**Tìm hiểu DB SQLServer**

1. **Mô hình kiến trúc DB, giải thích các thành phần bên trong**
2. **Mô hình kiến trúc của SQL Server 2008**

****

* **Kiến trúc SQL Server có thể được chia làm 3 phần:**
* *External Protocols*
* *Database Engine*
* *SQLOS API*

1. ***External Protocols***

Khi một ứng dụng giao tiếp với SQL Server Database Engine, giao diện trình ứng dụng (API) được tiếp xúc với định dạng giao thức được định nghĩa bởi Microsoft là bảng luồng dữ liệu (TDS). Net-libraries có trên cả server và client để đóng gói TDS bên trong một giao thức truyền thông tiêu chuẩn như TCP/IP, Named Pipes.

SQL Server cấu hình nhằm hỗ trợ nhiều giao thức cùng lúc đến từ các client khác nhau. Mỗi client sử dụng một giao thức duy nhất để kết nối. Nếu chương trình bên client không biết giao thức mà Server đang lắng nghe thì có thể cấu hình máy client thực hiện nhiều giao thức theo tuần tự.

*Có 4 loại giao thức bên ngoài của SQL Server:*

1. *Shared Memory*

* Giao thức đơn giản nhất, không có thiết lập cấu hình.
* Máy client sử dụng giao thức này có thể kết nối chỉ với một thể hiện của SQL Server đang chạy trên cùng một máy tính.
* Sử dụng giao thức này để xử lý sự cố khi có sự nghi ngờ các giao thức khác được cấu hình không chính xác.

1. *Named Pipes*

* Giao thức được phát triển cho các mạng nội bộ (LAN).
* Một phần của bộ nhớ được dùng để một tiến trình truyền thông tin tới tiến trình khác, do đó đầu ra của tiến trình này là đầu vào của tiến trình khác.
* Tiến trình thứ hai có thể ở trong nội bộ hoặc từ xa (remote).

1. *TCP/IP*

* Giao thức được sử dụng rộng rãi nhất trên Internet.
* TCP/IP có thể giao tiếp qua các mạng được kết nối của máy tính với các kiến trúc phần cứng và hệ điều hành khác nhau. Nó bao gồm việc định tuyến lưu lượng mạng và cung cấp các tính năng bảo mật tiên tiến.

1. *Virtual Interface Adapter(VIA)*

* Giao thức này làm việc với VIA hardware. Đây là một giao thức chuyên biệt, chi tiết về cấu hình có sẵn từ các nhà cung cấp phần cứng.
* Giao thức này đang được xem xét để gỡ bỏ trong phiên bản tương lai của Microsoft SQL Server.

1. ***Database Engine***

* Database Engine là dịch vụ cốt lõi dùng cho việc lưu trữ, xử lý và bảo vệ dữ liệu.
* Cung cấp việc kiểm soát truy cập và xử lý giao dịch nhanh chóng nhằm đáp ứng yêu cầu tiêu thụ dữ liệu của các ứng dụng.
* Tạo ra các cơ sở dữ liệu quan hệ để xử lý giao dịch trực tuyến hoặc phân tích xử lý dữ liệu trực tuyến, bao gồm việc tạo ra các bảng lưu trữ dữ liệu và các đối tượng cơ sở dữ liệu như indexes, views và stored procedures cho viewing, managing, securing data.
* Có thể chia thành 2 phần là: Query Processor và Storage Engine
  1. ***Query Processor***

1. *Parser*

* Kiểm tra các cú pháp phù hợp và dịch các lệnh T-SQL vào một định dạng nội bộ có thể vận hành được.
* Định dạng này là một cây truy vấn, nếu việc phân tích không công nhận cú pháp thì một lỗi cú pháp được đưa ra ngay lập tức để xác định nơi xảy ra lỗi. Tuy nhiên các thông báo lỗi cú pháp có thể không rõ ràng về nơi gây ra.
* Chỉ có các phân tích cú pháp câu lệnh mới có thể truy cập vào mã nguồn của câu lệnh.

1. *Optimizer*

* Truy vấn tối ưu cần cây truy vấn từ việc phân tích cú pháp câu lệnh và chuẩn bị cho việc thực hiện.
* Các câu lệnh không thể được tối ưu hóa như dòng điều khiển và các lệnh DDL, chúng được biên dịch vào trong một định dạng nội bộ.
* Các câu lệnh tối ưu hóa được đánh dấu và sau đó được truyền tới optimizer.
* Optimizer liên quan tới với lệnh DML: Select, Insert, Update và Delete và nó xác định cách xử lý tốt nhất trong nhiều cách khác nhau.

