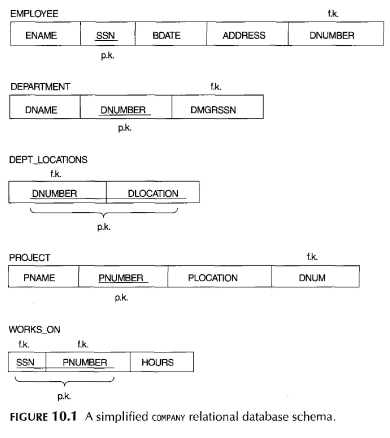
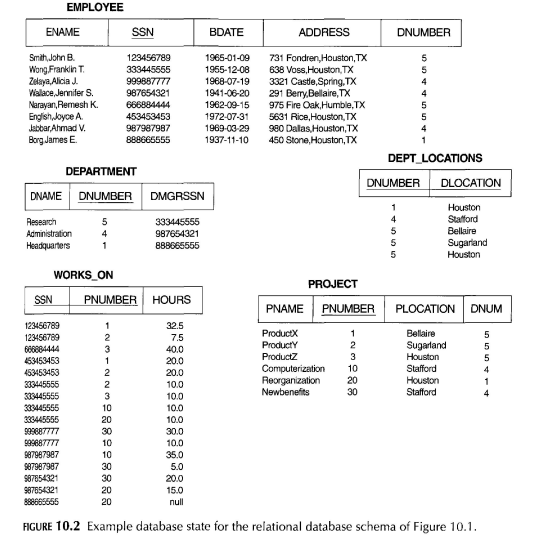
**10.1.1 semantics of the relation attributes**

Bất cứ khi nào ta nhóm các thuộc tính tạo thành 1 lược đồ quan hệ, chúng ta biết rằng các thuộc tính thuộc về 1 quan hệ thì có 1 ý nghĩa mang tính đời thực, và sự thể hiện thích hợp về chúng. Trong chương 5, chúng ta đã bàn về làm thế nào để các quan hệ có thể được thể hiện (hiểu) như 1 tập hợp các sự việc hoặc các báo cáo. Với nghĩa này- hoặc semantics, chỉ rõ làm thế nào để thể hiện các giá trị thuộc tính được lưu trữ trong 1 tuple của 1 quan hệ-trong từ khác, làm thế nào các giá trị thuộc tính trong 1 tuple quan hệ với những cái khác. Nếu mức thiết kế quan niệm được thiết kế kỹ càng theo một hệ thống sắp xếp vào các quan hệ, hầu hết các ngữ nghĩa sẽ được giải thích cho và kết quả thiết kế phải có ý nghĩa rõ ràng. Nói chung, càng dễ giải thích ý nghĩa của quan hệ thì quan hệ đó càng thiết kế tốt. để minh hoạ cho điều này,xem hình 10.1, một diễn tả đã rút gọn đơn giản về lược đồ dữ liệu quan hệ của 1 công ty hình 5.5 và hình 10.2 thể hiện 1 ví dụ của trạng thái quan hệ của 1 lược đồ này. Quan hệ EMPLOYEE là mỗi tuple là 1 nhận viên, với các giá trị thuộc tính: tên (ENAME), số bảo mật xã hội(CMND )(SSN), ngày sinh(BDATE), và địa chỉ(ADDRESS), và DNUMBER là mã của phòng ban mà nhân viên đó thuộc về. thuộc tính DNUMBER là 1 khoá ngoại tượng trung 1 quan hệ ngầm giữa EMPLOYEE và DEPARTMENT. Nghĩa logic(ngữ nghĩa học) của DEPARTMENT và lược đồ PROJECT thì cũng đơn giản: mỗi tuple của DEPARTMENT tượng trưng cho 1 thực thể phòng ban, và mỗi 1 tuple PROJECT tượng trung cho 1 thực thể đề án. Thuộc tính DMGRSSN liên kết 1 phòng ban với nhân viên của phòng ban đó, trong khi DNUM của PROJECt liên kết 1 đề án với phòng ban phụ trách đề án đó; cả 2 đều là thuộc tính khoá ngoại. ràng buộc cái mà nghĩa của thuộc tính của 1 quan hệ có thể dc giải thích với nó là 1 đơn vị đo lường theo đúng quy định: làm thế nào đề thiết kế lược đồ quan hệ hiệu quả.



