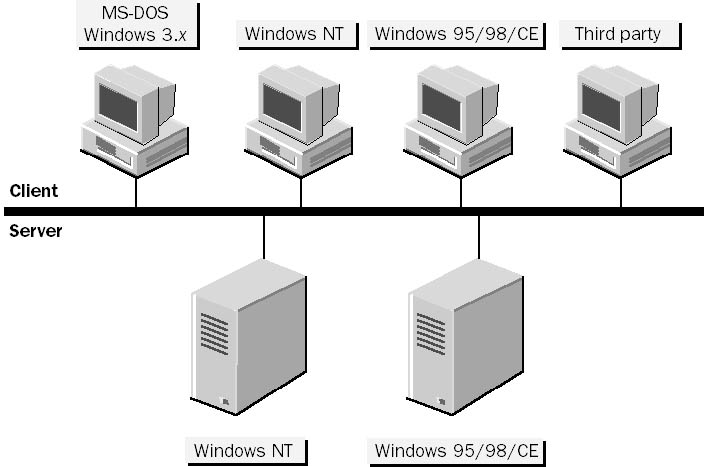
CHƯƠNG 1. TỔNG QUAN VỀ SQL SERVER

Mục đích: Biết được các thành phần thiết yếu của SQL SERVER

**I. KIẾN TRÚC MẠNG CỦA SQL SERVER**

Kiến trúc của SQL SERVER phân chia các ứng dụng truy xuất cơ sở dữ liệu qua bộ điều khiển cơ sở dữ liệu (database engine). SQL SERVER chạy trên hệ điều hành NT cho phép kết nối đến nhiều hệ thống client qua mạng LAN hay Ethernet. Hệ thống client thông thường là các PCs chạy trên phần mềm client của SQL SERVER. SQL SERVER hỗ trợ cho các client trên các hệ điều hành sau:



*Hình 1.1 : Hệ điều hành mà Client và Server của SQL Server có thể hoạt động*

- Windows 9x

- Windows NT, XP, 2Kx

- Macintos

- DOS

- OS/2

- UNIX

Bảng sau đây cho ta biết chi tiết về các hệ điều hành; Hệ điều hành nào hỗ trợ (hoặc không hỗ trợ) SQL Server

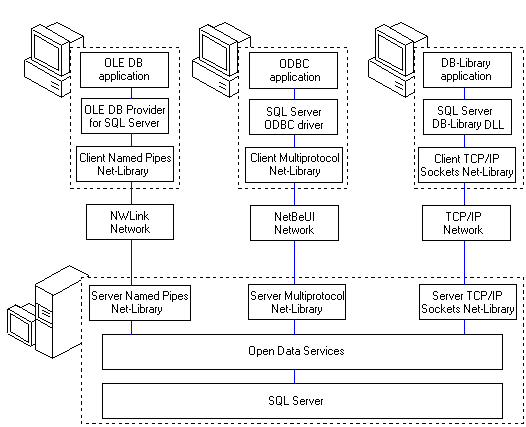
|  |  |  |
| --- | --- | --- |
| **Hệ điều hành** | **Server software** | **Client software** |
| Microsoft Windows 9x | Có; hoạt động như 1 application | Có |
| Microsoft Windows NT Workstation 4.0 or later | Có; hoạt động như 1 service | Có |
| Windows NT Server | Có; hoạt động như service | Có |
| Windows NT Server Enterprise Edition | Có; hoạt động như service | Có |
| Windows 3.*x* | No | Có (chỉ dùng cho SQL Server 6.0 và 6.5) |
| MS-DOS | No | Có (chỉ dùng cho SQL Server  versions 6.0 và 6.5) |
| Third party | No | Có, chẳng hạn như UNIX và Apple Macintosh |

Bộ điều khiển cơ sở dữ liệu SQL SERVER chạy trên WINDOWS NT hay WINDOWS 9x. Các user truy xuất cơ sở dữ liệu của SQL SERVER thông qua hệ thống client của nó. Nói cách khác, các client chạy trên các hệ thống client mạng của nó, trong khi đó, thành phần database server chỉ chạy trên hệ thống Server của SQL SERVER.

SQL SERVER sử dụng các mạng phổ biến như là Ethernet và Token Ring. SQL SERVER cũng sử dụng các giao thức phổ biến: TCP/IP, Named Pipe, IPX/SPX, Apple’s AppleTalk.

Một trong những thuận lợi chính của SQL SERVER là nó có thể hợp nhất với các công cụ phát triển client/server và các ứng dụng như MicroSoft Word, Excel, và Access. Cơ sở dữ liệu của SQL SERVER cũng có thể được truy xuất qua các ứng dụng : Visual Basic , Visual Foxpro, Visual C++, Delphi, PowerBuilder…

Cơ sở dữ liệu của SQL SERVER có thể được truy xuất với bộ điều khiển MicroSoft Jet Engine, Data Access Objects (DAO), Remote Data Objects (RDO), Activex Data Objects (ADO), ODBC, thư viện có sẵn của SQL SERVER, ….



SQL Server Computer

Client Computer

*Hình 1.2- Kiến trúc clent/server của SQL SERVER.*

Hình 1.2 minh họa chi tiết kiến trúc client/server của SQL SERVER. Trong phần trên của hình, ta thấy các ứng dụng client khác nhau sử dụng các giao tiếp lập trình ứng dụng để truy xuất dữ liệu mà SQL SERVER cung cấp (API- Application Programming Interface). SQL SERVER có 3 API truy xuất dữ liệu chính : OLE DB, ODBC, và DB Library. Đối với các client trên Windows, tất cả các API này được cài đặt như thư viện kết nối động ( dynamic link library – DLL) và chúng liên lạc với SQL SERVER thông qua thư viện mạng client. Thư viện mạng client sử dụng 1 phương thức liên lạc bên trong mạng (IPC – interprocess communication) để giao tiếp với thư viện mạng của server (server network library).

Thư viện mạng của server nhận gói dữ liệu gởi từ client và trao chúng cho các *dịch vụ mở dữ liệu*  ( Open Data Services – ODS ; bao gồm tập hợp các macro và hàm trong C++). Bản thân SQL SERVER là 1 ứng dụng ODS, nó chấp nhận lời gọi ODS, xử lý chúng, và trả kết quả về cho ODS.

