

## Phần 4 :Tự Luận

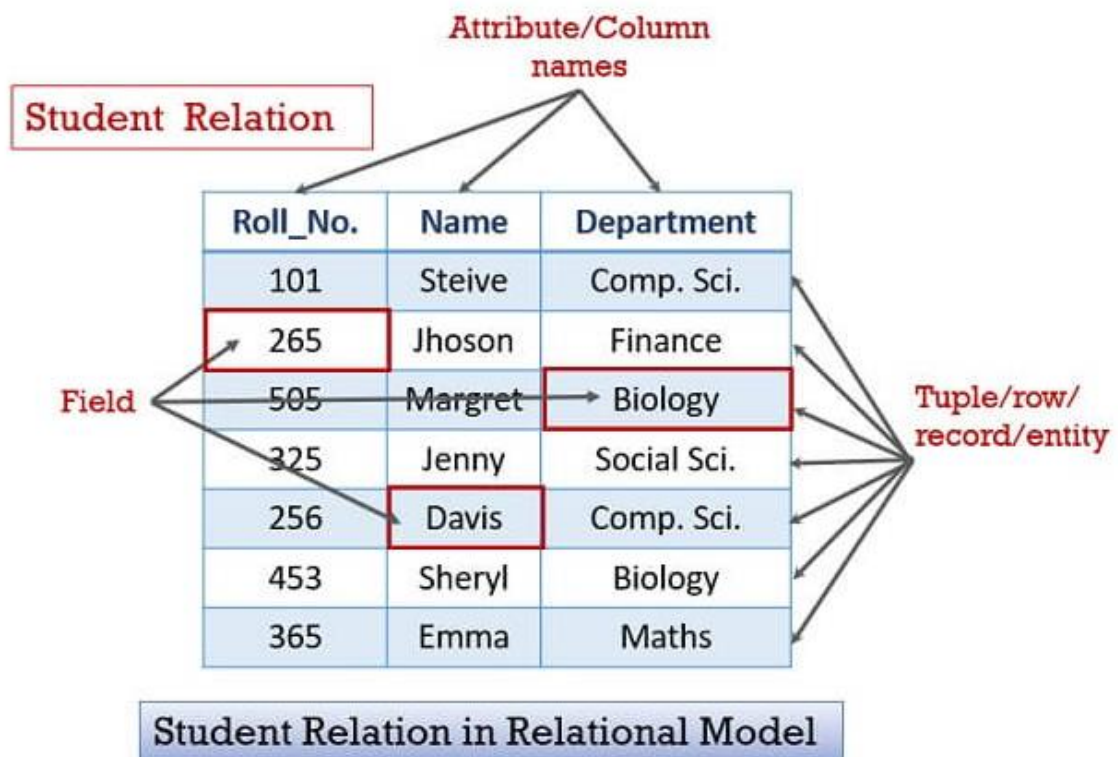
### 1. Các mô hình dữ liệu:

#### - Mô hình quan hệ (Relational Model):

+ Mô tả: Tổ chức dữ liệu thành các bảng có hàng và cột, mỗi bảng đại diện cho một thực thể. Đây là mô hình dữ liệu phổ biến nhất, áp dụng cho các hệ thống quản lý dữ liệu như hệ thống tài chính, nhân sự, quản lý khách hàng (CRM).

+ Ứng dụng: Phổ biến trong các hệ thống quản lý cơ sở dữ liệu như MySQL, PostgreSQL, và SQL Server. Sử dụng trong các hệ thống quản lý nhân sự, quản lý hàng hóa, hệ thống thanh toán.

+ Ví dụ ứng dụng: Trong hệ thống quản lý bán hàng, hệ thống có thể có các bảng “Sản phẩm”, “Khách hàng”, và “Đơn hàng” để lưu thông tin chi tiết từng đơn hàng của khách. Dữ liệu giữa các bảng được liên kết qua khoá ngoại để dễ dàng truy xuất và cập nhật khi cần.



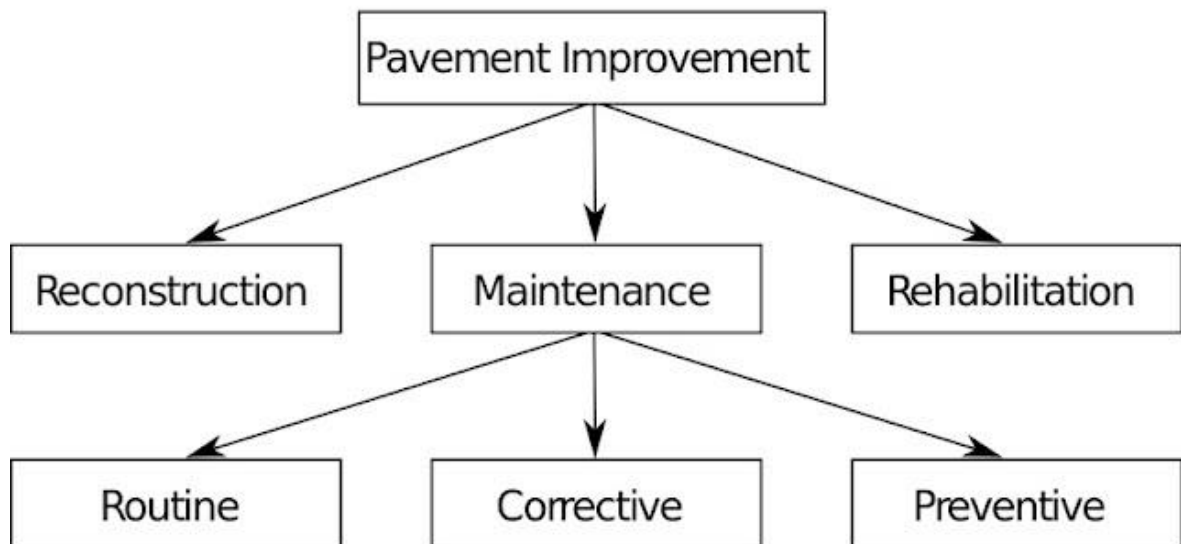
#### - Mô hình phân cấp (Hierarchical Model):

+ Mô tả: Cấu trúc dạng cây, trong đó mỗi node cha có thể có nhiều node con, nhưng mỗi node con chỉ có một node cha. Thường sử dụng cho các hệ thống thông tin phân cấp như hệ thống quản lý tài nguyên của tổ chức lớn.

