

## **CHƯƠNG 4:**

### **Kiến trúc lưu trữ cơ sở dữ liệu**

Tham khảo:                      **Chapter 4: SQL Server Storage Architecture**  
   **Chapter 5: SQL Server Databases**

#### **Mục tiêu chương 4:**

Học viên phải biết, hiểu và áp dụng được các nội dung sau:

- Kiến trúc lưu trữ các CSDL hệ thống và CSDL người dùng
- Kiến trúc các file lưu dữ liệu và lưu thông tin hoạt động hệ thống

#### **Log File và Data File**

Một database có 2 loại file: data files và log files

Data files chứa data (của table) và index

Log files thông tin về các hoạt động và giao tác, dùng cho việc giám sát hệ thống và phục hồi dữ liệu

#### **Data Files**

Hai loại data files: primary và secondary.

##### ■ *primary data file:*

- 1 csdl luôn có,
- chứa thông tin hệ thống của csdl (của user và của catalog csdl)
- chứa pointer chỉ đến các secondary data file
- có đuôi file là .mdf

##### ■ *secondary data file:*

- 1 csdl không bắt buộc phải có, user có thể tùy chọn tạo ra hay không
- chứa các đối tượng dữ liệu và user

data

- có thể lưu khác ổ đĩa với primary data file
- có đuôi file là .ndf

## Log Files

- Log files thông tin về các hoạt động và giao tác, dùng cho việc giám sát hệ thống và phục hồi dữ liệu
- 1 cơ sở dữ liệu có ít nhất 1 log file
- có thể tạo nhiều log file cho 1 cơ sở dữ liệu
- có đuôi file là .ldf

## Filegroups

- Một filegroup là 1 nhóm logic (không phải nhóm vật lý) các data file
- Tạo ra để quản lý data file dễ dàng hơn
- Thuận lợi và nhanh chóng cho việc bảo trì hệ thống và sao lưu / phục hồi dữ liệu
- Có 2 loại filegroups: primary và user-defined.
- *primary filegroup*
  - chứa primary data file và
  - chứa các secondary data không nằm các filegroup khác
  - chứa tất cả các system table
- *user-defined filegroup*
  - chứa các secondary files

- chứa các database objects
- 1 cơ sở dữ liệu chứa tối đa 32,766 user-defined filegroups.

## HỎI VUI:

### Quick Check

- How many log files can you have in each filegroup?

### *Quick Check Answer*

- None. Filegroups contain only data files.

Khi làm việc với filegroups (dùng SSMS hay lệnh *ALTER DATABASE*), cần chú ý đến các file- group properties sau:

- 1 cơ sở dữ liệu có 1 default filegroup. Khi ta tạo 1 database object mà không cho vào filegroup nào, thì mặc định sẽ thuộc default filegroup.
- Nên dùng các read-only filegroups nếu không cần thay đổi dữ liệu. Tất cả các file- groups đều có thể gán là read-only trừ primary filegroup.

Nếu có những table có mức cập nhật dữ liệu thường xuyên — như table *Order Details* — thì ta nên:

- Tạo cơ sở dữ liệu có nhiều secondary data files
- Lưu dữ liệu trên nhiều ổ đĩa khác nhau
- Nhóm các file vào 1 filegroup

- Lưu *Order Detail* table trong filegroup để các query truy vấn dữ liệu đồng thời trên nhiều đĩa

## Cấu hình Data Files and Log Files

Cấu hình lúc dùng:

- Lệnh *CREATE DATABASE*
- Lệnh *ALTER DATABASE*
- dùng SSMS

*Table 2-1 File Configuration Options*

Option	Description
Name	logical name
Filename	file name
Size	The size for the file. When you do not specify a size for the primary file, the database engine uses the size of the primary file on the model database. If you specify a secondary or log file without the size option, the database engine creates files that are 1 MB in size.
Maxsize	The maximum size for the file. If you do not specify maxsize or you specify the <i>UNLIMITED</i> value, the file grows until the drive is full. In SQL Server 2005, a log file has a maximum size of 2 terabytes, and

data files have a maximum size of 16 terabytes.

