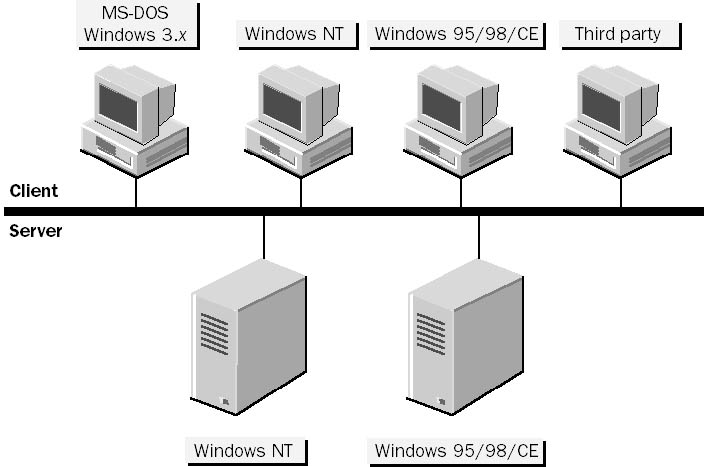
CHƯƠNG 1. TỔNG QUAN VỀ SQL SERVER

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:



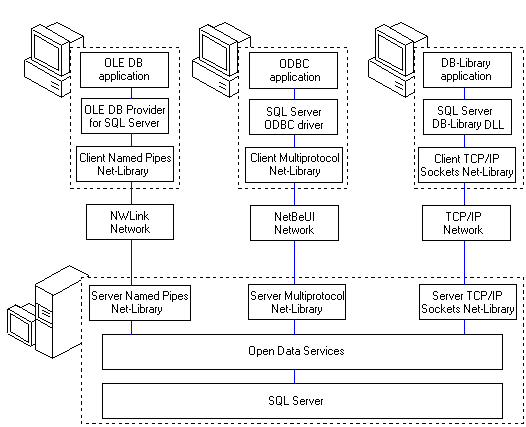
*Hình 1.1 : Hệ điều hành mà Client và Server của SQL Server có thể hoạt động*

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 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++, 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 cilent/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.

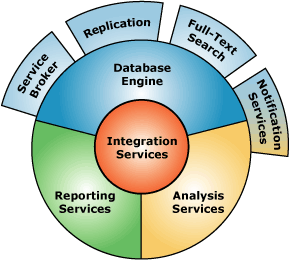
**Việc liên lạc giữa SQL server và các các client được thực hiện theo thứ tự như sau:**

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.

Quá trình trả lời từ SQL server cho client sẽ theo chuỗi các hành động ngược lại như trên .

**II. CÁC DỊCH VỤ TRONG SQL SERVER:**

SQL SERVER có các dịch vụ (Hình 1.3)



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

**1. Database Engine**: Thành phần này chịu trách nhiệm lưu trữ, bảo mật dữ liệu và xử lý giao dịch nhanh chóng.

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

Dịch vụ tìm kiếm Microsoft l một bộ máy tìm kiếm với chỉ mục 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 khác Full-text Search cũng không giới hạn chiều dài và dạng chuỗi tìm kiếm. Kỹ thuật này có khả năng tìm kiếm một ký tự , một từ, hay thậm chí cả một chuỗi.

