

Thiết kế và thực thi CSDL Sql Server.

Như đã trình bày ở các bài trước một trong những đặc điểm của SQL Server 2000 là Multiple-Instance nên khi nói đến một (SQL) Server nào đó là ta nói đến một Instance của SQL Server 2000, thông thường đó là Default Instance. Một Instance của SQL Server 2000 có 4 system databases và một hay nhiều user database. Các system databases bao gồm:

Cấu Trúc Của SQL Server

Như đã trình bày ở các bài trước một trong những đặc điểm của SQL Server 2000 là **Multiple-Instance** nên khi nói đến một (SQL) Server nào đó là ta nói đến một Instance của SQL Server 2000, thông thường đó là Default Instance. Một Instance của SQL Server 2000 có 4 system databases và một hay nhiều user database. Các system databases bao gồm:

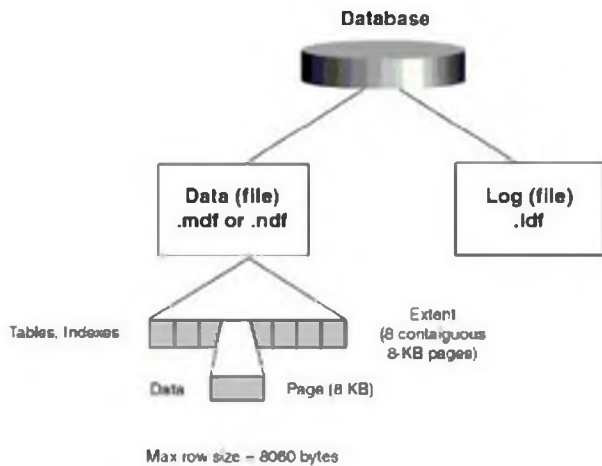
- Master : Chứa tất cả những thông tin cấp hệ thống (system-level information) bao gồm thông tin về các database khác trong hệ thống như vị trí của các data files, các login account và các thiết đặt cấu hình hệ thống của SQL Server (system configuration settings).
- Tempdb : Chứa tất cả những table hay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

Cấu Trúc Vật Lý Của Một SQL Server Database

Mỗi một database trong SQL Server đều chứa ít nhất một data file chính (primary), có thể có thêm một hay nhiều data file phụ (Secondary) và một transaction log file.

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

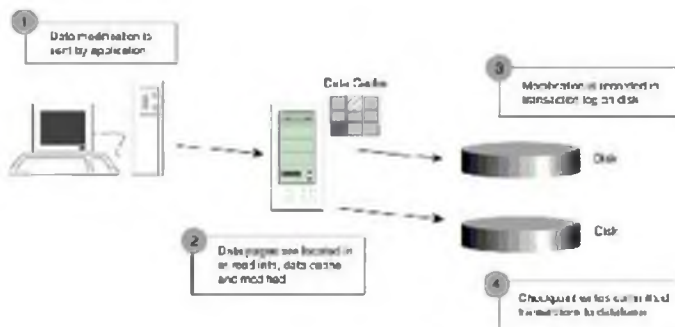
Data trong SQL Server được chứa thành từng Page 8KB và 8 page liên tục tạo thành một Extent như hình vẽ dưới đây:



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

Nguyên Tắc Hoạt Động Của Transaction Log Trong SQL Server

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



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

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

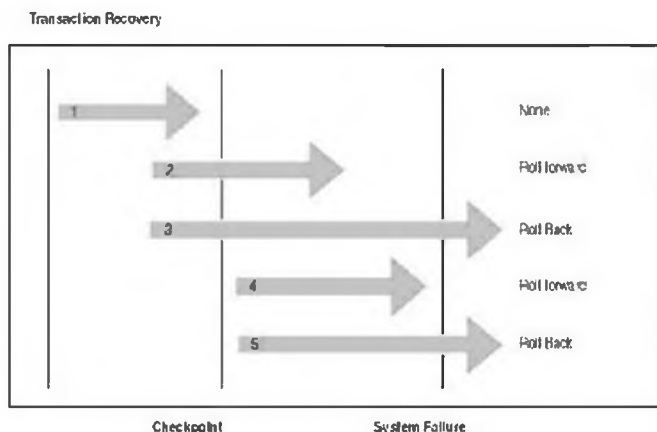
Chúng ta muốn chuyển một số tiền \$500 từ account A sang account B như vậy công việc này cần làm các bước sau:

1. Trừ \$500 từ account A

2. Cộng \$500 vào account B

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

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



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

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

Cấu Trúc Logic Của Một SQL Server Database

Hầu như mọi thứ trong SQL Server được tổ chức thành những objects ví dụ như tables, views, stored procedures, indexes, constraints.... Những system objects trong SQL Server thường có bắt đầu bằng chữ sys hay sp. Các objects trên sẽ được nghiên cứu lần lượt trong các bài sau do đó trong phần này chúng ta chỉ bàn sơ qua một số system object thông dụng trong SQL Server database mà thôi.

Một số System objects thường dùng:

System Stored Procedure	Ứng dụng
Sp_help ['object']	Cung cấp thông tin về một database object (table, view...) hay một data type.
Sp_helpdb ['database']	Cung cấp thông tin về một database cụ thể nào đó.
Sp_monitor	Cho biết độ bận rộn của SQL Server
Sp_spaceused ['object', 'updateusage']	Cung cấp thông tin về các khoảng trống đã được sử dụng cho một object nào đó
Sp_who ['login']	Cho biết thông tin về một SQL Server user

Ví dụ:

sp_helpdb 'Northwind' sẽ cho kết quả có dạng như bảng dưới đây

name	db_size	owner	dbid	created	status

Northwind	3.94 MB	sa	6	Aug 6 2000	Status=ONLINE, Updateability= READ_WRITE,

stored procedure sp_spaceused như ví dụ sau

USE Northwind

Go

sp_spaceused 'Customers'

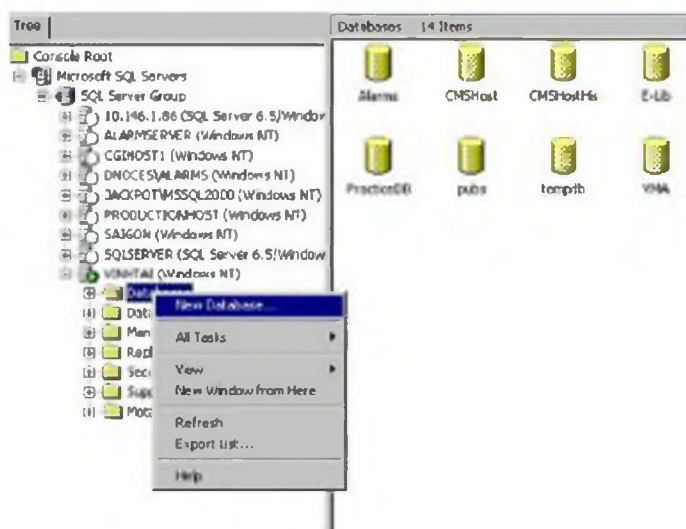
sẽ cho biết thông tin về table Customer:

name	rows	reserved	data	index_size	unused

Customers	91	104 KB	24 KB	80 KB	0 KB

Tạo Một User Database

Chúng ta có thể tạo một database dễ dàng dùng SQL Server Enterprise bằng cách right-click lên trên "database" và chọn "New Database" như hình vẽ sau:



Sau đó chúng ta chỉ việc đánh tên của database và click OK.

Ngoài ra đôi khi chúng ta cũng dùng SQL script để tạo một database. Khi đó ta phải chỉ rõ vị trí của primary data file và transaction log file.

Ví dụ:

USE master

GO

CREATE DATABASE Products

ON

(NAME = prods_dat,

FILENAME = 'c:\program files\microsoft SQL server\mssql\data\prods.mdf',

SIZE = 4,

MAXSIZE = 10,

```
FILEGROWTH = 1
```

```
)
```

```
GO
```

Trong ví dụ trên ta tạo một database tên là Products với logical file name là prods_dat và physical file name là prods.mdf, kích thước ban đầu là 4 MB và data file sẽ tự động tăng lên mỗi lần 1 MB cho tới tối đa là 10 MB. Nếu ta không chỉ định một transaction log file thì SQL sẽ tự động tạo ra 1 log file với kích thước ban đầu là 1 MB.