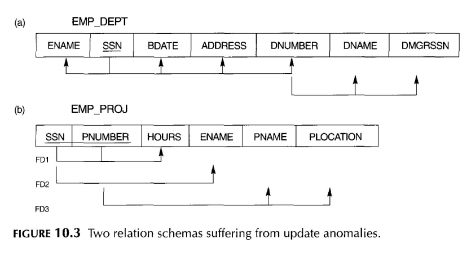
Ngữ nghĩa học của 2 lược đồ còn lại trong hình 10.1 hơi phức tập. mỗi tuple trong DEPT\_LOCATIONS cho 1 mã phòng ban(DNUMBER) và 1 trong những địa chỉ của phòng ban(DLOCATION). Mỗi tuple trong WORKS\_ON cho 1 số bảo mật xã hỗi(SSN), mã đề án của 1 trong những đề án mà nhân viên đó làm(PNUMBER), và số giờ mỗi tuần mà nhân viên đó làm cho đề án đó(HOURS). Tuy nhiên, cả 2 lược đồ có 1 xác định rõ ràng và sự thể hiện không nhập nhằng. lược đồ DEPT\_LOCATION tượng trưng cho 1 thuộc tính đa trị của DEPARTMENT, ngược lại WORKS\_ON tượng trưng cho 1 quan hệ M:N giữa EMPLOYEE và PROJECT. Do đó, tất cả lược đồ quan hệ trong hình 10.1 có thể dc xét dễ dàng để giải thích và như vậy\_\_\_\_\_\_ thật tốt từ quan điểm lập trường của 1 lược đồ rõ ràng.\_\_\_\_\_\_\_\_\_\_\_\_ như vậy ta có thể lập thành quy tắc sau.

**GUIDELINE 1.** Thiết ké 1 lược đồ quan hệ để làm rõ ý nghĩa của nó. Ta không được gộp các thuộc tính từ các loại thực thể phức tạp và gộp các loại vào 1 quan hệ. bằng trực giác, nếu 1 lược đồ quan hệ tương ứng với 1 loại thực thể hoặc một loại mối quan hệ, thì nó không phức tạp để giải thích ý nghĩa của nó. Vì thế, nếu 1 quan hệ tương ứng với 1 hỗn hợp các thực thể và các



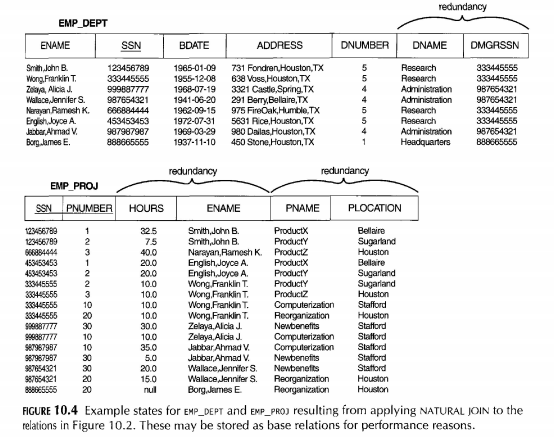
mối quan hệ phức tạp thì cho ra ngữ nghĩa không rõ ràng và các quan hệ không được giải thích 1 cách dễ dàng.

Những lược đồ quan hệ trong hình 10.3a và 10.3b cũng là những lược đồ rõ ràng. (người đọc có thể bỏ qua những dòng gạch chân trong các quan hệ; chúng được dùng để minh hoạ(làm rõ) cho ký hiệu phụ thuộc chức năng, đã đề cập đến trong mục 10.2). một tuple trong lược đồ quan hệ EMP\_DEPT

Trong hình 10.3a đại diện cho 1 nhân viên nhưng bao gồm thêm cả thông tin tên (DNAME) của phòng ban mà nhân viên đó thuộc và cả số bảo mật xã hôi(DMGRSSN) của phòng ban đó. Trong lược đồ EMP\_PROJ trong hình 10.3b, mỗi tuple gắn kết 1 nhân viên với 1 đề án nhưng cũng bao gồm thêm tên của phòng ban đó(DNAME), tên đề án(PNAME), và địa điềm của đề án đó(PLOCATION). Mặc dù 2 quan hệ này không có lỗi sai logic, nhưng chúng bị cho là thiết kế kém vì thiết kế này bị phạm nguyên tắc GUIDELINE 1 do cho các thuộc tính thuộc các thực thể có nghĩa trong thực tế khác nhau vào chung 1 quan hệ; EMP\_DEPT gộp các thuộc tính của các quan hệ: nhân viên và phòng ban, và EMP\_PROJ gộp các thuộc tính của nhân viên và đề án. Chúng có thể được dùng như xem())))(((, nhưng sẽ có vấn đề nếu được xem như lược đồ cơ sở, ta bàn luận trong phần dưới đây.

