**10.1 QUI TẮC THIẾT KẾ LƯỢC ĐỒ QUAN HỆ:**

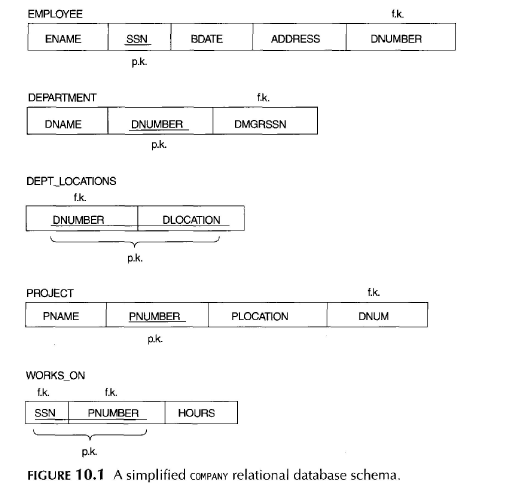
4 tiêu chuẩn chất lượng đánh giá thiết kế lược đồ quan hệ

* Ngữ nghĩa của thuộc tính
* Giảm sự trùng lắp dữ liệu trong bộ
* Giảm giá trị null trong bộ
* Ngăn chặn khả năng phát sinh những bộ sai

**10.1.1 Ngữ nghĩa của thuộc tính quan hệ:**

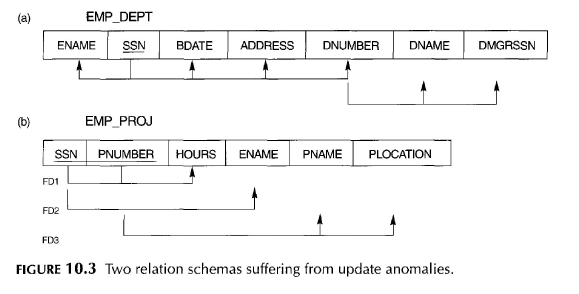
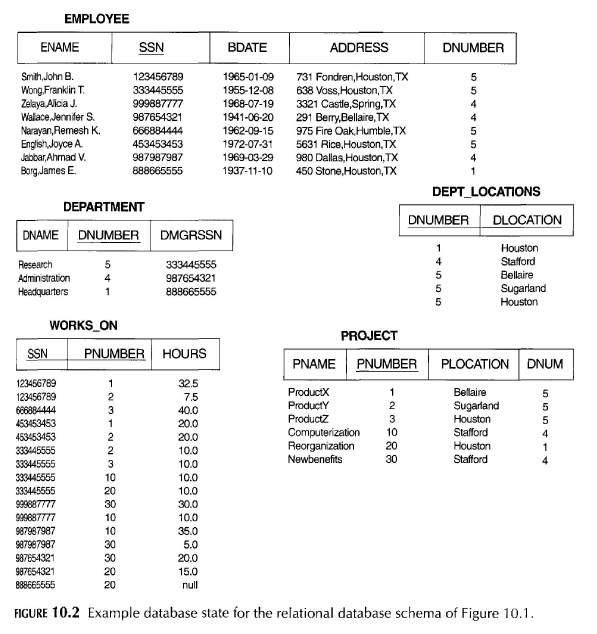
Bất cứ khi chúng ta nhóm các thuộc tính lại để tạo ra lược đồ quan hệ, ta cho rằng các thuộc tính đó thuộc về một quan hệ có ý nghĩa xác định trong thế giới thực và có cách hiểu riêng liên quan. Trong chương 5, chúng ta đã nói về cách mỗi quan hệ được hiểu như một tập các sự thật hay các phát biểu. Ngữ nghĩa (semantics) xác định cách hiểu giá trị thuộc tính chứa trong một bộ của quan hệ, nói cách khác giá trị thuộc tính trong bộ có liên quan với nhau. Nếu thiết kế quan niệm được thực hiện cẩn thận, được ánh xạ có hệ thống vào các quan hệ, hầu hết các ngữ nghĩa sẽ được chứa trong đó, và thiết kế kết quả sẽ có ý nghĩa rõ ràng.