**Filegrowth** Specifies the automatic growth allowed for the file. You can specify the value in ~~kilobytes, megabytes, gigabytes, or~~ terabytes; or as a percentage of the actual file size. If you specify a value of 0, the file will not grow.

---

- Tạo file lưu dữ liệu càng lớn càng ít bị phân mảnh dữ liệu
- Tạo file lưu dữ liệu lớn thì lâu bị vượt dung lượng file
- Có thể chỉnh file size tự tăng (autogrowth)
- Chú ý max size vượt khả năng lưu của ổ đĩa
- Nên tạo nhiều file ndf trên nhiều ổ đĩa (còn trống nhiều) để chống phân mảnh và truy xuất nhanh
  - **Cấu hình RAID tùy thuộc vào từng máy**
  - **RAID 10 nên dùng cho transaction log, data, và index files**
  - **Nếu muốn tiết kiệm chi phí, log files dùng RAID 10, data và index dùng RAID 5**

ĐỌC THÊM:

*Configuring Database Files with RAID Systems*

RAID levels used with SQL Server are 0, 1, 5, and 10.

## Best Practices

To configure data and log files for best performance, follow these best practices:

- To avoid disk contention, do not put data files on the same drive that contains the operating system files.
- Put transaction log files on a separate drive from data files. This split gives you the best performance by reducing disk contention between data and transaction log files.
- Put the *tempdb* database on a separate drive if possible, preferably on a RAID 10 or RAID 5 system. In environments in which there is intensive use of *tempdb* databases, you can get better performance by putting *tempdb* on a separate drive, which lets SQL Server perform *tempdb* operations in parallel with database operations.

### Câu hỏi ôn tập trắc nghiệm:

1. Which of the following statements can you use to create a filegroup?
  - A. *ALTER DATABASE ... ADD FILE*
  - B. *ALTER DATABASE ... MODIFY FILEGROUP*
  - C. *ALTER DATABASE ... ADD FILEGROUP*
  - D. *ALTER DATABASE ... REMOVE FILEGROUP*
2. You are in charge of designing the physical structure for your company's new server running SQL Server 2005. The server has the following characteristics: two disks in RAID 1, five disks in RAID 5, and another ten disks in RAID 5. Where should you store database files for the best performance?
  - A. Use RAID 1 to install the operating system. Use the first RAID 5 disk set to install SQL Server executable files and the second RAID 5 disk set to store database files.
  - B. Use RAID 1 to install the operating system. Use the first RAID 5 system to install SQL Server executable files and data and transaction log files. Use the second RAID

5 system to store database backups.

**C.** Use R AID 1 to install the operating system and SQL Server executable files.

Use the first R AID 5 system to store transaction log files.

Use the second

R AID 5 system to store data files.

**D.** Use the first R AID 5 system to install the operating system and SQL Server executable files. Store data files in the second R AID 5 system and log files in the R AID 1 system.

**3.** Which of the following are valid filegroup types? (Choose all that apply.)

**A.** Read-only

**B.** Write-only

**C.** Default

**D.** Primary

#### **Chương 4: (TIẾP THEO)**

#### **KIẾN TRÚC LƯU TRỮ CƠ SỞ DỮ LIỆU**

Kiến trúc Data file:

- Data file lưu: data + index + metadata
- Data file lưu theo cấu trúc gồm EXTENT và PAGE

Extent và Page:

- Một page có kích thước 8KB
- Một extent có kích thước 64KB, gồm 8 page liên tiếp nhau
- Có 2 loại extent: Uniform extent và Mixed extent
- Mixed extent: lưu dữ liệu của nhiều đối tượng (ví dụ của 2 bảng dữ liệu)
- Uniform extent: lưu dữ liệu của cùng một đối tượng (ví dụ của 1 bảng dữ liệu)