+ Ứng dụng: Sử dụng trong các hệ thống quản lý thông tin cá nhân, hệ thống quản lý tệp tin.

+ Ví dụ ứng dụng: Trong hệ thống quản lý tài nguyên công ty, ta có thể tổ chức dữ liệu theo cấu trúc phân cấp: Công ty > Bộ phận > Phòng ban > Nhân viên. Điều này giúp quản lý chi tiết thông tin nhân viên theo từng cấp bậc.

## Hierarchical Model

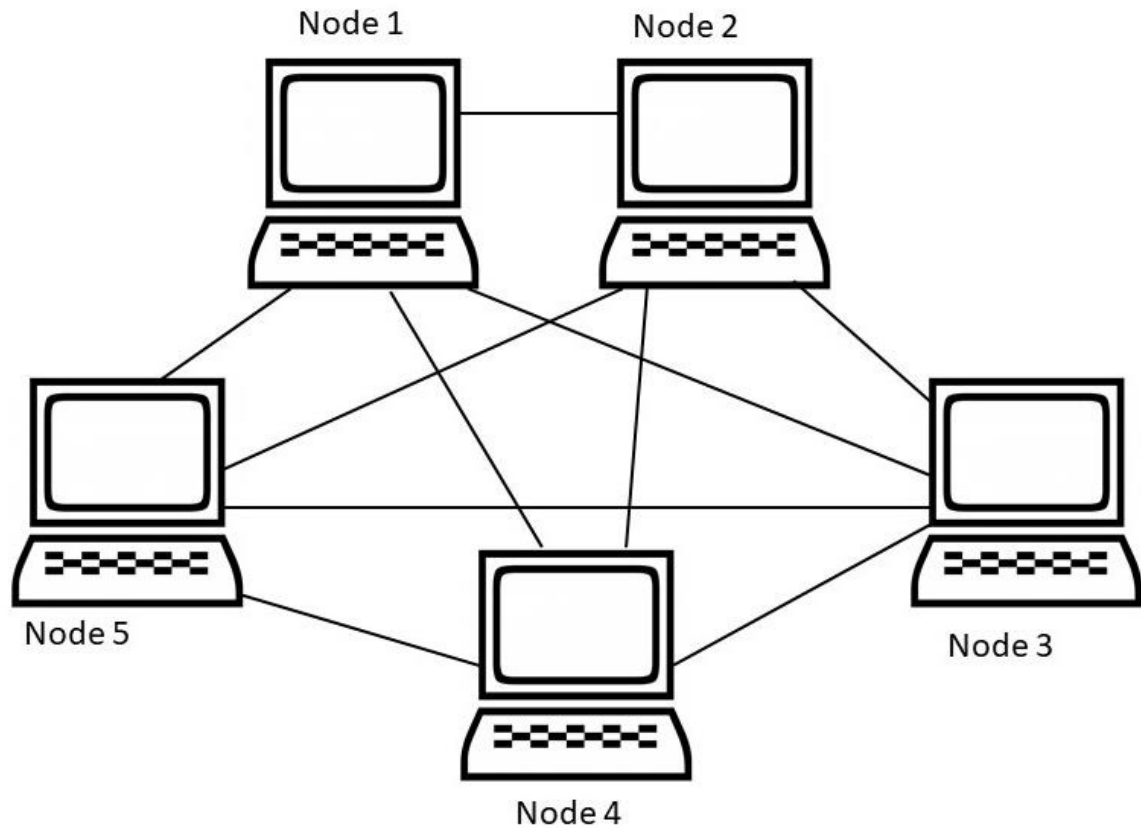


- Mô hình mạng (Network Model):

+ Mô tả: Tương tự như mô hình phân cấp nhưng mỗi node con có thể có nhiều node cha. Thích hợp với các ứng dụng như hệ thống quản lý mối quan hệ phức tạp trong công nghiệp.

+ Ứng dụng: Dùng trong các ứng dụng phức tạp với nhiều mối quan hệ như hệ thống quản lý sản xuất và quản lý vận tải.

+ Ví dụ ứng dụng: Trong hệ thống quản lý sản xuất, mỗi sản phẩm có thể có nhiều nhà cung cấp và mỗi nhà cung cấp cũng có thể cung cấp nhiều sản phẩm khác nhau. Các bảng dữ liệu “Nhà cung cấp” và “Sản phẩm” được liên kết với nhau qua các node trung gian.