Tóm lại, ngữ nghĩa của quan hệ càng dễ hiểu, thiết kế lược đồ quan hệ càng hiệu quả. Ví dụ trong Figure 10.1, một bản rút gọn của lược đồ dữ liệu quan hệ COMPANY hình 5.5, và 10.2, diễn tả một ví dụ của tình trạng quan hệ của lược đồ quan hệ này. Ý nghĩa của quan hệ EMPLOYEE khá đơn giản Mỗi bộ đại diện cho 1 nhân viên, với các giá trị tên, số SSN, ngày sinh, địa chỉ, mã phòng ban mà nhân viên đó làm việc. Thuộc tính DNUMBER là một khóa ngoại thể hiện mối quan hệ ngầm giữa nhân viên và phòng ban. Ngữ nghĩa của lược đồ DEPARTMENT và PROJECT cũng dế hiểu. Mỗi bộ của DEPARTMENT đại diện một thực thể phòng ban, và mỗi bộ của PROJECT là một thực thể đề án. Thuộc tính DMGRSSN liên quan giữa phòng ban và nhân viên trưởng phòng, trong khi đó DNUM của PROJECT diễn tả một đề án và phòng ban phụ trách, cả 2 đều là thuộc tính khóa ngoại. Mức độ dễ dàng giải thích ý nghĩa của các quan hệ chính là thước đo hiệu quả một thiết kế quan hệ có tốt hay không



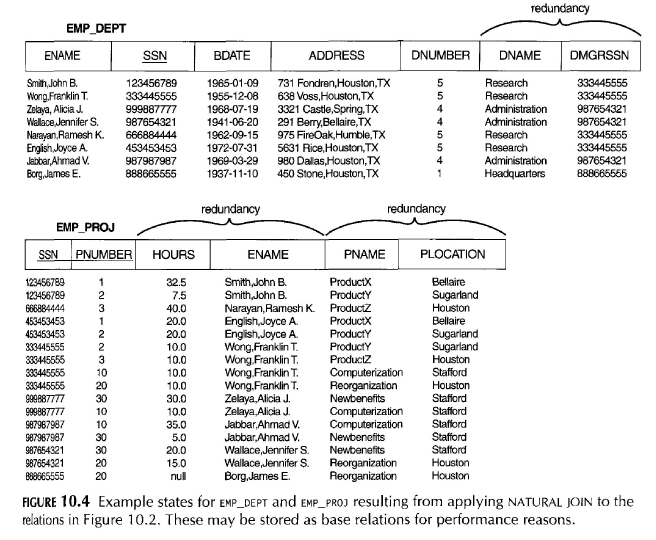
Ngữ nghĩa của 2 lược đồ quan hệ khác trong hinh 10.1 phức tạp hơn. Mỗi bộ trong DEPT\_LOCATION cho biết mã phòng ban (DNUMBER) và một trong những địa điểm của phòng ban (DLOCATION). Mỗi bộ trong WORKS\_ON xác định mỗi nhân viên một SSN, mã đề án của một trong những đề án mà nhân viên đó phụ trách (PNUMBER), và thời gian làm việc mỗi tuần cho đề án đó của nhân viên (HOURS). Tuy nhiên, cả 2 lược đồ đều rõ ràng và không mơ hồ. Lược đồ DEPT\_LOCATION thể hiện thuộc tính đa trị của DEPARTMENT, trong khi WORKS\_ON thể hiện mối quan hệ M:N giữa EMPLOYEE và PROJECT. Vì thế, tất cả lược đồ quan hệ trong hình 10.1 có thể được coi như dễ dàng giải thích và do dó tốt cho quan điểm có ngữ nghĩa rõ ràng. Chúng ta do đó tính toán đề ra qui tắc thiết kế lược đồ như sau:

**QUI TẮC 1**:Thiết kế một lược đồ quan hệ có ý nghĩa dễ hiểu. Đừng kết hợp các thuộc tính từ nhiều loại thực thể và quan hệ trong một quan hệ.Một cách trực giác, nếu một lược đồ quan hệ tương ứng một loại thực thể hoặc một loại quan hệ, hãy giải thích ý nghĩa của nó một cách rành mạch rõ ràng. Mặt khác, nếu quan hệ tương ứng một tập hợp của nhiều thực thể và quan hệ, có thể dẫn đến ngữ nghĩa mơ hồ và quan hệ không rõ ràng. Lược đồ quan hệ trong hinhg 10.3a, 10.3b có ngữ nghĩa rõ ràng. Một bộ trong lược đồ quan hệ hình 10.3a đại diện một nhân viên nhưng bao gồm thông tin mở rộng – tên phòng ban đang làm việc và số SSN của trưởng phòng ban. Hình 10.3b, mỗi bộ thể hiện quan hệ của nhân viên với đề án nhưng gồm tên nhân viên (ENAME) và tên đề án (PNAME), và địa điểm đề án. Tuy nhiên chúng được xem là một thiết kế tồi vì vi phạm qui tắc 1 bằng cách kết hợp thuộc tính từ nhiều thực thể riêng biệt, EMP\_DEPT kết hợp thuộc tính của nhân viên và phòng ban, EMP\_PROJ kết hợp nhân viên và đề án. Chúng được dùng như các khung nhing, nhưng lại gây rắc rối khi sử dụng như một quan hệ cơ bản, chúng ta sẽ thảo luận về vấn đề này ở phần sau.



**10.1.2 Dư thừa (Trùng lắp) dữ liệu trong bộ và cập nhật những bất thường:**

Một trong các mục tiêu của thiết kế lược đồ là hạn chế tối đa không gian lưu trữ của quan hệ cơ bản (và cả các file tương ứng). Ví dụ, so sánh không gian sử dụng bởi 2 quan hệ cơ bản EMPLOYEE vad DEPARTMENT trong hình 10.2 với quan hệ cơ bản EMP\_DEPT hình 10.4 - kết quả của phép kết tự nhiên của EMPOYEE và DEPARTMENT. Trong EMP\_DEPT, giá trị thuộc tính thuộc về một phòng ban cụ thể (DNUMBER, DNAME, DMGRSSN) được lặp lại cho bất kì một nhân viên nào làm việc cho phòng ban đó. Ngược lại, mỗi thông tin phòng ban xuất hiện chỉ một lần trong quan hệ DEPARTMENT hình 10.2. Chỉ có mã phòng ban được lặp lại trong quan hệ EMPLOYEE với mỗi nhân viên làm việc cho phòng ban đó. Kết luận tương tự vẫn đúng với quan hệ EMP\_PROJ, tăng quan hệ WORKS\_ON với thuộc tính mở rộng từ EMPLOYEE và PROJECT



Một lỗi nghiêm trọng khác khi sử dụng quan hệ trong hình 10.4 như quan hệ cơ bản là vấn đề cập nhật những bất thường. Những vấn đề đó có thể liệt vào thêm, xóa và chỉnh sửa những bất thường

*Thêm các bất thường:* thêm các bất thường có thể khác biệt thành 2 loại, được mô tả bằng 2 ví dụ dưới đây trên quan hệ EMP\_DEPT”

* để thêm một bộ nhân viên mới vào EMP\_DEPT, chúng ta phải bao gồm những giá trị của phòng ban mà nhân viên đó làm và cả giá trị null, (nếu nhân viên không làm việc cho phòng ban nào). Ví dụ để thêm 1 bộ mới cho 1 nhân viên làm việc ở phòng 5, chúng ta phải nhập thuộc tính giá trị của phòng 5 chính xác, vì thế chúng phải phù hợp với các giá trị của phòng 5 trong các bộ trước đó. Trong thiết kế hình 10.2, chúng ta không cần phải lo lắng vấn đề phù hợp vì ta chỉ nhập một mã phòng ban trong bộ nhân viên, các giá trị khác của phòng 5 được lưu chỉ 1 lần trong cơ sở dữ liệu, - 1 bộ đơn lẻ trong quan hệ DEPARTMENT
* khó khăn trong việc thêm mới 1 phòng ban mà chưa có nhân viên trong quan hệ EMP\_DEPT. Cách duy nhât để làm việc này là để giá trị null trong thuộc tính của nhân viên. Điều này sẽ gây rắc rối vì SSN là khóa chính của EMP\_DEPT, và mỗi bộ được giả định đại diện cho 1 nhân viên – không phải cho 1 thực thể phòng ban. Hơn thế nữa, khi nhân viên đầu tiên được thêm vào phòng ban, chúng ta không cần giá trị null nữa. Vấn đề này không xảy ra trong thiết kế hình 10.2, vì một phòng ban vẫn được nhập vào quan hệ DEPARTMENT dù có nhân viên hay không , và bất cứ khi nào một nhân viên được thêm vào phòng ban, một bộ tương ứng sẽ được chèn vào EMPLOYEE

