Hướng dẫn thực hiện dạng chuẩn 3NF

Tác giả: Fred Coulson

Mục lục

Về bản dịch

Người dịch: Phan Anh Vũ. Lớp ĐT12.K49. Trường ĐH Bách Khoa HN.
Email: virces931511@yahoo.com
Website: http://cntt.tv

Xin giành bản dịch này tặng anh em lớp ĐT12.K49 nói riêng, bà con khoa Điện tử Viễn thông K49 trường ĐH Bách Khoa HN nói *hơi riêng* với lời chúc anh em thi tốt môn Kỹ thuật phần mềm (thi lại tốn 5k đấy). Với ai không ôn thi môn KTPM nhưng quan tâm và bước đầu tìm cách chuẩn hóa CSDL của riêng mình, đây có lẽ sẽ là tài liệu bắt đầu tốt nhất.

Theo quan điểm của tôi thì đây là một tutorial rất thú vị, đề cập đến khá nhiều khía cạnh lắt léo trong quá trình chuẩn hóa. Tuy nhiên bản dịch vì nhiều lí do (tôi đang ôn thi Tư tưởng HCM *lần 1* chẳng hạn) nên chất lượng còn hạn chế, mong nhận được góp ý để hoàn thiện dần.

Cảm ơn đại ca Fred Coulson tốt bụng đã đồng ý cho dịch và phát tán tài liệu này với lời hứa sẽ host bản dịch trên trang của đại ca. Chúc đại ca sức khỏe, chụp nhiều ảnh đẹp và viết nhiều tutorial hay.

Còn bây giờ, nào mình cùng đi xe buýt, nào mình cùng đi thi nhé …

Giới thiệu

Đây là một hướng dẫn rất ngắn gọn giành cho những người mới bắt đầu bước vào lĩnh vực chuẩn hóa cơ sở dữ liệu. Vì rất khó để diễn tả bằng lời nên tôi dùng nhiều nhất có thể các hình ảnh, biểu đồ.

Để trình bày các qui tắc chính trong quá trình chuẩn hóa, tôi dựa theo ví dụ cổ điển về **Hóa đơn** (Invoice) và chuẩn hóa nó về dạng 3NF (Third Normal Form). Trong quá trình đó, chúng ta sẽ hình thành Sơ đồ liên kết thực thể (Entity Relationship Diagram - ERD) cho cơ sở dữ liệu.

Chú ý: Đây không phải là hướng dẫn chi tiết để thiết kế và thực thi một cơ sở dữ liệu thực tế. Bạn không phải làm theo tuyệt đối như các hình minh họa vì nó chỉ minh họa cho việc các dữ liệu thô được sắp xếp lại như thể nào trong quá trình chuẩn hóa.

Có thể có người không thích cách đó. Tôi cũng không trình bày các vấn đề liên quan đến điểm lợi, hại của việc chuẩn hóa. Ai quan tâm đến các chủ đề đó, xin xem danh sách tham khảo ở cuối tài liệu này.

Thường thì khi ai đó bắt tay vào thiết kế CSDL, trong đầu anh/cô ta đã có một mô hình chuẩn hóa phần nào rồi – chuẩn hóa là một cách tự nhiên để nhận ra mối quan hệ của dữ liệu và không cần kiến thức đặc biết về toán học, tập hợp … Trong thực tế, nhiều khi còn phải "phi chuẩn hóa" (de-normalize) CSDL – nhưng vấn đề này nằm ngoài nội dung bài viết.