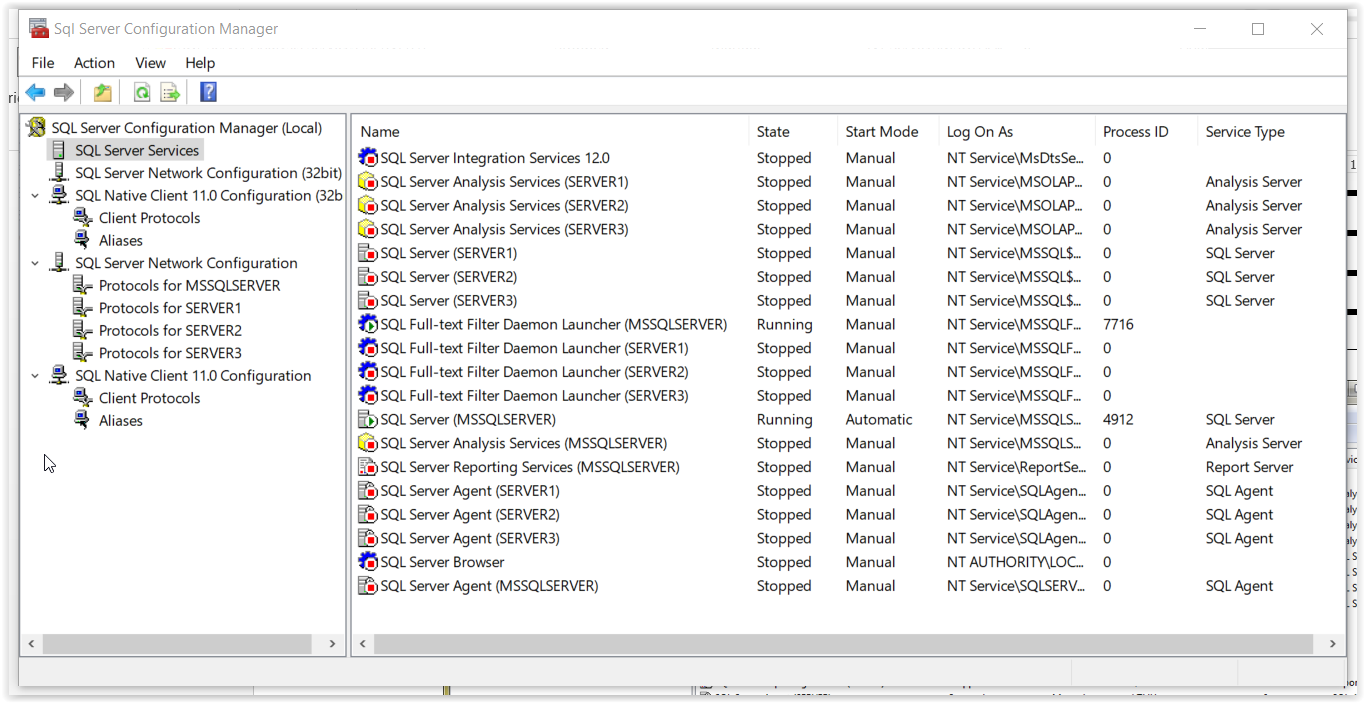
***3. Dịch vụ SQL Server* (SQL Server Service):** quản lý tất cả các file cơ sở dữ liệu. Nó có nhiệm vụ thi hành tất cả các phát biểu SQL và cấp phát tài nguyên hệ thống.

Bộ xử lý cơ sở dữ liệu SQL Server hoạt động như một dịch vụ trên Hệ điều hành Windows.

Khi nhiều Server của SQL server chạy trên cùng một máy tính thì mỗi Server có dịch vụ SQL Server của riêng nó.

Lưu ý: Muốn Start hay Stop một dịch vụ trong SQL Server, ta vào SQL Server Configuration Manager. Bảng dưới đây là các file của 5 version có trong dĩa C khi ta setup SQL Server:

| **SQL SERVER CONFIGURATION MANAGER** | |
| --- | --- |
| **Version** | **Path** |
| SQL Server 2019 | C:\Windows\SysWOW64\SQLServerManager15.msc |
| SQL Server 2017 | C:\Windows\SysWOW64\SQLServerManager14.msc |
| SQL Server 2016 | C:\Windows\SysWOW64\SQLServerManager13.msc |
| SQL Server 2014 (12.x) | C:\Windows\SysWOW64\SQLServerManager12.msc |
| SQL Server 2012 (11.x) | C:\Windows\SysWOW64\SQLServerManager11.msc |



SQL Server hỗ trợ 3 loại giao thức kết nối:

1. Shared Memory: client - SQL Server chạy trên cùng một máy, giao tiếp với nhau bằng shared memory protocol
2. TCP/IP: client - SQL Server tương tác với nhau ở mạng LAN, WAN
3. Named Pipes: client - SQL Server giao tiếp thông qua mạng LAN.

Dịch vụ SQL Server quản lý tất cả các 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 tài nguyên khác ngoài 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 .

***4. 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 này cũng cần thiết trong lệnh nhân bản database (Replication). Các thành phần SQL server thực hiện chức năng này là :

* **Jobs**:

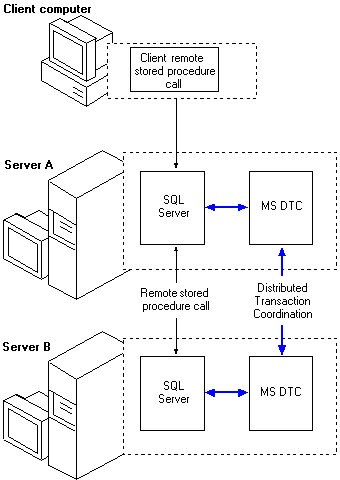
Định nghĩa các đối tượng mà đối tượng này bao gồm một hay nhiều bước thực thi. Các bước là các câu lệnh giao tác SQL sẽ được thực thi. Các công việc có thể được lên kế hoạch để thực thi tại một thời điểm chỉ định trước,