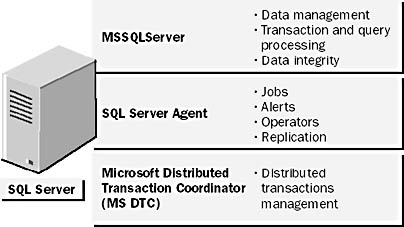
**Việc lin lạc giữa SQL server và các các client được thực hiện theo thứ tự như sau:**

1. Ứng dụng client gọi OLE DB, ODBC, thư viện DB, hay API chứa SQL. Hành động này khởi tạo: bộ cung cấp OLE DB (OLE DB provider ), ODBC driver, hay DB-Library DLL, để sử dụng cho các truyền thông của SQL server .
2. Bộ phận cung cấp OLE DB, ODBC driver hay DB-Library DLL gọi một thư viện mạng client .Và sau đó thư viện mạng client gọi tới một IPC API.
3. Lời gọi của Client đến IPC API được truyền đến một thư viện mạng server bằng một IPC nằm bên dưới. Nếu nó là một IPC cục bộ ,thì lời gọi được truyền đi bằng cách sử dụng Windows operating IPC như: bộ nhớ chia sẻ hay local named pipes. Nếu nó là một IPC mạng, chồng giao thức mạng trên client sẽ sử dụng mạng để lin lạc với chồng giao thức mạng trn server.
4. Thư viện mạng server chuyển những yêu cầu của client đến CSDL SQL server.

Qu trình trả lời từ SQL server cho client sẽ theo chuỗi cc hnh động ngược lại như trên .

**II. CÁC THÀNH PHẦN CƠ BẢN TRONG SQL SERVER:**

SQL SERVER có 4 thành phần cơ bản: (Hình 1.3)



***Hình 1.3*** *Các dịch vụ của SQL Server*

**1.  *Dịch vụ tìm kiếm (Full text Search Service):***

Dịch vụ tìm kiếm Microsoft l một bộ my tìm kiếm v chỉ mục tồn bằng văn bản (full-text).

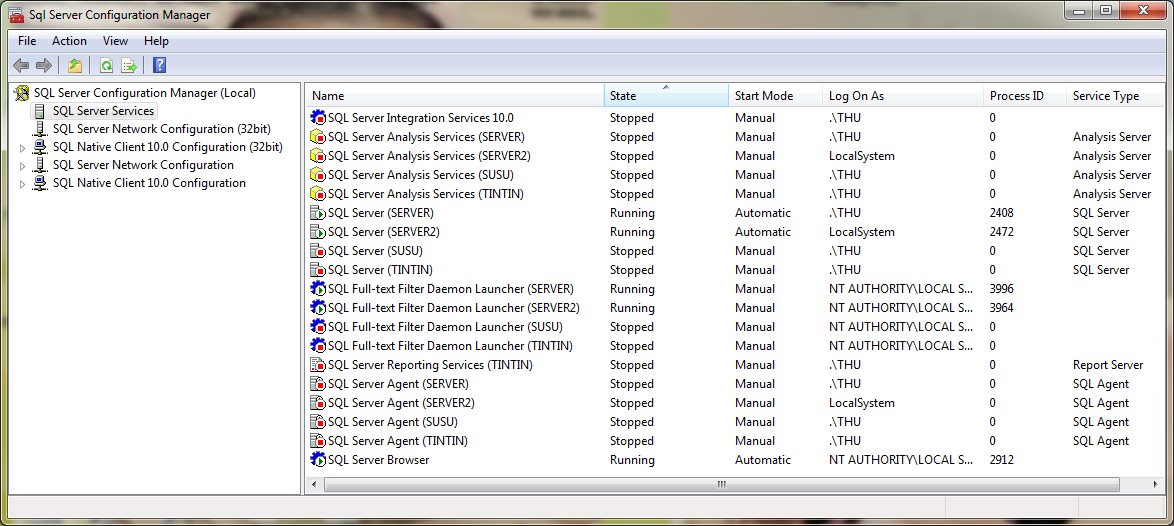
Sử dụng Full-text Search bạn có thể tăng thêm chỉ mục trong một chuỗi, đặc biệt là giữ được từ trong chuỗi tìm kiếm bạn cần ,mặt khc Full-text Search cũng khơng giới hạn chiều di v dạng chuỗi tìm kiếm. Kỹ thuật ny cĩ khả năng tìm kiếm một ký tự , một từ, hay thậm chí cả một chuỗi.

***2. Dịch vụ SQL Server* (SQL Server Service):** quản lý tất cả các file cơ sở dữ liệu. Nó có nhiệm vụ thi hành tất cả các phát biểu SQL và cấp phát tài nguyên hệ thống.

Bộ xử lý cơ sở dữ liệu SQL Server hoạt động như một dịch vụ trên Windows NT và Windows 2000. Nhưng nó không hoạt động như một dịch vụ trên Windows 98 vì hệ thống ny khơng hỗ trợ cc dịch vụ. SQL server cũng cĩ thể hoạt động như một file thực thi trên Windows NT và Windows 2000 .

Khi nhiều Server của SQL server chạy trn cng một my tính thì mỗi Server cĩ dịch vụ SQL Server của ring nĩ.

Lưu ý: Muốn Start hay Stop một dịch vụ trong SQL Server, ta chọn Start/Program/MicroSoft SQL Server / Configuration Tools/ SQL Server Configuration Manager .



Dịch vụ SQL server quản lý tất cả cc file trong cơ sở dữ liệu của các Server SQL server . Đây là thành phần xử lý tất cả các câu lệnh giao tác được gởi đến từ các ứng dụng client và server. SQL server cũng hỗ trợ cho các truy vấn phân tán lấy dữ liệu từ nhiều nguồn ti nguyn khc ngồi SQL server.

Dịch vụ SQL Server chỉ ra vị trí các tài nguyên cho nhiều người dùng.Nó cũng áp đặt các luật hoạt động (business rules) được định nghĩa trong các thủ tục lưu trữ và các trigger, để bảo đảm tính nhất quán của dữ liệu ,và ngăn chặn các vấn đề luận lý xảy ra như có hai người cùng cố gắng để cập nhật cùng một dữ liệu tại một thời điểm .

***3. Dịch vụ SQL Server Agent:***

SQL server Agent hỗ trợ các đặc tính cho phép lên kế hoạch sẵn cho các hoạt động theo chu kỳ trên SQL server, và khai báo đến người quản trị hệ thống các vấn đề xảy ra với server . Dịch vụ ny cũng cần thiết trong lệnh nhn bản database (Replication). Các thành phần SQL server thực hiện chức năng này là :

* **Jobs**:

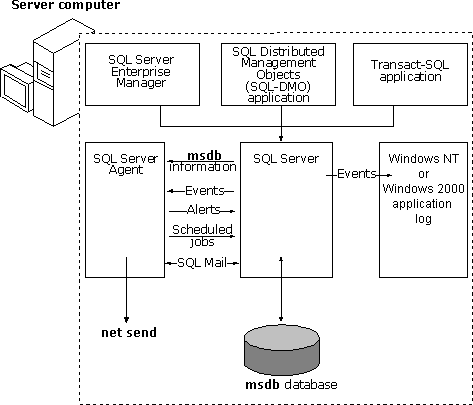
Định nghĩa các đối tượng mà đối tượng này bao gồm một hay nhiều bước thực thi. Các bước là các câu lệnh giao tác SQL sẽ được thực thi .Các công việc có thể được lên kế hoạch để thực thi tại một thời điểm chỉ định trước hay thực hiện theo các khoảng thời gian chu kỳ.

