phải dùng các ngôn ngữ lập trình khác.

### *VIII.1 CÁC HÀM VÀ BIỂU THỨC HỖ TRỢ TRUY VẤN*

#### *VIII.1.1 AVG:*

Trả về giá trị trung bình.

**VD: tìm giá tiền thuê nhà trung bình trong Hopdong**

**Select AVG (giatien) from Hopdong**

#### *VIII.1.2 Min :*

Trả về giá trị nhỏ nhất.

#### *VIII.1.3 Max :*

Trả về giá trị lớn nhất.

#### *VIII.1.4 Count*

Trả về số lượng bản ghi thỏa mãn.

**VD: Tìm số lượng các Hợp đồng có giá tiền > 10.000.000**

**Select Count (\*) from ViewHopdonglon**

#### *VIII.1.5 Sum:*

Trả về tổng giá trị của trường

### *VIII.2 CÁC HÀM XỬ LÝ CHUỖI*

#### *VIII.2.1 ASCII:*

Trả về mã ASCII của ký tự đầu tiên trong chuỗi. Vd: Tìm mã ASCII của ký tự t  
**print ASCII ('t')**

*VIII.2.2 Char:*

**chuyển đổi kiểu mã ASCII từ số nguyên sang dạng chuỗi:**

**VD: Tìm ký tự có mã ASCII là 65**

**print char (65)**

*VIII.2.3 Upper:*

**Chuyển đổi chuỗi sang chữ hoa**

*VIII.2.4 Lower:*

**Chuyển đổi chuỗi sang chữ thường**

*VIII.2.5 Len:*

**Trả về chiều dài chuỗi**

*VIII.2.6 Ltrim:*

**Bỏ các dấu trắng bên trái chuỗi**

**VD: print ltrim (' Viet Nam')**

*VIII.2.7 Rtrim:*

**Bỏ các dấu trắng bên phải chuỗi.**

*VIII.2.8 Left:*

**Trả về chuỗi con tính từ bên trái chuỗi**

**VD: print left ('Viet Nam ',3) Kết quả là chuỗi 'Vie'**

*VIII.2.9 Right:*

**Trả về chuỗi con tính từ bên phải chuỗi**

*VIII.2.10 CharIndex:*

**Trả về vị trí của một chuỗi con bên trong một chuỗi**

**VD : print CharIndex ('Nam','Viet Nam') Kết quả là 5**

*VIII.3 CÁC HÀM THỜI GIAN*

*VIII.3.1 GetDate():*

**Trả về ngày tháng năm của hệ thống:**

**VD: select 'Hôm nay là : ' =getDate()**

### VIII.3.2 *DatePart()*:

Trả về một phần của chuỗi ngày tháng năm

VD: select 'Hôm nay là ngày : ' =DatePart(d,getDate())

### VIII.3.3 *DateDiff()*:

Trả về số ngày giữa 2 khoảng thời gian

VD: select 'số năm thuê là: ' =DateDiff(y,NgayKetthuc, NgayBatdau)

### VIII.3.4 *Day()*:

Trả về ngày trong tháng.

### VIII.3.5 *Month()*:

Trả về tháng trong năm.

### VIII.3.6 *Year()*:

Trả về năm

## VIII.4 CÁC HÀM TOÁN HỌC

### VIII.4.1 *Square*:

Bình phương của một biểu thức. VD: bình phương của 3: Print square (3)

### VIII.4.2 *Sqrt*:

Căn bậc 2 của biểu thức

### VIII.4.3 *Round*:

Làm tròn một biểu thức

## VIII.5 CÁC HÀM CHUYỂN ĐỔI

### VIII.5.1 *Cast*:

Trả về giá trị có kiểu dữ liệu được định nghĩa

VD: Print cast (getDate() as Varchar(11))

### VIII.5.2 *Convert*:

Chuyển đổi giá trị có kiểu này sang kiểu dữ liệu khác nếu cho phép

VD: print convert (int, '255')

