Trường Đại Học Bách Khoa Tp.HCM
Hệ Đào Tạo Từ Xa
Khoa Khoa Học và Kỹ Thuật Máy Tính

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Chương 7. Các kỹ thuật khôi phục dữ liệu (data recovery)

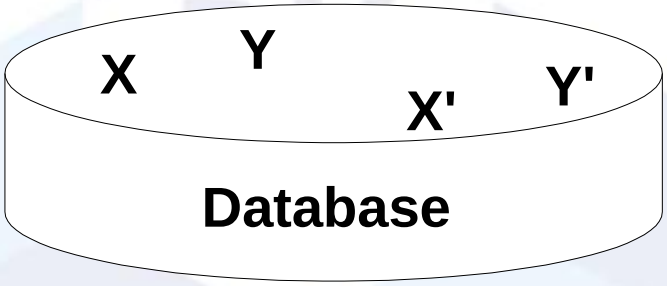
Bài 2 – Nâng cao

Bùi Hoài Thắng

Shadow paging (*)

Phân trang bóng hình (Shadow paging)

* Phân trang bóng hình (shadow paging):
  - Không ghi đè các AFIM lên các BFIM trên đĩa
  - Ghi vào một vị trí khác trên đĩa
* Tại mỗi thời điểm, một mục liệu có cả AFIM và BFIM (bản sao bóng hình của mục liệu) tại hai vị trí khác nhau trên đĩa

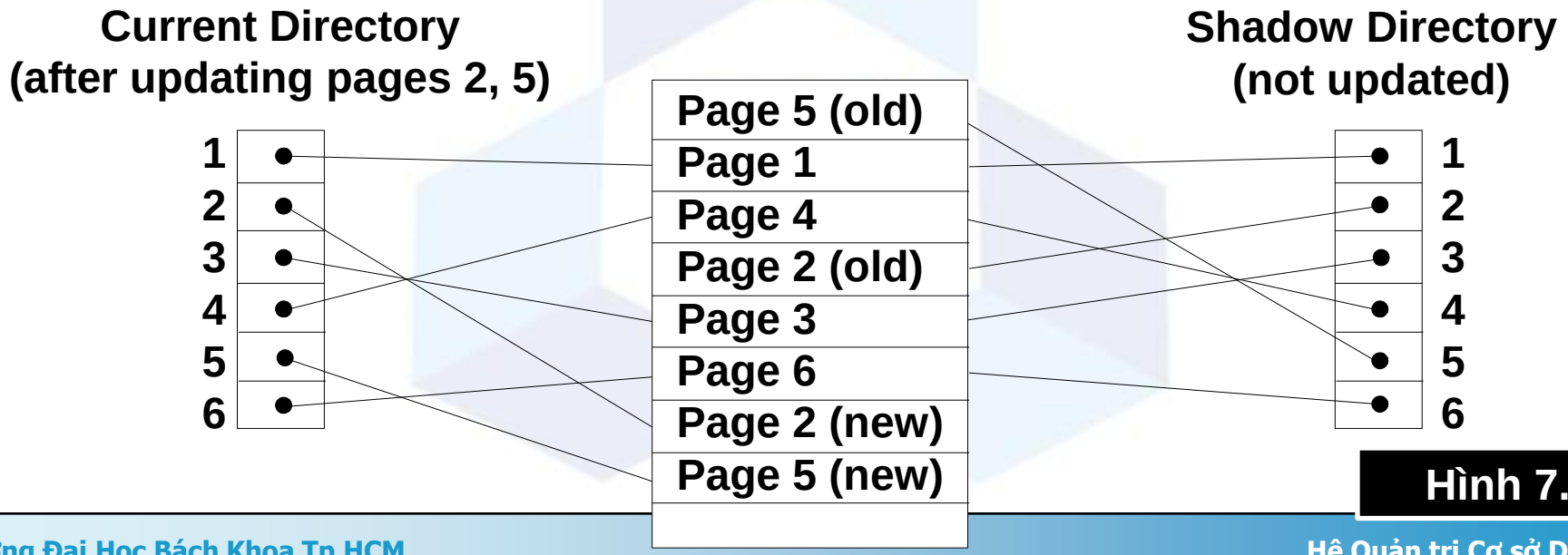


Hình 7.7

  - X và Y: bản sao bóng hình (Shadow copy) của các mục liệu X và Y
  - X' và Y': Bản sao hiện tại của các mục liệu

Phân trang bóng hình (tt.)

