

BÀI THỰC HÀNH SỐ 4:

Thiết kế và tạo lập CSDL

Mục tiêu bài học:

- *Tìm hiểu sự cần thiết của thiết kế CSDL*
- *Tìm hiểu quy trình thiết kế CSDL*
- *Vẽ sơ đồ Quan hệ - Thực thể theo mô tả thực thể*
- *Chuyển thiết kế CSDL sang các bảng dữ liệu*
- *Đưa các bảng về dạng chuẩn hóa 1, 2 và 3*

Phần I: Hướng Dẫn

Thiết kế CSDL

Như chúng ta đã biết, CSDL là một tập hợp các dữ liệu có liên quan đến nhau dưới dạng các bản ghi trong các bảng. Khi phát triển các hệ thống tin học hóa người phát triển không chỉ cần thiết kế các tiến trình xử lý của hệ thống mà còn phải quan tâm đến cách tổ chức dữ liệu. Quá trình này chính là thiết kế CSDL trong đó chỉ ra các loại dữ liệu được lưu trữ, lượng dữ liệu lưu trữ và cách tổ chức dữ liệu, v.v. Quá trình thiết kế CSDL chính là quá trình lập kế hoạch và đưa ra cấu trúc của dữ liệu. Vậy **tại sao lại cần phải thiết kế CSDL?** Câu trả lời là để có được một dự án hay một hệ thống thành công thì chúng ta không chỉ phải đảm bảo các tiến trình thực thi chính xác mà còn phải đảm bảo một cấu trúc dữ liệu hợp lý. Với việc xác định trước các yếu tố liên quan đến dữ liệu của môi trường xung quanh, chúng ta có thể tránh được các sai sót hay xung đột về sau.

Khi thiết kế một CSDL, chúng ta có thể phải dựa vào một hệ thống thực để mô hình hóa trong CSDL. Quá trình này bao gồm việc quyết định các bảng cần tạo, các trường dữ liệu cũng như mối quan hệ giữa các bảng. Nếu quá trình này được thực hiện một cách rõ ràng, tự nhiên và tự động thì rất tốt, nhưng thường thì không phải như vậy. Một CSDL được thiết kế tốt cần phải có thời gian, công sức để chuẩn bị, xây dựng và cải tiến.

Một CSDL được thiết kế theo mô hình quan hệ mang lại rất nhiều lợi ích. Dưới đây liệt kê một số lợi ích này:

- Giúp thêm mới, cập nhật, xóa dữ liệu hiệu quả hơn.
- Việc truy xuất tổng hợp dữ liệu và chiết xuất báo cáo hiệu quả hơn.
- Do CSDL tuân theo mô hình đã được thiết kế tốt, chúng ta có thể biết trước hoạt động của chúng.
- Với hầu hết dữ liệu được lưu trữ trong CSDL mà không phải trong ứng dụng, bản thân CSDL đã chứa đầy đủ thông tin.
- Dễ dàng thay đổi cấu trúc CSDL.

Như đề cập ở trên, thiết kế CSDL rất cần sự linh hoạt và sáng tạo. Dù là việc thiết kế CSDL cần phải theo đúng các mô hình chuẩn hóa và mô hình quan hệ, cuối cùng chúng ta vẫn phải đưa ra một thiết kế thể hiện được nghiệp vụ của doanh nghiệp. Lý thuyết thiết kế CSDL quan hệ thường đề cập đến những vấn đề cần tránh khi thiết kế nhưng lại không hướng dẫn chúng ta bắt đầu từ đâu và cách quản lý nghiệp vụ. Chính vì vậy ta cần phải hiểu rõ nghiệp vụ của doanh nghiệp (hay hoàn cảnh nghiệp vụ) mà chúng ta đang mô hình hóa. Một CSDL thiết kế tốt đòi hỏi người thiết kế phải hiểu rõ nghiệp vụ, cần có thời gian và kinh nghiệm.