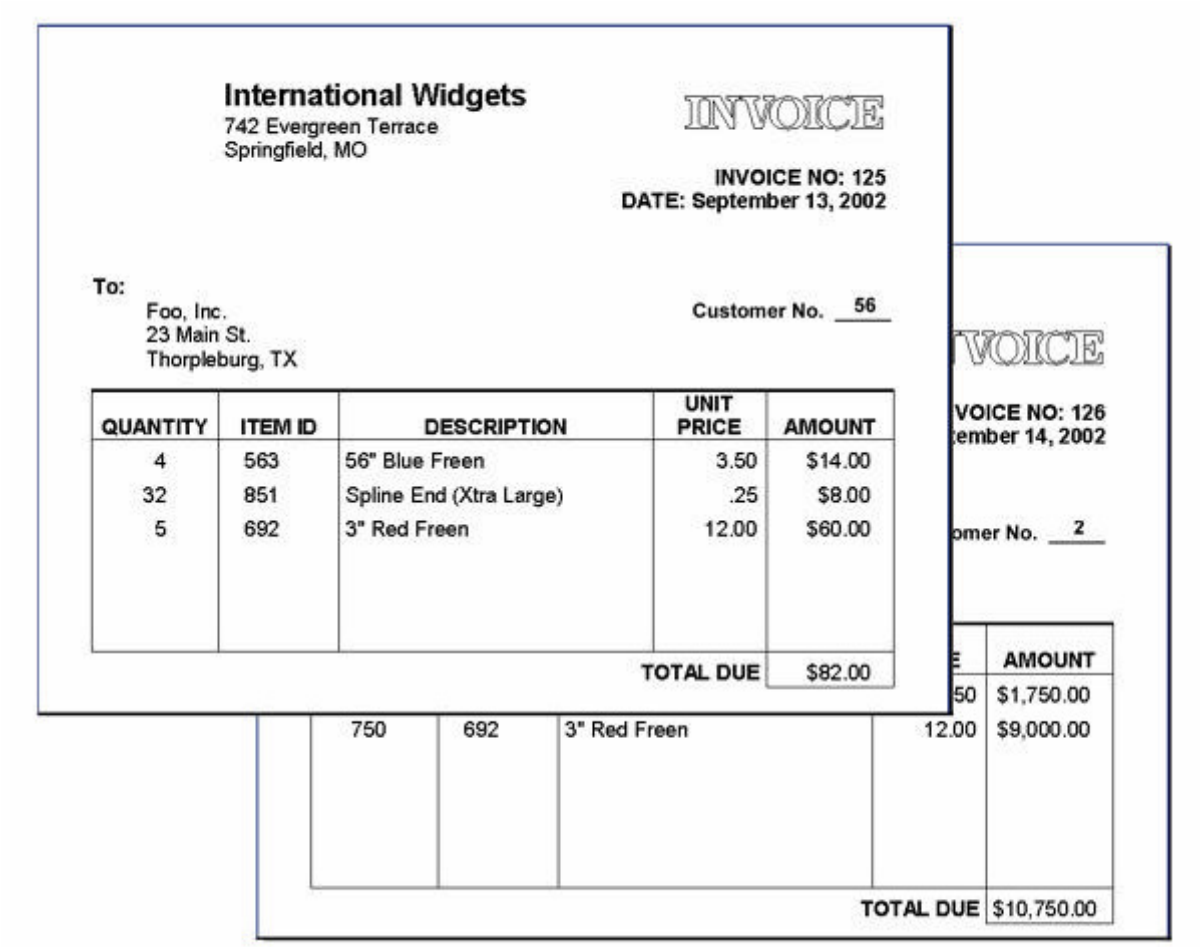
Để bắt đầu: Trước tiên, xin nhớ nằm lòng 3 qui tắc sau về các dạng chuẩn. Nhớ trước, bạn sẽ hiểu sau:

1. Không có phần tử/nhóm các phần tử lặp.
2. Không có phụ thuộc hàm không đầy đủ vào khóa ứng cử.
3. Không có phụ thuộc hàm vào các thuộc tính không khóa.

Bài toán: Quản lí Hóa đơn

Cho mẫu hóa đơn như **Hình A**).

Hình A: Hóa đơn



International Widgets
742 Evergreen Terrace
Springfield, MO

INVOICE

INVOICE NO: 125
DATE: September 13, 2002

To:
Foo, Inc.
23 Main St.
Thorpleburg, TX

Customer No. 56

QUANTITY	ITEM ID	DESCRIPTION	UNIT PRICE	AMOUNT
4	563	56" Blue Freen	3.50	$14.00
32	851	Spline End (Xtra Large)	.25	$8.00
5	692	3" Red Freen	12.00	$60.00
			TOTAL DUE	$82.00

VOICE
VOICE NO: 126
ember 14, 2002
omer No. 2

				AMOUNT
			50	$1,750.00
750	692	3" Red Freen	12.00	$9,000.00
			TOTAL DUE	$10,750.00

Đây là mẫu hóa đơn quen thuộc trong kinh doanh. Tất cả các thông tin trên đó đều quan trọng. Chúng ta đưa các thông tin đó vào CSDL như thế nào đây?

Ai đó chưa biết về CSDL quan hệ có thể đưa các thông tin đó vào spreadsheet trong Excel như sau:

Hình A-1: Bảng hóa đơn

orders.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Invoice No.	Date	Cust. No.	Cust. Name	Cust. Address	Cust. City	Cust. State	Item ID	Item Descri	Item Qty.	Item Price	Item Total	Order Total Price
2	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	563	56" Blue Fre	4	$ 3.50	$ 14.00	$ 82.00
3								851	Spline End (	32	$ 0.25	$ 8.00	$ 82.00
4								652	3" Red Free	5	$ 12.00	$ 60.00	$ 82.00
5	126	9/14/2002	2	Freens R Us	1600 Pennsylvania Avenu	Washington	DC	563	56" Blue Fre	500	$ 3.50	$1,750.00	$ 10,750.00
6								652	3" Red Free	750	$ 12.00	$9,000.00	$ 10,750.00

Không tồi! Bàng này ghi lại tất cả các đơn hàng được mua bởi tất cả các khách hàng. Nhưng điều gì xảy ra nếu ta muốn lấy các thông tin phức tạp như:

- Có bao nhiêu *3" Red Freens* mà *Freens R Us* đặt trong năm 2002?