* Dùng hai thư mục:
  - Thư mục hiện tại (current directory): các con trỏ đến các trang (mục liệu) đang được sử dụng (cập nhật)
  - Thư mục bóng hình (shadow directory): các con trỏ đến các trang (mục liệu) bóng hình



Hình 7.8

Khôi phục dùng phân trang bóng hình

* Để tháo gỡ các giao tác chưa hoàn thành, bỏ đi thư mục hiện tại và dùng lại thư mục bóng hình
* Để tái thực thi các giao tác đã commit, bỏ đi thư mục bóng hình
* Còn gọi là kỹ thuật NO-UNDO/NO-REDO
* Trong môi trường tương tranh, cần phải dùng sổ ghi (và checkpoint) để biết được trang/block nào được cập nhật bởi giao tác nào

Khôi phục dùng phân trang bóng hình (tt.)

* Ưu điểm:
  - Không cần sổ ghi trong môi trường đơn người dùng
  - Không cần BFIM và AFIM lưu trong sổ ghi (môi trường tương tranh) – NO-UNDO/NO-REDO
* Nhược điểm:
  - Phân mảnh dữ liệu khi các mảnh dữ liệu không nằm kề nhau
  - Phải quản lý vấn đề thu dọn các trang/khối bóng hình không còn dùng (khi giao tác commit)
  - …

Giải thuật khôi phục ARIES (*)

Khôi phục dùng giải thuật ARIES

* Giải thuật khôi phục ARIES dựa trên phương án steal/no-force (UNDO/REDO)
* Giải thuật dựa trên ba ý niệm chính:
  - WAL (Write Ahead Logging)
  - Tái lập lịch sử trong khi tái thực thi (redo):
    - ARIES truy tất cả các tác vụ của hệ thống CSDL trước khi có sự cố để tái dựng lại trạng thái CSDL khi sự cố xảy ra.
  - Ghi sổ các thay đổi trong khi tháo gỡ (undo):
    - Tránh cho ARIES lặp lại các tác vụ đã tháo gỡ (undo) thành công nếu có sự cố xảy ra trong quá trình khôi phục làm cho quá trình khôi phục tái khởi động

Khôi phục dùng giải thuật ARIES (tt.)

* Giải thuật ARIES gồm 3 bước:
  - Phân tích (Analysis):
    - Phát hiện ra các trang đã được cập nhật trong vùng đệm và tập các giao tác đang hoạt động tại thời điểm hệ thống gặp sự cố
    - Phát hiện điểm thích hợp trong sổ ghi để bắt đầu tác vụ tái thi hành (redo) đầu tiên
  - Tái thực thi (Redo): các tác vụ tái thực thi cần thiết được áp dụng
  - Tháo gỡ (Undo): sổ ghi được quét ngược để tháo gỡ các tác vụ cập nhật của các giao tác đang hoạt động (chưa commit)

Số ghi sổ tuần tự (Log Sequence Number – LSN)

- Giải thuật yêu cầu ghi sổ các tác vụ sau:
  - (a) cập nhật dữ liệu,
  - (b) commit giao tác,
  - (c) huỷ bỏ giao tác (abort),
  - (d) tháo gỡ (undo)
    - để tránh lặp lại trong quá trình tháo gỡ giao tác
  - (e) kết thúc giao tác (end)
- Mỗi mục sổ ghi được gán một giá trị LSN tăng dần duy nhất.
  - Cho biết địa chỉ đĩa chứa mục sổ ghi
- Mỗi một trang dữ liệu lưu LSN của mục sổ ghi cuối cùng cập nhật trang

Số ghi sổ tuần tự (tt.)