<div>Contact</div> <div>ContactID</div> <div>NameStyle</div> <div>Title</div> <div>FirstName</div> <div>MiddleName</div> <div>LastName</div>	<div>Contact</div> <div>ContactID</div> <div>NameStyle</div> <div>Title</div> <div>FirstName</div> <div>MiddleName</div> <div>LastName</div>	<div>Customer</div> <div>CustomerID</div> <div>TerritoryID</div> <div>AccountNumber</div> <div>CustomerType</div> <div>rowguid</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>Store</div> <div>CustomerID</div> <div>Name</div> <div>SalesPersonID</div> <div>Demographics</div> <div>rowguid</div> <div>ModifiedDate</div>	<div>Store</div> <div>CustomerID</div> <div>Name</div> <div>SalesPersonID</div> <div>Demographics</div> <div>rowguid</div> <div>ModifiedDate</div>	<div>SalesPerson</div> <div>SalesPersonID</div> <div>TerritoryID</div> <div>SalesQuota</div> <div>Bonus</div> <div>CommissionPot</div> <div>SalesYTD</div>	<div>SalesPerson</div> <div>SalesPersonID</div> <div>TerritoryID</div> <div>SalesQuota</div> <div>Bonus</div> <div>CommissionPot</div> <div>SalesYTD</div>
--	--	--	--	--	--	--	--

#### Mixed Extent

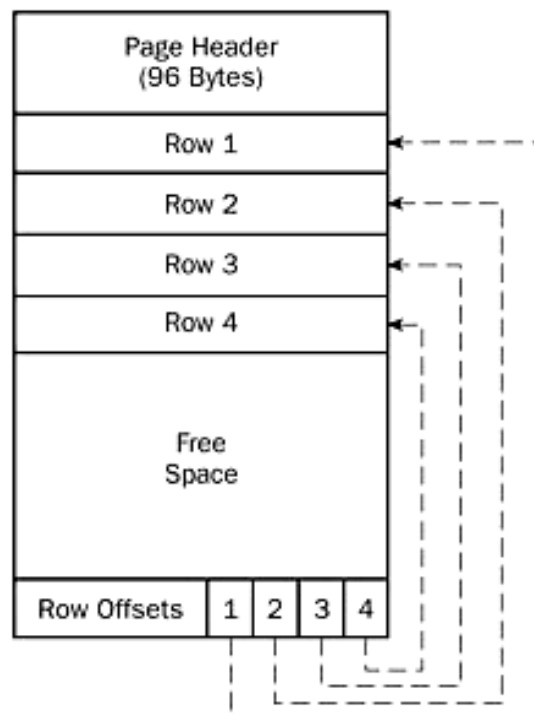
<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>	<div>CreditCard</div> <div>CreditCardID</div> <div>CardType</div> <div>CardNumber</div> <div>ExpMonth</div> <div>ExpYear</div> <div>ModifiedDate</div>
--	--	--	--	--	--	--	--

#### Uniform Extent

- Mỗi data file bắt đầu bằng Mixed extent chứa các page lưu thông tin quản lý nội dung của data file
- Page đầu tiên của extent đầu tiên (mixed extent) của data file gọi là File header page. File header page lưu: logical name, physical name, max size, growth increment, và filegroup của data file
- Không như mixed extent, mỗi page luôn chứa dữ liệu của cùng một đối tượng
- Mỗi page dành ra 96 byte lưu thông tin quản lý page, trong đó chứa: page number, loại dữ liệu trong page, không gian còn trống của page, và đối tượng của dữ liệu trong page
- Có nhiều loại page, phân loại theo dữ liệu của page:
  - o data page
  - o index page
  - o Text/Image Pages
  - o Global Allocation Map (GAM) and Secondary Global Allocation Map (SGAM) Pages
  - o Page Free Space (PFS) Pages
  - o Index Allocation Map (IAM) Pages
  - o Bulk Changed Map (BCM) Pages
  - o Differential Changed Map (DCM) Pages