* **Alerts:**

Các cảnh báo được đưa ra khi các sự kiện xảy ra,như khi lỗi xảy ra, hay khi một cơ sở dữ liệu đạt tới một giới hạn vì bộ nhớ trống cĩ sẵn khơng cịn đủ nữa. Cảnh báo có thể được xác định để đưa ra các hành động giải quyết như gởi một email hay thực hiện một công việc nào đó để giải quyết vấn đề xảy ra.

* Người điều hành(**Operators**):

Được nhận diện thông qua tài khoản mạng hay qua ID email. Đây có thể là đích đến của các cảnh báo bằng email hay câu lệnh được gởi qua mạng.



Hình 1.5: Minh hoạ hoạt động của các thành phần

***4. MSDTC*** (MicroSoft Distributed Transaction Coordinator- Điều phối các giao tác phân tán): quản lý các giao tác, có trách nhiệm điều phối các giao tác của cơ sở dữ liệu trên nhiều server.

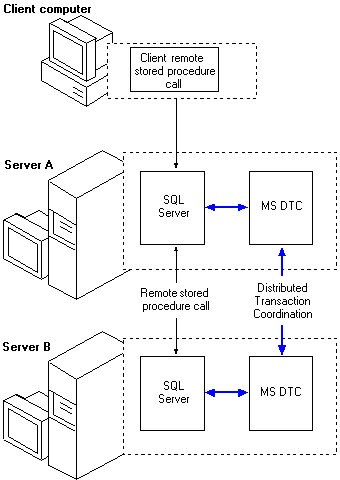
Bộ theo di hoạt động của giao tác phân tán Microsoft (The Microsoft Distributed Transaction Coordinator (MS DTC)) l một bộ quản lý giao tc cho php cc ứng dụng client tập hợp cc ti nguyn dữ liệu phn tn cho một giao tc. Cc MS DTC theo di việc chuyển cc giao tc phn tn đến nơi an toàn xuyên qua các server tham gia trong giao tc .

Một Microsoft SQL Server cĩ thể tham gia vo một giao tc phn tn bằng cch :

* Gọi các thủ tục lưu trữ (Stored Procedure) trong các server từ xa đang chạy SQL server.
* Nng cấp giao tc cục bộ thnh giao tc phn tn v tuyển cc server từ xa vo giao tc.
* Tạo các cập nhật phân tán để cập nhật dữ liệu trong nhiều tài nguyên dữ liệu OLE DB.

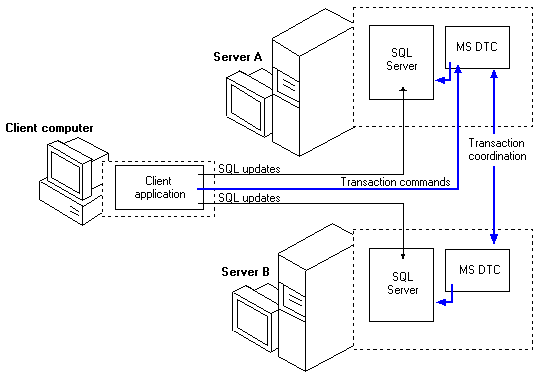
Nếu những ti nguyn dữ liệu OLE DB ny hỗ trợ giao diện giao tc phn tn OLE DB, thì SQL Server cĩ thể cũng tuyển chng vo giao tc phn tn.

Cc MS DTC theo di qu trình hồn thnh hợp lệ của cc giao tc để đảm bảo rằng các quá trình cập nhật trn tất cả cc server đều đồng nhất , hoặc tất cả đều bị xoá đi trong các trường hợp lỗi .



Hình 1.6.

Cc ứng dụng SQL server cũng cĩ thể gọi trực tiếp MS DTC để bắt đầu một giao tác ngay lập tức. Một hay nhiều các server đang chạy SQL server có thể được chỉ dẫn để tuyển vào một giao tác phân tán và hoàn tất theo đúng cách của giao tác MS DTC.



Hình 1.7

1. **Nhiều Server của Server SQL trn cng một my tính (Multiple Instances of SQL Server)**

**1. Khi niệm:**

SQL Server hỗ trợ nhiều Server của SQL Server chạy đồng thời trên cùng một máy tính. Mỗi Server của SQL server có cơ sở dữ liệu hệ thống của ring nĩ v cc cơ sở dữ liệu người dùng không được chia sẻ giữa các Server .Các ứng dụng có thể kết nối đến các Server trên một máy tính theo nhiều cách giống như dùng để kết nối các SQL Server đang chạy trên các máy tính khác nhau .

Cĩ hai loại Server SQL Server:

**Server mặc định**

Server mặc định của SQL Server hoạt động giống như các bộ xử lý cơ sở dữ liệu trong cc phin bản trước đó của SQL Server. Server mặc định được nhận diện bằng tn của my tính, khơng cĩ một tn ring biệt cho Server loại ny. Khi cc ứng dụng chỉ định tên máy tính để kết nối đến SQL server, thì cc thnh phần SQL Server client sẽ kết nối đến Server mặc định trên máy tính đó.

Chỉ cĩ thể cĩ một Server mặc định trn bất kỳ một my tính, Server mặc định cĩ thể l bất kỳ phin bản no của SQL Server.

**Các Server đặt tên**

Tất cả cc Server khc của bộ xử lý cơ sở dữ liệu ngồi Server mặc định được nhận diện bằng tn của chng.Cc ứng dụng phải cung cấp cả tn my tính v tn của Server đặt tn m chng muốn kết nối đến. Tên máy tính và tên Server được chỉ định có định dạng l *computer\_name*\*instance\_name*.

Cĩ thể cĩ nhiều Server đặt tn chạy trn một máy tính , nhưng chỉ cĩ SQL Server 2000 trở ln mới cĩ thể thực thi cc Server ny.

Khi bạn cài đặt nhiều Server , mỗi Server có một tập duy nhất:

* Hệ thống và CSDL người dùng
* Cc dịch vụ SQL Server và SQL Server Agent. Đối với các Server mặc định , tên của các dịch vụ vẫn là **MSSQLServer** v **MSSQLServerAgent**. Đối với các Server đặt tên , tên của các dịch vụ được đổi thành **MSSQL$*instancename***v **SQLAgent$*instancename***, chúng được bắt đầu và kết thúc độc lập với cc Server khc trn my tính. Cc bộ xử lý CSDL cho cc Server khc nhau được bắt đầu và kết thúc bằng cách sử dụng dịch vụ liên tác SQL Server. Các dịch vụ SQL Server Agent quản lý cc sự kiện xảy ra cho cc Server lin tc của bộ xử lý CSDL .