QuyTrìnhThiếtKếCSDL

20 bước dưới đây giúp chúng ta thiết kế tốt một CSDL:

1. Người thiết kế phải nghiên cứu kỹ nghiệp vụ của hệ thống sẽ phát triển. Bước này thường được thực hiện thông qua việc gặp mặt nói chuyện và đặt câu hỏi với những người sẽ sử dụng hệ thống.
2. Viết lên giấy mục đích cơ bản của hệ thống. Ví dụ, ta có thể viết “Hệ thống này sẽ được dùng để xử lý đơn đặt hàng của khách hàng và theo dõi chúng để phục vụ cho các nghiệp vụ kế toán và lưu kho.” Thêm vào đó, ta có thể liệt kê các yêu cầu của hệ thống. Các yêu cầu này sẽ giúp chúng ta xây dựng cấu trúc của CSDL và các nghiệp vụ. Ví dụ, ta có thể xây dựng một danh sách các yêu cầu như “Phải có khả năng truy xuất địa chỉ của khách hàng để gửi thư.”
3. Xây dựng các mẫu biểu nhập liệu tạm (lên giấy). (Nếu trong quá trình thiết kế các bảng xuất hiện các ý tưởng về nghiệp vụ, ta nên ghi lại vào danh sách các yêu cầu như ở bước 2.) Cách tiếp cận cụ thể tùy thuộc vào trạng thái của hệ thống hiện tại.
 - Nếu hệ thống hiện tại chưa được tin học hóa, ta có thể dùng hệ thống giấy tờ sẵn có để thiết kế nhập các bảng dựa vào các biểu mẫu sẵn có. Các mẫu biểu này sẽ được chuẩn hóa về sau.
 - Nếu CSDL phải được chuyển đổi từ hệ thống tin học hiện tại, dùng các bảng có sẵn để bắt đầu.
 - Nếu ta phải xây dựng hệ thống từ đầu (ví dụ: cho một doanh nghiệp hoàn toàn mới), phác thảo qua trên giấy các biểu mẫu người sử dụng có thể dùng.
4. Dựa vào các biểu mẫu xây dựng ở bước 3, ta có thể phác thảo các bảng lên giấy. Nếu dữ liệu chưa được chuẩn hóa ngay, ta có thể bắt đầu bằng cách tạo ra một bảng dữ liệu lớn, phi chuẩn rồi sau đó tiến hành các bước chuẩn hóa.
5. Ta có thể tham khảo các giấy tờ có sẵn hoặc các báo cáo từ những hệ thống cũ. Đối với những hệ thống hiện tại không đáp ứng được yêu cầu người sử dụng thường các báo cáo quan trọng sẽ bị thiếu. Ta có thể tạo nhập các báo cáo này lên giấy.
6. Tiếp theo, phải đảm bảo rằng các bảng dữ liệu tạo ở bước 4 có chứa các dữ liệu của các mẫu biểu ở bước 5. Nếu các thông tin này chưa có, cần phải thêm vào bảng hoặc tạo ra bảng dữ liệu mới.
7. Trên giấy, ta đưa các bản ghi dữ liệu vào bảng đã phác thảo, cố gắng dùng dữ liệu thật nếu có thể.
8. Bây giờ ta có thể bắt đầu quá trình chuẩn hóa. Đầu tiên là xác định các khóa ứng viên cho mỗi bảng rồi chọn ra khóa chính. Chú ý phải chọn khóa chính nhỏ nhất, ổn định, đơn giản và phổ biến. Tốt nhất là mỗi bảng phải có một khóa chính!
9. Sau đó ta cần phải chọn khóa ngoại hoặc thêm vào bảng liên quan khóa ngoại nếu cần thiết. Tiếp theo ta thiết lập mối quan hệ giữa các bảng, chú ý phân biệt các quan hệ 1-1 hay 1-nhiều. Nếu tồn tại quan hệ nhiều-nhiều ta cần tạo các bảng quan hệ.
10. Bước tiếp theo ta xác định xem các bảng hiện đã ở dạng chuẩn một chưa. Các trường dữ liệu có đảm bảo tính đơn nhất chưa? Có tồn tại các nhóm dữ liệu lặp lại? Đưa dữ liệu về dạng chuẩn 1 (1NF).
11. Kiểm tra xem các bảng đã ở dạng chuẩn 2. Mỗi bảng chỉ mô tả một thực thể? Các trường không phải khóa chính đã phụ thuộc hoàn toàn vào khóa chính? Hay nói cách khác, trường khóa chính có thể được dùng để truy xuất các trường của bảng? Phân rã để được dạng chuẩn 2 (2NF). Nếu bảng có khóa chính tổng hợp ta cần phân rã bằng cách chia khóa chính này và các trường phụ thuộc vào mỗi phần của khóa chính vào mỗi bảng.