**10.1.2 thông tin thừa trong tuples và update dữ liệu sai**

1 mục đích của thiết kế lược đồ là để làm giảm không gian lưu trữ bộ nhớ cho các quan hệ cơ sở. việc nhóm các thuộc tính vào các lược đồ quan hệ sẻ tạo ra ảnh hưởng lớn đến khong gian lưu trữ. Ví dụ, so sánh không gian lưu trữ được dùng cho 2 quan hệ cơ sở: EMPLOYEE và DEPARTMENT trong hình 10.2 với quan hệ cơ sở trong 10.4, cái mà là kết quả của phép kết tự nhiện của EMPLOYEE và DEPARTMENT. trong EMP\_DEPT, các giá trị thuộc tính liên quan đến một phòng ban cụ thể (DNUMBER, DNAME, DMGRdqa) được lặp đi lặp lại cho mỗi nhân viên làm việc cho phòng ban đó. Ngược lại, thông tin của mỗi phòng ban chỉ xuất hiện 1 lận trong quan hệ DEPARTMENT trong hình 10.2. chỉ có thuộc tính number(DNUMBER) là lặp lại trong quan hệ EMPLOYEE ở các nhân viên mà làm việc cho phòng ban đó. Tương tự trong quan hệ EMP\_PROJ (hình 10.4), làm tănng quan hệ WORDS\_ON với các thuộc tính bổ sung từ quanhệ EMPLOYEE và PROJECT.



Vấn đế quan trọng khác gặp phải khi dùng quan hệ trong hình 10.4 làm quan hệ cơ sở là update anomalies(update sai). Những vấn đề này được chia thành: ensertion anomalies, deletion anomalies, and modification anomalies.

* Để insert 1 tuple nhân viên vào EMP\_DEPT, ta phải insert cả các giá trị thuộc tính của phòng ban mà nhân viên đó thuộc hoặc để giá trị null( nếu nhân viên đó chưa làm việc cho phong ban nào). Ví dụ, để insert cho 1 tuple nhân viên mà làm trong phòng ban số 5 , ta phải nhập vào giá trị thuộc tính 5 của phòng ban đó một cách chính xác nhất quán với các giá trị phòng ban 5 trong những tuples EMP\_DEPT khác. Trong thiết kế của hình 10.2, ta khong phải lo lắng về vấn đề đồng nhất bởi vì ta chỉ nhập số phòng ban trong tuple nhân viên; tất cả các giá trị thuộc tính khác của phòng ban có số phòng ban: 5 thì được lưu 1 lần trong dữ liệu, như 1 tuple trong quan hệ DEPARTMENT.
* Thật khó để insert 1 phòng ban mới vào quan hệ EMP\_DEPT mà chưa có nhân viên nào thuộc phòng ban đó. Cách duy nhát là đặt giá trị null vào các thuộc tính của nhân viên trong EMP\_DEPT. điều này gây ra vấn đề là vì SSN là khoá chính của EMP\_DEPT, và mỗi tuple cần để đại diên cho 1 thực thể entity – không phải cho thực thể phòng ban. Hơn nữa, khi một nhân viên đầu tiên được phân công cho phòng ban đó, ta không cần tuple này với giá trị null bất kỳ. vấn để này không xảy ra với thiết kế của hình 10.2, vì 1 phòng ban được nhập vào quan hệ DEPARTMENT khong cần có nhân viên nào làm việc cho nó, và bất cứ khi nào 1 nhân viên được phân công cho phòng ban đó thì tương ứng có EMPLOYEE để insert vào.
* Deletion anomalies. Vấn đề deletion anomalies có liên quan đến vấn đề insert anomaly thứ 2 đã bàn ở trên. Nếu ta xoá 1 nhân viên trong bảng EMP\_DEPT mà không may nhân viên đó là nhân viên duy nhất làm việc cho 1 phòng ban nào đó, thì thông tin liên quan đến phòng ban đó sẽ bị mất. vấn đề này thì không xảy ra với dữ liệu trong hình 10.2 bởi vì tuples của DEPARTMENT được lưu trữ ở nơi riêng biệt.
* Modification anomalies. Trong EMP\_DEPT, nếu ta thay đổi giá trị của 1 trong những thuộc tính của 1 phòng ban nào đó—ví dụ: quản lý của phòng 5—ta phải udate các tuple của tất cả nhân viên mà làm trong phòng ban đó; mặt khác, dữ liệu sẽ trở nên không đồng nhất. nếu ta udate bị lỗi một vài tuple, thì cùng 1 phòng ban sẽ có cho ra nhiểu giá trị manager khác nhau trong tuples nhân viên, gây ra sai. Dựa vào 3 anomalies trước, ta có guideline sau.