* Mỗi mục sổ ghi chứa:
  - LSN của tác vụ trước của cùng giao tác
    - Giống danh sách liên kết ngược
  - Danh định của giao tác (Transaction ID)
  - Kiểu mục sổ ghi
* Nếu là tác vụ write_item thì cần thêm:
  - Danh định của trang dữ liệu (Page ID) chứa mục liệu
  - Chiều dài của mục liệu
  - Vị trí của mục liệu trong trang (offset)
  - BFIM của mục liệu
  - AFIM của mục liệu

Transaction table và Dirty Page table

* Transaction table:
  - Mỗi mục trong bảng cho một giao tác đang hoạt động
  - Gồm: Transaction ID, trạng thái của giao tác (status) và LSN của mục sổ ghi gần nhất của giao tác
* Dirty Page table:
  - Mỗi mục trong bảng cho một trang đã cập nhật trong vùng đệm
  - Gồm: Page ID và LSN của mục sổ ghi **đầu tiên** cập nhật trang này
* Các bảng này được lưu vào trong sổ ghi trong quá trình lập điểm kiểm tra (checkpoint)

Quá trình tạo điểm kiểm tra

- ARIES dùng kỹ thuật tạo điểm kiểm tra sau:
  - (1) Ghi mục *begin_checkpoint* vào sổ ghi
  - (2) Ghi mục *end_checkpoint* vào sổ ghi khi quá trình kết thúc. Đồng thời, thêm nội dung của *transaction table* và *dirty page table* vào cuối sổ ghi
  - (3) Ghi LSN của mục *begin_checkpoint* vào một tập tin đặc biệt.
    - Tập tin này sẽ được dùng để lấy điểm kiểm tra cuối cùng
- ARIES dùng kỹ thuật điểm kiểm tra mờ (fuzzy checkpointing)
- Trong quá trình checkpoint, không cần phải lưu tất cả vùng đệm xuống đĩa
  - Dùng sổ ghi để khôi phục

Giải thuật ARIES

* Bước phân tích (Analysis phase):
  - Bắt đầu từ mục *begin_checkpoint* và tiếp tục đến mục *end_checkpoint* trong sổ ghi
  - Lấy các thông tin của transaction table và dirty page table (kế tiếp mục *end_checkpoint*)
  - Quét đến cuối sổ ghi (trước sự cố xảy ra) để tái lập transaction table và dirty page table

Giải thuật ARIES (tt.)

* Tái lập transaction table và dirty page table
  - Nếu có mục *end_transaction* của một giao tác trong transaction table, giao tác này sẽ bị loại khỏi bảng
  - Nếu có một mục sổ ghi của một giao tác, giao tác này sẽ được thêm vào transaction table (nếu chưa có trong bảng) hoặc là LSN cuối cùng của giao tác này sẽ được cập nhật
  - Nếu có tác vụ thay đổi ở một trang, trang này sẽ được thêm vào dirty page table (nếu chưa có) với LSN của tác vụ này

Giải thuật ARIES (tt.)

* Bước tái thực thi (Redo phase):
  - Bắt đầu tại vị trí trong sổ ghi khi tất cả các trang đã được cập nhật đã được ghi vào đĩa
  - Duyệt đến cuối sổ ghi, các thay đổi ở các trang trong dirty page table sẽ được tái thực thi
* Tìm vị trí bắt đầu redo:
  - M là giá trị LSN nhỏ nhất của tất cả các mục trong dirty page table
  - Tại sao?

Giải thuật ARIES (tt.)

* Cần kiểm tra ở bước tái thực thi để tránh trường hợp ghi nhận các giá trị đã cũ:
  - Nếu có tác vụ ghi vào trang không còn trong dirty page table, nghĩa là trang này đã được lưu xuống đĩa, không cần tái thực thi tác vụ này
  - Nếu LSN của tác vụ ghi vào trang có trong dirty page table (hoặc mới đọc lên từ đĩa) nhỏ hơn LSN lưu trong trang, nghĩa là trang này đã từng được lưu xuống đĩa trước đó và đã được cập nhật sau tác vụ hiện tại, không cần tái thực thi tác vụ này

Giải thuật ARIES (tt.)