12. Kiểm tra xem các bảng đã ở dạng chuẩn 3. Có tồn tại các trường tính toán hay không? Có tồn tại sự phụ thuộc lẫn nhau của các trường không phải là khóa chính. Loại bỏ các trường tính toán. Loại bỏ các trường phụ thuộc lẫn nhau bằng cách tạo ra các bảng phụ tra cứu.
13. Dùng các bảng đã được chuẩn hóa ở bước 12, ta có thể xây dựng mối quan hệ giữa các bảng.
14. Tạo các bảng dùng Microsoft SQL Server. Nếu ta dùng Microsoft SQL Server, tạo quan hệ giữa các bảng bằng Enterprise Manager (dùng công cụ trợ giúp tạo sơ đồ quan hệ). Nhập dữ liệu mẫu vào các bảng.
15. Tạo ra các truy vấn, biểu nhập liệu và các báo cáo mẫu. Khi tạo các đối tượng này, các thiếu sót trong thiết kế sẽ xuất hiện. Sửa chữa, cập nhật thiết kế nếu cần thiết.
16. Tham khảo ý kiến người dùng bằng cách yêu cầu họ đánh giá các biểu nhập liệu và các báo cáo. Chúng có đáp ứng được yêu cầu của người sử dụng không? Nếu không ta cần phải sửa lại. Chú ý chuẩn hóa lại dữ liệu nếu cần thiết (các bước 8-12).
17. Đến đây ta có thể trở lại màn hình thiết kế bảng để thêm vào các ràng buộc về nghiệp vụ.
18. Tạo các biểu nhập liệu, báo cáo và truy vấn cuối. Phát triển ứng dụng. Sửa lại thiết kế nếu thấy cần thiết.
19. Yêu cầu người dùng chạy thử hệ thống. Cập nhật thiết kế nếu cần thiết.
20. Cuối cùng hệ thống đã sẵn sàng để triển khai.

Phá vỡ quy tắc: Khi nào cần phi chuẩn hóa

Đôi khi ta cần phá vỡ các quy tắc chuẩn hóa để tạo ra CSDL không chuẩn. Quá trình này thường do yêu cầu tăng hiệu năng của ứng dụng hoặc do người dùng yêu cầu. Tuy nhiên, nếu ta bỏ qua các quy tắc và quyết định phi chuẩn hóa dữ liệu, ta cần theo các hướng dẫn sau:

- Cần phải có lập luận hợp lý cho việc phi chuẩn.
- Ý thức được sự đánh đổi của quyết định trên.
- Cần phải ghi lại chi tiết quy trình phi chuẩn.
- Sửa đổi ứng dụng nếu cần thiết để tránh các sai sót.

Chú ý: Thiết kế CSDL là một thành phần quan trọng trong việc thiết kế ứng dụng. Nếu ta dành thời gian thiết kế CSDL hợp lý, ta đã tạo ra nền tảng ổn định cho việc phát triển ứng dụng.

Giai đoạn xây dựng khái niệm giúp ta có cái nhìn thực tế về CSDL cần thiết kế. Giai đoạn logic bao gồm các bước thiết kế cấu trúc CSDL, cấu trúc các bảng thông qua việc xây dựng mô hình quan hệ - thực thể (sẽ được giới thiệu dưới đây). Giai đoạn thiết kế vật lý bao gồm việc tạo các file vật lý trên thiết bị, thao tác và hiển thị dữ liệu.

Để thiết kế CSDL logic, ta có thể dùng một số cách. Một trong những cách tiếp cận mà chúng ta sẽ nghiên cứu là “Mô hình quan hệ-thực thể”. Ta sẽ thiết kế CSDL dùng mô hình E-R, trong đó các sơ đồ quan hệ thực thể sẽ được dùng để thể hiện mô hình logic của CSDL.

Mô hình quan hệ-thực thể

Mô hình quan hệ thực thể thể hiện toàn bộ hệ thống thông qua các thực thể có quan hệ với nhau.