- Tổng số *56" Blue Freens* được bán ở Texas?
- Những sản phẩm nào được bán vào ngày 14 tháng 7 năm 2003?

Bảng trên càng nhiều thông tin thì việc trả lời các câu hỏi trên càng khó khăn. Trong nỗ lực đưa dữ liệu về trạng thái mong muốn để trả lời các câu hỏi kiểu như trên, chúng ta đang bắt đầu việc **chuẩn hóa CSDL** (normalization).

Dạng chuẩn thứ 1 (1NF): Không có phần tử/nhóm phần tử lặp

Nhìn vào hàng 2, 3, 4 của bảng trong Hình A-1, ta thấy tất cả các dữ liệu liên quan đến một hóa đơn (Invoice #125). Theo thuật ngữ CSDL, nhóm các hàng này được gọi là một hàng đơn CSDL (a single *database row)*. Một hàng đơn CSDL ở đây được tạo bởi ba hàng trong bảng ở Hình A-1.

Dạng chuẩn 1NF muốn chúng ta triệt tiêu các phần tử lặp. Chúng là các phần tử nào?

Một lần nữa, để ý hóa đơn đầu tiên (#125) trong Hình A-1. Ô H2, H3, và H4 chứa một danh sách các số Item ID. Đây là một cột trong hàng CSDL đầu tiên của chúng ta. Tương tự, I2-I4 hình thành một cột khác; tương tự với J2-J4, K2-K4, L2-L4, và M2-M4. Các cột trong CSDL thường được gọi là **thuộc tính** (attributes) (hàng/cột có cách gọi khác là bộ/thuộc tính).

Để ý thấy các cột này chứa danh sách các giá trị. Rõ ràng là các danh sách như thế vi phạm luật chuẩn 1NF: 1NF không cho phép danh sách hay chuỗi giá trị như vậy tồn tại trong một cột CSDL. 1NF đòi hỏi **tính nguyên tố** - tức là sự không thể phân chia một thuộc tính thành các phần nhỏ hơn.

Vì thế chúng ta cần phải loại bỏ sự lặp lại thông tin về **item** trong hàng giành cho Hóa đơn #125. Trong Hình A-1, đó là các ô sau:

- Từ H2 đến M2
- Từ H3 đến M3
- Từ H4 đến M4

Tương tự, chúng ta cũng thấy hiện tượng trùng lặp dữ liệu trong hàng giành cho Hóa đơn #126. Chúng ta có thể chuẩn hóa sang dạng 1NF để đạt được tính nguyên tố một cách dễ dàng như sau – cho mỗi *item* một hàng riêng biệt (thường gọi là cách làm phẳng).

Hình A-2: làm phẳng bảng dữ liệu

orders.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Invoice No.	Date	Cust. No.	Cust. Name	Cust. Address	Cust. City	Cust. State	Item ID	Item Descri	Item Qty.	Item Price	Item Total	Order Total Price
2	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	563	56" Blue Fre	4	$ 3.50	$ 14.00	$ 82.00
3	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	851	Spline End (	32	$ 0.25	$ 8.00	$ 82.00
4	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	652	3" Red Free	5	$ 12.00	$ 60.00	$ 82.00
5	126	9/14/2002	2	Freens R Us	1600 Pennsylvania Avenu	Washington	DC	563	56" Blue Fre	500	$ 3.50	$1,750.00	$ 10,750.00
6	126	9/14/2002	2	Freens R Us	1600 Pennsylvania Avenu	Washington	DC	652	3" Red Free	750	$ 12.00	$9,000.00	$ 10,750.00

Có thể có ai đó phản đối: Chúng ta đang cố gắng làm giảm sự trùng lặp dữ liệu, nhưng ở đây thậm chí chúng ta đang làm ngược lại! Dữ liệu về Khách hàng bị trùng lặp!

Xin đừng lo lắng về điều đó. Sự trùng lặp đó sẽ được giải quyết khi chúng ta đi tới dạng chuẩn 3NF. Xin hãy kiên nhẫn; đây là một bước chúng ta cần phải đi qua để đến kết quả cuối cùng.

Đến đây chúng ta mới chỉ đi được một nửa chặng đường để đạt được dạng chuẩn 1NF. Dạng chuẩn 1NF giải quyết 2 vấn đề:

1. Mỗi hàng phải không chứa nhóm lặp (**Tính nguyên tố**).
2. Mỗi hàng phải có một thuộc tính nhận dạng duy nhất (**Khóa chính**)

Chúng ta đã giải quyết xong tính nguyên tố. Để giải quyết vấn đề Khóa Chinh, chúng ta cần phải chuyển toàn bộ dữ liệu sang một hệ quản trị CSDL quan hệ (RDBMS). Ở đây, tôi dùng Microsoft Access để tạo bảng **orders** như **Hình B**:

Hình B: Bảng orders

orders : Table

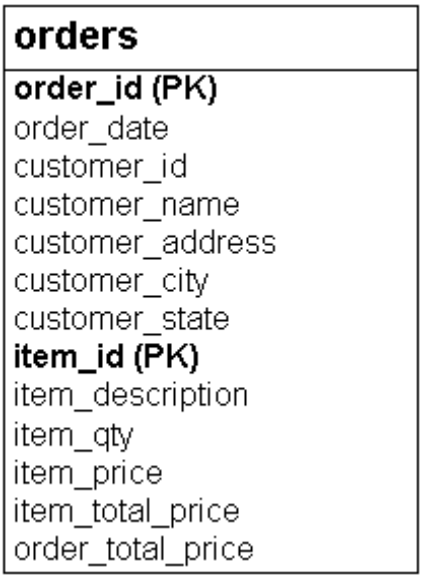
order_id	order_date	customer_id	customer_name	customer_addre	customer_city	customer_	item_id	item_description	item_qty	item_price	item_total_price	order_total_price
125	9/13/2002	56	Foo, Inc.	23 Main St., Th	Thorpleburg	TX	563	56" Blue Freen	4	$3.50	$14.00	$82.00
125	9/13/2002	56	Foo, Inc.	23 Main St., Th	Thorpleburg	TX	851	Spline End (Xtra	32	$0.25	$8.00	$82.00
125	9/13/2002	56	Foo, Inc.	23 Main St., Th	Thorpleburg	TX	652	3" Red Freen	5	$12.00	$60.00	$82.00
126	9/14/2002	2	Freens R Us	1600 Pennsylva	Washington	DC	563	56" Blue Freen	500	$3.50	$1,750.00	$10,750.00
126	9/14/2002	2	Freens R Us	1600 Pennsylva	Washington	DC	652	3" Red Freen	750	$12.00	$9,000.00	$10,750.00

Record: 6 of 6

Bảng này cũng khá giống bảng trong Excel nhưng điểm khác là trong RDBMS, chúng ta có thể tạo một **khóa chính**. Khóa chính là một cột (hoặc nhóm cột) giúp xác định duy nhất một hàng. Như nhìn thấy trong hình B, không có một cột đơn nào có thể dùng để xác định duy nhất các hàng. Tuy nhiên, nếu chúng ta kết hợp 2 cột **order_id** và **item_id** thì được: không có hai hàng nào có giá trị order_id và item_id giống nhau. Vì thế, kế hợp hai cột đó với nhau, chúng ta có khóa chính của bảng Orders. Chúng ta gọi hai cột đó là **khóa gộp (concatened key).**

*Một giá trị, giúp xác định duy nhất một hàng gọi là **khóa chính**. Khi giá trị đó được tạo bởi hơn một cột thì ta gọi chúng là **concatenated primary key**.*

Cấu trúc của bảng Order có thể được biểu diễn trong **Hình C** ở bên:
Hai thuộc tính hình thành nên khóa chính được kí hiệu **PK**. Hình C cũng chính là **Lược đồ liên kết thực thể** (Entity Relationship Diagram - or ERD).
CSDL của chúng ta bây giờ đã thỏa mãn hai yêu cầu của 1NF: **tính nguyên tố** và **tính duy nhất**. Đó là hai điều kiện cơ bản nhất của CSDL quan hệ.

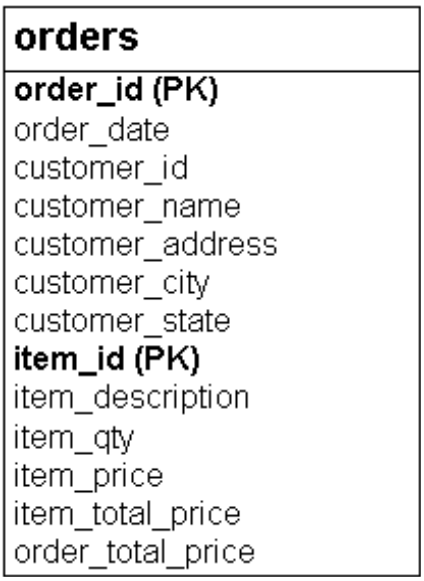


Tiếp theo là gì?

Dạng chuẩn thứ 2 (2NF): Không có phụ thuộc hàm không đầy đủ vào khóa chính.

Bầy giờ, chúng ta tìm các phụ thuộc hàm không đầy đủ vào khóa chính để loại bỏ chúng. Với các bảng có khóa chính được tạo bởi hơn một thuộc tính, các thuộc tính không nằm trong khóa chính phải phụ thuộc hàm đầy đủ vào khóa chính mà không được phép tồn tại các thuộc tính chỉ phụ thuộc vào một phần của khóa chính. Nếu có thuộc tính nào chỉ phụ thuộc một phần vào khóa chính thì bảng đó chưa đạt dạng chuẩn 2NF.

