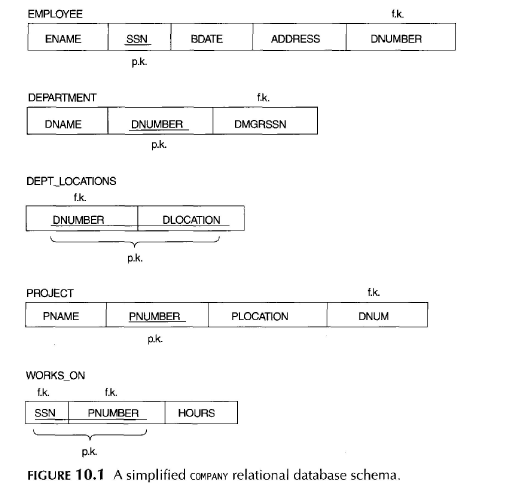
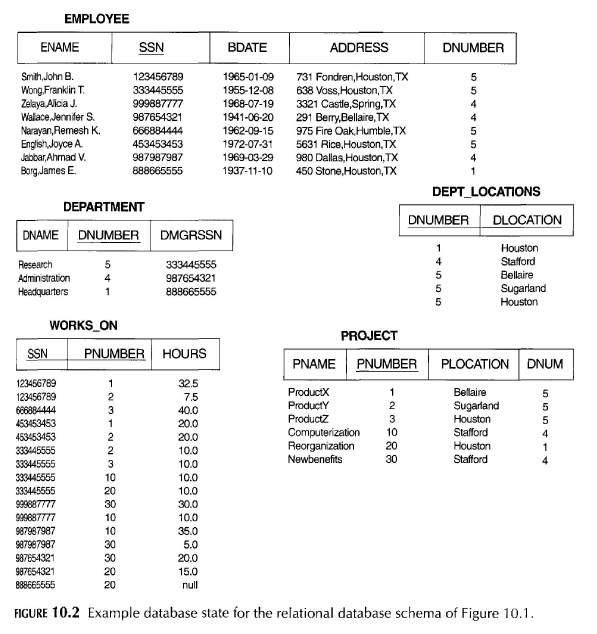
**TOPIC: CÁC HƯỚNG DẪN KHÔNG CHÍNH THỨC VỀ THIẾT KẾ LƯỢC ĐỒ**

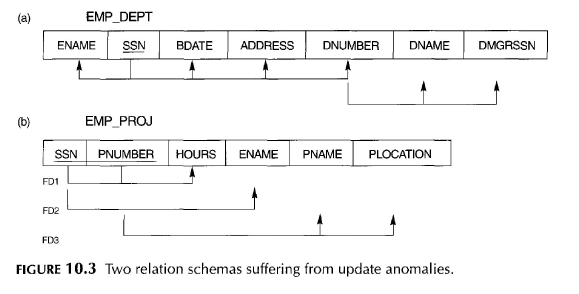
**Những vẫn đề sẽ thảo luận trong topic này là:**

* Ngữ nghĩa của thuộc tính.
* Giảm các giá trị dư thừa trong bộ dữ liệu.
* Giảm giá trị NULL trong bộ dữ liệu.
* Loại bỏ khả năng tạo ra bộ dữ liệu ảo.
* Các biện pháp này không phải lúc nào cũng độc lập từng biện pháp một , mà có khi sẽ hòa lẫn vào nhau.