## **CHƯƠNG IX. THỦ TỤC THƯỜNG TRÚ STORED PROCEDURE**

### *IX.1 GIỚI THIỆU CHUNG*

Thủ tục thường trú là một đối tượng được tạo thành từ các lệnh T-SQL. Nó được lưu trữ như là một phần của CSDL. Khi cần thực hiện một chuyển tác (Transaction) trong CSDL, đôi khi chúng ta thực hiện chúng bằng cách dùng các Script, hoặc chúng ta có thể lưu chúng trong các hàm gọi là Stored Procedure (SP) và gọi hàm này. SP cũng giống như khái niệm hàm trong ngôn ngữ lập trình.

SP có một số ưu điểm:

Có thể tái sử dụng cho nhiều ứng dụng thay vì phải viết lặp lại một đoạn script cho mỗi ứng dụng.

Được thi hành ngay trong SQL Server (khi một ngôn ngữ lập trình gọi đến SP).

Được hưởng các chế độ Security của SQL Server.

Ta có thể tạo một SP bằng EM hoặc bằng QA.

### *IX.2 TẠO MỘT SP*

Vd: Tính tổng diện tích và doanh thu cho thuê nhà trong khoảng thời gian từ a đến b.

```
CREATE PROCEDURE Doanhthu
```

```
@dauthang smalldatetime,
```

```
@cuoithang smalldatetime
```

```
AS
```

```
select sum(dientich) as dientich,sum(giatien) as doanhthu
```

```
from hopdong inner join nha on hopdong.NhaID=nha.NhaID
```

```
inner join coquan on hopdong.CoquanID=coquan.CoquanID
```

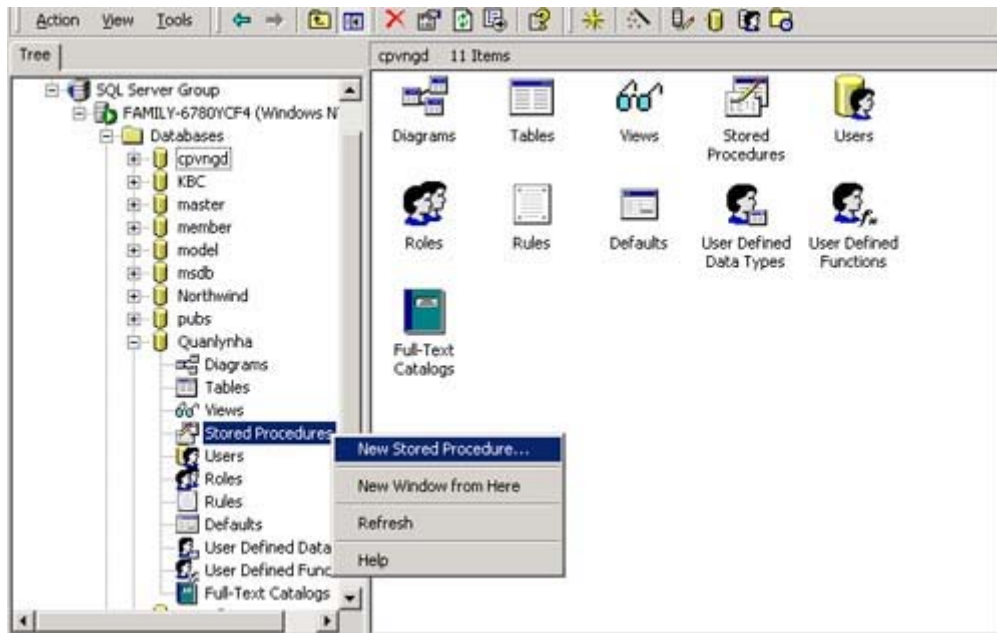
```
where (ngayketthuc >= @dauthang ) and (ngaybatdau<=@cuoithang)
```

```
Return
```