Để hiểu rõ mô hình này và các thành phần của nó, ta xem một hệ thống cụ thể xử lý Khách hàng-Đơn đặt hàng (Customer-Order). Một khách hàng đặt nhiều hóa đơn hàng. Nhân viên xử lý đơn hàng kiểm tra hàng trong kho và xử lý đơn hàng. Với mỗi đơn đặt hàng, một mã đơn duy nhất được tạo. Các chi tiết đơn hàng như địa chỉ giao hàng, ngày giao hàng, ngày đặt hàng được ghi lại. Các chi tiết đơn hàng được lưu trong một bảng với các thông tin về các mặt hàng như tên hàng và giá.

Ta cần xác định các thực thể, các thuộc tính của chúng và mối quan hệ giữa các thực thể.

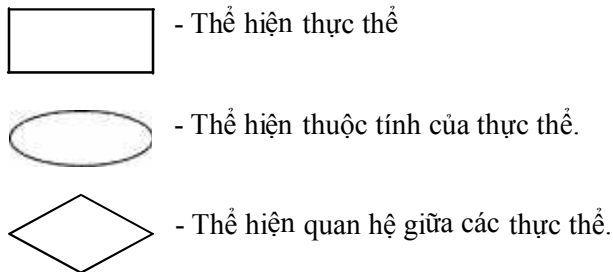
Trong ví dụ này “Khách hàng”(“Customer”), “Đơn đặt hàng” (“Order”) và “mặt hàng” (“Item”) là các thực thể. Các thực thể này cùng các thuộc tính được xác định và liệt kê trong bảng dưới đây.

Thực thể	Thuộc tính
‘Customer’	Customer Code, Customer Name, Address, Phone No.
‘Order’	Order No, Order Date, Customer Code, Item Code, Qty Ordered
‘Item’	Item Code, Item Name, Rate, Quantity On Hand, ReOrder Level

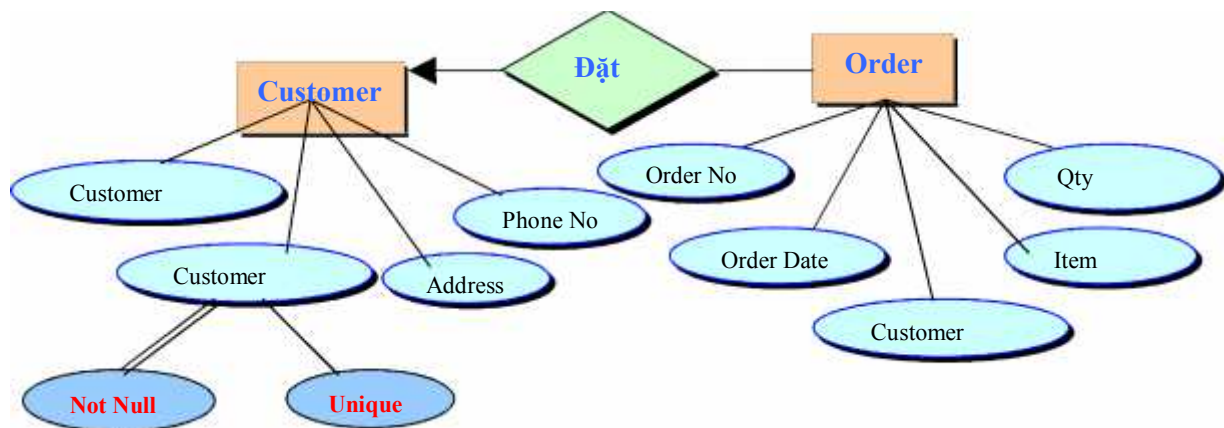
Các quan hệ tồn tại giữa các thực thể được mô tả dưới đây:

- Một ‘Customer’ **đặt** một ‘Order’
- Một ‘Order’ **chứa** nhiều ‘Items’

Ta dùng các biểu tượng sau để vẽ biểu đồ quan hệ thực thể (ERD).



Mô hình quan hệ thực thể của ví dụ trên được thể hiện như trong hình 2.1.



Hình 2.1: Sơ đồ quan hệ thực thể

Sơ đồ quan hệ thực thể trên mô tả quan hệ giữa ‘Customer’ và ‘Order’, đồng thời cũng thể hiện các thuộc tính của mỗi thực thể. Chú ý thuộc tính ‘Customer Code’ của thực thể ‘Customer’ có thêm hai thuộc tính khác là ‘Not Null’ and ‘Unique’. Cũng theo cách đó các thuộc tính khác cũng có thể được liên kết với các thuộc tính của các thực thể theo mô hình như trên.