* **Alerts:**

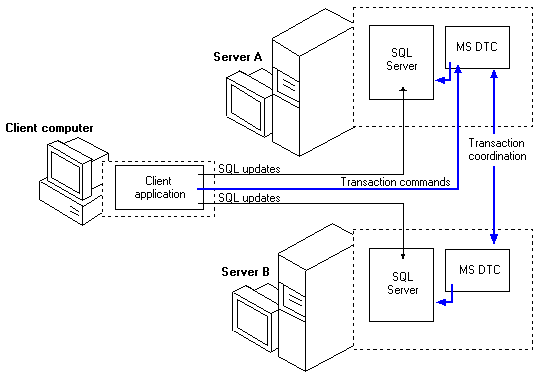
Các cảnh báo được đưa ra khi các sự kiện xảy ra, như khi lỗi xảy ra, hay khi một cơ sở dữ liệu đạt tới một giới hạn vì bộ nhớ trống có sẵn không còn đủ nữa. Cảnh báo có thể được xác định để đưa ra các hành động giải quyết như gởi một email hay thực hiện một công việc nào đó để giải quyết vấn đề xảy ra.

***5. MSDTC*** (MicroSoft Distributed Transaction Coordinator- Điều phối các lệnh trong 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 dõi hoạt động của giao tác phân tán Microsoft (The Microsoft Distributed Transaction Coordinator (MS DTC)) l một bộ quản lý giao tác cho phép các ứng dụng client tập hợp các tài nguyên dữ liệu phân tán cho một giao tác. Các MS DTC theo dõi việc chuyển các giao tác phân tán đến nơi an toàn xuyên qua các server tham gia trong giao tác

. 

Các ứng dụng SQL server cũng có 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.



1. **Nhiều Server của Server SQL trên cùng một máy tính (Multiple Instances of SQL Server)**

**1. Khái 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 riêng nó và các 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 .

Có hai loại Server SQL Server:

**Server mặc định (Default Instance )**

Server mặc định của SQL Server hoạt động giống như các bộ xử lý cơ sở dữ liệu trong các phiên bản trước đó của SQL Server. Server mặc định được nhận diện bằng tên của máy tính, không có một tên riêng biệt cho Server loại này. Khi các ứng dụng chỉ định tên máy tính để kết nối đến SQL server, thì các thành phần SQL Server client sẽ kết nối đến Server mặc định trên máy tính đó.

**Các Server đặt tên (Named Instance)**

Tất cả các Server khác của bộ xử lý cơ sở dữ liệu ngoài Server mặc định được nhận diện bằng tên của chúng. Các ứng dụng phải cung cấp cả tên máy tính và tên của Server đặt tên mà chúng muốn kết nối đến. Tên máy tính và tên Server được chỉ định có định dạng : *computer\_name*\*instance\_name*.

Có thể có nhiều Server đặt tên chạy trên một máy tính. Khi bạn cài đặt nhiều Server, mỗi Server có một tên duy nhất.

***Làm 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 làm việc với Server mặc định. Nếu chỉ định cả *computername*\*instancename* thì ta sẽ làm việc với Server đặt tên.

* **SQL Server Management Studio.**

Bằng cách sử dụng SQL Server Management Studio bạn có thể đăng ký mỗi Server với các quyền hạn cho phép. Sau khi một Server được đăng ký, bạn có thể tạo, chỉnh sửa hay xóa bỏ các đối tượng trong các CSDL quan hệ với Server đó .

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

Biến hệ thống @@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*.

**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.

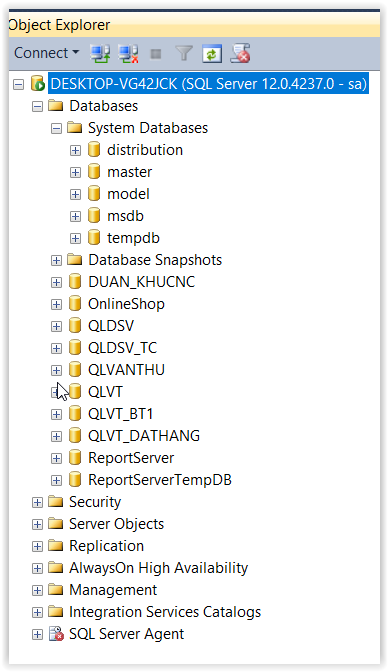
Transact-SQL thường được dùng trong các công việc để quản trị cơ sở dữ liệu như tạo bảng, tạo field, xóa field, xóa bảng..viết các thủ tục. Nó còn cho phép thay đổi cấu hình của SQL SERVER. Transact-SQL có 3 loại:

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

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

- 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.