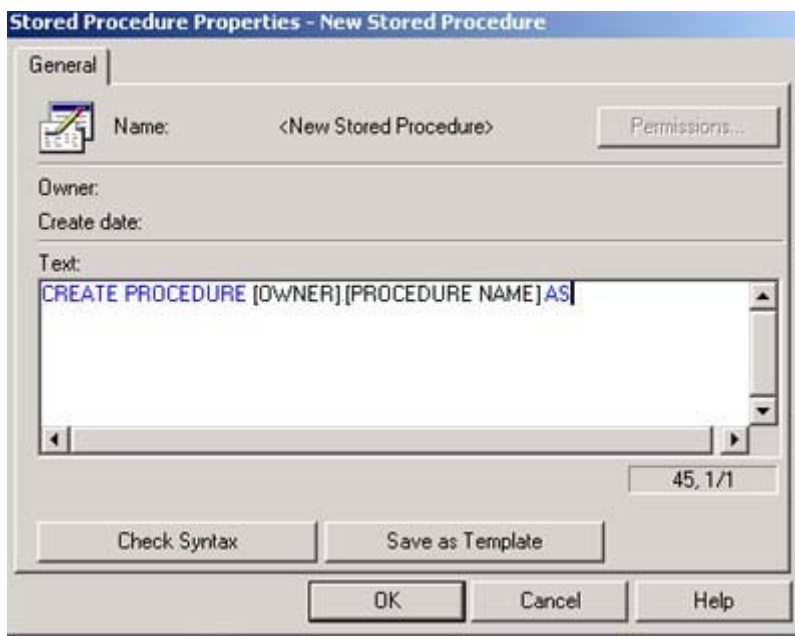
```
GO
```

Ta có thể dùng QA để tạo SP bằng cách gõ Script trên trong cửa sổ Query. Cũng có thể dùng EM:





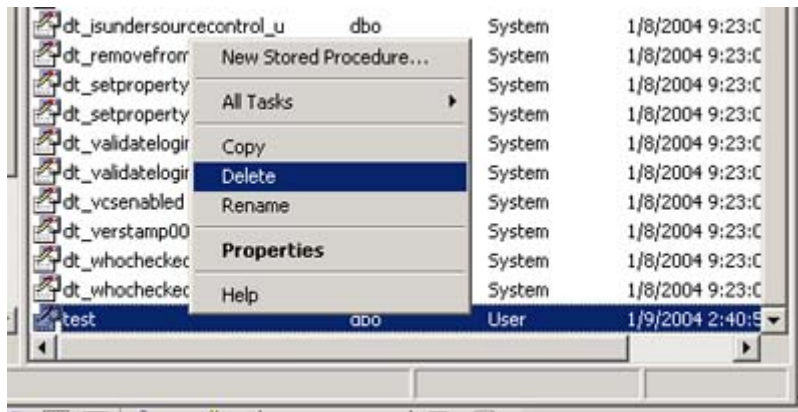
Gõ Script vào cửa sổ Text, chọn Check Syntax để kiểm tra xem SP có đúng cú pháp không. Chọn OK để Save SP.



### *IX.3 XOÁ MỘT SP*

**DROP PROCEDURE Doanhthu**

Với EM, Chọn SP cần xoá và thực hiện thao tác Delete



## IX.4 THAM SỐ TRONG SP

Khi thực hiện một SP, ta có thể truyền tham số để thông báo cho SP nên làm gì trong CSDL. Tham số có thể do người dùng truyền vào hoặc một SP hay ứng dụng khác gửi đến. Ở ví dụ trên SP có 2 tham số là @dauthang smalldatetime, @cuoithang smalldatetime.

Có 2 loại tham số: tham số nhập vào input parameter và tham số kết xuất output parameter.

### IX.4.1 Khai báo tham số:

ta khai báo như trong ví dụ :

```
CREATE PROCEDURE Doanhthu
```

```
@dauthang smalldatetime,
```

```
@cuoithang smalldatetime
```

```
AS
```

...