Các loại quan hệ

Khi khách hàng đặt hàng ta thấy rằng một khách hàng có thể thực hiện nhiều đơn đặt hàng và một đơn đặt hàng có thể chứa nhiều mặt hàng. Trong những trường hợp như vậy ta nói mối quan hệ giữa ‘Customer’ và ‘Order’ là Một-Nhiều.

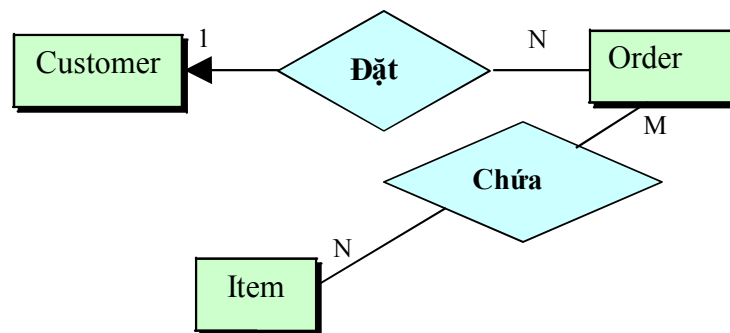
Mặt khác, quan hệ giữa ‘Item’ và ‘Item Code’ là Một-Một, vì một mặt hàng chỉ có một mã hàng mà thôi.

Cũng như vậy, quan hệ giữa *Order* và *Item* là Nhiều-Nhiều. Trong trường hợp này, một đơn hàng có thể có nhiều mặt hàng và một mặt hàng có thể thuộc nhiều đơn hàng.

Vậy với những điểm ở trên ta thấy tồn tại 3 loại quan hệ:

- Một – Một (được thể hiện trong ERD là 1:1)
- Một – Nhiều hay Nhiều – Một (trong ERD tương ứng là 1:N hay N:1)
- Nhiều – Nhiều (trong ERD là M:N)

Sơ đồ ERD thể hiện các loại quan hệ được trình bày trong hình 2.2.



Hình 2.2: Sơ đồ ERD hiển thị các loại quan hệ

Khi sơ đồ ERD được hoàn thành và người dùng đồng ý ta có thể bắt đầu tạo bảng dữ liệu. Vậy mối quan hệ giữa ERD và thiết kế bảng là gì? ERD nằm ở vị trí nào trong bức tranh thiết kế CSDL? Câu trả lời là các thực thể xác định được sẽ được thể hiện thành bảng, các thuộc tính trở thành các trường dữ liệu của các bảng tương ứng.

Sau khi đã xác định đầy đủ các thực thể và các thuộc tính, ta có thể chuyển thiết kế sang CSDL và chuẩn bị các bảng logic cho nó.

Ví dụ, các bảng cho ví dụ trên có thể được thể hiện từ thiết kế như dưới đây:

OrderID	Int
CustomerID	Char
EmployeeID	Char
OrderDate	Datetime
Datetime	Datetime
RequiredDate	Datetime
ShippedDate	Datetime

Bảng 2.1: Bảng Orders

ItemID	Int
ItemName	Char
SupplierID	Char
CategoryID	Char
QuantityPerUnit	Int
UnitPrice	Int
UnitsInStock	Int
UnitsOnOrder	Int
ReorderLevel	Int
Discontinued	Int

Bảng 2.2: Bảng Item

Chuẩn hóa

Chuẩn hóa là một kỹ thuật giúp người thiết kế nhóm các dữ liệu và đặt chúng trong các bảng phù hợp. Do vậy việc chuẩn hóa một CSDL là hết sức quan trọng trước khi ta bắt đầu làm việc với nó. Các dạng chuẩn có những quy tắc chỉ rõ các yêu cầu tạo một CSDL quan hệ.

Ba dạng chuẩn được mô tả như sau:

Dạng chuẩn 1(1NF)

Bảng dữ liệu thỏa mãn các đặc tính của một quan hệ (relation) được coi là ở dạng chuẩn 1. Một relation không thể chứa các thuộc tính tập hợp hay có nhiều giá trị. Dưới đây là quy tắc của dạng chuẩn 1.