Lưu Ý:

Khi tạo ra một database chúng ta cũng phải lưu ý một số điểm sau: Đối với các hệ thống nhỏ mà ở đó vấn đề tốc độ của server không thuộc loại nhạy cảm thì chúng ta thường chọn các giá trị mặc định (default) cho Initial size, Automatically growth file. Nhưng trên một số production server của các hệ thống lớn kích thước của database phải được người DBA ước lượng trước tùy theo tầm cỡ của business, và thông thường người ta không chọn Autogrowth (tự động tăng trưởng) và Autoshrink (tự động nén). Câu hỏi được đặt ra ở đây là vì sao ta không để SQL Server chọn một giá trị khởi đầu cho datafile và sau đó khi cần thì nó sẽ tự động nở rộng ra mà lại phải ước lượng trước? Nguyên nhân là nếu chọn Autogrowth (hay Autoshrink) thì chúng ta có thể sẽ gặp 2 vấn đề sau:

- Performance hit: Ảnh hưởng đáng kể đến khả năng làm việc của SQL Server. Do nó phải thường xuyên kiểm tra xem có đủ khoảng trống cần thiết hay không và nếu không đủ nó sẽ phải mở rộng bằng cách dành thêm khoảng trống từ đĩa cứng và chính quá trình này sẽ làm chậm đi hoạt động của SQL Server.
- Disk fragmentation : Việc mở rộng trên cũng sẽ làm cho data không được liên tục mà chứa ở nhiều nơi khác nhau trong đĩa cứng điều này cũng gây ảnh hưởng lên tốc độ làm việc của SQL Server.

Trong các hệ thống lớn người ta có thể dự đoán trước kích thước của database bằng cách tính toán kích thước của các tables, đây cũng chỉ là kích thước ước đoán mà thôi (xin xem "Estimating the size of a database" trong SQL Books Online để biết thêm về cách tính) và sau đó thường xuyên dùng một số câu lệnh SQL (thường dùng các câu lệnh bắt đầu bằng DBCC .Phần này sẽ được bàn qua trong các bài sau) kiểm tra xem có đủ khoảng trống hay không nếu không đủ ta có thể chọn một thời điểm mà SQL server ít bận rộn nhất (như ban đêm hay sau giờ làm việc) để nở rộng data file như thế sẽ không làm ảnh hưởng tới performance của Server.

Chú ý giả sử ta dành sẵn 2 GB cho datafile, khi dùng Window Explorer để xem ta sẽ thấy kích thước của file là 2 GB nhưng data thực tế có thể chỉ chiếm vài chục MB mà thôi.

Những Điểm Cần Lưu Ý Khi Thiết Kế Một Database

Trong phạm vi bài này chúng ta không thể nói sâu về lý thuyết thiết kế database mà chỉ đưa ra một vài lời khuyên mà bạn nên tuân theo khi thiết kế.

Trước hết bạn phải nắm vững về các loại data type. Ví dụ bạn phải biết rõ sự khác biệt giữa char(10), nchar(10) varchar(10), nvarchar(10). Loại dữ liệu Char là một loại string có kích thước cố định nghĩa là trong ví dụ trên nếu data đưa vào "This is a really long character string" (lớn hơn 10 ký tự) thì SQL Server sẽ tự động cắt phần đuôi và ta chỉ còn "This is a". Tương tự nếu string đưa vào nhỏ hơn 10 thì SQL sẽ thêm khoảng trống vào phía sau cho đủ 10 ký tự. Ngược lại loại varchar sẽ không thêm các khoảng trống phía sau khi string đưa vào ít hơn 10. Còn loại data bắt đầu bằng chữ n chứa dữ liệu dạng unicode.

Một lưu ý khác là trong SQL Server ta có các loại Integer như : tinyint, smallint, int, bigint. Trong đó kích thước từng loại tương ứng là 1,2,4,8 bytes. Nghĩa là loại smallint tương đương với Integer và loại int tương đương với Long trong VB.