### IX.4.2 Cung cấp giá trị mặc định:

nếu cần ta có thể khởi tạo giá trị mặc định cho tham số:

```
CREATE PROCEDURE Doanhthu
```

```
@dauthang smalldatetime = '1/1/2004',
```

```
@cuoithang smalldatetime
```

```
AS
```

...

### IX.4.3 Tham số output:

Nếu cần xuất giá trị ra ngoài khi SP thực thi xong, hoặc sử dụng kết quả của SP làm tham số đầu vào cho SP khác thì khai báo thuộc tính Output.

**CREATE PROCEDURE test**

**@X OUTPUT**

**AS**

...

### *IX.5 CÁC LỆNH ĐIỀU KHIỂN*

Cũng giống như các ngôn ngữ lập trình, ta có thể viết các lệnh điều khiển trong SP nói riêng và trong các Script của SQL Server nói chung.

#### *IX.5.1 Lệnh điều khiển IF ... ELSE*

**Cú pháp:**

**IF**

**Begin**

**End**

**ELSE**

**Begin**

**End**

#### *IX.5.2 Lệnh điều khiển CASE*

Cũng giống như Select Case trong Visual Basic hoặc ASP

**Cú pháp:**

**CASE**

**When .....then .....**

**When ..... then .....**

**[ELSE .....]**

**END**

**VD:**

**CASE DatePart(d,NgàyBatdau)**

**When 1 then 'SUN'**

**When 2 then 'MON'**

**When 3 then 'TUE'**

**When 4 then 'WED'**

**When 5 then 'THU'**

**When 6 then 'FRI'**

**When 7 then 'SUN'**

**END**

### *IX.5.3 Lệnh điều khiển WHILE*

**Cú pháp:**

**WHILE**

**Begin**

**End**

### *IX.5.4 Lệnh điều khiển WAITFOR:*

**SQL Server tạm dừng một thời gian trước khi xử lý các lệnh tiếp theo.**

**Cú pháp:**

**WAITFOR**

**DELAY**

**hoặc**

**WAITFOR**

**TIME**

# CHƯƠNG X. HÀM NGƯỜI DÙNG VÀ HÀM HỆ THỐNG

## X.1 KHÁI NIỆM HÀM NGƯỜI DÙNG

SQL Server cho phép người dùng tạo ra các hàm theo những tiêu chuẩn riêng, phù hợp với yêu cầu sử dụng. User Defined Function giống như Stored Procedure. Nó bao gồm các lệnh T-SQL kết hợp tạo nên hàm và có thể được gọi thực thi như là một đơn vị độc lập.

Một số hạn chế của Hàm người dùng so với SP là tham số không được mang thuộc tính output, nghĩa là giá trị tham số không được truyền ra ngoài hàm. Thay vào đó muốn trả về giá trị ta phải dùng return.

## X.2 TẠO VÀ XOÁ HÀM NGƯỜI DÙNG

### X.2.1 Tạo hàm người dùng

```
CREATE FUNCTION [OWNER].[FUNCTION NAME] (PARAMETER LIST)
RETURNS (return_type_spec) AS
BEGIN
(FUNCTION BODY)
END
```

### X.2.2 Xoá hàm người dùng

```
DROP FUNCTION <FUNCTION NAME>
```

## X.3 HÀM HỆ THỐNG

Hàm người dùng được tạo ra trong một CSDL và chỉ có hiệu lực trong CSDL đó. Nếu hàm có tính phổ biến sử dụng chung trong nhiều CSDL thì ta nên tạo nó như là hàm hệ thống.