Một quan hệ được coi là ở dạng chuẩn 1 khi và chỉ khi các miền giá trị của quan hệ chứa các giá trị đơn nhất.

Dạng chuẩn 2(2NF)

Điều kiện để đạt được dạng chuẩn 2 là bảng dữ liệu phải ở dạng chuẩn 1. Mục đích của dạng chuẩn 2 là đảm bảo rằng thông tin chứa trong quan hệ chỉ mô tả một thực thể duy nhất.

Chú ý: Một quan hệ ở dạng chuẩn 2 khi và chỉ khi nó ở dạng chuẩn 1 và tất cả các trường không phải khóa chính phải phụ thuộc hoàn toàn vào khóa chính của quan hệ.

Để hiểu rõ hơn định nghĩa dạng chuẩn 2 ở trên ta cần định nghĩa khái niệm *key attribute*. Mỗi thuộc tính của quan hệ tham gia vào ít nhất một khóa ứng viên (candidate key) được coi là *key attribute* của quan hệ. Tất cả các thuộc tính khác được gọi là *non-key*.

Dạng chuẩn 2 quy định rằng tất cả các thuộc tính không phải thành phần của khóa ứng viên phải phụ thuộc hoàn toàn vào khóa ứng viên.

Dạng chuẩn 3(3NF)

Mặc dù dạng chuẩn 2 đã loại bỏ được các bất thường có thể xuất hiện trong các bảng chưa ở dạng chuẩn 1, nhưng không phải đã loại bỏ được tất cả và cần thiết phải thực hiện dạng chuẩn hóa tiếp theo để đảm bảo loại bỏ hết những bất thường đó. Các tính bất thường này có thể xảy ra

do dạng chuẩn 2 có thể chứa những thuộc tính không liên quan trực tiếp đến thực thể được mô tả bởi các khóa ứng viên trong quan hệ. Dạng chuẩn 3 được mô tả dưới đây:

Chú ý: Một quan hệ R được coi là ở dạng chuẩn 3 khi và chỉ khi nó ở dạng chuẩn 2 và các thuộc tính không phải là khóa chính của R phải phụ thuộc vào mỗi khóa ứng viên của R.

Để hiểu rõ hơn về dạng chuẩn 3 chúng ta tìm hiểu khái niệm *sự phụ thuộc ngoại suy (transitive dependence)*, khái niệm này dựa vào một trong những tiên đề của Armstrong. Coi *A*, *B* và *C* là ba thuộc tính của quan hệ *R*, ta có các mối quan hệ sau $A \rightarrow B$ và $B \rightarrow C$. Từ các mối quan hệ này ta suy ra là $A \rightarrow C$. Như đã đề cập ở trên, mỗi quan hệ giữa $A \rightarrow C$ là phụ thuộc ngoại suy (*transitive*).

Dạng chuẩn 3 khác với dạng chuẩn 2 ở chỗ tất cả các thuộc tính không phải khóa chính trong dạng chuẩn 3 phải phụ thuộc trực tiếp vào khóa ứng viên của mỗi quan hệ. Nếu có thuộc tính phụ thuộc vào trường khóa theo quan hệ ngoại suy điều đó có ý nghĩa là các thuộc tính đó mô tả không chỉ trường khóa mà còn mô tả thuộc tính không phải khóa. Do đó thông tin không phụ thuộc trực tiếp vào khóa chính mặc dù rõ ràng thuộc tính này có quan hệ với khóa chính.

Các quy tắc xây dựng CSDL chuẩn hóa

Ba quy tắc đơn giản mô tả dưới đây giúp thiết kế các bảng dữ liệu chuẩn hóa đáp ứng các yêu cầu trên.

Quy tắc 1- Loại bỏ các nhóm dữ liệu lặp lại

Tạo bảng riêng cho mỗi tập thuộc tính lặp lại và gán cho mỗi bảng một khóa chính. Chúng ta xem lại ví dụ về CSDL Customers-Order.