Để hiểu rõ, chúng ta xem xét từng thuộc tính của bảng **Orders**. Với mỗi thuộc tính, chúng ta sẽ đặt câu hỏi: *Liệu thuộc tính này có thể tồn tại mà không cần một hay nhiều thuộc tính nào đó nằm trong khóa chính không?* Nếu câu trả lời là "có" – dù chỉ một – thì bảng chưa đạt chuẩn 2NF.



Xem lại Hình C ở bên để nhớ lại cấu trúc bảng. Đầu tiên, nhắc lại ý nghĩa của hai thuộc tính làm nên khóa chính:

- **order_id** xác định duy nhất một hóa đơn.
- **item_id** xác định duy nhất một *item* trong kho. Đây có thể là mã số của linh kiện, mã số hàng trong kho, số SKU, sô UPC, …

Chúng ta sẽ không phân tích hai thuộc tính đó (vì chúng là thành phần của khóa chính). Bây giờ, ta sẽ xem xét các thuộc tính còn lại.

order_date là ngày lập hóa đơn. Rõ ràng là thuộc tính này phụ thuộc vào **order_id**; ngày lập hóa đơn thì phải liên quan đến hóa đơn, nếu không nó chỉ là một ngày bình thường. Nhưng ngày lập hóa đơn có thể tồn tại mà không cần **item_id**?

Câu trả lời đơn giản là có thể: ngày hóa đơn chỉ phụ thuộc vào order_id, không phụ thuộc vào item_id. Một số có thể phản đối, cho rằng làm như thế tức là có thể tạo ra một hóa đơn mà không có *item* nào (một hóa đơn rỗng). Nhưng đó không phải là vấn đề của chúng ta. Chúng ta đang xem xét ở đây là liệu một hóa đơn nào đó, lập vào một ngày nào đó có phụ thuộc vào một *item* nào đó không? Rõ ràng là không. Vấn đề làm sao để không tồn tại hóa đơn rỗng là một "qui tắc nghiệp vụ" (business rule) được thực hiện, kiểm tra mở chương trình; đó không phải vấn đề mà chuẩn hóa giải quyết.

Như vậy, order_date không thỏa mãn dạng chuẩn 2NF. ✘

Do đó, bảng Orders không đạt 2NF. Bây giờ hãy xem xét các thuộc tính còn lại. Chúng ta cần tìm tất cả các thuộc tính không thỏa mãn 2NF để xử lí.

customer_id là số ID của khách hàng. Nó có phụ thuộc vào **order_id**? Không: một khách hàng có thể tồn tại mà không cần mua hàng. Nó có phụ thuộc vào **item_id**? Không: với cùng lí do. Đây là một điều thú vị: customer_id (cùng với các thuộc tính customer_* khác) không phụ thuộc vào customer_id lẫn order_id, tức là không phụ thuộc vào bất cứ thuộc tính nào của khóa chính). Chúng ta sẽ làm gì với chúng? Chúng ta chỉ quan tâm tới chúng khi xem xét dạng chuẩn 3NF. Bây giờ chúng ta đánh dấu chúng là "không rõ ràng" (unknown). ?

item_description là miêu tả về hàng hóa. Rõ rang là nó phụ thuộc vào **item_id**. Nhưng nó có thể tồn tại mà không cần **order_id**? Có! Một *item* có thể tồn tại trong kho mãi mãi, mà không bao giờ được bán … Nó độc lập với hóa đơn. Như vậy, item_description không thỏa điều kiện của 2NF. ✗

item_qty là số lượng một mặt hàng được yêu cầu trong một hóa đơn. Rõ rang thuộc tính này phụ thuộc vào cả hai thuộc tính của khóa chính. Chúng ta chỉ có thể nói "5 cái máy tính" hay "6 cái TV" chứ không thể nói "10 cái không gì cả" (ít nhất là trong thiết kế CSDL). Số lượng một hàng hóa được yêu cầu trong một hóa đơn không thể tồn tại không có hóa đơn. Như vậy thuộc tính này thỏa mãn 2NF. ✓

item_price tương tự như **item_description**. Nó chỉ phụ thuộc vào **item_id** mà không phụ thuộc vào **order_id**, nên nó không thỏa mãn 2NF. ✗

item_total_price hơi đặc biệt. Một mặt, nó có vẻ như phụ thuộc vào cả **order_id** lẫn **item_id**, tức là thỏa mãn 2NF. Mặt khác, nó là một giá trị rút ra từ item_qty và item_price. Chúng ta sẽ xử lí thể nào? Trong thực tế, trường này không liên quan đến CSDL của chúng ta. Nó có thể dễ dàng được tạo ra ben ngoài CSDL; thêm nó vào CSDL sẽ gây dư thừa. Do đó chúng ta sẽ bỏ nó đi.

order_total_price, là tổng tất cả các item_total_price lại là một giá trị rút ra nữa nên chúng ta sẽ bỏ thuộc tính này.