1. **Ngữ nghĩa của các thuộc tính.**

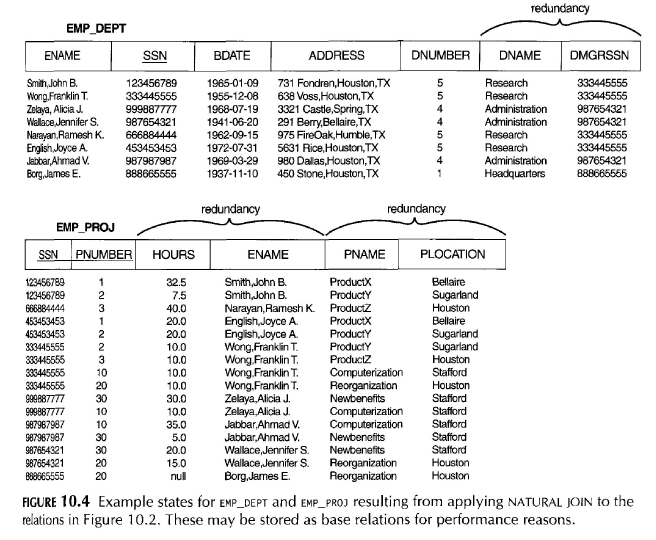
* Bất cứ khi nào khi ta nhóm các thuộc tính để tạo thành một lược đồ quan hệ, ta giả định rằng các thuộc tính trong mối quan hệ có một ý nghĩa nhất định bên ngoài thế giới thực và kết hợp phù hợp với quan hệ đó.
* Ý nghĩa hay ngữ nghĩa, quy định cụ thể như thế nào để giải thích các thuộc tính giá trị được lưu trữ trong một bộ của quan hệ, nói cách khác là làm thế nào mà các giá trị thuộc tính được lưu trữ trong bộ dữ liệu liên quan đến nhau. Nếu các thiết kế khái niệm được thực hiện một cách cẩn thận, sau đó là một sơ đồ hệ thống các quan hệ ngữ nghĩa sẽ được tính toán và kết quả thiết kế nên một ý nghĩa rõ ràng.
* Ví dụ:
* Hình 10.1 miêu tả lược đồ lưu trữ thông tin của COMPANY , và hình 10.2 miêu tả thực trạng dữ liệu. Ý nghĩa của lược đồ quan hệ EMPLOYEE khá đơn giản: Mỗi bộ dữ liệu đại diện cho một nhân viên, với các giá trị của tên nhân viên (ENAME) , số an sinh xã hội (SSN), ngày sinh (BDATE), và mã số của bộ phận làm việc nhân viên (DNUMBER). Thuộc tính (DNUMBER) là khóa ngoại đại diện cho mối quan hệ giữa EMPLOYEE và DEPARTMENT. Ý nghĩa của lược đồ DEPARTMENT và PROJECT cũng đơn giản. Mỗi bộ dữ liệu của DEPARTMENT đại diện cho một thực thể bộ phận và mỗi PROJECT là đại diện cho một thực thể dự án. Thuộc tính DMGRSSN của quan hệ DEPARTMENT thể hiện mối quan hệ mỗi bộ phận có một nhân viên quản lý, trong khi DNUM của PROJECT thể hiện một dự án được bộ phận quản lý dự án đó, cả 2 thuộc tính đều là khóa ngoại. Việc dễ dàng để giải thích ý nghĩa của các thuộc tính của một mối quan hệ là biện pháp không chính thức của việc thiết kế quan hệ tốt.
* Ngữ nghĩa của hai lược đồ liên quan khác trong hình 10.1 thì hơi phức tạp hơn. Mỗi bộ dữ liệu trong DEPT\_LOCATIONS cung cấp cho một mã bộ phận (DNUMBER) và một trong những vị trí của bộ phận. Mỗi bộ dữ liệu trong WORK\_ON có một mã số an sinh xã hội của nhân viên (SSN), mã số dự án của một trong các dự án mà nhân viên tham gia (PNUMBER), và số giờ mỗi tuần người lao động làm việc trên dự án đó (HOUR). Tuy nhiên cả hai lược đồ có một giải thích rõ ràng và dễ hiểu. Lược đồ DEPT\_LOCATIONS đại diện cho một thuộc tính đa giá trị của DEPARTMENT, trong khi WORK\_ON đại diện chộ một quan hệ M:N giữa EMPLOYEE và PROJECT. Do đó, tất cả các lược đồ quan hệ trong hình 10.1 có thể được coi là dễ dàng giải thích và do đó có quan điểm có ngữ nghĩa không rõ ràng. Như vậy chúng ta có các hướng dẫn không chính thức như sau:
  + **HƯỚNG DẪN 1:**  Thiết kế một lược đồ quan hệ để dễ dàng để giải thích ý nghĩa của nó. Không kết hợp các thuộc tính từ nhiều loại thực thể thành một mối quan hệ duy nhất. Bằng trực giác, nếu một lược đồ quan hệ tương ứng với một loại thực thể hoặc các loại mối quan hệ một, sẽ là đơn giản để giải thích ý nghĩa của nó. Nếu không, nếu mối quan hệ tương ứng với một tập hợp của nhiều thực thể và mối quan hệ, sự mơ hồ ngữ nghĩa sẽ cho kết quả và mối quan hệ không thể được giải thích một cách dễ dàng.
* Các lược đồ quan trong hình 10.3a và 10.3b cũng có ngữ nghĩa rõ ràng.(Chúng ta nên bỏ qua các dòng theo quan hệ bây giờ, chúng được sử dụng để minh họa cho ký hiệu phụ thuộc chức năng, thảo luận tại mục 10,2) Một bộ dữ liệu trong lược đồ quan hệEMP\_DEPT hình 10.3a đại diện cho một nhân viên duy nhưng bao gồm thêm thông tin - cụ thể là, tên(DNAME) của bộ phận màcác nhân viên làm việc và số an xã hội (DMGRSSN) của người quản lý bộ phận.Đối với mối EMP\_PROJ của hình 10.3b, mỗi bộ dữ liệu liên quan một nhân viên cho một dự án mà còn bao gồmtên nhân viên (ENAME), tên dự án (PNAME), và địa điểm dự án(PLOCATION).Mặc dù không có gì sai một cách logic với hai mối quan hệ, họ được coi là thiết kế người nghèo bởi vì họ vi phạmHướng dẫn 1 thuộc tính pha từ các tổ chức thế giới khác biệtthực sự EMP\_DEPT hỗn hợp các thuộc của nhân viên và các phòng ban, và EMP\_PROJ hỗn hợp thuộc tính của nhân viên vàcác dự án. Họ có thể được sử dụng như xem, nhưng chúng gây ra vấn đề khi sử dụng như là cơ sở quan hệ, khi chúng tôi thảo luận trong phần sau.