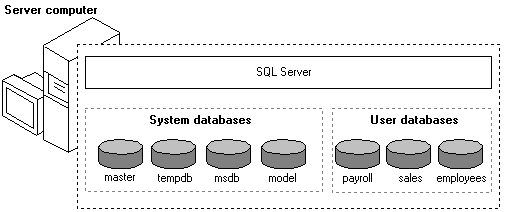
**V. KIẾN TRÚC CƠ SỞ DỮ LIỆU TRONG SQL SERVER:**



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

1. **Server:** 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 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. Mặc 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 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,.Ví dụ, một số Table của **master** chứa các thông tin hệ thống như:

|  |  |
| --- | --- |
| **Tên Table** | **Mô tả** |
| sysconfigures | Mỗi row chứa tuỳ chọn cấu hình được thiết lập sẵn bởi SQL Server |
| sysdatabases | Mỗi row chứa thông tin một cơ sở dữ liệu của SQL Server. Khi vừa được cài đặt, sysdatabase được khởi tạo với thông tin của các cơ sở dữ liệu **master, model, msdb, tempdb** |
| syslogins | Mỗi row chứa một login account để đăng nhập vào SQL Server |
| sysusers | Mỗi row chứa tên một user có thể truy cập tới cơ sở dữ liệu. |
| sysservers | Mỗi row chứa một server mà SQL Server có thể truy cập tới. |
| sysobjects | Mỗi row chứa tên 1 đối tượng trong cơ sở dữ liệu: tên user table (xtype =’U’), tên view (xtype =’V’), tên Stored Procedure (xtype =’P’), … |

Tip: Không được thay đổi master database, và nên tạo bản backup của master database thường xuyên.

⮚ **Model Database**: là 1 template cơ sở dữ liệu mà SQL SERVER dùng để hỗ trợ tạo cơ sở dữ liệu mới. 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 (thiết lập mặc định, 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 về tập sao lưu
* **user\_name:** tên của user thực hiện quá trình sao lưu
* **database\_creation\_date:** ngày & giờ tạo cơ sở dữ liệu gốc
* **backup\_start\_date:** ngày & giờ tiến trình sao lưu bắt đầu
* **backup\_finish\_date**: ngày & 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 dùng chung, 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 cho 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 không hỗ trợ user cập nhật trực tiếp những thông tin trong Table hệ thống.

3. **Các đối tượng cơ sở dữ liệu**: Cơ sở dữ liệu của SQL SERVER có các đối tượng để lưu trữ dữ liệu, các ràng buộc về dữ liệu (Diagrams, Table, Column), và các đối tượng xử lý dữ liệu( View, Stored Procedures, User Defined Function, Trigger ).

a. Table: là thành phần lưu trữ dữ liệu chính của cơ sở dữ liệu. 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 có 2 thành 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(mã), họ, tên…
* Hàng : mỗi hàng biểu diễn cho một sự xuất hiện riêng lẻ của đối tượng được xây dựng trong Table, ví dụ như trong Table CONGTY có một hàng đại diện cho một công ty.

SQL Server có 3 loại Table: Table hệ thống, Table tạm và 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 cập nhật những Table hệ thống này một cách trực tiếp. Những Table hệ thống có thể thay đổi từ phiên bản này đến phiên bản khác.

|  |  |  |
| --- | --- | --- |
| Sysallocations | Sysdevices | Syscolumns |
| Sysmembers | syslanguages | sysprocesses |
| Syscharsets | Syslockinfo | sysremotelogins |
| Sysconfigs | Syslogins | sysservers |
| Syscurconfigs | Sysusers | Sysdatabases |

Các table hệ thống trong cơ sở dữ liệu master

***⮚*Table tạm :**

SQL Server hỗ trợ những Table tạm. Những Table này có tên bắt đầu với một ký hiệu (#). Nếu một Table tạm không bị xóa trước khi 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**.