*Xóa những bất thường*: Vấn đề trong việc xóa cũng tương tự như trường hợp thêm các bất thường thứ 2. Nếu ta xóa một nhân viên từ quan hệ EMP\_DEPT, mà nhân viên đó là nhân viên cuối cùng làm việc cho một phòng ban cụ thể, thì khi xóa thông tin về phòng ban cũng mất theo. Vấn đề này không xảy ra trong hình 10.2 vì các bộ của DEPARTMENT được lưu trữ riêng.

*Chỉnh sửa các bất thường:* Trong EMP\_DEPT, nếu ta thay đổi giá trị của một thuộc tính ở một phòng ban - ví dụ phòng 5 - chúng ta phải cập nhật các bộ của tất cả nhân viên làm việc cho phòng ban đó, mặt khác, cơ sở dữ liệu sẽ trở nên không phù hợp. Nếu thất bại trong việc thay đổi một vài bộ, sẽ dẫn đến sai sót.

Dựa vào 3 bất thường trên, chúng ta đúc kết ra qui tắc 2

**Qui tắc 2:** thiết kế lược đồ quan hệ cơ bản không có những bất thường trong thêm, xóa, chỉnh sửa. Nếu có bất thường nào, hãy ghi chú rõ ràng và chắc chắn chương trình cập nhật cơ sở dữ liệu đúng đắn

Qui tắc 2 phải phù hợp (nói cách khác là phát biểu lại) qui tắc 1. Chúng ta còn cần thấy sự cần thiết của một phương pháp đánh giá một thiết kế có đạt qui tắc hay không. Từ phần 10.2 đến 10.4 cung cấp những khái niệm hình thức cần thiết. Đôi lúc chúng ta có thể vi phạm các qui tắc để cải thiện hiệu suất của các câu truy vấn. Ví dụ, nếu một truy vấn quan trọng tìm thông tin liên quan đến một phòng ban của nhân viên cùng với thuộc tính của nhân viên, lược đồ EMP\_DEPT có thể được dùng như một quan hệ cơ bản. Tuy nhiên những bất thường trong EMP\_DEPT phải được chú ý và kể đến (ví dụ cách dùng triggers hoặc stored procedure có thể tự động cập nhật) vì thế, bất cứ khi nào quan hệ cơ bản được cập nhât, chúng ta không thể không kết thuc sự không phù hợp. Một cách tổng quát, nên sử dụng những quan hệ cơ bản tự do bất thường và xác định các view chứa sự kết hợp cho của các quan hệ thường xuyên được truy vấn. Điều này sẽ làm giảm số thuật ngữ JOIN (phép kết) trong câu truy vấn, và làm câu truy vấn đơn giản hơn và trong nhiều trường hợp cải thiện hiệu suất.

**10.1.3 Giá trị NULL trong bộ**

Trong một vào thiết kế lược đồ, chúng ta có thể nhóm nhiều thuộc tính lại với nhau vào một “fat” (lớn - nhiều) quan hệ. Nếu nhiều thuộc tính không có trong tất cả các bộ thì chúng ta sẽ có nhiều có nhiều giá trị null trong bộ. Điều này có thể gây lãng phí không gian lưu trữ và có thể dẫn đến những rắc rối trong việc hiểu lầm ý nghĩa của thuộc tính và việc xác định phép kết ở mức vật lí. Vấn đề khác với giá trị null là việc tính toán chúng khi thực hiện các phép toán tính tổng số như COUNT hay SUM. Thêm vào đó, giá trị null có thể có nhiều ý nghĩa diễn dịch