1. **Giảm các giá trị dư thừa trong bộ dữ liệu.**

* Một trong những mục tiêu thiết kế lược đồ là giảm thiểu không gian lưu trữ được sử dụng bởi những quan hệ cơ bản(và do các tập tin tương ứng) . Nhóm các thuộc tính vào lược đồ quan hệ có ảnh hưởng đáng kể không gian lưu trữ.
* Ví dụ: So sánh không gian lưu trữ khi sử dụng bởi 2 bảng: EMPLOYEE & DEPARTMENT trong hình 10.2 với một bảng EMP\_DEPT trong hình 10.4, kết quả việc áp dụng phép kết tự nhiên cho EMPLOYEE và DEPARTMENT. Ở bảng EMP\_DEPT, các giá trị thuộc tính liên quan đến mỗi bộ phận cụ thể (DNUMBER, DNAME, DMGRSSN) bị lặp đi lặp lại cho mỗi nhân viên làm việc ở bộ phận đó. Ngược lại thông tin mỗi bộ phận chỉ xuất hiện một lần duy nhất trong quan hệ DEPARTMENT ở hình 10.2, chỉ có mã số bộ phận (DNUMBER) được lập đi lập lại trong quan hệ EMPLOYEE cho mỗi nhân viên làm việc trong bộ phận đó. Ý kiến tương tự áp dụng đối với quan hệ EMP\_PROJ (Hình 10.4) làm tăng mối quan hệ WORK\_ON với các thuộc tính bổ sung từ EMPLOYEE & PROJECT.