🡪Mục đích tối ưu hóa truy vấn, xác định một kế hoạch tối ưu nhất trong các kế hoạch. Xác định kế hoạch có chi phí ít nhất hoặc hiệu quả nhất dựa trên các số liệu như yêu cầu về bộ nhớ, CPU sử dụng và số lượng yêu cầu I/O.

1. *SQL Manager*

* Liên quan đến việc quản lý các quy trình đã lưu trữ và các kế hoạch của chúng.
* Xác định khi nào thì một quy trình đã lưu trữ cần phải biên dịch lại, nó quản lý bộ nhớ của các kế hoạch về quy trình để các quá trình khác có thể tái sử dụng chúng.
* Trong SQL Server, việc tái sử dụng các kế hoạch theo nhiều cách khác nhau có thể tiết kiệm nhưng trong một số trường hợp việc tái sử dụng này không thực sự tốt.

1. *Database Manager*

Quản lý quyền truy cập vào các siêu dữ liệu cần thiết để biên dịch và tối ưu hóa truy vấn

1. *Query Executor*

Chạy các kế hoạch thực hiện tối ưu hóa quy trình, hoạt động như một điều phối cho tất cả các lệnh trong kế hoạch thực hiện.

* 1. ***Storage Engine***

Bao gồm tất cả các thành phần liên quan đến việc xử lý dữ liệu hiện thời trong cơ sở dữ liệu cũng như một số Utilities và Access Methods

1. *Transaction Services*

* Hỗ trợ việc xử lý giao dịch, giúp các nhà phát triển viết và quản lý các thành phần dựa trên ứng dụng phân tán.
* Bao gồm mã cho kết nối, thư mục, bảo mật, quy trình và quản lý thread, cũng như các kết nối cơ sở dữ liệu cần thiết để tạo ra một ứng dụng nhận biết giao dịch.

1. *File Manager*

Quản lý các data file, log file trong cơ sở dữ liệu

1. *Buffer Manager*

* Là thành phần quan trọng trong việc làm cho I/O đạt hiệu quả cao.
* Buffer Manager bao gồm 2 cơ chế:
* Quản lý bộ đệm để truy cập và cập nhật các trang của cơ sở dữ liệu
* Buffer cache hay còn gọi là Buffer pool để giảm thiểu tập tin cơ sở dữ liệu I/O
* Buffer Manager quản lý các chức năng cho việc đọc dữ liệu hoặc chỉ số trang từ các tập tin cơ sở dữ liệu trên đĩa vào bộ nhớ đệm và ghi các trang đã chỉnh sửa vào lại đĩa.
* Quản lý tài nguyên để kiểm soát toàn bộ việc sử dụng bộ nhớ và trong các nền tảng 32bit thì kiểm soát việc sử dụng không gian địa chỉ.
* Quản lý cơ sở dữ liệu và hệ điều hành SQL Server cho các hoạt động file I/O ở mức độ thấp
* Quản lý các truy cập vật lý đến các trang cơ sở dữ liệu trên đĩa
* Quản lý log

1. *Lock Manager*

* Để xem hoặc thay đổi các thuộc tính cho một thể hiện của Analysis Services.
* Chịu trách nhiệm thiết lập và giải phóng lock, sắp xếp thứ tự ưu tiên của các lock.
* Cho biết một phiên truy cập như thế nào là bị lock.
* Ngăn chặn truy cập, thẩm định xem cách nào để quản lý truy cập mà dữ liệu không bị ảnh hưởng bởi phiên của chính lock.
* Cung cấp khóa thích hợp cho mỗi tài nguyên dựa vào loại truy cập và mức độ cô lập của giao dịch được thiết lập.
* Cung cấp khóa nếu không xảy ra xung đột giữa khóa bởi các giao dịch khác.