* Bước tháo gỡ (Undo phase):
  - Bắt đầu từ cuối sổ ghi và tiếp tục dò ngược để tiến hành các tác vụ tháo gỡ (undo)
  - Chỉ tháo gỡ cho các tác vụ của các giao tác có trong transaction table
  - Mỗi khi tháo gỡ một tác vụ, nó ghi vào sổ ghi một mục *undo*

Ví dụ về ARIES

(a)

LSN	LAST-LSN	TRAN-ID	TYPE	PAGE-ID	Other Info.
1	0	T1	update	C	-----
2	0	T2	update	B	-----
3	1	T1	commit		-----
4	begin checkpoint				
5	end checkpoint				
6	0	T3	update	A	-----
7	2	T2	update	C	-----
8	7	T2	commit		-----

Tại thời điểm checkpoint

(b)

TRANSACTION TABLE

TRANSACTION ID	LAST LSN	STATUS
T1	3	commit
T2	2	in progress

DIRTY PAGE TABLE

PAGE ID	LSN
C	1
B	2

Sau khi phân tích xong

(c)

TRANSACTION TABLE

TRANSACTION ID	LAST LSN	STATUS
T1	3	commit
T2	8	commit
T3	6	in progress

DIRTY PAGE TABLE

PAGE ID	LSN
C	1
B	2
A	6

Hình 7.9

Các vấn đề khôi phục khác

Khôi phục trong hệ thống đa CSDL

- Một hệ thống đa CSDL (multidatabase system) là một hệ CSDL phân bố (distributed database system) đặc biệt:
  - Mỗi nút có thể là một CSDL quan hệ trên nền tảng khác nhau (HĐH, DBMS, …)
  - Một giao tác có thể thi hành theo kiểu phân bố
- Giao tác chỉ có thể commit nếu tất cả các nút đã commit thành công
  - Dùng chế độ commit hai-pha (two-phase commit - 2PC)
  - Nếu một trong các nút thất bại, cả giao tác phải bị huỷ bỏ
  - Mỗi nút thực thi quá trình khôi phục độc lập

Khôi phục khi có tai hoạ

* Dùng Sao lưu và phục hồi (Backup & Restore)
* Các chiến lược sao lưu
  - Tự tìm hiểu

Sao lưu và khôi phục trong SQL-Server

Transaction Log

- Mỗi một CSDL trong SQL-Server có một transaction log lưu tất cả các giao tác và các thay đổi từ các giao tác đó.
  - Là một bộ phận tối quan trọng của CSDL được dùng để khôi phục CSDL về trạng thái nhất quán khi có sự cố
- Hỗ trợ các tác vụ:
  - Khôi phục các giao tác riêng lẻ
  - Khôi phục tất cả các giao tác chưa hoàn thành khi SQL-Server khởi động
  - Quay xuôi một CSDL/file/filegroup hoặc page đã được phục hồi (từ sao lưu) về (trước) điểm xảy ra sự cố
  - Hỗ trợ nhân bản (replication)
  - …

Transaction Log (tt.)

* Log file = tập tin tuần tự các log record
* Log record:
  - LSN
  - Trans. ID
  - Backward Pointer
  - BFIM: opt
  - AFIM: opt
* Log type:
  - start và end của mỗi giao tác
  - data modification (insert, update, or delete)
  - Cấp phát/thu hồi các extent và page
  - Tạo/huỷ các table/index
  - …

Lệnh CHECKPOINT

* `CHECKPOINT [ checkpoint_duration ]`
* Ghi tất cả các dirty page vào đĩa
* SQL-Server tự động thực hiện lệnh này trước khi sao lưu SQL và một số trường hợp khác (ví dụ như khi transaction log đầy đến 70%)