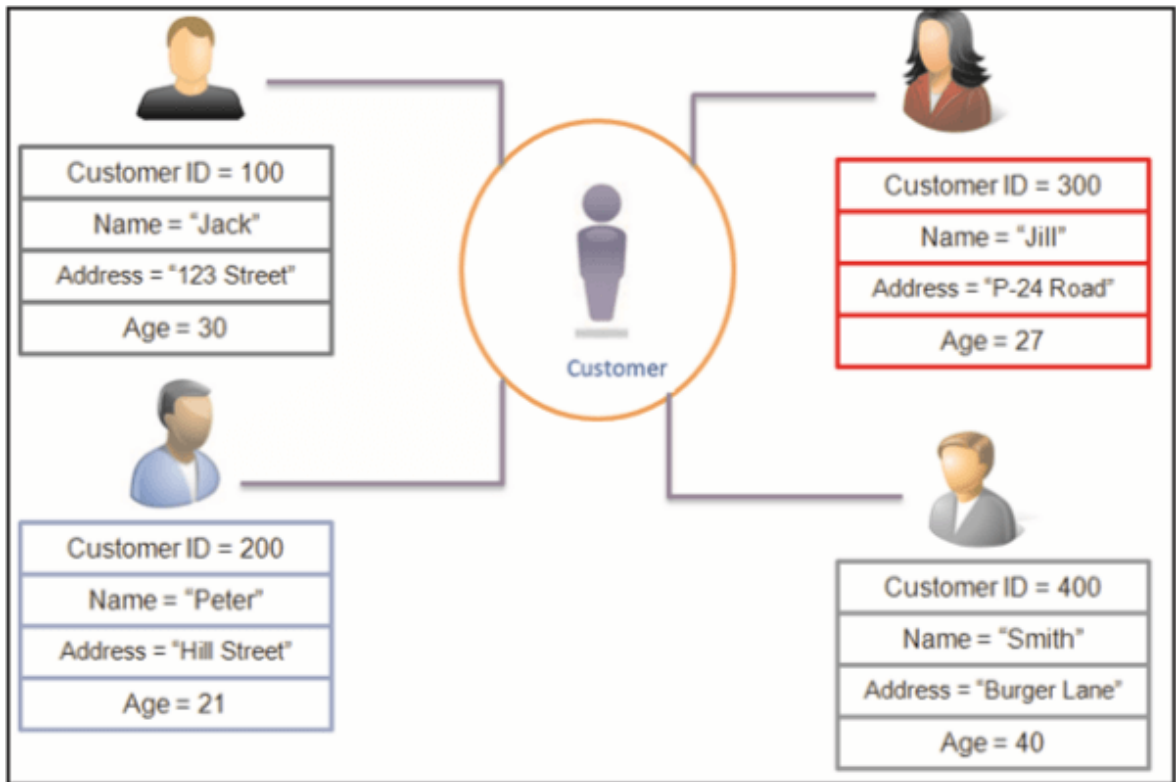


- Mô hình đối tượng (Object-Oriented Model):

+ Mô tả: Dữ liệu được lưu dưới dạng các đối tượng, tương tự như trong lập trình hướng đối tượng. Được sử dụng trong các hệ thống xử lý đa phương tiện hoặc các ứng dụng phức tạp.

+ Ứng dụng: Dùng trong các hệ thống phần mềm phức tạp như hệ thống CAD/CAM, hệ thống quản lý dự án.

+ Ví dụ ứng dụng: Trong hệ thống CAD/CAM, các bản vẽ thiết kế thường chứa các đối tượng như đường, vòng cung, và hình học khác. Mỗi đối tượng có thuộc tính riêng và được liên kết với nhau theo cách thức linh hoạt, phù hợp cho mô hình dữ liệu hướng đối tượng.

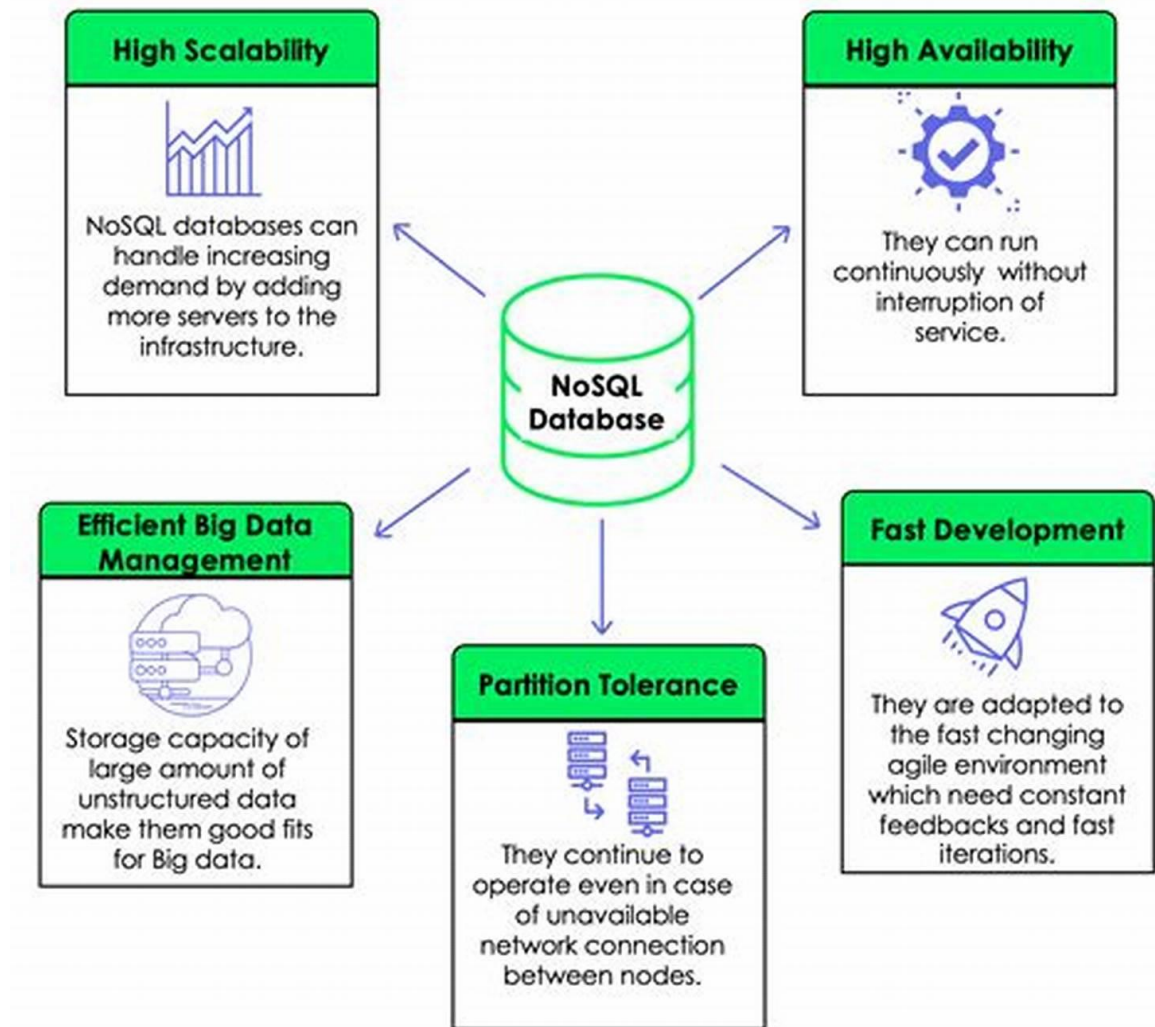


- Mô hình tài liệu (Document Model - NoSQL):

+ Mô tả: Lưu dữ liệu trong các tài liệu JSON hoặc XML. Mỗi tài liệu có thể chứa cấu trúc dữ liệu phức tạp và khác nhau.

+ Ứng dụng: Thường được áp dụng trong các ứng dụng web và các hệ thống cần linh hoạt về cấu trúc dữ liệu như MongoDB.

+ Ví dụ ứng dụng: Một ứng dụng thương mại điện tử có thể sử dụng MongoDB để lưu thông tin sản phẩm. Mỗi tài liệu sẽ lưu trữ thông tin một sản phẩm, bao gồm các thuộc tính như tên, giá, mô tả, ảnh và đánh giá. Các sản phẩm có thể có số lượng thuộc tính khác nhau, linh hoạt và dễ dàng mở rộng.