**GUIDELINE 2**. Thiết kế lược đồ quan hệ cơ sở để khong có insertion, deletion hoặc modification anomalies tồn tại trong các quan hệ. nếu có bất cứ 1 anomalies nào thì ghi chú chúng một cách rõ ràng và chắc chắn rằng chương trình update dữ liệu phải chính xác.

Trong 1 vài quan điểm thì Guideline 2 phù hợp với guideline 1. Ta cũng có thể thấy sự cần thiết cho 1 cách tiếp cận hình thức để đánh giá liệu một thiết kế đáp ứng những guideline này. Mục 10.2 đến 10.4 cung cấp những nội dung hình thức cần thiết. nó quan trọng để ghi chú các guidelines có thể thường bị phạm để cải tiến các câu lệnh query. Ví dụ, nếu 1 câu query quan trọng trả về thông tin của phòng ban mà 1 nhân viên đó thuộc về và các thuộc tính của nhân viên đó , thì lược đồ EMP\_DEPT có thể được dùng như lược đồ cơ sở. tuy nhiên, anomalies trong EMP\_DEPT phải được ghi nhớ và tính đến( ví dụ, dùng trigger hoặc lưu procedures để nó tự động update cho mình), bất cứ khi nào lược đồ quan hệ được update, ta không được kết thúc lệnh khi nó chứa mâu thuẫn. nói chung, nó thích hợp đễ dùng các quan hệ khong anomaly và các câu lệnh quan trọng thường hay dùng để xem mà có phép kết. điều này làm giàm đi phép kết trong query, làm cho nó đơn giản hoá câu lệnh query và trong nhiều trường hợp cải tiến câu lệnh.

10.1.3 giá trị null trong tuples

Trong một vài thiết kế lược đồ ta có thể nhóm nhiều thuộc tính lại vào 1 quan hệ “fat”. Nếu nhiều thuộc tính không có giá trị trong tất cả tuples trong lược đồ, ta kết thúc bằng giá trị null trong các tuples đó. Điều này có thể làm phí không gian lưu trữ và có thể dẫn đến vấn đề không hiểu nghĩa của các thuộc tính và với phép kết ở mức logic. 1 vấn đề khác với giá trị null là bằng cách nào để đếm được chúng khi dùng các phép sum, count. Hơn nữa, null có thể có sự thể hiện phức tạp:

* Thuộc tính không có trong tuple này.
* Không biết Giá trị thuộc tính cho tuple này.
* Giá trị được biết đến nhưng vắng mặt; đó là nó chưa được ghi vào.

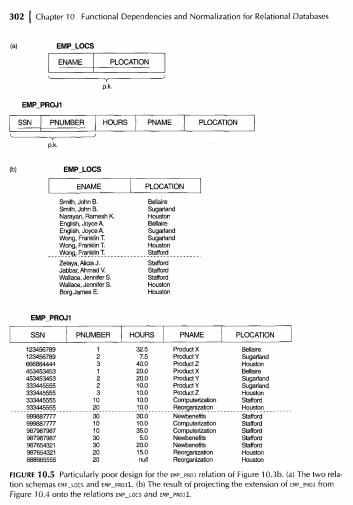
Có cùng đại diên cho tất cả giá trị null giàn xếp những nghĩa khác nhau mà chúng có. Vì thế , ta có thể guideline sau:

**GUIDELINE 3.** Hơn có thể, tránh việc đặt các thuộc tính trong 1 quan hệ cơ sở mà các giá trị của nó có thể thường xuyên bị null. Nều những giá trị null không thể tranh được , chắc rằng chúng chỉ đặt vào các trường hơp ngoại lệ và không chỉ đến phần đông của các tuples trong lược đồ.

Việc Dùng không gian hiệu quả và việc tránh phép kết là 2 tiêu chuẩn quan trọng hơn bất cứ nhận định nào khác mà xác định có gộp các cột mà có thể có giá trị null trong 1 quan hệ hay có quan hê riêng lẽ cho những cột đó. Ví dụ, nếu chỉ có 10% nhân viê có cơ quan cá nhân, ta thấy ít nên không nên gộp thuộc tính OFFICE\_NUMBER vào quan hệ EMPLOYEE; hơn nữa, 1 quan hệ EMP\_OFFICES( ESSN, OFFICE\_NUMBER) có thể được tạo để gộp các tuples chỉ cho các nhân viên có cơ quan cá nhân.