***Cc thnh phần được chia sẻ(Shared Components)***

Các thành phần sau đây được chia sẻ giữa tất cả các Server đang chạy trên cùng một máy tính:

* Tất cả cc tiện ích của SQL Server: Ta cĩ thể sử dụng một bản sao SQL Server Management Studio để điều khiển các đối tượng trong tất cả cc Server trn my tính. v sử dụng một bản sao của SQL Server Network Configuration để quản lý các địa chỉ mạng của tất cả các Server .
* Một bản sao duy nhất cc cu truy vấn bằng tiếng Anh v cc dịch vụ server Microsoft Analysis .

**2. Truyền thơng với nhiều Server(Communicating with Multiple Instances)**

***a. Sử dụng nhiều Server(Using Multiple Instances):***

Việc chạy nhiều Server trên cùng một máy tính mở rộng các khả năng của SQL Server .

Việc sử dụng một Server của SQL Server trn một server production đưa ra những lợi ích:

* Chỉ cần quản trị một Server .
* Khơng cần nhn bản hay xử lý overhead .

Chạy nhiều Server của SQL Server trn cng một my tính thì tốt nhất khi :

* Các hệ thống khác nhau phải được bảo đảm cô lập khỏi những hệ thống khác .
* Cần hỗ trợ nhiều CSDL kiểm tra v mở rộng .
* Cần chạy nhiều ứng dụng trên một desktop, và mỗi ứng dụng cần cài đặt một Server tách biệt của SQL Server 2000 Desktop Engine.

***b. Lm việc với nhiều Server (Working with Multiple Instances)***

Các Server đặt tên hoạt động gần giống như các Server mặc định.Chỉ khác là phải có cả tên máy tính và tên của Server để nhận diện một Server đặt tên. Nếu chỉ định tên máy tính,thì ta sẽ lm việc với Server mặc định. Nếu chỉ định cả *computername*\*instancename* thì ta sẽ lm việc với Server đặt tên.

* **Service Manager**

Khi một Server chỉ định được khởi động , bất kỳ CSDL nào được tạo ra trong Server đó sẽ sẵn sng cho bất kỳ một ứng dụng no kết nối đến nĩ bằng cch sử dụng một ID xc thực cho php truy cập vo CSDL.

* **SQL Server Management Studio.**

Bằng cch sử dụng SQL Server Management Studio bạn cĩ thể đăng ký mỗi Server với cc quyền hạn cho php. Sau khi một Server được đăng ký, bạn cĩ thể tạo, chỉnh sửa hay xố bỏ các đối tượng trong cc CSDL quan hệ với Server đó .Bạn cũng cĩ thể tạo ,chỉnh sửa ,hay xố bỏ cc dịch vụ về thay đổi dữ liệu ,về nhn bản và các đối tượng SQL Server Agent của Server đó.

* **Cc ứng dụng.**

Trong một ứng dụng khi bạn chỉ định *computername* l thơng số tn server trong một yu cầu kết nối ,thì bạn sẽ được kết nối đến server mặc định trên máy tính đó.

Bạn cĩ thể truy cập bất kỳ CSDL no trong Server mặc định m bạn cĩ quyền truy cập. Khi bạn được kết nối đến một Server chỉ định ,các đối tượng trong cc CSDL trong Server khc cĩ thể chỉ được truy cập thơng qua cc truy vấn phn tn .

Các ứng dụng SQL DMO có thể quản lý các Server của SQL Server bằng cách sử dụng đối tượng **SQLServer2** .

Các truy vấn phân tán và các server liên kết: Các chỉ định của các truy vấn phân tán và server liên kết sử dụng *computername*\*instancename* để nhận diện các Server đặt tên và sử dụng *computername* để nhận diện các Server mặc định.

***Nhận diện cc Server (Identifying Instances)***

Các bộ đếm giám sát hoạt động, các Profiler event , và Windows events trong Log Event Viewer nhận diện Server của SQL Server mà chúng kết nối đến .

Chức năng @@SERVERNAME nhận diện tên của Server ở dạng *servername*\*instancename* khi kết nối đến một Server đặt tên .Nếu kết nối đến một Server mặc định @@SERVERNAME chỉ nhận diện *servername*.

Mặc d cc chuỗi trả về bởi @@SERVERNAME và SERVERNAME sử dụng cùng dạng ,nhưng thông tin chúng báo cáo có thể là khác nhau :

* Chuỗi báo cáo bởi @@SERVERNAME bị ảnh hưởng bởi các hành động của **sp\_addserver** v **sp\_dropserver** ,nhưng chuỗi được báo cáo bởi SERVERNAME thì khơng.
* SERVERNAME tự động báo cáo các thay đổi về tên mạng của máy tính ,và @@SERVERNAME thì khơng nếu **sp**\_**dropserver** v **sp**\_**addserver** không được sử dụng để thay đổi tên mà nó báo cáo.
* Lưu ý: ta cĩ thể dng ***select serverproperty('servername') để*** in ra tn server.

**IV. TRANSACT-SQL ( T-SQL)**

SQL (Structured Query Language) là 1 ngôn ngữ cho phép truy xuất dữ liệu trong 1 cơ sở dữ liệu quan hệ. Trong SQL SERVER, SQL được gọi là Transact-SQL; Transact SQL cũng tuân thủ các cú pháp của SQL chuẩn; Ngoài ra, nó còn cung cấp 1 số option mở rộng giúp ta truy vấn dữ liệu dễ dàng.

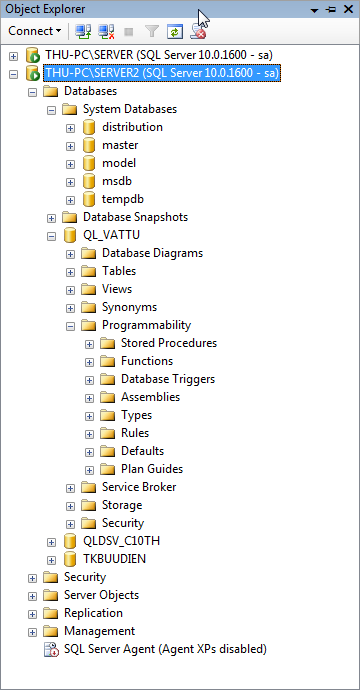
Transact-SQL thường được dùng trong các công việc để quản trị cơ sở dữ liệu như tạo bảng, tạo field, xóa field, xóa bảng..viết các thủ tục . Nó còn cho phép thay đổi cấu hình của SQL SERVER. Transact-SQL có 3 loại:

- Data Definition Language (DDL): create database, create table, create view ….

- Data Manipulation Language (DML): select, update, delete, insert into

- Data Control Language (DCL): dùng để điều khiển cho phép truy xuất dến các đối tượng cơ sở dữ liệu qua các lệnh GRANT và REVOKE.