Các mô hình khôi phục

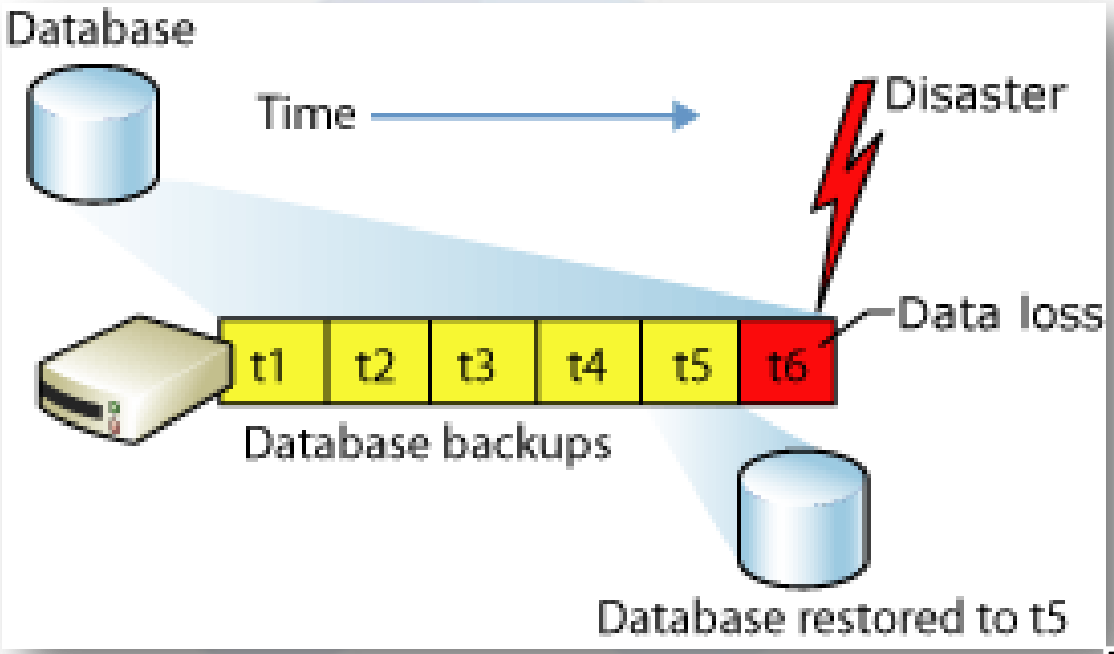
* Đơn giản (Simple recovery model):
  - Không cần sao lưu log
  - Chỉ có thể khôi phục đến một bản sao lưu gần nhất
    - Mất các dữ liệu từ lần sao lưu gần nhất đến hiện tại
* Đầy đủ (Full recovery model):
  - Cần phải sao lưu log
  - Có thể khôi phục CSDL đến một thời điểm nào đó trước khi khôi phục (có sự cố xảy ra)
* Ghi sổ gộp (Bulk logged recovery model):
  - Giống full recovery model nhưng các tác vụ gộp (nạp gộp dữ liệu – bulk import) sẽ được ghi dạng gộp

Các chế độ sao lưu dữ liệu

* Full backup:
  - Sao lưu toàn bộ dữ liệu của một CSDL/file/filegroup/log
* Differential backup:
  - Dựa trên lần full backup cuối
  - Chỉ các thay đổi mới được sao lưu
  - Khi phục hồi, dữ liệu full backup sẽ được phục hồi trước, sau đó các thay đổi trong differential backup gần nhất sẽ được dùng