Có 2 loại Table tạm:

* Table tạm cục bộ: tên của những Table này bắt đầu với một ký hiệu (#). Những Table này chỉ hữu hình với những kết nối tạo ra chúng.
* Table tạm toàn cục: tên của những Table này bắt đầu với 2 ký hiệu (##). Những Table này hữu hình với tất cả các kết nối. Nếu những Table này không bị huỷ trước khi kết nối tạo ra chúng bị ngắt, chúng sẽ tự động bị huỷ ngay sau khi user tạo ra nó kết thúc kết nối.

Lưu ý: Lệnh tạo table tạm không được đặt trong View

***⮚* Làm việc với Table của user:**

User làm việc với dữ liệu trong Table bằng cách sử dụng ngôn ngữ thao tác dữ liệu (DML) của phát biểu SQL

-- Lấy danh sách tất cả nhân viên có tên Smith:

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

**FROM** employee

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

-- Xóa một nhân viên:

**DELETE** employee

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

-- Thêm một nhân viên mới:

**INSERT INTO** employee

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

-- Thay đổi tên nhân viên:

**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 | Date | 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 | Time |  |  |  |

SQL Server cũng hỗ trợ kiểu Table 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 birthday cho phép giá 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 viên giúp bảo đảm rằng những cột lưu 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 giá trị kiểu nchar, kiểu int , kiểu decimal trong cùng một cột.

***Giá 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 không giống với một ký tự khoảng trắng (blank) hay một ký tự 0. Khoảng trắng thực chất l một ký tự hợp lệ và 0 là một số hợp lệ. NULL biểu diễn một cách đơn giản việc chúng ta không biết giá trị này là cái gì. NULL cũng khác với một chuỗi có độ dài bằng 0. Nếu định nghĩa một cột có mệnh đề NOT NULL, nghĩa là không thể thêm những hàng có giá 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. Để tính tóan ta phải dùng hàm IsNull( thamsố1, thamsố2) để chuyển NULL về 1 giá 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 (mặc định theo khóa chính của Table): 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.
* 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. Có tối đa 1024 noncluster index trong 1 table. Loại chỉ mục này được dùng để tăng hiệu suất truy vấn, hoặc thay cho Order By trong lệnh Select (Order By thuộc 1 trong các trường hợp làm Select chậm)

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 hàng trong một Table: ví dụ, cho phép chỉ xem những nhân viên hiện còn đang làm việc tại công ty .
* 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 phòng, số điện thoại làm việc, và phòng ban trong Table Nhân viên, nhưng không cho phép họ xem bất kỳ cột nào chứa thông tin lương.
* Nhóm những cột từ nhiều Table để xem như một Table.
* Thống kê thông tin thay cho những chi tiết cung cấp: ví dụ, biểu diễn tổng của một cột, hoặc giá 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 cơ sở (base table) của View đó. Trong ví dụ này, V\_CT\_DATHANG **titleview** trong cơ sở dữ liệu **QLVT** là một View lọc ra dữ liệu từ 3 Table cơ sở để biểu diễn một Table ảo của dữ liệu các đơn đặt hàng, và chi tiết của đơn đặt hàng:

CREATE VIEW V\_CT\_DATHANG

AS

SELECT DH.MasoDDH, NV.MANV, HO, TEN, MAVT, CT.SOLUONG, CT.DONGIA,

TRIGIA=CT.SOLUONG \* CT.DONGIA

FROM DonDatHang DH

INNER JOIN NhanVien NV ON DH.MANV=NV.MANV

INNER JOIN CTDDH CT ON DH.MasoDDH=CT.MasoDDH

Ta có thể tham chiếu đến **V\_CT\_DATHANG** trong những phát biểu tương tự như tham chiếu đến một Table:

**SELECT** \*

**FROM** titleview

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: Thay đổi tên của nhân viên có mã NV =4

UPDATE V\_CT\_DATHANG

SET HO=N'Thái Hoàng'

WHERE MANV=4

Nhưng chúng ta không thể cập nhật dữ liệu đối với những View có hàm thống kê.

Lưu ý: **View không chấp nhận tham số** từ user truyền vào.

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

Ví dụ: tạo một Rule mà khi thực thi thì có 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**

Muốn bỏ Rule đã áp dụng trên field :  
**sp\_unbindrule** 'cust\_sample.cust\_id'

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

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, sau đó liên kết nó với cột trong table 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 trong các tables 1 cách nhanh chóng.

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.

**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**

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 ảo, cấp quyền, cập nhật 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, hoặc tự động thực thi 1 nhiệm vụ nào đó để đảm bảo sự nhất quán về dữ liệu.

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

Microsoft SQL Server hỗ trợ 2 loại hàm:

* 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.