* Thuộc tính không áp dụng đối với bộ này
* Giá trị thuộc tính của bộ chưa xác định
* Giá trị xác định nhưng vắng mặt, chưa ghi lại

Có chung 1 thể hiện cho tất cả các giá trị null có thể dung hòa ý nghĩa khác nhau mà chúng có. Vì thế chúng ta khẳng định một qui tắc khác

**Qui tắc 3:** Cố gắng tránh đặt các thuộc tính thường có giá trị null trong quan hệ cơ bản. Nếu giá trị null là không thể tránh thì, chắc chắc rằng chúng chỉ áp dụng đối với những trường hợp ngoại lệ và không áp dụng đối với hầu hết các bộ trong quan hệ

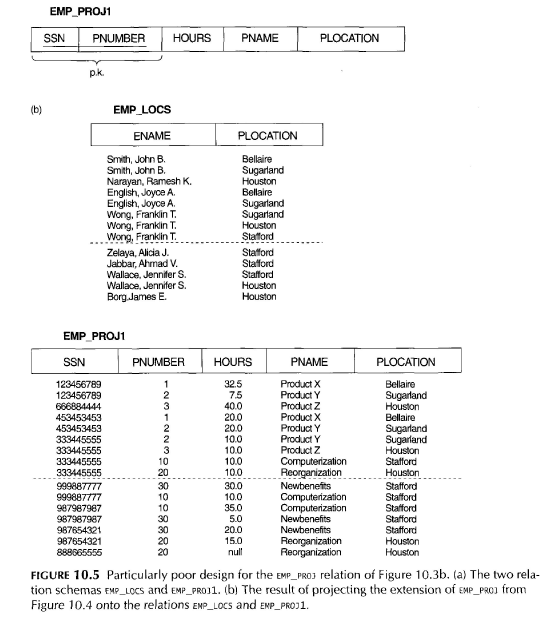
Sử dụng không gian một cách hiệu quả và tránh phép kết là 2 tiêu chuẩn quan trọng hơn hết khi quyết định có bao gồm những cột chứa giá trị null hay có một quan hệ riêng lẻ cho những cột đó (với khóa thích hợp). Ví dụ, nếu chỉ có 10% nhân viên có phòng làm việc riêng, thì chỉ có ít sự cần thiết để chứa một cột số phòng làm việc OFFICE\_NUMBER trong quan hệ EMPLOYEE, thay vào đó, một quan hệ EMP\_OFFICES (ESSN, OFFICE\_NUMBER) có thể được tạo ra để chứa các bộ mà nhân viên có phòng làm việc riêng

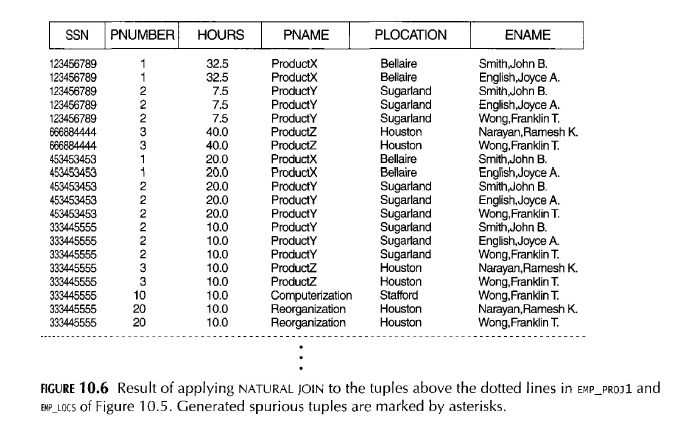
**10.1.4 Sự tạo thành những bộ giả**

Xem xét 2 lược đồ quan hệ EMP\_LOCS và EMP\_PROJ1 trong hình 10.5a, có thể được sử dụng thay cho quan hệ EMP\_PROJ trong hình 10.3b. Một bộ trong EMP\_LOCS cho biết một nhân viên có tên ENAME làm việc cho các đề án mà địa điểm đề án là PLOCATION. Một bộ trong EMP\_PROJ1 cho biết một nhân viên có số SSN và số giờ làm việc mỗi tuần của nhân viên trong dự án có tên, mã, địa điểm là PNAME, PNUMBER, PLOCATION. Hình 10.5b thể hiện tình trạng quan hệ của EMP\_LOCS và EMP\_PROJ1 tương ứng quan hệ của hình 10.4 có thể đạt được bằng cách áp dụng các phép toán thích hợp của PROJECT đối với EMP\_PROJ (không xét những dòng chấm trong hình 10.5)