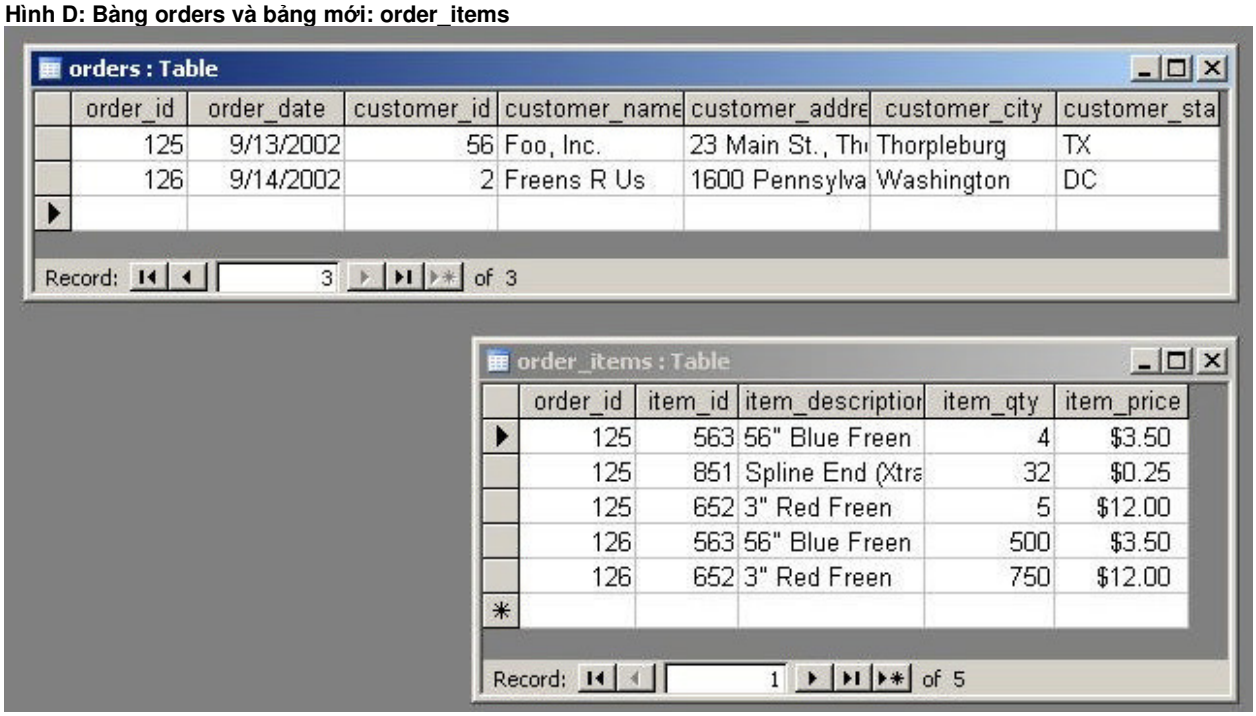
Đây là bảng phân tích 2NF của chúng ta:

Chúng ta sẽ làm gì với một bảng không thỏa mãn 2NF như thế?

Trước tiên, lấy ra nửa sau của khóa chính (**item_id**) và đưa nó vào một bảng khác. Các thuộc tính khác phụ thuộc vào item_id – đầy đủ hoặc không đầy đủ - cũng đưa luôn vào bảng mới này. Chúng ta sẽ gọi bảng mới này là **order_items** (xem **Hình D**).

Các thuộc tính còn lại – gồm các thuộc tính chỉ phụ thuộc vào nửa đầu của khóa chính (order_id) và các thuộc tính chưa xác định – giữ nguyên.

Hình D: Bảng orders và bảng mới: order_items

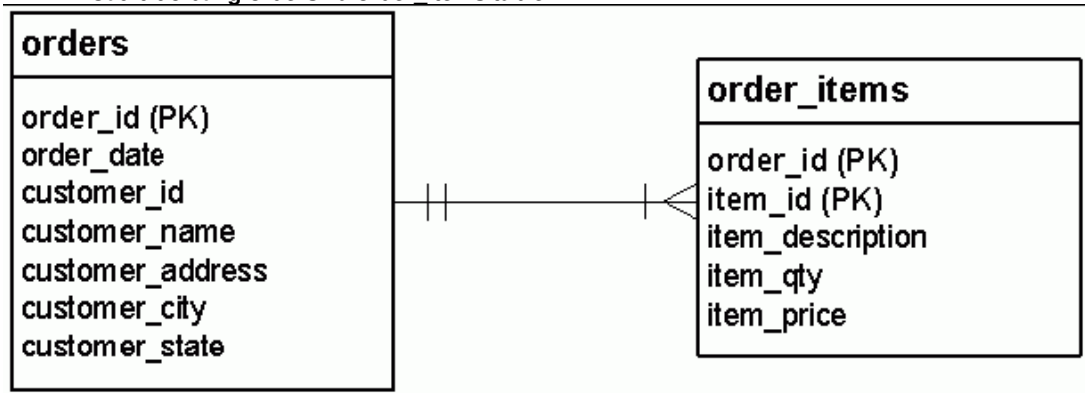


Có một vài điểm cần chú ý:

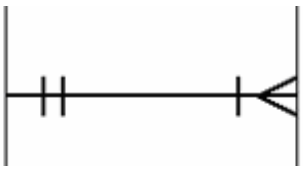
1. Chúng ta phải đưa thuộc tính **order_id** vào bảng **order_items** (để xác định xem mỗi order_item thuộc về order nào.
2. Bảng **orders** có ít thuộc tính hơn trước.
3. Khóa chính của bảng **orders** chỉ gồm một thuộc tính: **order_id**.
4. Khóa chính của bảng **order_items** gồm hai thuộc tính.