* Một vấn đề ngiêm trọng với việc sử dụng các mối quan hệ trong hình 10.4 như quan hệ cơ bản là vấn đề của bất thường trong cập nhật. Đây có thể phân loại thành chen, xóa, sửa bất thường:
  + **Bất thường trong thao tác chèn:** thao tác chèn gặp bất thương được chia làm 2 loại, minh họa cụ thể bằng các ví dụ cụ thể sau đây:
    - Để chèn một bộ dữ liệu nhân viên mới vào EMP\_DEPT với mã số bộ phận là 5, chúng ta phải bao gồm cả giá trị thuộc tính cho DEPARTMENT mà nhân viên đó làm việc, hoặc là để giá trị null nếu nhân viên đó không làm việc cho phòng ban nào. Chúng ta phải nhập các giá trị thuộc tính của bộ phận 5 một cách chính xác để phù hợp với các giá trị cho bộ phận 5 trong các bộ dữ liệu khác trong EMP\_DEPT. Trong thiết kế của hình 10.2 chúng ta không phải lo lắng về vấn đề nhất quán này bởi vì chúng ta chỉ nhập mã số bộ phận 5, chúng được ghi 1 lần trong cơ sở dữ liệu, như bộ dữ liệu duy nhất trong quan hệ DEPARTMENT.
    - Đó là khó khăn để chèn một bộ phận mới không có nhân viên nào trong mối quan hệ EMP\_DEPT. Cách duy nhất để làm đều này là đặt các giá trị null trong các thuộc tính cho nhân viên. Điều này gây ra một vấn đề SSN là khóa chính EMP\_DEPT, và mỗi bộ dữ liệu được cho là đại diện cho một thực thể nhân viên không phải là một thực thể bộ phận. Hơn nữa, các khi các nhân viên đầu tiên được phân công vào bộ phận, chúng ta không cần điều này, bộ dữ liêu với giá trị null và vấn đề không xảy ra ở hình 10.2. Bởi vì một bộ phận nào trong mối quan hệ DEPARTMENT có hay không có nhân viên làm việc cho nó, và bất cứ khi nào mọt nhân viên được phân vào bộ phận đó, một bộ tương ứng được lắp vào EMPLOYEE.
  + **Bất thường trong thao tác xóa:** Các vân đề bất thường trong thao tác xóa liên quan đến tình hình bất thường được đề cập trong ví dụ thứ 2 ở phần trên. Nếu chúng ta xóa từ EMP\_DEPT một bộ dữ liệu nhân viên, vấn đề này xảy khi đó là nhân viên đại diện cuối cùng làm việc cho bộ phận cụ thể, các thông tin liên quan mà bộ phận bị mất từ cơ sở dữ liệu. Vấn đề này không xảy ra trong cơ sở dữ liệu cảu hình 10.2 vì các dữ bộ dữ liệu của DEPARTMENT được lưu trữ riêng.
  + **Bất thường trong thao tác chỉnh sửa:** Trong EMP\_DEPT, nếu chúng ta thay đổi giá trị của một trong những thuộc tính của bộ phận cụ thể, người quản lý của bộ phận 5, chúng ta cần phải cập nhật các bộ dữ liệu của tất cả nhân viên làm việc trong bộ phận đó, nếu không các cơ sở dữ liệu trở nên không phù hợp, không nhất quán. Nếu chúng ta không cập nhật một số bộ dữ liệu, bộ phận tương tự sẽ hiển thị với hai giá trị khác nhau cho người quản lý trong các bộ nhân viên khác nhau, sẽ là sai.
* **Dựa vào các 3 thường trên đây, chúng ta có hướng dẫn sau đây:**

**HƯỚNG DẪN 2:** Thiết kế các lược đồ quan hệ để không chèn, xóa, hoặc sửa đổi bất thường có mặt trong mối quan hệ. Nếu có bất kỳ bất thường nào thì ta lưu ý rõ rằng và đảm bảo rằng các chương trình mà cập nhật cơ sở dữ liệu sẽ hoạt động một cách chính xác.