Khi đã là hàm hệ thống thì ta có thể gọi hàm như các hàm của SQL Server thông thường, ví dụ: hàm GetDate(), DatePart()

### X.3.1 Tạo hàm hệ thống từ hàm người dùng:

Tạo hàm trong CSDL Master (là 1 trong 6 CSDL sẵn có của SQL Server ngay từ lúc cài đặt) thay vì tạo hàm trong CSDL đang dùng (như CSDL Quanlynha)

Dùng tiền tố fn\_TenHam

Thay đổi Owner của hàm (sử dụng SP sp\_changeobjectowner hoặc dùng EM)

## CHƯƠNG XI. TRIGGER

### *XI.1 KHÁI NIỆM:*

Trigger là một dạng SP đặc biệt dùng để đáp ứng một sự kiện cụ thể (Insert, Update, Delete). Nó bao gồm một đoạn Script được gắn vào bảng dữ liệu và tự động thực thi khi có một sự kiện xảy ra tương ứng với Trigger được gắn cho sự kiện đó (trong khi SP chỉ thực thi khi được gọi ra) .

Trigger không có tham số và giá trị trả về.

Chỉ nên sử dụng Trigger trong trường hợp cần thiết, chẳng hạn trong những trường hợp sau:

- Referential Integrity: ràng buộc toàn vẹn dữ liệu cho phù hợp với mô hình quan hệ cơ sở dữ liệu.
- Audit Trails: kiểm soát dữ liệu hiện tại khi có thay đổi đến giá trị trong bản ghi của bảng.
- Check: kiểm tra dữ liệu nhập vào có phù hợp với mối liên hệ giữa các bảng với nhau.
- Kiểm chứng khi xoá bản ghi

### *XI.2 PHÂN LOẠI*

Có thể chia Trigger thành 3 loại theo sự kiện mà nó đáp ứng:

- Insert Trigger.
- Update Trigger.
- Delete Trigger.
- Tổng hợp cả 3.

### *XI.3 TẠO TRIGGER*

#### *XI.3.1 Cú pháp*

Tạo Trigger giống như SP, và nó phải được tạo ra cho bảng cụ thể

Ví dụ: Tạo một Trigger cho bảng Hopdong để kiểm tra ràng buộc dữ liệu với bảng Nha (không thể tạo hợp đồng cho một nhà chưa nhập vào CSDL)

```
CREATE TRIGGER Ins ON [dbo].[Hopdong]
```

```
FOR INSERT
```

```
AS
```

```
if not exists
```

```

( select 'true' from inserted where inserted.nhaid in (select nhaid from nha)
)
begin
raiserror ('Không thể tạo Hợp đồng này vì Nhà chưa được nhập trong table
Nhà',16,1)
rollback Tran
end