## 2. Phân tích sự khác biệt giữa SQL và NoSQL:

- Khái niệm:

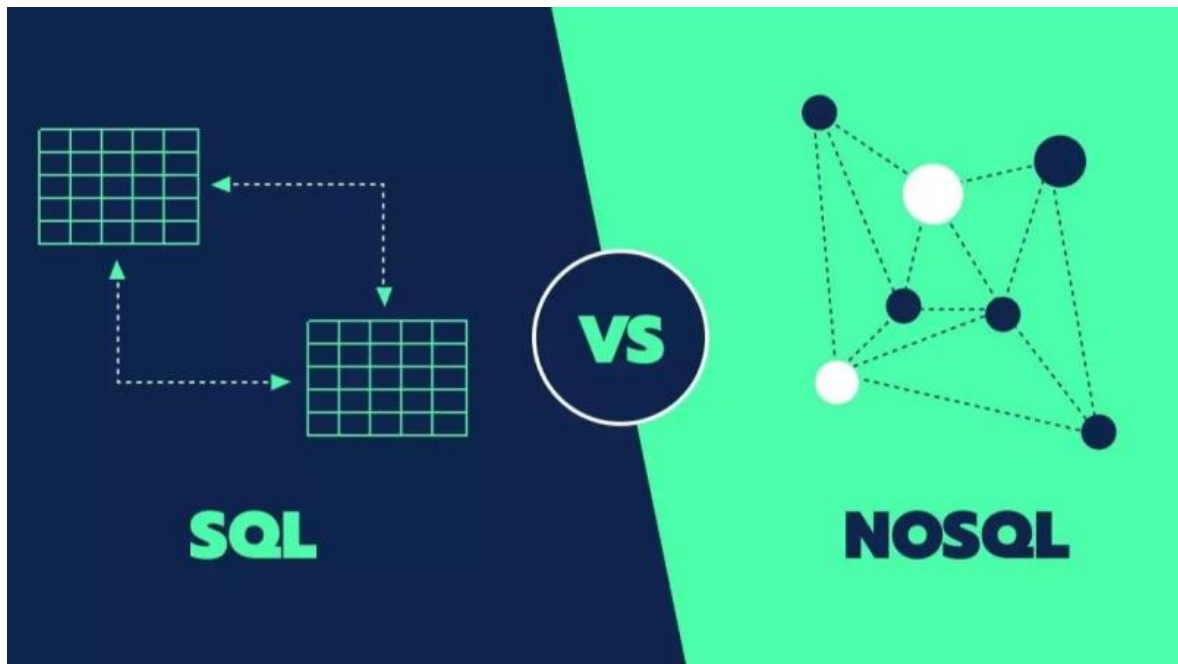
+ SQL: Dùng cho cơ sở dữ liệu quan hệ, dữ liệu được tổ chức trong các bảng với cấu trúc chặt chẽ. Thích hợp cho các hệ thống yêu cầu tuân thủ ACID (Atomicity, Consistency, Isolation, Durability) và dữ liệu có cấu trúc rõ ràng. Ví dụ: ngân hàng, ứng dụng kế toán.

- NoSQL: Dùng cho các cơ sở dữ liệu không quan hệ, lưu dữ liệu dưới dạng tài liệu, key-value, đồ thị hoặc cột. Phù hợp cho dữ liệu không có cấu trúc hoặc cấu trúc thay đổi nhanh, yêu cầu mở rộng ngang, như trong các ứng dụng mạng xã hội, phân tích big data.

- Khi nào sử dụng:

+SQL: Khi cần tính nhất quán cao, dữ liệu có cấu trúc và không thay đổi thường xuyên.

+ NoSQL: Khi cần mở rộng nhanh, dữ liệu thay đổi linh hoạt, và không yêu cầu tính nhất quán tức thời.



Câu 3 : Bạn hiểu thế nào về khái niệm "Chuẩn hóa dữ liệu"? Hãy mô tả các bước cơ bản trong quá trình chuẩn hóa?

Khái niệm chuẩn hóa dữ liệu :

Chuẩn hóa cơ sở dữ liệu là một phương pháp khoa học để phân tách (scientific method of breaking down) một bảng có cấu trúc phức tạp (complex table structures) thành những bảng có cấu trúc đơn giản (simple table structures) theo những quy luật đảm bảo (certain rule) không làm mất thông tin dữ liệu. Kết quả là sẽ làm giảm bớt sự dư thừa và loại bỏ những sự cố mâu thuẫn về dữ liệu, tiết kiệm được không gian lưu trữ. Một số dạng chuẩn hóa dữ liệu thông dụng là:

Dạng chuẩn thứ nhất (First Normal Form - 1NF)

Dạng chuẩn thứ hai (Second Normal Form - 2NF)

Dạng chuẩn thứ ba (Third Normal Form - 3NF)

Dạng chuẩn Boyce-Codd (Boyce-Codd Normal Form - BCNF)

Chuẩn hóa dữ liệu còn có ý nghĩa khác ngoài việc lưu trữ dữ liệu trên các máy tính. Đối với các văn bản, việc chuẩn hóa dữ liệu có thể làm cho văn bản trở nên dễ đọc hơn không vướng vào những trường hợp về hiển thị.

\* Ví dụ: Tồn tại những dòng mà chỉ có 1 dấu. (dấu chấm) do có khoảng trắng trước dấu chấm cuối cùng. Chuẩn hóa dữ liệu khiến cho máy vi tính có thể hiểu được ý nghĩa trong câu nếu những từ ngữ được viết đúng chuẩn.



Các bước cơ bản trong quá trình chuẩn hóa:

Chuẩn hóa là một kỹ thuật thiết kế cơ sở dữ liệu, được sử dụng để thiết kế một bảng cơ sở dữ liệu quan hệ lên đến dạng chuẩn hóa cao hơn. Quá trình này diễn ra theo tiến trình và không thể đạt được mức chuẩn hóa cơ sở dữ liệu cao hơn trừ khi các mức trước đó đã được thỏa mãn.

Điều đó có nghĩa là, khi có dữ liệu ở dạng không chuẩn hóa (ít chuẩn hóa nhất) và muốn đạt được mức chuẩn hóa cao nhất, bước đầu tiên sẽ là đảm bảo tuân thủ dạng chuẩn hóa thứ nhất, bước thứ hai sẽ là đảm bảo thỏa mãn dạng chuẩn hóa thứ hai, và cứ tiếp tục như vậy theo thứ tự đã đề cập ở trên, cho đến khi dữ liệu tuân thủ dạng chuẩn hóa thứ sáu.