1. *Utilities*

* *Bulk Load*
* Chèn dữ liệu từ tập tin dữ liệu bên ngoài vào một bảng không rỗng trong cơ sở dữ liệu.
* SQL hỗ trợ 3 phương thức :
* *BULK INSERT:* như một câu lệnh T-SQL thực hiện trong quy trình SQL Server, chia sẻ cùng một không gian địa chỉ bộ nhớ. Việc sử dụng này không cần xác định một luồng dữ liệu, có thể định nghĩa cả nguồn và đích trong phạm vi nhiệm vụ của chính nó. Dùng để tải một tập tin bên ngoài vào bảng cơ sở dữ liệu bằng cách sử dụng một định dạng đã được chỉ định.
* *BCP:* phân tích các dữ liệu và thực hiện việc chuyển đổi dữ liệu sang định dạng lưu trữ được chỉ định trong quy trình của client.
* *DTS:* phương thức này sẽ được SQL Server loại bỏ ở những phiên bản sau
* *DBCC*
* Ngôn ngữ lập trình T-SQL cung cấp câu lệnh DBCC có tác dụng như các lệnh điều khiển cơ sở dữ liệu SQL Server.
* Duy trì các nhiệm vụ trên cơ sở dữ liệu, index hay filegroup
* Kích hoạt các cờ theo dõi hoặc loại bỏ một DLL khỏi bộ nhớ
* Thu thập và hiển thị các loại thông tin
* Xác nhận các hoạt động trên một cơ sở dữ liệu, bảng, index, catalog, filegroup hoặc phân chia các trang cơ sở dữ liệu.
* *Backup/Restore*
* Sao lưu và khôi phục là cần thiết đối với dữ liệu
* Cung cấp một biện pháp bảo vệ dữ liệu quan trọng được lưu trữ trong cơ sở dữ liệu
* Giảm thiểu nguy cơ mất mát dữ liệu khi gặp thất bại hay sự cố trong quá trình chạy

1. *Access Methods*

* Theo dõi các trang logic trong cơ sở dữ liệu được truy cập như thế nào.
* Giúp xác định hiệu suất truy vẫn có được cải thiện bằng các thêm hoặc sửa đổi các chỉ số hay bằng cách viết lại câu truy vấn.
* Có thể được đùng dể theo dõi số lượng dữ liệu, chỉ số và không gian trống trong cơ sở dữ liệu, qua đó cho ta thấy khối lượng dữ liệu và độ phân mảnh của chúng. Nếu quá nhiều phân mảnh có thể làm giảm hiệu suất hoạt động.

*Bao gồm giám sát các hạng mục:*

* *Rows:* giám sát dữ liệu được lấy ra hay đẩy vào trong một row
* *Indexes:* quét và lấy bản ghi dữ liệu từ chỉ số. Việc thực hiện các lệnh tác động lên dữ liệu có thể mất nhiều thời gian để giữ các chỉ số. Cần không gian lưu trữ chỉ số trong cơ sở dữ liệu, vì vậy nên chú ý đến hiệu suất làm việc với chi phí về không gian lưu trữ và xử lý.
* *Versions*
* *Pages:* là đơn vị cơ bản của I/O mà SQL Server sử dụng. Pages tồn tại để lưu trữ các bản ghi. Một database page tương đương với 8KB của một data file.
* *Allocations:* số lượng trang được cấp phát và chưa được cấp phát

1. ***SQLOS API***

* SQLOS cung cấp một API để các tầng khác gọi nó khi cần yêu cầu tài nguyên cho vận hành.
* API cho phép các nhà phát triển làm việc mà không cần chú ý đến nền tảng về phần cứng bên dưới.
* Có thể chia SQLOS API thành hai phần: SQLOS và External Components(Hosting API)
  1. ***SQLOS***

Là thành phần cơ bản của SQL Server, nó thực hiện chức năng thường được kết hợp với hệ điều hành như: Lock Manager, Synchronization Services, Thread Scheduler, Buffer Pool, Memory Manager, I/O Manager, Deadlock

1. *Lock Manager*

* Để xem hoặc thay đổi các thuộc tính cho một thể hiện của Analysis Services.
* Chịu trách nhiệm thiết lập và giải phóng lock, sắp xếp thứ tự ưu tiên của các lock.
* Cho biết một phiên truy cập như thế nào là bị lock.
* Ngăn chặn truy cập, thẩm định xem cách nào để quản lý truy cập mà dữ liệu không bị ảnh hưởng bởi phiên của chính lock.
* Cung cấp khóa thích hợp cho mỗi tài nguyên dựa vào loại truy cập và mức độ cô lập của giao dịch được thiết lập.
* Cung cấp khóa nếu không xảy ra xung đột giữa khóa bởi các giao dịch khác.

1. *Synchronization Services*

* Đồng bộ hóa dữ liệu bằng cách sao chép các dữ liệu và siêu dữ liệu của một cơ sở dữ liệu trên máy chủ nguồn đến một cơ sở dữ liệu trên máy chủ đích.
* Có thể thực hiện đồng bộ hóa bất cứ khi nào ứng dụng yêu cầu bằng cách gọi phương thức Synchronize hoặc việc đồng bộ có thể xảy ra tự động khi có vấn đề xảy ra với một đồng bộ cụ thể nào đó.