Order ID 1..n
Orderdate
CustomerID 1..n
Orderdate
Requireddate
Shippeddate
Quantity
Discount
Company name
Address
City
Product ID 1..n
Productname
UnitPrice

Bảng 2.3: Dữ liệu chưa chuẩn hóa trong CSDL ví dụ

Trong danh sách dữ liệu gốc, mỗi một mô tả về khách hàng theo sau là một danh sách các thuộc tính liên quan đến khách hàng đó. Một khách hàng có thể đặt 10 món hàng nhưng cũng có người chỉ đặt 1 món hàng. Để tìm lời giải cho câu hỏi, "Khách hàng A có mua món hàng B hay không?" trước tiên chúng ta cần tìm các chi tiết đặt hàng của khách hàng A, sau đó ta phải duyệt qua danh sách hàng hóa đó. Đây là phương pháp không hiệu quả và rất rườm rà.

Vấn đề này được giải quyết nếu ta đưa dữ liệu về chi tiết đơn hàng sang một bảng khác. Tách các nhóm dữ liệu lặp lại trong thông tin về chi tiết đơn hàng của khách hàng sẽ cho ta dữ liệu ở dạng chuẩn 1. Trường *Customer id* trong bảng *Orders* trùng với khóa chính trong bảng *Customer*, có ta khóa ngoại liên kết hai bảng. Lúc này ta có thể giải quyết câu hỏi trên bằng cách truy xuất trực tiếp: kiểm tra xem giá trị customer id của khách hàng A và giá trị ID của mặt hàng B có cùng nằm trong bảng *Orders* hay không.

Customer Table

Customer ID	<i>Primary Key</i>
Company Name	
Address	
City	

Orders Table

Customer ID	<i>Foreign Key</i>
Order ID Item	
ID Itemname	
Unitprice	
Order date	
Required date	
...	

Bảng 2.4: DẠNG CHUẨN 1

[Chú ý trong bảng *Orders*, trường *Customer ID* là khóa ngoại.]

Quy tắc 2- Loại bỏ dữ liệu dư thừa

Nếu một trường chỉ phụ thuộc vào một phần của trường khóa chính chứa nhiều giá trị, đưa dữ liệu đó sang một bảng.

Một phần của bảng *Orders* như sau:

BẢNG ORDERS				
Order ID	Customer ID	ItemID	Itemname	Unitprice
1	A	101	Rivets	10
2	B	102	Bolts	15
3	C	101	Rivets	10
4	D	101	Rivets	10

Bảng 2.5: Bảng Order Details

Với cấu trúc bảng *Orders* trên, các chi tiết mặt hàng xuất hiện lặp lại với mỗi khách hàng đặt cùng một món hàng. Sẽ khả thi hơn nếu ta chỉ lưu trường *Item ID*.

Nếu ta muốn định nghĩa lại một mặt hàng, có nghĩa là phải cung cấp giá trị *itemID* mới, sự thay đổi này phải thực hiện cho mọi đơn đặt hàng có chứa mặt hàng đó! Nếu chúng ta bỏ qua một số bản ghi, ta sẽ có các đơn đặt hàng có cùng mặt hàng nhưng lại khác ID. Đây được gọi là **Cập nhật bất thường (Update anomaly)**.

Giả sử mặt hàng cuối cùng hết hàng và không được sản xuất tiếp – nghĩa là mặt hàng đó đã lỗi thời. Các bản ghi đó sẽ bị xóa khỏi CSDL và như vậy mặt hàng đó sẽ không được lưu ở bất kỳ đâu thậm chí không được lưu trong lịch sử mua hàng của khách hàng! Đây được gọi là **Xóa bất thường (Delete anomaly)**. Để tránh lỗi này ta cần có **dạng chuẩn 2**.

Để đạt được dạng chuẩn 2, tách các trường phụ thuộc vào cả 2 phần của khóa chính khỏi các trường chỉ phụ thuộc vào trường Item ID. Kết quả ta được 2 bảng: bảng “**Items**” chứa tên, giá và các chi tiết khác của mặt hàng; và bảng “**Orders**” liệt kê danh sách các mặt hàng đặt bởi mỗi khách hàng.

Customer Table

CustomerID	<i>Primary Key</i>
CompanyName	
Address	
City	

Orders Table

CustomerID	<i>Foreign Key</i>
OrderID	
ItemID	
Required date	
Shippeddate	
Quantity	
Discount	

Item Table

ItemID	<i>Primary Key</i>
Itemname	
Unitprice	

Bảng 2.6: DẠNG CHUẨN 2

Quy tắc 2 – Loại bỏ các trường không phụ thuộc vào khóa chính

Nếu các trường không tham gia mô tả khóa chính, tách chúng sang một bảng khác.