Sau đây là cấu trúc các bảng (**Hình E**):

Hình E: Cấu trúc bảng orders và order_items table



Nếu bạn chưa quen đọc Lược đồ liên kết thực thể thì xin để ý vào đường nối giữa hai bảng. Đường nối này



có nghĩa là:

- *Mỗi order có thể có một hoặc nhiều order-item, nhưng phải có ít nhất một;*
- *Mỗi order-item có thể thuộc về một và chỉ một order.*

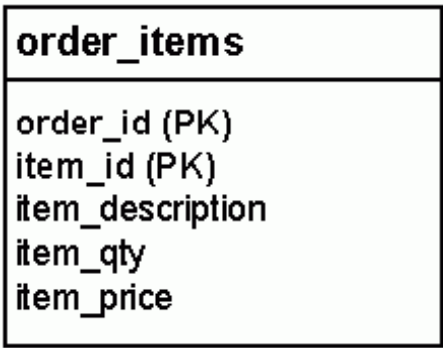
Dạng chuẩn thứ 2 (2NF): Pha thứ II

Nếu bạn nghĩ rằng chúng ta đã đạt chuẩn 2NF thì chờ đã, vẫn còn!

Nhớ rằng dạng chuẩn 2NF áp dụng cho các bảng có khóa chính hợp thành bởi hơn một thuộc tính. Bây giờ bảng **orders** có khóa chính là khóa đơn, bảng này đã dạt dạng chuẩn 2NF. Xin chúc mừng!

Tuy nhiên, bây giờ, bảng **order_items** lại có khóa chính tạo bởi hai thuộc tính. Chúng ta lại phải phân tích xem nó đã đạt 2NF chưa. Chúng ta lại làm theo cách cũ, với mỗi thuộc tính, đặt ra câu hỏi: *Liệu thuộc tính này có thể tồn tại mà không cần một hay nhiều thuộc tính nào đó nằm trong khóa chính không?*

Bên cạnh là Hình F, biểu diễn cấu trúc của bảng **order_items**. Bây giờ chúng ta lần lượt xem xét các thuộc tính không khóa.

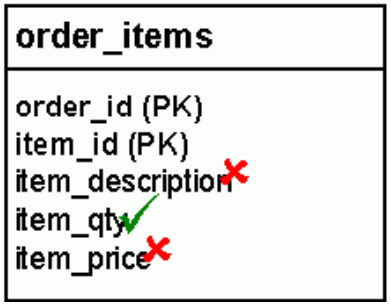


item_description phụ thuộc vào **item_id**, nhưng không phụ thuộc vào order_id. Do đó, thuộc tính này không đạt chuẩn 2NF (ngạc nhiên?) ✗

item_qty phụ thuộc vào cả hai thuộc tính của khóa chính nên thuộc tính này thỏa mãn chuẩn 2NF. ✓

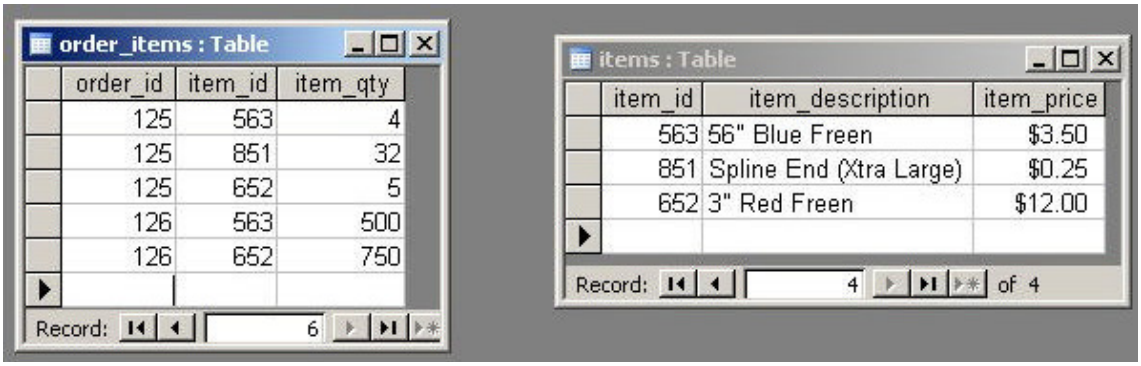
item_price chỉ phụ thuộc vào **item_id** mà không phụ thuộc vào **order_id**, nên nó vi phạm điều kiện của chuẩn 2NF. ✗