```

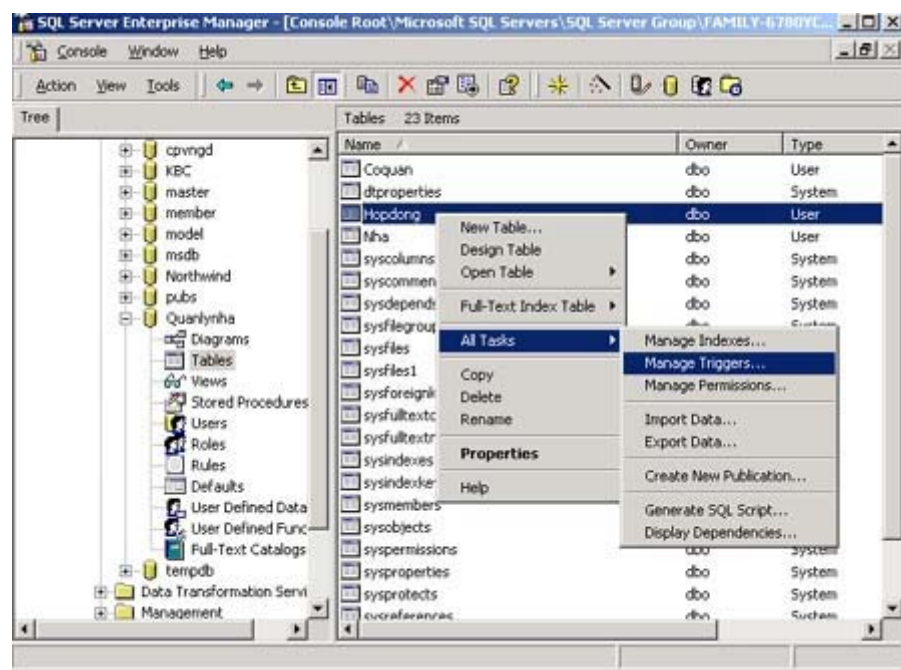
**Diễn giải:**

**On:** chỉ ra Trigger đang viết cho bảng hay khung nhìn nào. Nếu có từ khoá after thì không hỗ trợ view.

**For Insert /For Update / For Delete:** dùng khai báo đối với Insert Trigger/Update Trigger, Delete Trigger.

**AS:** giống như SP, nó chỉ định cho SQL Server biết điểm bắt đầu, sau AS là Script

### *XI.3.2 Tạo Trigger bằng EM.*



Sau khi một bảng đã được gắn Trigger, nó sẽ tự động thực thi với sự kiện tương ứng. Chẳng hạn như ví dụ dưới đây, Trigger Ins được thực thi khi người dùng Insert một bản ghi mới vào bảng Hopdong. Nó đưa ra thông báo lỗi vì Nhà có NhaID =100 chưa tồn tại trong bảng Nha.

SQL Server Enterprise Manager - [4>Data in Table 'Hopdong' in 'Quandynha' on 'FAMILY-67801124']

Console Window Help

HopdongID	NhaID	CoquanID	NgayBatdau	NgayKetthuc	Giatri
3	3	2	1/1/2000	5/1/2005	20000000
?	100	2	1/1/2001	1/1/2006	15000000
*					

SQL Server Enterprise Manager

[Microsoft][ODBC SQL Server Driver][SQL Server] không thể tạo Hợp đồng này vì Nhà chưa được nhập trong table Nhà

OK Help



## **CHƯƠNG XII. CHUYỂN TÁC (TRANSACTION) VÀ KHOÁ (LOCK)**

### *XII.1 KHÁI NIỆM CHUYỂN TÁC*

Xuất phát từ nhu cầu khi giao tiếp với dữ liệu có những trường hợp chúng ta mong muốn hoặc đồng thời nhiều sự kiện diễn ra, hoặc không sự kiện nào. Ví dụ: thao tác chuyển tiền bao gồm 2 sự kiện: rút tiền từ tài khoản A và cộng vào tài khoản B, yêu cầu hoặc cả 2 thao tác đều phải hoàn thành, hoặc không thao tác nào hoàn thành. Đó chính là một Chuyển tác.

Các lệnh:

**Begin:** bắt đầu một chuyển tác; **Commit:** xác nhận chuyển tác đã hoàn thành.

**Rollback:** quay ngược chuyển tác. (Huỷ toàn bộ các sự kiện của chuyển tác)

**Save :** đánh dấu một điểm cho phép Rollback chỉ một phần chuyển tác.

Một ví dụ là chúng ta đã thực hiện chuyển tác ngầm Rollback TRAN trong Trigger Ins ở chương Trigger, khi nhận thấy có lỗi trong ràng buộc dữ liệu thì huỷ toàn bộ quá trình Insert dữ liệu:

```
if not exists
```

```
( select 'true' from inserted where inserted.nhaid in (select nhaid from nha)
```

```
)
```

```
begin
```

```
raiserror ('Không thể tạo Hợp đồng này vì Nhà chưa được nhập trong table Nhà',16,1)
```

```
rollback Tran
```

```
end
```

### *XII.2 KHOÁ (LOCK) VÀ ĐỒNG HÀNH (CONCURRENCY)*

Khi có nhiều chuyển tác cùng thực hiện trên một đối tượng, người ta gọi là đồng hành. Để hiểu hơn, đồng hành là hiện tượng tại một thời điểm có nhiều người cùng kết nối và truy cập một đối tượng của CSDL. Trong nhiều trường hợp điều này sẽ xảy ra xung đột, ví dụ nhiều người đồng thời cố gắng sửa dữ liệu trong một bảng.