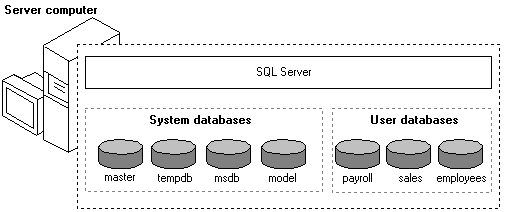
**V. KIẾN TRÚC CƠ SỞ DỮ LIỆU TRONG SQL SERVER:**



*Hình 1.8: Kiến trúc cơ sở dữ liệu trong Sql Server*

1. **Server**: Nhân của cơ sở dữ liệu SQL SERVER là server, hay bộ điều khiển cơ sở dữ liệu. Bộ điều khiển cơ sở dữ liệu có nhiệm vụ xử lý các yêu cầu về cơ sở dữ liệu, và trả kết quả sau khi xử lý cho hệ thống client.

2. **Cơ sở dữ liệu** : Mỗi SQL SERVER chứa nhiều cơ sở dữ liệu, trong đó, mỗi cơ sở dữ liệu được duy trì trong 1 hoặc nhiều file . Ngầm định, tiến trình cài đặt SQL SERVER tạo 4 cơ sở dữ liệu hệ thống: master, model, msdb, và tempdb. Mỗi cơ sở dữ liệu sẽ có 1 file nhật ký (log file) tương ứng để chứa các giao tác trên cơ sở dữ liệu. Ta có 2 loại cơ sở dữ liệu : SystemDatabase và User database



* **master:** cơ sở dữ liệu ghi tất cả những thông tin mức hệ thống của hệ thống SQL Server. Nó ghi tất cả login accounts và tất cả những lựa chọn cấu hình hệ thống. **master** là một cơ sở dữ liệu ghi sự tồn tại của tất cả những cơ sở dữ liệu khác, bao gồm vị trí của file cơ sở dữ liệu, thông tin khởi tạo cho SQL Server, luôn có sự sao lưu cho **master**.

Ví dụ, một số Table của **master** chứa cc thơng tin hệ thống như:

|  |  |
| --- | --- |
| **Tn Table** | **Miu tả** |
| sysconfigures | Mỗi hng chứa tuỳ chọn cấu hình được thiết lập bởi một user |
| sysdatabases | Mỗi hàng chứa thông tin một cơ sở dữ liệu của MS SQL Server. Khi vừa được cài đặt, sysdatabase được khởi tạo với thông tin của các cơ sở dữ liệu **master, model, msdb, v** **tempdb** |
| syslogins | Mỗi hng chứa một login account |
| sysprocesses | Chứa thơng tin về những tiến trình được thực thi trên Server |
| sysservers | Mỗi hàng chứa một server mà SQL Server có thể truy cập tới như một nguồn dữ liệu OLE DB. |

Tip: Không được thay đổi master database, và luôn luôn tạo bản backup của master database.

⮚ **Model Database**: là 1 template mà SQL SERVER dùng để tạo cơ sở dữ liệu mới. Bởi vì tempdb được tạo mỗi lần SQL SERVER khởi tạo, model database phải luôn luôn hiện hữu. Khi 1 cơ sở dữ liệu của user được tạo, SQL SERVER sẽ tạo 1 bản copy của model database. Model database chứa các bảng hệ thống được dùng trong mỗi database.

Các table hệ thống theo dõi các option của cơ sở dữ liệu (default setting, quyền hạn của user (user authority, và các ràng buộc trong cơ sở dữ liệu) . Thay đổi model database sẽ ảnh hưởng tới tất cả các cơ sở dữ liệu được tạo mới.

Tip: Ta có thể trao các giá trị ngầm định, ràng buộc toàn vẹn, và quyền truy xuất của user cho các cơ sở dữ liệu mới bằng cách thay đổi model database.

* **msdb:** được dùng bởi SQL Server Agent để lập lịch cho alert và job, và ghi những toán tử. Ví dụ như Table **backupset** chứa thông tin các tập sao lưu (backup set) với các trường như:
* **name:** tên tập sao lưu
* **description:** những mô tả tập sao lưu
* **user\_name:** tn của user thực hiện qu trình sao lưu
* **database\_creation\_date:** ngày & giờ tạo cơ sở dữ liệu gốc
* **backup\_start\_date:** ngy & giờ tiến trình sao lưu bắt đầu
* **backup\_finish\_date**: ngy & giờ tiến trình sao lưu hoàn tất
* **type:** kiểu sao lưu

⮚ **Tempdb database**: để SQL SERVER lưu trữ các table tạm. Nó được tạo lại mỗi lần SQL SERVER khởi động. Tempdb là 1 tài nguyên công cộng, do đó, tất cả các user đều có quyền truy xuất tới nó. Tất cả các table được tạo trong tempdb đều được tự động xóa khi user thoát khỏi SQL SERVER.

Mỗi cơ sở dữ liệu trong SQL Server chứa những Table hệ thống ghi dữ liệu cần thiết bởi những thành phần SQL Server. Sự điều hành thành công của SQL Server tuỳ thuộc vào sự toàn vẹn thông tin trong những Table hệ thống; vì thế, Microsoft khơng hỗ trợ user cập nhật trực tiếp những thơng tin trong Table hệ thống.

Microsoft cung cấp một tập hoàn chỉnh những công cụ quản trị mà cho phép user quản trị đầy đủ hệ thống của họ và quản lý tất cả user v những đối tượng trong cơ sở dữ liệu. User có thể sử dụng những tiện ích quản trị như SQL Server Enterprise Manager, quản lý trực tiếp hệ thống. Những lập trình vin cĩ thể sử dụng SQL\_DMO API để bao gồm những chức năng hoàn chỉnh cho việc quản trị SQL Server trong ứng dụng của họ. Những lập trình vin xy dựng những đoạn m Transact-SQL v SP cĩ thể sử dụng SP hệ thống v những pht biểu Transact-SQL DDL để hỗ trợ tất cả những hàm quản trị trong hệ thống của họ.

3. **Các đối tượng cơ sở dữ liệu**: Cơ sở dữ liệu của SQL SERVER dùng các đối tượng để lưu trữ dữ liệu; các đối tượng bao gồm: Diagrams, Table, Column, View, Constraint, Rule, Default, Trigger (bẫy lỗi), Stored Procedures, User Defined Function và User Defined Data Type

a. Table: là thành phần lưu trữ dữ liệu chính của SQL SERVER. Tất cả dữ liệu trong cơ sở dữ liệu của SQL Server được chứa trong Table. Mỗi Table biểu diễn một vài loại đối tượng có ý nghĩa đối với user. Ví dụ như trong cơ sở dữ liệu trường học sẽ tìm thấy một Table LOP, một Table GIAOVIEN, một Table SINHVIEN…

Những Table SQL Server cĩ 2 thnh phần chính :

* Cột : mỗi cột biểu diễn một thuộc tính nào đó của đối tượng được xây dựng trong Table, ví dụ như trong Table có những cột cho ID(m), mu sắc, trọng lượng…
* Hng : mỗi hng biểu diễn cho một sự xuất hiện ring lẻ của đối tượng được xây dựng trong Table, ví dụ như trong Table CONGTY cĩ một hng đại diện cho một cơng ty.