Sao lưu dùng mô hình đơn giản

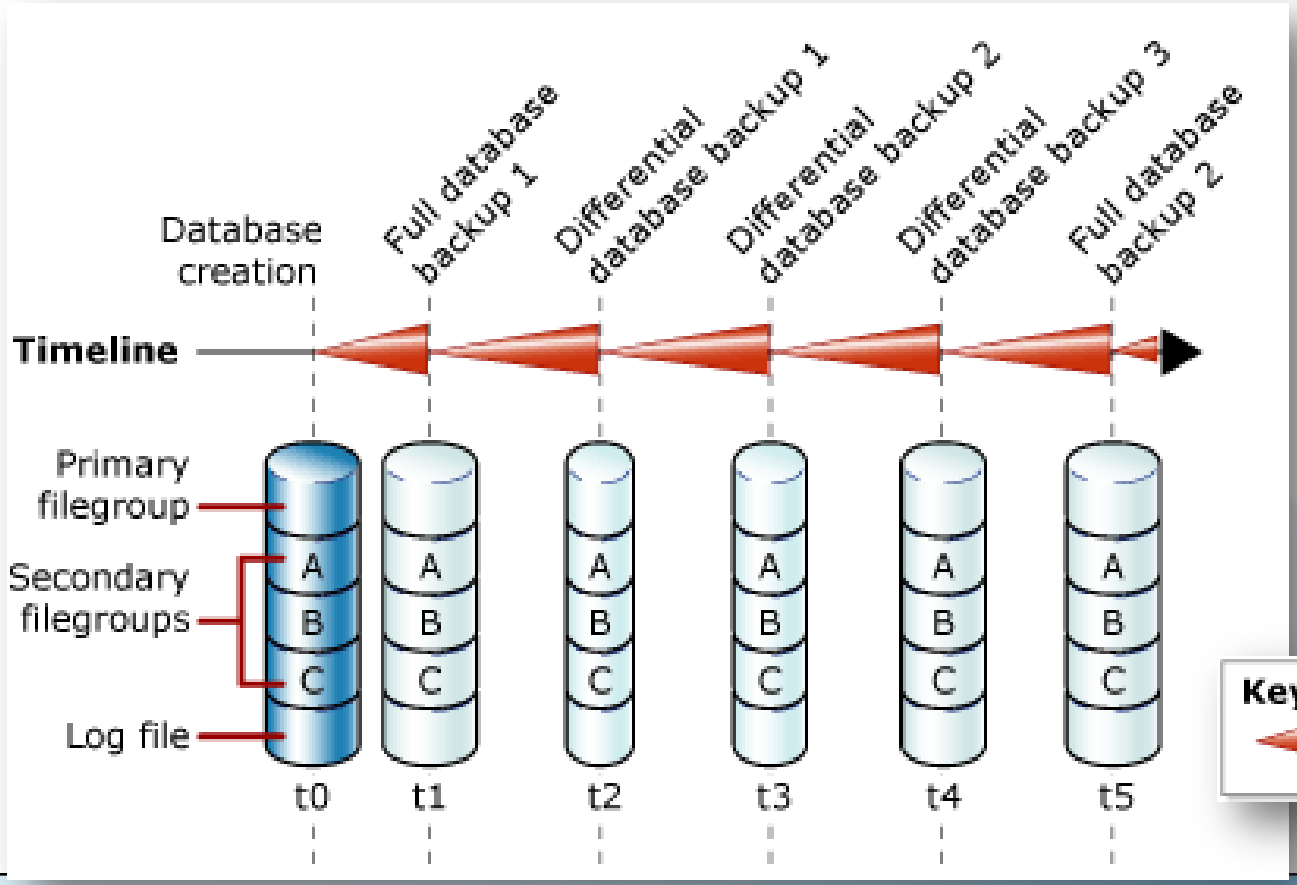
* Dùng full backup



Hình 7.10

Sao lưu dùng mô hình đơn giản (tt.)