Tuy nhiên, điều đáng chú ý là các dạng chuẩn vượt quá 4NF chủ yếu mang tính học thuật, vì các vấn đề chúng tồn tại để giải quyết hiếm khi xuất hiện trong thực tế. <sup>[11]</sup>

Dữ liệu trong ví dụ sau đây được thiết kế có chủ đích để mâu thuẫn với hầu hết các dạng chuẩn. Trong thực tế, thường có thể bỏ qua một số bước chuẩn hóa vì dữ liệu đã được chuẩn hóa ở một mức độ nào đó. Việc sửa lỗi vi phạm một dạng chuẩn hóa cũng thường sửa lỗi vi phạm một dạng chuẩn hóa cao hơn. Trong ví dụ, một bảng đã được chọn để chuẩn hóa ở mỗi bước, nghĩa là ở cuối, một số bảng có thể không được chuẩn hóa đủ.

Ví dụ: Cho một bảng cơ sở dữ liệu tồn tại với cấu trúc sau:



Tiêu đề	Tác giả	Quốc tịch của tác giả	Định dạng	Giá	Chủ thể	Trang	Độ dày	Nhà xuất bản	Quốc gia xuất bản	ID thể loại	Tên thể loại
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Chad Russell	người Mỹ	Bìa cứng	49,99	MySQL	520	Dày	Bảo chí	Hoa Kỳ	1	Hướng dẫn
					Cơ sở dữ liệu						
					Thiết kế						

Trong ví dụ này, giả định rằng mỗi cuốn sách chỉ có một tác giả.

Một bảng tuân thủ mô hình quan hệ có khóa chính xác định duy nhất một hàng. Trong ví dụ của chúng tôi, khóa chính là khóa tổng hợp của **{Title, Format}** (được chỉ ra bằng gạch chân):

<u>Tiêu đề</u>	Tác giả	Quốc tịch của tác giả	<u>Định dạng</u>	Giá	Chủ thể	Trang	Độ dày	Nhà xuất bản	Quốc gia xuất bản	ID thể loại	Tên thể loại
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Chad Russell	người Mỹ	Bìa cứng	49,99	MySQL	520	Dày	Bảo chí	Hoa Kỳ	1	Hướng dẫn
					Cơ sở dữ liệu						
					Thiết kế						

**Thỏa mãn 1NF:**

Trong dạng chuẩn đầu tiên, mỗi trường chứa một giá trị duy nhất. Một trường không được chứa một tập hợp các giá trị hoặc một bản ghi lồng nhau.

**Chủ đề** chứa một tập hợp các giá trị chủ đề, nghĩa là nó không tuân thủ.

Để giải quyết vấn đề, các chủ đề được trích xuất vào một bảng **Chủ đề** riêng biệt :



<u>Tiêu đề</u>	Tác giả	Quốc tịch của tác giả	<u>Định dạng</u>	Giá	Trang	Độ dày	Nhà xuất bản	Quốc gia xuất bản	ID thể loại	Tên thể loại
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Chad Russell	người Mỹ	Bìa cứng	49,99	520	Dày	Báo chí	Hoa Kỳ	1	Hướng dẫn

Tiêu đề - Chủ đề

<u>Tiêu đề</u>	<u>Tên chủ đề</u>
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	MySQL
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Cơ sở dữ liệu
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Thiết kế

**Thỏa mãn 2NF:** Đảm bảo bảng đáp ứng 1NF và mỗi cột không khóa phụ thuộc hoàn toàn vào khóa chính.

Hãy nhớ rằng bảng **Book** bên dưới có khóa tổng hợp là **{Title, Format}** , khóa này sẽ không thỏa mãn 2NF nếu một số tập con của khóa đó là một định thức. Tại thời điểm này trong thiết kế của chúng ta, khóa **chưa** được hoàn thiện là khóa chính , do đó nó được gọi là khóa ứng viên . Hãy xem xét bảng sau:

<u>Tiêu đề</u>	<u>Định dạng</u>	Tác giả	Quốc tịch của tác giả	Giá	Trang	Độ dày	Nhà xuất bản	Quốc gia xuất bản	ID thể loại	Tên thể loại
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Bìa cứng	Chad Russell	người Mỹ	49,99	520	Dày	Báo chí	Hoa Kỳ	1	Hướng dẫn
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Sách điện tử	Chad Russell	người Mỹ	22,34	520	Dày	Báo chí	Hoa Kỳ	1	Hướng dẫn
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	Sách điện tử	EFCodd	Anh	13,88	538	Dày	Addison-Wesley	Hoa Kỳ	2	Khoa học phổ thông
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	Bìa mềm	EFCodd	Anh	39,99	538	Dày	Addison-Wesley	Hoa Kỳ	2	Khoa học phổ thông

Tất cả các thuộc tính không phải là một phần của khóa ứng viên đều phụ thuộc vào *Title* , nhưng chỉ có *Price* cũng phụ thuộc vào *Format* . Để tuân thủ 2NF và loại bỏ các bản sao, mọi thuộc tính không phải khóa ứng viên phải phụ thuộc vào toàn bộ khóa ứng viên, không chỉ một phần của khóa đó.

Để chuẩn hóa bảng này, hãy biến **{Title}** thành một khóa ứng viên (đơn giản) (khóa chính) sao cho mọi thuộc tính không phải khóa ứng viên đều phụ thuộc vào toàn bộ

khóa ứng viên và xóa *Price* vào một bảng riêng để có thể duy trì sự phụ thuộc của nó vào *Format*

Sách

<u>Tiêu đề</u>	Tác giả	Quốc tịch của tác giả	Trang	Độ dày	Nhà xuất bản	Quốc gia xuất bản	ID thể loại	Tên thể loại
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Chad Russell	người Mỹ	520	Dày	Báo chí	Hoa Kỳ	1	Hướng dẫn
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	EFCodd	Anh	538	Dày	Addison-Wesley	Hoa Kỳ	2	Khoa học phổ thông

Giá

<u>Tiêu đề</u>	<u>Định dạng</u>	Giá
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Bìa cứng	49,99
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Sách điện tử	22,34
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	Sách điện tử	13,88
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	Bìa mềm	39,99