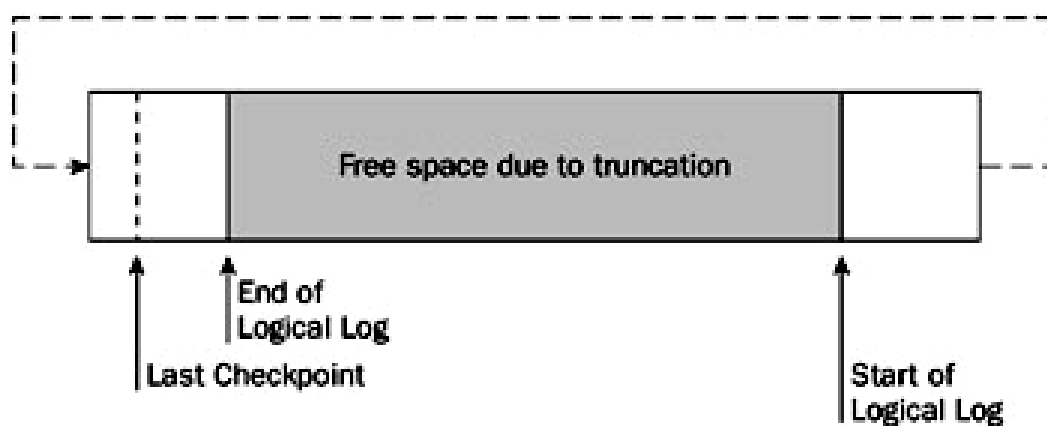


Data page:



Transaction Log file

- Là file binary
- Không thể xem nội dung bằng các chương trình xử lý văn bản bình thường (như Notepad)
- Có thể có 1 hay nhiều file ldf, thường chỉ có 1 file
- Cấu trúc file ldf do SQL Server tạo tự động và không cho điều chỉnh



Lập phương án lưu cơ sở dữ liệu:

- Thông thường, ta khó dự đoán kích cỡ và mức tăng kích cỡ của 1 cơ sở dữ liệu
- Kích cỡ mặc định thường không thích hợp, chỉ nên dùng học hay demo mà thôi
- Thường người viết ứng dụng yêu cầu các kích cỡ cho csdl của họ
- Nếu phải tính kích cỡ thì ta có thể thực hiện 7 bước để dự đoán (xem tài liệu tham khảo)
- Cách tính thực dụng: dùng csdl lưu dữ liệu thử, sau đó nhân dung lượng lên 1.5 lần
- Dữ liệu thử phải bao gồm tất cả các đối tượng dữ liệu sẽ dùng

**Đọc thêm:**

One such technique that I am sure you will encounter is calculating a database size requirement by calculating table sizes. It looks like this:

1. Add up the total number of bytes used by the fixed-length columns in the table.
2. Average the total number of bytes used by the variable-length columns in the table.
3. Add the number from Step 1 to the number calculated in Step 2.
4. Divide 8092 (the maximum amount of data bytes in a page) by the number calculated in Step 3, and round down to the nearest whole number. This is number of rows that will fit on a single page. Remember that rows cannot span pages, which is why you round down.
5. Divide the total number of expected rows by the number of rows per page calculated in Step 4. This is the total number of data pages expected to support the table.
6. Multiply the number calculated in Step 5 by 8192 (the size of data page). This is the total number of bytes required for the table.
7. Repeat the process for every table in the database.

Sounds like fun doesn't it? Here's a tip: Don't do it. The results from this algorithm are misleading at best. The calculation doesn't take into account variables that affect storage space, such as the number of

indexes, the fill-factor used on the indexes, and data fragmentation, just to name a few. So, why did I even bother to explain the process? Because it does give insight to size considerations and because, as I mentioned earlier, you will most likely encounter this technique and I wanted to make sure you knew its limitations.