OrderID
CustomerID
ItemID
Requireddate
Shippeddate
Quantity
Discount

Bảng 2.7: Bảng Orders

Bảng Orders thỏa mãn dạng chuẩn 1 do không còn chứa các dữ liệu lặp lại. Nó cũng thỏa mãn dạng chuẩn 2 do không tồn tại khóa chính nhiều giá trị. Nhưng khóa chính của bảng là Order ID, và các trường **quantity** và **discount** của mỗi mặt hàng không cần được lưu trong bảng này. Để đạt được dạng chuẩn 3, chúng phải được đưa sang một bảng khác. Do các trường này mô tả chi tiết đơn hàng nên bảng mới là Order Details phù hợp nhất để lưu dữ liệu này. Do các trường này mô tả mặt hàng và chi tiết đơn hàng nên khóa chính sẽ là kết hợp của hai trường Item ID và Order ID.

Customer Table

CustomerID	<i>Primary Key</i>
CompanyName	
Address	
City	

Orders Table

CustomerID	<i>Foreign Key</i>
OrderID	
ItemID	
Requireddate	
Shippeddate	

Item Table

ItemID	<i>Primary Key</i>
Itemname	
Unitprice	

Order Details

OrderID
CustomerID
ItemID
Requireddate
Shippeddate
Quantity
Discount

Bảng 2.8: DẠNG CHUẨN 3

1. Liệt kê các thuộc tính của thực thể ‘Order’ và ‘Item’ sau đó vẽ sơ đồ ERD để thể hiện mối quan hệ giữa 2 thực thể này.
2. Trong hệ quản trị thư viện, người ta có thể mượn và trả sách trong khoảng thời gian 7 ngày. Người trả mượn sẽ bị phạt \$2 một ngày. Sách được phân loại dựa theo nội dung (ví dụ: ‘Fiction’, ‘Romance’, ‘Computers’ v.v.). Các thông tin về mượn sách cần được lưu trữ. Hãy vẽ sơ đồ ERD cho hệ thống này và thiết kế CSDL.
3. Chuẩn hóa CSDL sau dùng 3 quy tắc chuẩn hóa

Bouquet No
Bouquet Name
No of Flowers
Flower Name
Price of flower
Value of bouquet
Company Name
Company Address
Delivery date

Phần III

Xem bài toán sau – dịch vụ bán vé cho một hãng hàng không. Dịch vụ bán vé của hãng hàng không **Fly Safe Airways** cung cấp dịch vụ đặt vé, hủy vé và quản lý lịch các chuyến bay. Trước hết ta xác định các yêu cầu về đặt và hủy vé. Có một số khía cạnh ta cần xem xét. Trước hết, **Fly Safe Airways** cung cấp các chuyến bay nội địa (Domestic) và quốc tế (International). **Fly Safe Airways** cung cấp dịch vụ đặt vé cho các hãng hàng không sau:

- Lufthansa
- Cathay Pacific
- British Airways
- Malaysian Airlines

Các hạng vé trên các hãng này như sau:

Các chuyến bay nội địa có các hạng **Economy** và **Executive**, các chuyến bay quốc tế có các hạng **Economy**, **Executive** và **First Class**.

Với các chuyến bay quốc tế hay nội địa, các món ăn phục vụ cũng khác nhau. Đối với các chuyến bay nội địa, các món ăn gồm có: **Vegetarian** và **Non-Vegetarian**. Các chuyến bay quốc tế được phục vụ các món: **Vegetarian**, **Non-Vegetarian**, **Chinese Vegetarian**, **Chinese Non-Vegetarian**, **Continental Vegetarian**, **Continental Non-Vegetarian**.

Một số dịch vụ khác như **Child Care**, **Nurse** và **Wheel chair** cũng được cung cấp trên các chuyến bay nội địa và quốc tế.

Để đảm bảo các chức năng của hệ thống được thực thi một cách hợp lý ta cần phải quản lý rất nhiều thông tin và thực hiện các thay đổi khi cần thiết.

1. Xác định các thực thể trong trường hợp cụ thể trên và vẽ sơ đồ quan hệ thực thể.
2. Thiết kế các bảng cho trường hợp cụ thể trên.