**Thỏa mãn 3NF:** Đảm bảo bảng đáp ứng 2NF và không có phụ thuộc bắc cầu giữa các cột không khóa.

Bảng **Book** vẫn có một phụ thuộc hàm transitive ( $\{ \text{Author Nationality} \}$  phụ thuộc vào  $\{ \text{Author} \}$ , phụ thuộc vào  $\{ \text{Title} \}$ ). Các vi phạm tương tự cũng tồn tại đối với publisher ( $\{ \text{Publisher Country} \}$  phụ thuộc vào  $\{ \text{Publisher} \}$ , phụ thuộc vào  $\{ \text{Title} \}$ ) và đối với genre ( $\{ \text{Genre Name} \}$  phụ thuộc vào  $\{ \text{Genre ID} \}$ , phụ thuộc vào  $\{ \text{Title} \}$ ). Do đó, bảng **Book** không ở dạng 3NF. Để giải quyết vấn đề này, chúng ta có thể đặt  $\{ \text{Author Nationality} \}$ ,  $\{ \text{Publisher Country} \}$  và  $\{ \text{Genre Name} \}$  vào các bảng tương ứng của chúng, do đó loại bỏ các phụ thuộc hàm transitive:

#### Sách

<u>Tiêu đề</u>	<u>Tác giả</u>	<u>Trang</u>	<u>Độ dày</u>	<u>Nhà xuất bản</u>	<u>ID thẻ loại</u>
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Chad Russell	520	Dày	Báo chí	1
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	EFCodd	538	Dày	Addison-Wesley	2

#### Giá

<u>Tiêu đề</u>	<u>Định dạng</u>	<u>Giá</u>
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Bìa cứng	49,99
Bắt đầu thiết kế và tối ưu hóa cơ sở dữ liệu MySQL	Sách điện tử	22,34
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	Sách điện tử	13,88
Mô hình quan hệ cho quản lý cơ sở dữ liệu: Phiên bản 2	Bìa mềm	39,99

5. Bạn có thể nêu ra một số phương pháp sao lưu và phục hồi dữ liệu? Hãy giải thích tầm quan trọng của việc này trong quản trị cơ sở dữ liệu

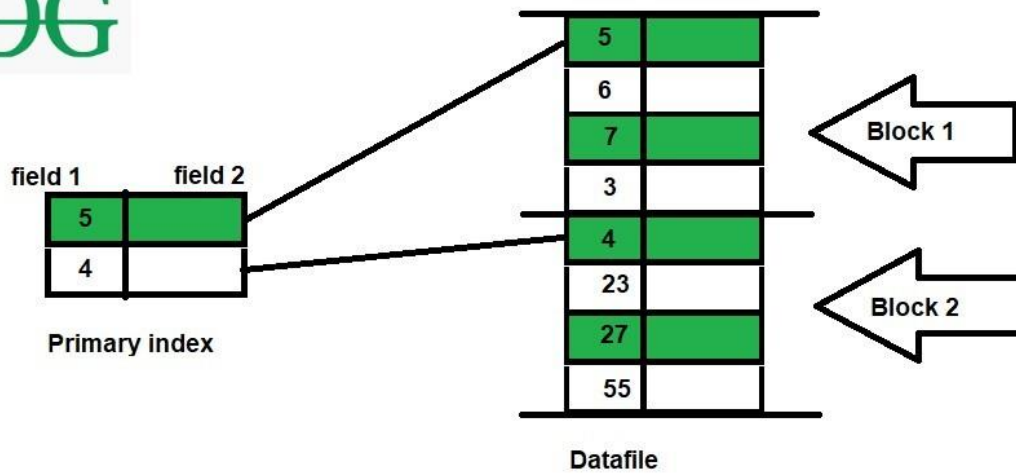
Khái niệm:

Sao lưu và phục hồi dữ liệu là quá trình bảo vệ và khôi phục dữ liệu quan trọng trong trường hợp xảy ra sự cố như lỗi hệ thống, tấn công mạng hoặc thiên tai. Việc này có vai trò quan trọng trong quản trị cơ sở dữ liệu vì nó đảm bảo tính liên tục, an toàn và khả dụng của dữ liệu. Dưới đây là một số phương pháp sao lưu và phục hồi phổ biến:

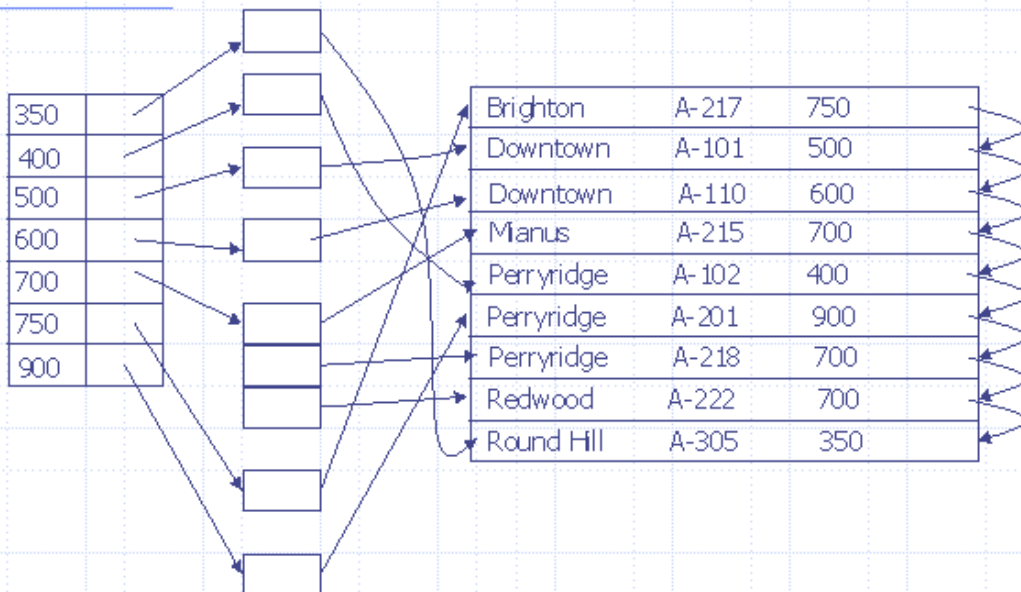
#### 4. Giải thích các loại chỉ mục trong Cơ sở dữ liệu và lợi ích của việc sử dụng chúng trong truy vấn dữ liệu.