SQL Server cĩ 3 loại Table: Table hệ thống, Table tạm v Table của user.

**⮚Những Table hệ thống:**

SQL Server lưu trữ dữ liệu định nghĩa cấu hình của server và tất cả những Table của nó trong một tập những Table đặc biệt được biết đến như là những Table hệ thống (bắt đầu bởi sys). User không thể truy vấn hoặc cập nhật những Table hệ thống này một cách trực tiếp trừ khi không có cách nào khác để lấy dữ liệu được yêu cầu bởi một ứng dụng. Chỉ có SQL Server tham chiếu đến những Table hệ thống trong những hồi đáp cho những lệnh quản trị được đưa ra bởi user. Những Table hệ thống có thể thay đổi từ phiên bản này đến phiên bản khác.

|  |  |  |
| --- | --- | --- |
| Sysallocations | Sysdevices | Sysperinfo |
| Sysaltfile | syslanguages | sysprocesses |
| Syscharsets | Syslockinfo | sysremotelogins |
| Sysconfigs | Syslogins | sysservers |
| Syscurconfigs | sysmessages |  |
| Sysdatabases | sysoledbusers |  |

Các table hệ thống trong cơ sở dữ liệu master

***⮚*Table tạm :**

SQL Server hỗ trợ những Table tạm. Những Table này có những cái tên bắt đầu với một ký hiệu số (#). Nếu một Table tạm không biến mất khi một user ngắt kết nối, SQL Server tự động huỷ Table tạm này. Những Table tạm không được lưu trữ trong cơ sở dữ liệu hiện hành, chúng được lưu trữ trong cơ sở dữ liệu **tempdb**.

Cĩ 2 loại Table tạm:

* Table tạm cục bộ: tên của những Table này bắt đầu với một ký hiệu số (#). Những Table ny chỉ hữu hình với những kết nối tạo ra chng.
* Table tạm toàn cục: tên của những Table này bắt đầu với 2 ký hiệu số (##). Những Table ny hữu hình với tất cả cc kết nối. Nếu những Table ny khơng bị huỷ tường minh trước khi kết nối tạo ra chúng bị ngắt, chúng sẽ bị huỷ ngay sau khi tất cả những tác vụ khác ngừng tham chiếu đến chúng. Không có những tác vụ mới nào có thể tham chiếu đến Table tạm toàn cục sau khi kết nối tạo ra chúng bị ngắt. Sự kết hợp giữa một tác vụ và một Table luôn bị huỷ khi phát biểu hiện hành thực hiện hoàn tất; hơn nữa, những Table tạm toàn cục thường bị huỷ sau khi kết nối tạo ra chng bị ngắt.
* Lưu ý: Lệnh tạo table tạm không được đặt trong View

***⮚* Lm việc với Table của user:**

User lm việc với dữ liệu trong Table bằng cch sử dụng ngơn ngữ thao tc dữ liệu (DML) của pht biểu SQL

-- Lấy danh sch tất cả nhn vin cĩ tn Smith:

**SELECT** emp\_first\_name, emp\_last\_name

**FROM** employee

**WHERE** emp\_last\_name = 'Smith'

-- Xố một nhn vin:

**DELETE** employee

**WHERE** emp\_id = 'OP123'

-- Thm một nhn vin mới:

**INSERT INTO** employee

**VALUES** ( 'OP456', 'Dean', 'Straight', '01/01/1960')

-- Thay đổi tn nhn vin:

**UPDATE** employee

**SET** emp\_last\_name = 'Smith'

**WHERE** emp\_id = 'OP456'.

b. Column: Mỗi table có nhiều cột liên hệ với nhau. Mỗi cột (field) sẽ có tên và kiểu dữ liệu tương ứng. Dưới đây là các kiểu dữ liệu được dùng trong SQL SERVER:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Binary |  | Bit | Char | datetime |
| Decimal | Float | Image | Int, Bigint | Money |
| Nchar | ntext | nvarchar | Numeric | Real |
| Smalldatetime | smallint | Smallmoney | SQL\_variant | sysname |
| Text | timestamp | Tinyint | varbinary | varchar |
| Uniqueidentifier |  |  |  |  |

SQL Server cũng hỗ trợ cho Table kiểu dữ liệu cơ bản mà có thể được dùng để lưu trữ tập kết quả của một phát biểu SQL. Kiểu dữ liệu Table này không thể dùng cho một cột trong Table. Nó chỉ có thể dùng cho những tham biến của Transact-SQL hoặc những giá trị trả về của một hàm do user định nghĩa.

User cũng có thể tạo ra những kiểu dữ liệu của họ do chính user định nghĩa, ví dụ như:

-- Tạo kiểu dữ liệu một ngy sinh nhật m cho php gi trị NULL.

**EXEC** **sp\_addtype** birthday, datetime, 'NULL'

**GO**

-- Tạo một Table sử dụng kiểu dữ liệu mới.

**CREATE TABLE** employee

(emp\_id char(5),

emp\_first\_name char(30),

emp\_last\_name char(40),

emp\_birthday birthday)

Một kiểu dữ liệu do user định nghĩa tạo ra một cấu trúc Table có ý nghĩa hơn cho những người lập trình v gip bảo đảm rằng những cột nắm giữ các lớp dữ liệu giống nhau có cùng kiểu dữ liệu cơ bản.

Kiểu dữ liệu **SQL-variant** của SQL Server là một kiểu dữ liệu đặc biệt mà cho phép lưu trữ những giá trị của nhiều kiểu dữ liệu cơ bản trong cùng một cột. Ví dụ có thể lưu trữ những gi trị kiểu nchar, kiểu int v kiểu decimal trong cng một cột.

***Gi trị NULL:***

Giá trị NULL là một giá trị đặc biệt trong cơ sở dữ liệu, nó biểu diễn khái niệm về một giá trị chưa biết. NULL khơng giống với một ký tự khoảng trắng (blank) hay một ký tự 0. Khoảng trắng thực chất l một ký tự hợp lệ v 0 l một số hợp lệ. NULL biểu diễn một cch đơn giản việc chúng ta không biết giá trị này là cái gì. NULL cũng khc với một chuỗi cĩ độ dài bằng 0. Nếu định nghĩa một cột có mệnh đề NOT NULL, nghĩa là khơng thể thm những hng cĩ gi trị NULL trên cột đó. Nếu định nghĩa một cột có từ khoá NULL, nó chấp nhận những giá trị NULL.