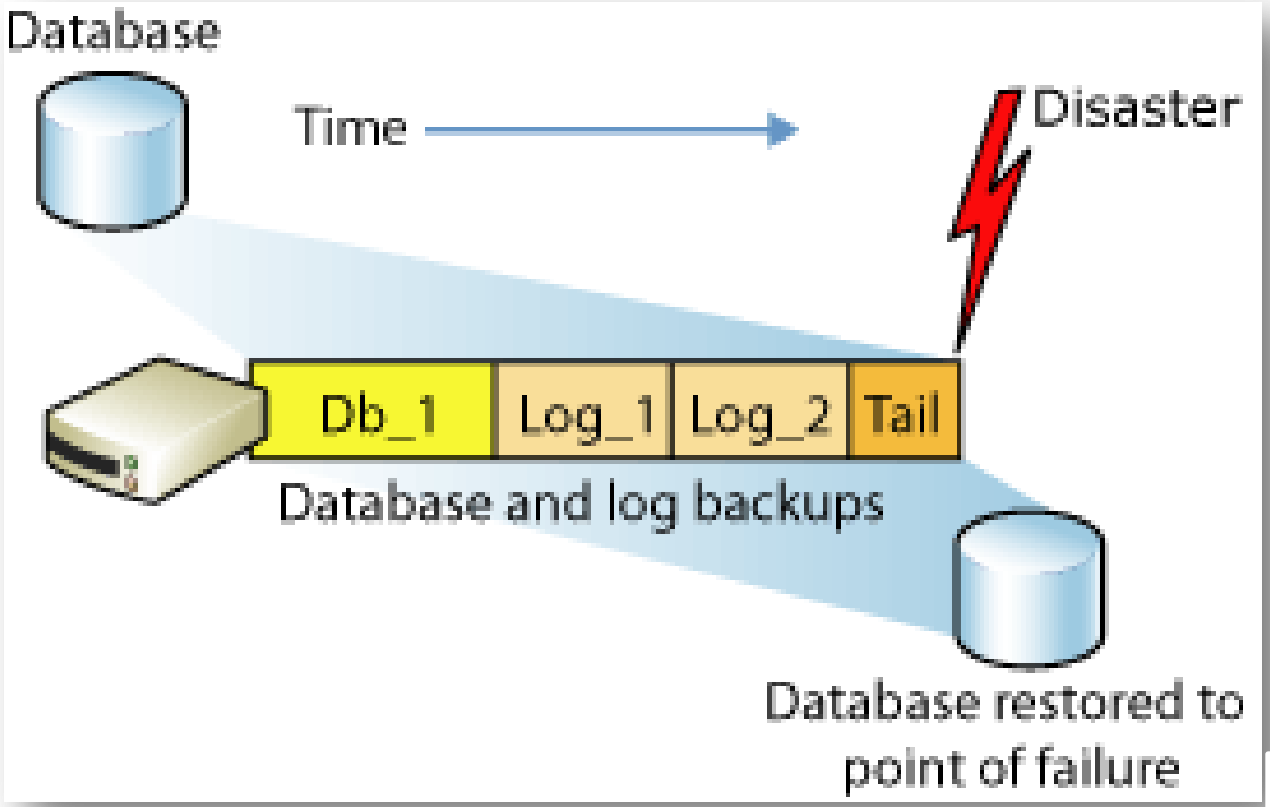
* Dùng differential backup



Hình 7.11

Sao lưu dùng mô hình đầy đủ

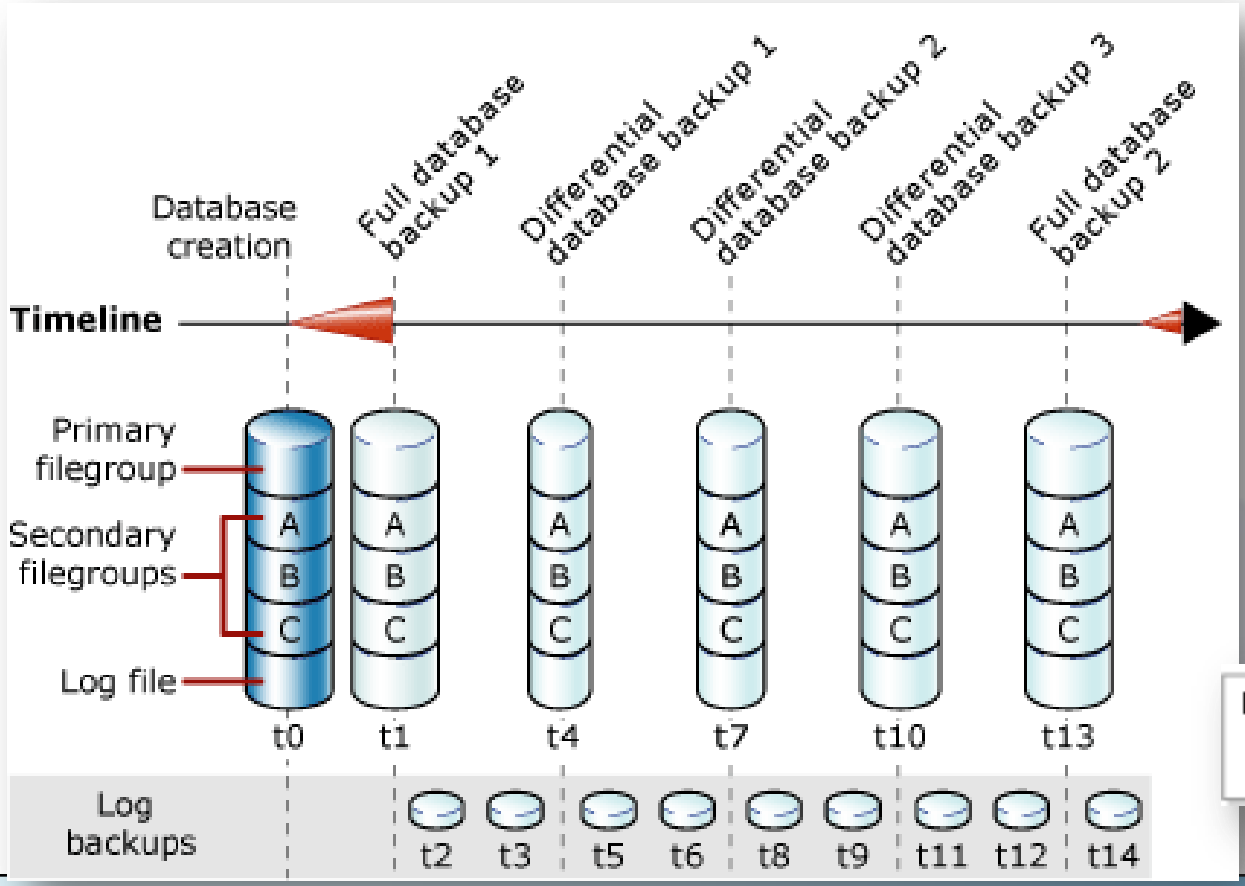
* Dùng database backup và log backup



Hình 7.12

Sao lưu dùng mô hình đầy đủ (tt.)