* Hướng dẫn thứ hai là phù hợp với một trường hợp, trình bày lại những định hướng đầu tiên. Chúng ta cũng có thể thấy sự cần thiết cho một cách tiếp cận chính thức để đánh giá liệu thiết kế đáp ứng các nguyên tắc này. Phần 10.2 thông qua 10,4 cung cấp các khái niệm chính thức cần thiết. Điều quan trọng là cần lưu ý rằng các nguyên tắc này đôi khi có thể phải bị vi phạm để cải thiện hiệu suất của các truy vấn nhất định. Ví dụ, nếu một truy vấn lấy thông tin quan trọng liên quan đến các bộ phận của một nhân viên cùng với các thuộc tính của nhân viên, các lược đồ EMP\_DEPT có thể được sử dụng như là một mối quan hệ cơ sở. Tuy nhiên, bất thường trong EMP\_DEPT phải được ghi nhận và tính toán (ví dụ, bằng cách sử dụng trigger hoặc stored procedures sẽ cập nhật tự động), bất cứ khi nào mối quan hệ cơ sở được cập nhật, chúng tôi không kết thúc với sự mâu thuẫn. Nói chung, nó được khuyến khích sử dụng các mối quan hệ bất thường cơ sở và để xác định quan điểm bao gồm sự tham gia cho việc đặt các thuộc tính tham chiếu thường xuyên trong các truy vấn quan trọng. Điều này làm giảm số lượng các điều kiện JOIN quy định trong truy vấn, làm cho nó đơn giản để viết các truy vấn một cách chính xác, và trong nhiều trường hợp, nó cải thiện hiệu suất.

1. **Giảm các giá trị NULL trong bộ dữ liệu.**

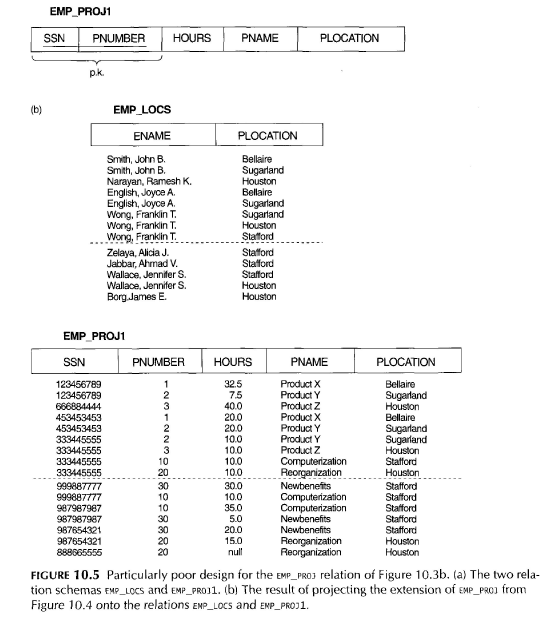
* Trong một số thiết kế lược đồ chúng ta có thể nhóm nhiều thuộc tính lại với nhau tạo thành một quan hệ “fat”. Nếu các thuộc tính không áp dụng cho tất cả các bộ dữ liệu liên quan, chúng ta kết thúc bằng null trong những bộ dữ liệu. Điều này có thể lãng phí không gian ở cấp độ lưu trữ và cũng có thể dẫn đến các vấn đề về sự hiểu ý nghĩa của các thuộc tính và quy định cụ thể JOIN hoạt động ở mức độ hợp lý. Một vấn đề khác với null là làm thế nào tính toán cho các bộ dữ liệu đó khi các hàm COUNT hoặc SUM được áp dụng. Hơn nữa null có thể mang nhiều ý nghĩa, ví dụ như sau:
  + Thuộc tính không được áp dụng cho bộ dữ liệu này.
  + Giá trị thuộc tính này cho bộ dữ liệu là giá trị unknown.
  + Giá trị này được biết đến nhưng không có, có nghĩa là nó không được ghi nhận.