Việc cho phép những giá trị NULL trong một cột có thể làm tăng sự phức tạp của những so sánh luận lý sử dụng trong cột đó. Chuẩn SQL-92 nĩi r rằng bất kỳ sự so snh no với gi trị NULL đều không ước lượng được là đúng hay sai mà chỉ ước lượng được là không xác định (Unknown). Do đó, để tính tốn ta phải dng hm IsNull( thamsố1, thamsố2) để chuyển NULL về 1 gía trị xác định.

Hàm IsNull sẽ kiểm tra nếu thamsố1=NULL thì hàm sẽ trả về giá trị của thamsố2; ngược lại thì hàm sẽ trả về giá trị của thamsố1.

c. Index: để tăng tốc độ truy xuất dữ liệu trong SQL SERVER. Có 2 kiểu index trong SQL SERVER:

* Cluster Index: yêu cầu SQL SERVER lưu trữ dữ liệu trong table cơ sở theo thứ tự như cluster index. Phụ thuộc vào phương thức truy xuất dữ liệu, mà ta có thể cải thiện đáng kể tốc độ truy xuất. Mỗi table chỉ có duy nhất 1 cluster index. Loại chỉ mục này là loại chỉ mục ngầm định
* Noncluster Index: không thay đổi cách thức dữ liệu được lưu trữ trong base table. Một noncluster index có thể có 1 hay nhiều field, cùng với 1 con trỏ chỉ tới dữ liệu được chứa trong table. Nó có tối đa 1024 noncluster index trong 1 table.

d. View: Một View có thể hiểu là một Table ảo hoặc là một truy vấn (query) được lưu trữ. View trả về một tập kết quả của phát biểu SELECT tạo thành Table ảo. User có thể dùng những Table ảo này bằng cách tham chiếu đến tên View trong những phát biểu Transact-SQL tương tự như tham chiếu một Table. Một View được dùng để thực hiện bất kỳ hoặc tất cả các chức năng sau:

* Giới hạn một user chỉ đọc được những hng trong một Table: ví dụ, cho php một nhn vin chỉ xem những hng ghi cơng việc của họ trong một Table labor-tracking.
* Giới hạn một user chỉ định những cột: ví dụ, cho phép những nhân viên không làm việc trên Table lương được xem những cột: tên, văn phịng, số điện thoại làm việc, và phịng ban trong Table Nhân viên, nhưng không cho phép họ xem bất kỳ cột nào chứa thông tin lương hoặc thông tin cá nhân.
* Nhóm những cột từ nhiều Table để xem như một Table.
* Thống k thơng tin thay cho những chi tiết cung cấp: ví dụ, biểu diễn tổng của một cột, hoặc gi trị lớn nhất / nhỏ nhất của một cột.

Những View được tạo bằng việc định nghĩa phát biểu SELECT sẽ tìm kiếm dữ liệu cho View đó hiển thị và những Table được tham chiếu bởi phát biểu SELECT đó gọi là những Table gốc của View đó. Trong ví dụ này, **titleview** trong cơ sở dữ liệu **pubs** là một View lọc ra dữ liệu từ 3 Table gốc để biểu diễn một Table ảo của dữ liệu cần thiết:

**CREATE VIEW** titleview

**AS**

**SELECT** title, au\_ord, au\_lname, price, ytd\_sales, pub\_id

**FROM** authors AS a

**JOIN** titleauthor **AS** ta **ON** (a.au\_id = ta.au\_id)

**JOIN** titles **AS** t **ON** (t.title\_id = ta.title\_id)

Có thể tham chiếu đến **titleview** trong những phát biểu tương tự như tham chiếu đến một Table

**SELECT** \*

**FROM** titleview

Những View có thể được dùng để phân chia dữ liệu trên nhiều cơ sở dữ liệu hoặc nhiều Server của Microsoft SQL Server.

View trong tất cả những phiên bản của SQL Server đều có thể cập nhật (có thể thực hiện những phát biểu UPDATE, DELETE hoặc INSERT), cùng với những thay đổi ảnh hưởng đến một vài Table gốc mà được tham chiếu bởi View, ví dụ:

--Tăng giá cho nhà xuất bản ‘0736’ thêm 10%

**UPDATE** titleview

**SET** price = price \* 1.10

**WHERE** pub\_id = '0736'

**GO**

Nhưng chúng ta không thể cập nhật dữ liệu đối với những View có nhiều Table kết nối với nhau (có phát biểu INNER JOIN), Trigger INSTEAD OF được sử dụng để khắc phục những trường hợp này.

e. Constraint (Ràng buộc): đảm bảo sự toàn vẹn về dữ liệu trong table. Ràng buộc thường được đưa vào khi ta tạo table, và chúng tồn tại dưới 2 mức: table và column. SQL SERVER hỗ trợ 5 loại ràng buộc sau:

i/ Primary Key : khóa chính

ii/ Foreign key : khóa ngoại

iii/ Unique key: khóa duy nhất; nó không cho phép dữ liệu trùng trên 1 cột nào đó; không giống như Primary key, Unique key cho phép dữ liệu null.

iv/ Check: ràng buộc về miền giá trị

v/ Not Null: yêu cầu dữ liệu trong 1 field không được chứa giá trị Null.

f. Rule: giới hạn các giá trị đưa vào field. Nhưng không giống như Check, 1 rule có thể giới hạn dữ liệu qua 1 biểu thức điều kiện hay qua 1 danh sách các giá trị. Và cũng không giống như Check, ta chỉ có 1 rule trên 1 column. Trong khi 1 cột được ràng buộc chỉ với 1 rule, thì 1 rule có thể áp dụng cho nhiều cột.

Ví dụ ny tạo một Rule m khi thực thi thì cĩ chức năng giới hạn cust\_id trong phạm vi từ 0 đến 10000.