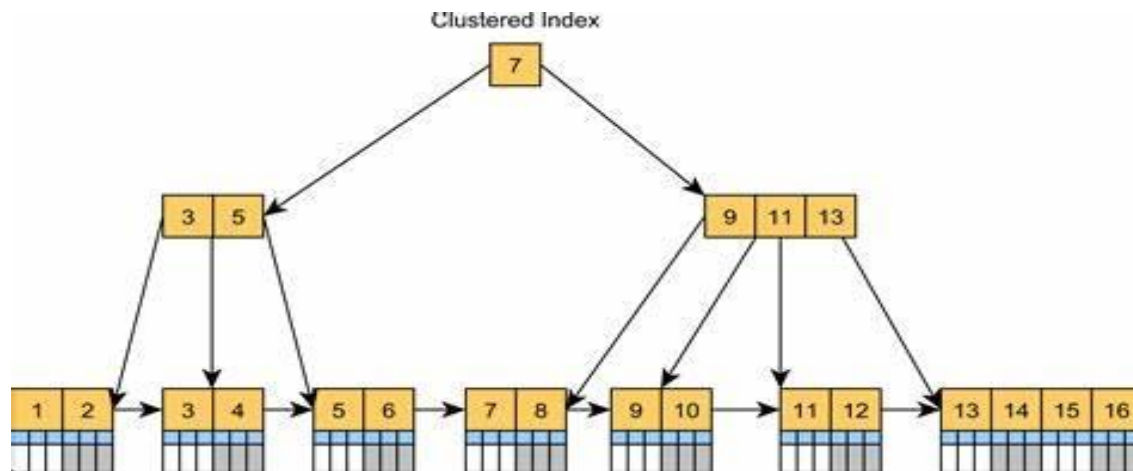
- Các loại chỉ mục trong Cơ sở dữ liệu và lợi ích:

- + Primary Index: Chỉ mục được tạo dựa trên khóa chính của bảng.
  - + Secondary Index: Chỉ mục được tạo dựa trên các cột không phải là khóa chính, giúp tăng tốc độ truy vấn không dựa vào khóa chính.
  - + Clustered Index: Lưu trữ dữ liệu vật lý theo thứ tự của chỉ mục.
  - + Non-Clustered Index: Chỉ mục chứa tham chiếu đến dữ liệu, không thay đổi thứ tự vật lý của dữ liệu.
- Lợi ích: Chỉ mục giúp tăng tốc độ truy vấn bằng cách giảm thiểu số lượng bản ghi cần duyệt, đặc biệt hữu ích khi truy vấn dữ liệu từ các bảng lớn.



## Secondary Index





Key	Record Locator
1	-
2	-
3	-
4	-
5	-
6	-
7	-
8	-

Nonclustered Index

EmployeeID	PassportNumber	<u>ExpiryDate</u>
3	A5423215	NULL
5	A5423215	NULL
2	A5423215	NULL
8	A5423215	NULL
1	A5423215	NULL
4	A5423215	NULL
6	A5423215	NULL
7	A5423215	NULL

Table

5. Bạn có thể nêu ra một số phương pháp sao lưu và phục hồi dữ liệu? Hãy giải thích tầm quan trọng của việc này trong quản trị cơ sở dữ liệu.

**Một số phương pháp sao lưu và phục hồi dữ liệu:**

### 1. Sao lưu đầy đủ (Full Backup)

- **Phương pháp:** Sao lưu toàn bộ dữ liệu của hệ thống tại thời điểm sao lưu.
- **Ưu điểm:** Dễ dàng phục hồi toàn bộ dữ liệu khi có sự cố xảy ra.
- **Nhược điểm:** Tốn nhiều thời gian và dung lượng lưu trữ, thường được thực hiện định kỳ (hàng tuần, hàng tháng).
- **Ứng dụng:** Phù hợp cho dữ liệu ít thay đổi hoặc làm cơ sở để tạo các bản sao lưu khác.

- **Ví dụ:** Một công ty kế toán thực hiện sao lưu đầy đủ vào cuối tuần để lưu trữ toàn bộ dữ liệu khách hàng, báo cáo tài chính và hồ sơ. Bản sao lưu đầy đủ này được lưu trữ trên một ổ cứng ngoài và đặt tại kho dữ liệu. Nếu có sự cố xảy ra, họ có thể khôi phục toàn bộ dữ liệu từ bản sao lưu cuối tuần trước.

## 2. Sao lưu gia tăng (Incremental Backup)

- **Phương pháp:** Sao lưu dữ liệu thay đổi kể từ lần sao lưu gần nhất (dù là sao lưu đầy đủ hay sao lưu gia tăng).
- **Ưu điểm:** Tiết kiệm thời gian và dung lượng lưu trữ, thường được thực hiện hàng ngày hoặc thậm chí hàng giờ.
- **Nhược điểm:** Phục hồi dữ liệu có thể mất thời gian hơn, vì cần kết hợp nhiều bản sao lưu để phục hồi đầy đủ.
- **Ứng dụng:** Hữu ích trong các hệ thống thường xuyên thay đổi dữ liệu, như trang thương mại điện tử.
- **Ví dụ:** Một công ty thương mại điện tử thực hiện sao lưu gia tăng hàng ngày sau khi có một bản sao lưu đầy đủ vào chủ nhật. Mỗi ngày, chỉ các thay đổi kể từ lần sao lưu gần nhất mới được sao lưu, giảm thiểu thời gian và dung lượng. Khi cần khôi phục, hệ thống sẽ sử dụng bản sao lưu đầy đủ từ chủ nhật và lần lượt các bản sao lưu gia tăng trong tuần.

## 3. Sao lưu vi sai (Differential Backup)

- **Phương pháp:** Sao lưu dữ liệu thay đổi kể từ lần sao lưu đầy đủ gần nhất.
- **Ưu điểm:** Dễ dàng và nhanh chóng phục hồi vì chỉ cần bản sao lưu đầy đủ và một bản sao lưu vi sai mới nhất.
- **Nhược điểm:** Tốn nhiều dung lượng hơn so với sao lưu gia tăng vì lượng dữ liệu tăng dần.
- **Ứng dụng:** Tối ưu cho những hệ thống cần phục hồi nhanh chóng mà không tốn quá nhiều không gian.
- **Ví dụ:** Một bệnh viện thực hiện sao lưu đầy đủ toàn bộ dữ liệu bệnh án vào ngày đầu tiên của tháng. Sau đó, họ thực hiện sao lưu vi sai hàng tuần, chỉ sao lưu dữ liệu đã thay đổi kể từ bản sao lưu đầy đủ. Nếu cần khôi phục vào cuối tháng, bệnh viện sẽ dùng bản sao lưu đầy đủ và bản sao lưu vi sai mới nhất, giảm thiểu thời gian khôi phục so với sao lưu gia tăng.