Giả sử chúng ta sử dụng EMP\_PROJ1 và EMP\_LOCS như một quan hệ cơ bản thay cho EMP\_PROJ. Điều này sẽ tạo ra một thiết kế lược đồ tồi, vì chúng ta không thể phục hồi thông tin gốc ở EMP\_PROJ từ EMP\_PROJ1 và EMP\_LOCS. Nếu ta cố gắng thực hiện phép kết tự nhiên trên EMP\_PROJ1 và EMP\_LOCS, kết quả sẽ tạo r nhiều bộ hơn tập bộ gốc của EMP\_PROJ. Trong hình 10.6 thể hiện kết quả của việc áp dụng phép kết chỉ trên các bộ ở trên đường dấu chấm ở hình 10.5b (để giảm kích thước của quan hệ kết quả). Các bộ mở rộng không có trong EMP\_PROJ được gọi là bộ giả bởi chúng đại diẹn cho thông tin giả hoặc sai lệch, không có căn cứ. Các bộ giả được đánh dấu (\*) trong hình 10.6

Phân rã EMP\_PROJ thành EMP\_LOCS và EMP\_PROJ1 không được mong đợi, vì khi chúng ta kết hợp lại bằng phép kết tự nhiên, chúng ta không thể có được thông tin gốc chính xác. Điều này là bởi vì trong trường hợp PLOCATION là thuộc tính liên quan giữa EMP\_LOCS và EMP\_PROJ1 và PLOCATION không phải là khóa chính của EMP\_LOCS và EMP\_PROJ1. Bây giờ ta có thể khẳng định thêm một qui tắc thiết kế





**Qui tắc 4:** Thiết kế lược đồ quan hệ có thể kết lại bằng điều kiện bằng trên các thuộc tính là khóa chính hoặc khóa ngoại sao cho có thể bảo đảm không xuất hiện các bộ giả. Tránh quan hệ chứa các thuộc tính giống nhau mà không phải là sự kết hợp (không phải khóa ngoại, hay khóa chính), bời vì kết hợp trên các thuộc tính đó có thể tạo ra bộ giả

Qui tắc thông thường này rõ ràng cần thiết được khẳng định một các chính thức. Trong chương 11, chúng ta sẽ bàn về một điều kiện quan trọng, được gọi là tính chất kết không cộng hợp được (hay ít mất mác), đảm bảo các phép kết không sinh ra các bộ giả

**10.1.5 TỔNG KẾT VÀ CÁC TRANH LUẬN VỀ QUI TẮC THIẾT KẾ**

Trong phần 10.1.1 -> 10.1.4, chúng ta đã bàn về các trường hợp dẫn đến các lược đồ quan hệ mơ hồ, và đề xuất các qui tắc thông thường để có một thiết kế quan hệ tốt. Vấn đề mà ta chỉ ra, có thể được tìm thất bởi các công cụ phân tích hỗ trợ, như sau:

* Các bất thường có thể dẫn đến công việc dư thừa được thực hiện suốt quá trình thêm, sửa một quan hệ và có thể khiến mất mác thông tin trong suốt quá trình xóa
* Tiêu tốn không gian lưu trữ do giá trị null và khó khăn trong các phép toán tập hợp và phép kết do giá trị null
* Sự sản sinh các bộ giả hay không có căn cứ trong phép kết trên các quan hệ cơ bản liên quan không đúng cách

Trong phần còn lại của chương chúng ta sẽ trình bày về các khái niệm và lí thuyết quan trọng có thể được sử dụng để xác định mặt tốt và mặt xấu của một lược đồ quan hệ. Chúng ta sẽ thảo luận về phụ thuộc hàm như một công cụ để phân tích. Sau đó sẽ xác định 3 dạng chuẩn và dạng chuẩn Boyce-Codd cho lược đồ quan hệ.

***Translation: JML***