**CREATE RULE** id\_chk **AS** @id **BETWEEN** 0 and 10000

**GO**

**CREATE TABLE** cust\_sample

(

cust\_id int **PRIMARY KEY**,

cust\_name char(50),

cust\_address char(50),

cust\_credit\_limit money,

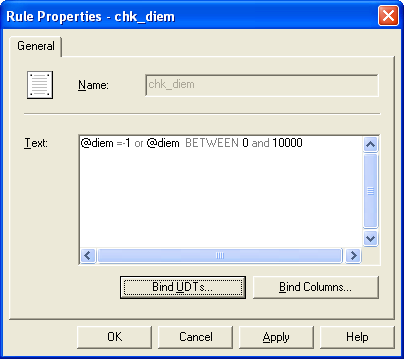
)

**GO**

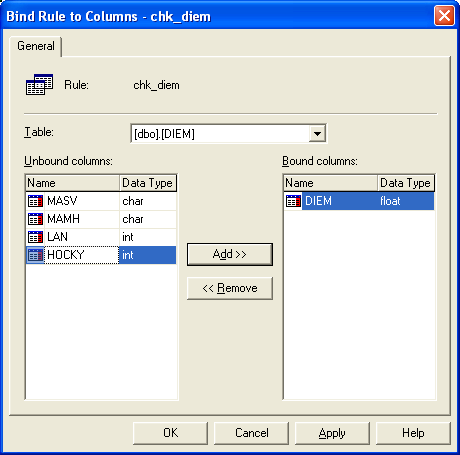
**sp\_bindrule** id\_chk, 'cust\_sample.cust\_id'

**GO**

Ta có thể tạo Rule qua giao diện như sau: Right click trên Rules, chọn New Rule:



Click Apply, sau đó click Bind Columns, chọn field muốn gắn Rule chk\_diem vào, click Apply.



g. Default: dùng để gán giá trị ngầm định cho 1 field khi field đó chưa có dữ liệu.

Giá trị Default có thể là : Hằng số, hàm được xây dựng sẵn, biểu thức toán học

Có 2 cách để áp dụng Default:

* Tạo một định nghĩa Default dùng từ khoá DEFAULT trong CREATE TABLE để ấn định biểu thức hằng là Default trên một cột. Đây là một phương pháp chuẩn thường được sử dụng.
* Tạo một đối tượng Default sử dụng phát biểu CREATE DEFAULT và kết nó vào cột qua SP hệ thống **sp\_bindefault.** Đây là tính năng cho phép có thể thay đổi giá trị mặc định cho hàng loạt các cột 1 cách dễ dàng .

Ví dụ sau đây tạo ra một Table sử dụng một trong những loại Default. Nó tạo ra một đối tượng Default để ấn định Default cho một danh hiệu, và gắn đối tượng Default đó vào cột. Sau đó nó kiểm tra việc chèn thêm mà không xác định giá trị cho những cột với những Default và tìm kiếm những hng kiểm tra để xác nhận lại những Default được áp dụng.

**USE** pubs

**GO**

**CREATE TABLE** test\_defaults

(keycol smallint,

process\_id smallint **DEFAULT** @@SPID, --định nghĩa Default

date\_ins datetime **DEFAULT** getdate(), --định nghĩa Default

mathcol smallint **DEFAULT** 10 \* 2, --định nghĩa Default

char1 char(3),

char2 char(3) **DEFAULT** 'xyz') --định nghĩa Default

**GO**

/\* Chỉ minh họa, thay cho sử dụng định nghĩa Default.\*/

**CREATE DEFAULT** abc\_const **AS** 'abc'

**GO**

**sp\_bindefault** abc\_const, 'test\_defaults.char1'

**GO**

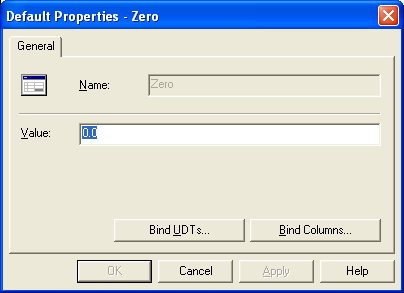
**INSERT INTO** test\_defaults(keycol) **VALUES** (1)

**GO**

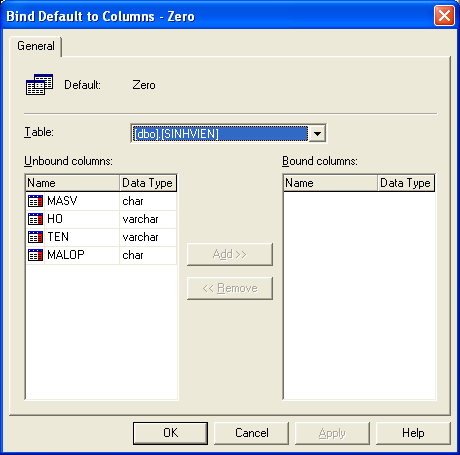
**SELECT** \* **FROM** test\_defaults

**GO**

⮚ Để tạo đối tượng Default bằmg giao diện, ta right click trên mục Defaults, chọn New Default:



Theo ví dụ trên, ta định nghĩa 1 default Zero có giá trị 0.00. Muốn kết Default Zero vào field nào, ta click **Bind Columns.**

****

Chọn field và click Add để chuyển field đó qua danh sách Bound Columns.

h. Thủ tục (Stored Procedure): là 1 nhóm các phát biểu Transact-SQL đã được compile thành 1 chương trình con. Thủ tục là 1 công cụ rất mạnh và linh hoạt được dùng để thực hiện việc quản trị cơ sở dữ liệu, cũng như thao tác trên dữ liệu như tạo table, cấp quyền, update dữ liệu…

Khi 1 thủ tục đã được compile, SQL Server sẽ tối ưu việc truy xuất dữ liệu mà thủ tục sẽ thực hiện. Các thủ tục có thể trả về các tham số, 1 tập các giá trị, hoặc đơn giản chỉ thi hành 1 công việc tự động nào đó ngầm trong hệ thống. Một thủ tục có thể được dùng chung cho nhiều user. Mỗi thủ tục có thể nhận tối đa 1024 tham số, và nó có thể được thực hiện trên 1 hệ thống SQL Server cục bộ hay từ xa.

i. Trigger: Một trigger là 1 thủ tục được tự động thực hiện khi ta thay đổi dữ liệu trong 1 table của SQL SERVER qua các lệnh Update, Insert, hay Delete. Giống như 1 thủ tục thông thường, 1 trigger sẽ chứa 1 tập các phát biểu của Transact-SQL. Trigger thường được dùng để kiểm tra các ràng buộc toàn vẹn trong cơ sở dữ liệu. Sự kiểm tra này tinh vi hơn kiểm tra trong khi tạo table.

j. Hàm do người dùng định nghĩa (User-defined Functions):

Những hm trong ngơn ngữ lập trình l những thủ tục con được dùng để đóng gói những nguyên lý thực hiện một cch thường xuyên. Một đoạn m bất kỳ phải được thực hiện sự kết hợp luận lý chặt chẽ trong một hm m cĩ thể gọi một hm thay vì lặp lại tất cả những nguyn lý của hm đó.

Microsoft SQL Server hỗ trợ 2 loại hm:

* Những hàm được cài đặt sẵn (Built-in Functions): có sẵn của hệ thống, ví dụ như IsNull, getdate, datediff, dateadd…
* Những hàm người dùng định nghĩa (User-defined Functions): cho phép định nghĩa những hàm Transact-SQL qua phát biểu CREATE FUNCTION.

Những hàm do người dùng định nghĩa có thể không có hoặc có tham số vào , và trả về một giá trị đơn giản như int, char, decimal…hoặc có thể trả về 1 tập các records.