10.1.4 sự phát sinh của tuples giả

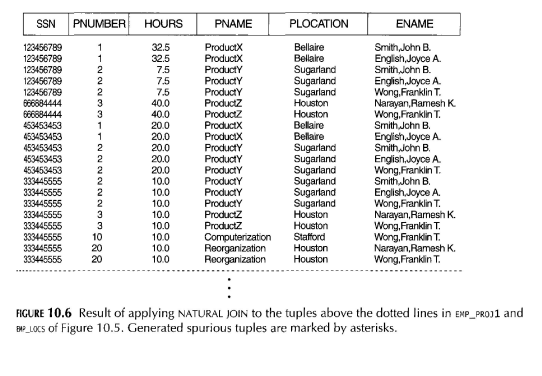
Xét 2 lược đồ quan hệ EMP\_LOCS và EMP\_PROJ1 trong hình 10.5a, chúng cố thể được dùng thay vì chỉ có lược đồ EMP\_PROJ đơn lẻ trong hình 10.3b. 1 tuple trong EMP\_LOCS có nghĩa là người nhân viên có tên là ENAME làm việc cho 1 số đền án mà địa điểm của đề án đ1o là PLOCATION.



1 tuple trong EMP\_PROJ1 có nghĩa là người nhân viên mà có social security number là SSN làm việc HOURS cho 1 tuần trến 1 đề án có tên , số và địa điểm: PNAME, PNUMBER, và PLOCATION. Hình 10.5 cho biết quan trạng thái quan hệ của EMP\_LOCS và EMP\_PROJ1 tương ứng với EMP\_PROJ trong hình 10.4 mà được thu bởi phép chiếu trong PROJECT.

Giả sử rằng ta đã dùng EMP\_PROJ1 và EMP\_LOCS như quan hệ cơ sở thay vì dùng EMP\_PROJ. điều này là 1 thiết kết kém, vì ta không thể thu lại được thông tin mà đã thường có trong EMP\_PROJ từ EMP\_PROJ1 và EMP\_LOCS. Nếu ta cố gắng bằng 1 phép kế tự nhiên trên EMP\_PROJ1 và EMP\_LOCS, kết quả tạo ra nhiều tuples hơn tập hợp các tuples trong RMP\_PROJ. trong hình 10.6, kết quả được gán phép kết đó là những dòng tuple trên dòng kẻ của hình 10.5b. thêm các tuples mà đã không có trong EMP\_PROJ được gọi là các tuples giả vì chúng hiện diên giả tạo hoặc thông tin sai không có giá trị. Tuples giả được đánh dấu bằng dầu (\*) trong hình 10.6.

Việc phân tích EMP\_PROJ thành EMP\_LOCS và EMP\_PROJ1 là không được chuộng vì khi ta kết chúng lại bằng phép kết tự nhiên thì ta không có được kết quả với các thông tin chính xác như ban đầu. điều này bởi vì trong trường hợp PLOCATION là thuộc tính mà kết nối EMP\_LOCS và EMP\_PROJ1, và PLOCATION không là khoá chính hay khoá ngoại của EMP\_LOCS hoặc EMP\_PROJ1. Ta có thể xem guideline sau:



**Guideline 4**. Thiết kế lược đồ quan hệ để chúng có thể được kết với điều kiện ngang bằng trên các thuộc tính mà là khoá chính hay khoá ngoại từ một vài quan điểm mà đảm bảo không có tuple giả được hình thành. Tránh các quan hệ mà chứa việc hợp các thuộc tính mà là các kết hợp, vì việc kết trên những thuộc tính đó có thể gây ra tuples giả.

Guideline này cần 1 cách hiển nhiên để được bắt đầu một cách chính thức hơn.

**10.1.5 tóm tắt và bàn luận về guideline.**

Trong phần 10.1.1 đến 10.1.4, ta bàn luận một cách kỹ lưỡng tình huống mà dẫn đến lược đồ quan hệ mơ hồ, và ta đề xuất guidelines cho việc thiết kế quan hệ tốt

. vấn đề ta chỉ ra mà có thể được xác định mà không có công cụ cho người phân tích, là:

* Anomalies mà gây ra việc thừa được làm tròng quá trình insertion vào và modification vào 1 quan hệ, và có thể gây ra mất dữ liệu trong deletion từ quan hệ.
* Bỏ trống vùng lưu trữ dẫn đến null và các thao tác kết hợp thực thi khó và kết dẫn đến giá trị null.
* Sinh ra dữ liệu không có giá trị và giả trong phép kết trên kết nối không thích hợp các quan hệ cơ sở.

Phần cuối của chương này ta trình bày nội dung hình thức và lý thuyết mà có thể thường dùng để xác định tốt và kém của quan hệ riêng lẻ chính xác hơn.đầu tiên ta bàn phần phụ thuộc hàm như một công cụ cho nhà phân tích.