Để kiểm soát người dùng thực hiện những chuyển tác đồng hành, ngăn ngừa xung đột, đụng độ, người ta dùng khoá. Có nghĩa là một người không thể làm gì để sửa đổi một đối tượng khi có một người khác đang truy cập nó trước người

**đó. Shared lock: Thực tế có nhiều kiểu khoá. Khoá căn bản nhất là khoá chia sẻ (Shared lock). Nó cho phép đọc dữ liệu và không cho phép thay đổi bất kỳ thuộc tính nào của dữ liệu.**

## **CHƯƠNG XIII. SỬ DỤNG NGÔN NGỮ LẬP TRÌNH TƯƠNG TÁC VỚI SQL SERVER**

SQL Server thuần túy chứa CSDL. Để cho phép người sử dụng khai thác được CSDL SQL Server, cần xây dựng những ứng dụng được phát triển bằng ngôn ngữ lập trình.

### *XIII.1 LẬP TRÌNH BẰNG ASP*

Sử dụng ASP để thao tác với CSDL nói chung và SQL Server nói riêng được đề cập kỹ trong giáo trình môn “Lập trình Web bằng ASP”. Ở đây chúng ta đề cập đến một số khía cạnh thực hành.

#### *XIII.1.1 Kết nối đến CSDL SQL Server:*

Dùng đối tượng Connection

‘tạo một Connection

```
Set Conn = Server.CreateObject("ADODB.Connection")
```

‘ Khởi tạo chuỗi kết nối

```
strConn = "driver={SQL Server}; server=127.0.0.1; uid=iitm; pwd=test;  
database=quanlynha"
```

‘ Kết nối đến CSDL

```
Conn.Open strConn
```

#### *XIII.1.2 Thực thi câu lệnh SQL:*

```
conn.execute "update hopdong set giatien=1000000 where hopdongid=1"
```

#### *XIII.1.3 Thao tác với các bản ghi của một bảng:*

Dùng đối tượng Recordset

```
<%
```

‘ câu lệnh SQL

```
sqlString= "select * from Hopdong"
```

‘ Tạo đối tượng Recordset

```
Set RS = Server.CreateObject("ADODB.Recordset")
```

‘ thực thi câu lệnh SQL thông qua đối tượng kết nối Conn, kết quả trả về cho đối tượng Recordset.

```
Rs.Open sqlString,Conn
```

‘ duyệt từng bản ghi trong bảng Hopdong bằng Recordset, hiện thi dạng bảng trên ‘giao diện Web.

```
<table>
```

```
<%
```

‘ vòng lặp do while để duyệt tất cả các bản ghi

```
do while not Rs.EOF
```

```
%>
```

```
<tr>
```

```
<td><%=Rs("Hopdongid")%></td>
```

```
<td><%=Rs("NhaID")%></td>
```

```
<td><%=Rs("CoquanID")%></td>
```

```
<td><%=Rs("NgayBatdau")%></td>
```

```
<td><%=Rs("NgayKetthuc")%></td>
```

```
<td><%=Rs("Giatien")%></td>
```

```
</tr>
```

```
<%
```

‘ dịch chuyển đến bản ghi kế tiếp

```
RS.MoveNext
```

**Loop**

```
RS.close
```

```
%>
```

```
</table>
```

*XIII.1.4 Cập nhật vào CSDL bằng Recordset:*

```
sqlString ="select * from hopdong"
```

```
Set RS1 = Server.CreateObject("ADODB.Recordset")
```

```
'de co the update
```

```
adOpenDynamic=2
```

```
adLockPessimistic=2
```

```
Rs1.Open sqlString,Conn,adOpenDynamic,adLockPessimistic
```

```
do while not Rs1.EOF
```