## 4. Sao lưu bản chụp (Snapshot Backup)

- **Phương pháp:** Tạo một bản chụp nhanh dữ liệu tại một thời điểm cụ thể, thường là ở mức phân cứng hoặc hệ thống.
- **Ưu điểm:** Nhanh chóng và ít ảnh hưởng đến hiệu suất hệ thống.

- **Nhược điểm:** Chủ yếu lưu trữ các trạng thái ngắn hạn, không phải là giải pháp lâu dài.
- **Ứng dụng:** Thích hợp cho các môi trường ảo hóa và hệ thống lưu trữ dựa trên đám mây.
- **Ví dụ:** Một ngân hàng tạo các bản chụp nhanh dữ liệu hàng giờ trong hệ thống quản lý tài khoản để bảo vệ dữ liệu giao dịch. Nếu có sự cố, như lỗi hệ thống hay giao dịch không chính xác, ngân hàng có thể khôi phục nhanh chóng từ một bản chụp gần đây.

## 5. Sao lưu trực tuyến và ngoại tuyến (Online & Offline Backup)

- **Online Backup:** Sao lưu trong khi hệ thống vẫn hoạt động, đảm bảo tính sẵn sàng của dịch vụ. Tuy nhiên, có thể ảnh hưởng đến hiệu suất.
- **Offline Backup:** Sao lưu khi hệ thống không hoạt động, giảm thiểu rủi ro hỏng dữ liệu nhưng làm gián đoạn dịch vụ.
- **Ứng dụng:** Tùy vào yêu cầu của hệ thống, như dịch vụ liên tục (online) hoặc khi bảo trì định kỳ (offline).

### Ví dụ:

- **Online Backup:** Một công ty công nghệ lưu trữ dữ liệu trực tiếp trên nền tảng đám mây như AWS hoặc Google Cloud, giúp họ truy cập và sao lưu liên tục mà không gián đoạn dịch vụ.
- **Offline Backup:** Trong khi đó, một văn phòng luật có thể thực hiện sao lưu ngoại tuyến vào mỗi cuối ngày làm việc để lưu trữ các tài liệu mật, bảo vệ dữ liệu khỏi tấn công trực tuyến.

## 6. Sao lưu tại chỗ và sao lưu từ xa (Onsite & Offsite Backup)

- **Onsite Backup:** Sao lưu dữ liệu tại vị trí vật lý của hệ thống, giúp truy cập nhanh chóng khi cần phục hồi.
- **Offsite Backup:** Sao lưu dữ liệu ở một vị trí từ xa (có thể là một trung tâm dữ liệu khác hoặc lưu trữ đám mây), đảm bảo an toàn trong trường hợp xảy ra thiên tai hoặc tấn công vật lý.
- **Ứng dụng:** Thường kết hợp cả hai loại để đảm bảo an toàn và tính sẵn sàng cao nhất.

### Ví dụ:



- **Onsite Backup:** Một doanh nghiệp nhỏ lưu trữ bản sao lưu trên một máy chủ nội bộ đặt tại văn phòng để có thể nhanh chóng phục hồi trong trường hợp lỗi nhỏ.
- **Offsite Backup:** Doanh nghiệp này cũng lưu bản sao lưu từ xa trên đám mây, đảm bảo rằng ngay cả khi có hỏa hoạn hoặc thiên tai tại văn phòng, dữ liệu vẫn được bảo vệ an toàn ở vị trí khác.
- 

## 7. Sao lưu dữ liệu theo thời gian thực (Real-time Backup)

- **Phương pháp:** Sao lưu ngay lập tức các thay đổi dữ liệu lên hệ thống dự phòng, giúp phục hồi gần như ngay lập tức khi có sự cố.
- **Ưu điểm:** Đảm bảo tính liên tục và an toàn cao, thường được áp dụng cho dữ liệu nhạy cảm.
- **Nhược điểm:** Tốn kém về tài nguyên và chi phí.
- **Ứng dụng:** Hệ thống tài chính, ngân hàng, và các tổ chức yêu cầu dữ liệu phải được bảo mật và phục hồi ngay lập tức.
- **Ví dụ:** Một công ty tài chính yêu cầu sao lưu dữ liệu giao dịch theo thời gian thực. Dữ liệu được đồng bộ ngay lập tức lên một máy chủ dự phòng. Khi máy chủ chính gặp sự cố, dữ liệu vẫn được bảo toàn trên máy chủ dự phòng, đảm bảo không có giao dịch nào bị mất.

## Tầm Quan Trọng Của Sao Lưu và Phục Hồi Dữ Liệu trong Quản Trị Cơ Sở Dữ Liệu

1. **Bảo vệ dữ liệu quan trọng:** Sao lưu giúp bảo vệ các dữ liệu quan trọng khỏi mất mát do lỗi phần cứng, sự cố hệ thống, hoặc tấn công mạng.
2. **Đảm bảo tính liên tục của hệ thống:** Phục hồi dữ liệu kịp thời cho phép các tổ chức duy trì hoạt động mà không bị gián đoạn, giúp đảm bảo dịch vụ và tránh tổn thất kinh tế.
3. **Phục hồi nhanh chóng sau sự cố:** Khi xảy ra sự cố, có thể phục hồi dữ liệu ngay lập tức hoặc trong thời gian ngắn, giảm thiểu ảnh hưởng đến hoạt động của doanh nghiệp.
4. **Tuân thủ quy định và luật pháp:** Trong nhiều ngành công nghiệp, các quy định pháp lý yêu cầu phải sao lưu dữ liệu và duy trì khả năng phục hồi để đảm bảo an toàn thông tin.
5. **Bảo vệ khỏi các rủi ro an ninh mạng:** Sao lưu và phục hồi giúp đối phó với các tấn công ransomware, khi hacker mã hóa dữ liệu và yêu cầu tiền chuộc, nhờ có dữ liệu sao lưu, có thể khôi phục hệ thống mà không cần trả tiền chuộc.