Khi thiết kế table nên:

- Có ít nhất một cột thuộc loại ID dùng để xác định một record dễ dàng.
- Chỉ chứa data của một entity (một thực thể)

Trong ví dụ sau thông tin về Sách và Nhà Xuất Bản được chứa trong cùng một table

Books

BookID	Title	Publisher	PubState	PubCity	PubCountry
1	Inside SQL Server 2000	Microsoft Press	CA	Berkely	USA
2	Windows 2000 Server	New Riders	MA	Boston	USA
3	Beginning Visual Basic 6.0	Wrox	CA	Berkely	USA

Ta nên tách ra thành table Books và table Publisher như sau:

Books

BookID	Title	PublisherID
1	Inside SQL Server 2000	P1
2	Windows 2000 Server	P2
3	Beginning Visual Basic 6.0	P3

và

Publishers

PublisherID	Publisher	PubState	PubCity	PubCountry
P1	Microsoft Press	CA	Berkely	USA
P2	New Riders	MA	Boston	USA
P3	Wrox	CA	Berkely	USA

- Tránh dùng cột có chứa NULL và nên luôn có giá trị Default cho các cột
- Tránh lặp lại một giá trị hay cột nào đó

Ví dụ một cuốn sách có thể được viết bởi hơn một tác giả và như thế ta có thể dùng một trong 2 cách sau để chứa data:

Books

BookID	Title	Authors
1	Inside SQL Server 2000	John Brown
2	Windows 2000 Server	Matthew Bortniker, Rick Johnson
3	Beginning Visual Basic 6.0	Peter Wright, James Moon, John Brown

hay

Books

BookID	Title	Author1	Author2	Author3
1	Inside SQL Server 2000	John Brown	Null	Null
2	Windows 2000 Server	Matthew Bortniker	Rick Johnson	Null
3	Beginning Visual Basic 6.0	Peter Wright	James Moon	John Brown

Tuy nhiên việc lập đi lập lại cột Author sẽ tạo nhiều vấn đề sau này. Chẳng hạn như nếu cuốn sách có nhiều hơn 3 tác giả thì chúng ta sẽ gặp phiền phức ngay....Trong ví dụ này ta nên chặt ra thành 3 table như sau:

Books

BookID	Title
1	Inside SQL Server 2000
2	Windows 2000 Server

Authors

AuthID	First Name	Last Name
A1	John	Brown
A2	Matthew	Bortniker
A3	Rick	Johnson
A4	Peter	Wright
A5	James	Moon

AuthorBook

BookID	AuthID
1	A1
2	A2
2	A3
3	A4
3	A5
3	A1

Ngoài ra một trong những điều quan trọng là phải biết rõ quan hệ (Relationship) giữa các table:

- One-to-One Relationships : trong mỗi quan hệ này thì một hàng bên table A không thể liên kết với hơn 1 hàng bên table B và ngược lại.
- One-to-Many Relationships : trong mỗi quan hệ này thì một hàng bên table A có thể liên kết với nhiều hàng bên table B.
- Many-to-Many Relationships : trong mỗi quan hệ này thì một hàng bên table A có thể liên kết với nhiều hàng bên table B và một hàng bên table B cũng có thể liên kết với nhiều hàng bên table A. Như ta thấy trong ví dụ trên một cuốn sách có thể được viết bởi nhiều tác giả và một tác giả cũng có thể viết nhiều cuốn sách. Do đó mỗi quan hệ giữa Books và Authors là quan hệ Many to Many. Trong trường hợp này người ta thường dùng một table trung gian để giải quyết vấn đề (table AuthorBook).

Để có một database tương đối hoàn hảo nghĩa là thiết kế sao cho data chứa trong database không thừa không thiếu bạn cần biết thêm về các thủ thuật Normalization. Tuy nhiên trong phạm vi khóa học này chúng tôi không muốn bàn sâu hơn về đề tài này, bạn có thể xem thêm trong các sách dạy lý thuyết cơ sở dữ liệu.

Tóm lại trong bài này chúng ta đã tìm hiểu về cấu trúc của một SQL Server database và một số vấn đề cần biết khi thiết kế một database. Trong bài sau chúng ta sẽ bàn về Backup và Restore database như thế nào.