‘ lấy giá tiền thuê nhà từ trường Giatien

```
tempGiatien=rs1("giatien")
```

‘ tăng giá tiền thuê nhà lên gấp đôi rồi cập nhật vào trường Giatien

**rs1("giatien")=tempGiatien\*2**

**‘ xác nhận việc cập nhật**

**rs1.Update**

**rs1.MoveNext**

**loop**

**Rs1.Close**

### *XIII.1.5 Thực thi một SP của SQL Server*

Trong ví dụ ở chương Stored Procedure ta đã tạo một SP tên “Doanhthu” trong CSDL Quanlynha

**CREATE PROCEDURE Doanhthu**

**@dauthang smalldatetime,**

**@cuoithang smalldatetime**

**AS**

**...**

**Gọi thực hiện SP đó từ ASP:**

**‘ gán giá trị cho các biến**

**dauthang=”1/1/2004”      ‘month/date/year**

**cuoithang=”1/31/2004”**

**‘câu lệnh sql gọi SP có tham số**

**sql=”execute doanhthu ” + dauthang + “,” + cuoithang**

**Set RS = Server.CreateObject("ADODB.Recordset")**

**RS.Open sql,Conn**

### *XIII.2 LẬP TRÌNH BẰNG VISUAL BASIC*

Sử dụng Visual Basic để thao tác với CSDL nói chung và SQL Server nói riêng được đề cập kỹ trong giáo trình môn “Lập trình VISUAL BASIC”. Về căn bản cũng tương đối giống ASP.

#### *XIII.2.1 Kết nối đến CSDL:*

**‘Khai báo biến Connection**

**Dim Conn as New ADO**

**‘ chuỗi kết nối CSDL**

**Dim StrConn as String**

**StrConn="Provider=SQLOLEDB.1;Persist Security = False; uid=iitm;  
pwd=test; Initial Catalog=Quanlynha;server=127.0.0.1 ;"**

**‘ Tạo kết nối:**

**Set Conn=CreateObject (“ADODB.Connection”)**

**Conn.Open StrConn**

*XIII.2.2 Thực hiện câu lệnh SQL:*

**Conn.Execute “Delete from hopdong where hopdongid=2”**

*XIII.2.3 Thực thi SP*

**Conn.Execute “Doanhthu ‘&”1/1/2004”&” ‘, ’&”1/31/2004”&”””**

*XIII.2.4 Thao tác với bản ghi*

**‘ Khai báo biến Recordset**

**Dim RS**

**‘ Tạo đối tượng Recordset**

**Set RS=CreateObject (“ADODB.Recordset”)**

**‘ Thực thi câu lệnh SQL cho kết quả trả về Recordset:**

**Rs.open “Select \* from Hopdong”, Conn, 2, 2**

**‘ Insert dữ liệu**

**RS.Addnew**

**RS(“NhaID”)="2”**

**RS(“NhaCoquanID”)="3”**

**RS(“NgàyBatdau”)="1/1/2004”**

**RS(“NgàyKetthuc”)="1/1/2006”**

**RS(“Giatien”)="5000000”**

**RS.Update**

**‘ Đóng kết nối**

**RS.Close**

**Conn.Close**

**Set RS=nothing**

**Set Conn= nothing**



## **CHƯƠNG XIV.      MỘT SỐ VẤN ĐỀ NÂNG CAO.**

*XIV.1          SECURITY*

*XIV.2          CHUYỂN TÁC VÀ TRUY VẤN PHÂN TÁN*

*XIV.3          XML*

*XIV.4          TÌM KIẾM FULL TEXT SEARCH*

*XIV.5          DỊCH VỤ PHÂN TÍCH ANALYSIS SERVICES*

*XIV.6          DATABASE REPLICATE*

*XIV.7          ENGLISH QUERY*