1. *Thread Scheduler*

* Thread là một quy trình nhỏ hoặc là một phần của quy trình. Ví dụ như chạy một câu Slect đơn giản cũng có thể coi là một quy trình và nó có thể chạy trong các bộ vi xử lý như một thread duy nhất.
* Mỗi thread sẽ nhận được một lượng nhỏ thời gian trong các bộ vi xử lý, sau đó nó phải di chuyển khỏi bộ vi xử lý để cung cấp thời gian cho các thread khác và nó phải chờ đợi trong thời gian xử lý để hoàn thành nhiệm vụ.
* SQL Server lập kế hoạch để quản lý việc xử lý và di chuyển các thread thay vì hệ điều hành.

1. *Buffer Pool*

* SQL Server tự động phát triển và thu nhỏ kích thước của bộ đệm dữ liệu tùy thuộc vào báo cáo về tải trong bộ nhớ vật lý của hệ điều hành.
* Buffer Pool trong SQL SERVER được chia thành nhiều sections, trong đó quan trọng nhất là:
* *Buffer cache:* giữ các trang dữ liệu trong bộ nhớ để các dữ liệu thường xuyên được truy cập có thể được lấy ra từ bộ nhớ cache. Giúp tối ưu hóa hiệu suất bằng việc giảm thiểu số hoạt động I/O đã yêu cầu.
* *Procedure cache:* Giữ các thủ tục được lưu trữ và kế hoạch thực hiện truy vấn để giảm thiểu số lần kế hoạch truy vấn phải được tạo ra

1. *Memory Manager*

* Giám sát việc sử dụng bộ nhớ máy chủ nhằm đánh giá hoạt động của người sử dụng và việc sử dụng tài nguyên, từ đó có thể xác định hiệu suất tắc nghẽn.
* Giám sát cải thiện hiệu suất truy vấn bằng cách thêm bộ nhớ hay trường hợp bộ nhớ sử dụng không đủ cần phải lấy dữ liệu ra từ đĩa.

1. *I/O Manager*

* Giám sát trạng thái I/O cho database file bao gồm các log file.
* Có thể theo dõi số lần I/O được thực hiện với một tập tin, theo dõi thời gian mà người dùng phải chờ để đọc và ghi vào tập tin
* Xác định được các tập tin đang gặp phải một số lượng lớn hoạt động I/O

1. *Worker Threads*

* *Deadlock Monitor*
* Một deadlock xảy ra khi có 2 hoặc nhiều công việc xảy ra xung đột khóa với nhau trên một nguồn tài nguyên.
* SQL Server tự động phát hiện deadlock và lựa chọn chấm dứt hoạt động của một trong các task nhằm phá bỏ sự xung đột.
* *Resource Monitor*
* Đây là công cụ nhanh chóng giúp giám sát việc thực hiện I/O ở cấp độ tập tin.
* Cho phép giám sát việc đọc, ghi, tổng số byte/s, thời gian phản hồi đối với từng tập tin.
* *Lazy Writer*
* Có thể điều chỉnh số lượng bộ đệm trong Buffer Pool nếu quản lý bộ nhớ động được kích hoạt.
* Hai mục đích chính là:
* Đảm bảo rằng một số bộ đệm xác định là trống trong Buffer Pool để chúng có thể được cấp phát cho việc sử dụng bởi máy chủ.
* Giám sát việc sử dụng bộ nhớ đã cam kết bởi Buffer Pool và điều chỉnh khi cần thiết để bộ nhớ vật lý đủ trống cho việc phòng ngừa Windows từ paging.
* *Scheduler Monitor*

SQL Server đưa ra khái niệm Scheduler SOS để sắp xếp khối lượng công việc xử lý bằng cách sử dụng một thuật toán lập lịch phối hợp. Nghĩa là thay vì ép buộc một thread đem lại thì nó cho phép thread tự nguyện đưa ra.

* 1. ***External Components***

Sử dụng Hosting API để giao tiếp

1. *CLR*

* Các Procedure, Function, Trigger được mã hóa trong CLR. Trong nhiều trường hợp CLR nhanh hơn T-SQL.
* CLR được dùng để hoàn thành các công việc mà T-SQL không làm được hoặc T-SQL làm được nhưng tiêu tốn quá nhiều tài nguyên.
* CLR có thể thực hiện các nhiệm vụ phức tạp đối với T-SQL.
* Việc thực hiện CLR cung cấp độ bảo mật hơn thủ tục lưu trữ mở rộng

1. *MDAC*

* Microsoft Data Access Components (MDAC) là tập hợp các thành phần phần mềm được dùng để cung cấp cho ứng dụng một cách truy cập dữ liệu từ các nguồn lưu trữ khác nhau.
* Thành phần của MDAC bao gồm các mảnh cốt lõi của OLE DB, ODBC, ADO và RDS.

🡪MDAC dùng để truy cập dữ liệu