**HƯỚNG DẪN 3:**

* Càng tránh càng tốt, tránh đặt các thuộc tính tỏng mối quan hệ cơ bản có giá trị thường xuyên có thể nhận giá trị null. Nếu null là không thể tránh khỏi, hãy chắc chắn rằng nó chỉ áp dụng trong trường hợp đặc biệt và không áp dụng với phần lớn các bộ dữ liệu trong mối quan hệ.

1. Sử dụng không gian một cách hiệu quả và tránh JOIN là hai tiêu chí trong xác định có bao gồm các cột có thể null trong một mối quan hệ hoặc một mối quan hệ riêng biệt (với các cột thích hợp). Ví dụ chỉ có 10% nhân viên văn phòng cá nhân, có ít lý do để bao gồm một thuộc tính OFFICE\_NUMBER trong mối quan hệ EMPLOYEE, thay vào đó một EMP\_OFFICE mối quan hệ (ESSN, OFFICE\_NUMBER) có thể được tạo ra để bao gồm các bộ dữ liệu cho các nhân viên với văn phòng cá nhân.**Loại bỏ khả năng tạo ra bộ dữ liệu ảo.**
2. **Phát sinh bộ dữ liệu giả:**

* Hãy xem xét hai EMP\_LOCS lược đồ quan hệ và EMP\_PROJ1 trong hình 10.5a, có thể được sử dụng thay vì mối quan hệ EMP\_PROJ duy nhất của 10.3b hình.Một bộ dữ liệu trong EMP\_LOCS có nghĩa là các nhân viên có tên là công trình ENAME trên một số dự án có vị trí là PLOCATION.Một bộ dữ liệu trong EMP\_PROJ1 có nghĩa là nhân viên có số an sinh xã hội là SSN làm việc HOURS mỗi tuần trên các dự án có tên, mã số và vị trí là PNAME, PNUMBER, và PLOCATION. Hình 10.5b cho thấy trạng thái liên quan EMP\_LOCS và EMP\_PROJ1 tương ứng với mối quan hệ EMP\_PROJ hình 10,4, thu được bằng cách áp dụng dự án thích hợp (bi) hoạt động EMP\_PROJ (bỏ qua các đường chấm chấm trong hình 10.5b tại thời điểm này).



* Giả sử chúng ta sử dụng EMP\_PROJ và EMP\_LOCS như các mối quan hệ cơ sở thay vì EMP\_PROJ. Điều này tạo ra một thiết kế giản đồ đặc biệt, bởi vì chúng ta không thể khôi phục lại các thông tin ban đầu được EMP\_PROJ từ EMP\_PROJ1 và EMP\_LOCS. Nếu chúng ta cố gắng thực hiện NATURSL JOIN EMP\_PROJ1 và EMP\_LOCS, kết quả tạo ra bộ dữ liệu nhiều hơn nữa so với các thiết lập ban đầu của các bộ dữ liệu trong EMP\_PROJ. Trong hình 10.5b được thể hiện (để giảm kích thước của mối quan hệ kết quả). Tuples thêm rằng không phải trong EMP\_PROJ được gọi là bộ dữ liệu giả bởi vì họ đại diện cho thông tin giả mạo hoặc sai lầm đó là không hợp lệ. Các bộ dữ liệu giả mạo được đánh dấu bằng dấu hoa thị ở Hình 10.6.  
  Tách EMP\_PROJ thành EMP\_LOCS và EMP\_PROJ1 không được ưa chuộng bởi vì, khi chúng ta JOIN chúng trở lại bằng cách sử dụng NATURAL JOIN, chúng ta không nhận được thông tin chính xác ban đầu. Điều này là bởi vì trong PLOCATION trường hợp là các thuộc tính có liên quan EMP\_LOCS và EMP\_PROJ1, và PLOCATION là một khóa chính cũng khôngphải là một chìa khóa nước ngoài tại một trong hai EMP\_LOChoặc EMP\_PROJ1. Lúc này chúng ta có nêu hướng dẫn thiết kế không chính thức khác.