* Dùng database backup và log backup



Hình 7.13

Lệnh BACKUP

* Sao lưu cả CSDL:

```
BACKUP DATABASE {database_name | @database_name_var }
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { DIFFERENTIAL | <general_WITH_options> [
,...n ] } ]
```

* Sao lưu file/filegroup:

```
BACKUP DATABASE {database_name | @database_name_var }
 <file_or_filegroup> [ ,...n ]
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { DIFFERENTIAL | <general_WITH_options> [
,...n ] } ]
```

Lệnh BACKUP (tt.)

* Sao lưu một phần CSDL:

```
BACKUP DATABASE {database_name | @database_name_var}
 READ_WRITE_FILEGROUPS [ , <read_only_filegroup> [
,...n ] ]
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { DIFFERENTIAL | <general_WITH_options> [
,...n ] } ]
```

* Sao lưu Transaction log:

```
BACKUP LOG { database_name | @database_name_var }
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { <general_WITH_options> | <log-
specific_optionspec> } [ ,...n ] ]
```

Ví dụ lệnh BACKUP

* Sao lưu một CSDL vào một stripe set:

```
BACKUP DATABASE AdventureWorks2008R2
TO
DISK='X:\Backups\AdventureWorks2008R2_1.bak',
DISK='Y:\Backups\AdventureWorks2008R2_2.bak',
DISK='Z:\Backups\AdventureWorks2008R2_3.bak'
WITH FORMAT,
MEDIANAME='AdventureWorks2008R2StripedSet0',
MEDIADESCRIPTION = 'Striped media set for
AdventureWorks2008R2 database;
GO
```

Lệnh RESTORE

* Phục hồi một CSDL từ một full database backup
* Phục hồi một phần của một CSDL
* Phục hồi các file/filegroup của một CSDL
* Phục hồi các trang (page) của một CSDL
* Phục hồi một transaction log vào một CSDL
* Đảo ngược một CSDL về một thời điểm dùng database snapshot

Lệnh RESTORE (tt.)

* Phục hồi một CSDL từ một full database backup

```
RESTORE DATABASE {database_name | @database_name_var}
 [FROM <backup_device> [ ,...n ]]
 [WITH
   {
    [RECOVERY | NORECOVERY | STANDBY =
      {standby_file_name | @standby_file_name_var }
    ]
   | ,  <general_WITH_options> [ ,...n ]
   | , <replication_WITH_option>
   | , <change_data_capture_WITH_option>
   | , <service_broker_WITH options>
   | , <point_in_time_WITH_options—RESTORE_DATABASE>
   } [ ,...n ]
```

Ví dụ lệnh RESTORE

* Phục hồi trọn vẹn một CSDL:

```
RESTORE DATABASE AdventureWorks2008R2
FROM AdventureWorks2008R2Backups
```

* Phục hồi từ differential backup:

```
RESTORE DATABASE AdventureWorks2008R2
FROM DISK = 'Z:\Backups\AdventureWorks2008R2.bak'
WITH FILE = 6
NORECOVERY;

RESTORE DATABASE AdventureWorks2008R2
FROM DISK = 'Z:\Backups\AdventureWorks2008R2.bak'
WITH FILE = 9
RECOVERY;
```

Sixth backup set on the device

Ninth backup set on the device

Tóm tắt nội dung

* Các khái niệm khôi phục dữ liệu
* Các kỹ thuật khôi phục dữ liệu dựa vào sự cập nhật trì hoãn
* Các kỹ thuật khôi phục dữ liệu dựa vào sự cập nhật tức thời
* Shadow paging
* Giải thuật khôi phục ARIES
* Sao chép dự phòng
* Sao lưu và khôi phục trong SQL-Server