**HƯỚNG DẪN 4:**

* Thiết kế lược đồ quan hệ để chúng có thể được tham gia với điều kiện như nhau trên các thuộc tính có khóa chính hoặc khóa ngoại,  với một cách mà đảm bảo rằng không có bộ dữ liệu giả mạo được tạo ra.Tránh quan hệ có chứa các thuộc tính kết hợp mà không phải là (khoá ngoại, khóa chính) kết hợp, bởi vì tham gia vào các thuộc tính như vậy có thể tạo các bộ dữ liệu giả mạo.  
  Hướng dẫn không chính thức này rõ ràng cần phải được quy định chính thức. Trong chương 11, chúng ta thảo luận một điều kiệnchính thức, được gọi là nonadditive (hoặc losseless) tham gia tài sản, đảm bảo chắc chắn rằng tham gia không sản xuất các bộ dữ liệu giả mạo.
* Trong phần 10.1.1.1 qua 10.1.4, chúng ta không chính thức thảo luận về tình huống dẫn đến các lược đồ quan hệ có vấn đề, và chúng ta đề nghị hướng dẫn không chính thức cho một thiết kếquan hệ tốt.Những vấn đề chúng ta đã chỉ ra, có thể được pháthiện mà không có công cụ bổ sung phân tích, như sau:  
  Bất thường gây ra dư thừa được trong quá trình chèn vào và sửa đổi, bổ sung một mối quan hệ, và có thể gây ra mất thông tin tình cờ trong một xóa từ một quan hệ.  
  Lãng phí không gian lưu trữ do null và khó khăn trong việc thực hiện các hoạt động tập hợp và tham gia do các giá trị null.  
  Thế hệ dữ liệu không hợp lệ và giả mạo trong quá trình tham giavào mối quan hệ không đúng cơ sở liên quan.  
  Trong phần còn lại của chương này, chúng ta trình bày các khái niệm chính thức và không chính thức lý thuyết có thể được sử dụng để xác định "tốt" và "thứ xấu" các lược đồ quan hệ cá nhânchính xác hơn. Chúng tôi lần đầu tiên thảo luận về sự phụ thuộcchức năng như một công cụ để phân tích. Sau đó, chúng tôi chỉ định ba bình thường froms thêm bình thường và Boyce-Coddhình thức bình thường (BCNF) với các lược đồ khẩu phần. TrongChaper 11, chúng ta xác định thêm các hình thức bình thường đómà là dựa trên các loại bổ sung dữ liệu phụ thuộc được gọi làphụ thuộc nhiều giá trị và tham gia phụ thuộc.
* **10,2 CHỨC NĂNG phụ thuộc.**Khái niệm đơn giản quan trọng nhất trong lý thuyết thiết kế sơ đồmối quan hệ phụ thuộc, chức năng). Phần này chúng tôi chính thức xác định khái niệm, và trong mục 10,3 chúng ta thấy làm thếnào nó có thể được sử dụng để xác định các hình thức bình thường cho shcemas mối quan hệ bình thường.  
    
  **10.2.1 Định nghĩa của chức năng,**  
  Một phụ thuộc chức năng là một hạn chế bettween hai bộ thuộctính từ cơ sở dữ liệu. Giả sử rằng cơ sở dữ liệu quan hệ của chúng tôi. Cơ sở dữ liệu giản đồ đã n attribule A1, A2, A3, A4,A1, ... An, cho phép chúng ta nghĩ về toàn bộ cơ sở dữ liệu như là một lược đồ quan hệ phổ quát R = {Ai}