Chúng ta có bảng phân tích như sau:



Bây giờ, chúng ta lấy ra các thuộc tính không thỏa mãn điều kiện của chuẩn 2NF và đưa vào một bảng mới. Chúng ta gọi bảng mới này là bảng **items**:

Hình G: Bảng order_items và bảng items



order_items : Table

order_id	item_id	item_qty
125	563	4
125	851	32
125	652	5
126	563	500
126	652	750

Record: 6

items : Table

item_id	item_description	item_price
563	56" Blue Freen	$3.50
851	Spline End (Xtra Large)	$0.25
652	3" Red Freen	$12.00

Record: 4 of 4

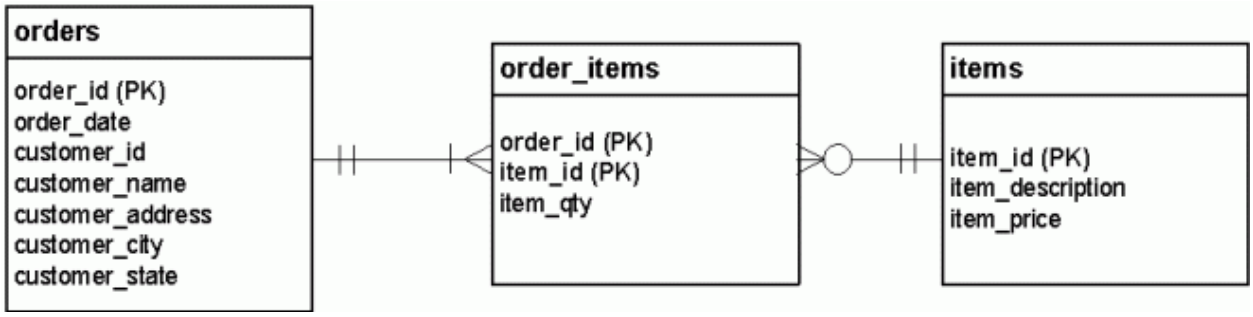
Khoan đã, có gì đó không ổn. Lúc trước, sau khi kiểm tra các điều kiện của chuẩn 2NF, chúng ta lấy ra tất cả các thuộc tính phụ thuộc vào item_id và đưa vào một bảng mới. Lần này, chúng ta lại lấy ra các thuộc tính không đạt chuẩn 2NF: nói cách khác, giữa nguyên **item_qty.** Tại sao? Lần này có gì khác mà lại làm như vậy?

Điểm khác nhau là ở chỗ: trong lần trước, chúng ta đưa thuộc tính khóa **item_id** ra khỏi bảng **orders**, là do quan hệ một-nhiều giữa orders và order-items. Do đó, thuộc tính **item_qty** phải đi cùng **item_id** vào bảng mới.

Lần này, **item_id** không được đưa ra khỏi bảng **order_items** là do quan hệ nhiều-một giữa order-items và items. Do đó, vì item_qty không vi phạm chuẩn 2NF nên nó được giữ lại bảng có khóa chính gồm hai thuộc tính.

Để hiểu rõ hơn, có thể xem ERD mới:

Hình H:



Đường nối giữa bảng **items** và bảng **order_items** nghĩa là:

- Mỗi item có thể nằm trong một số hóa đơn hoặc không nằm trong hóa đơn nào.
- Mỗi order-item có thể liên quan đến một và chỉ một item.

Hai quan hệ trên là ví dụ cho quan hệ một-nhiều. Ba bảng này, xem xét một cách toàn diện, là cách chúng ta biểu diễn quan hệ nhiều-nhiều: *Một order nào đó có thể có nhiều item; một item nào đó có thể thuộc về nhiều order.*

Nhớ rằng lần này, chúng ta không đưa thuộc tính khóa order_id vào bảng mới. Lí do là mỗi item cụ thể, không cần phải biết nó thuộc về order nào. Bảng order_items lưu trữ

những thông tin dó thông qua hai thuộc tính order_id và item_id. Hai thuộc tính này khi đứng kết hợp với nhau thì tạo thành khóa chính cho bảng order_items, nhưng khi đứng riêng rẽ, chúng là các khóa ngoại (foreign keys) trỏ tới các hàng trong các bảng khác. Chúng ta sẽ nói nhiều hơn về khóa ngoại trong phần 3NF.

Cũng chú ý rằng, bảng mới không có khóa chính hợp thành bởi nhiều thuộc tính nên nó thỏa mãn điều kiện của dạng chuẩn 2NF. Đến đây, CSDL của chúng ta đã đạt dạng chuẩn 2NF!

Dạng chuẩn thứ 3 (3NF): Không có phụ thuộc hàm vào thuộc tính không khóa.

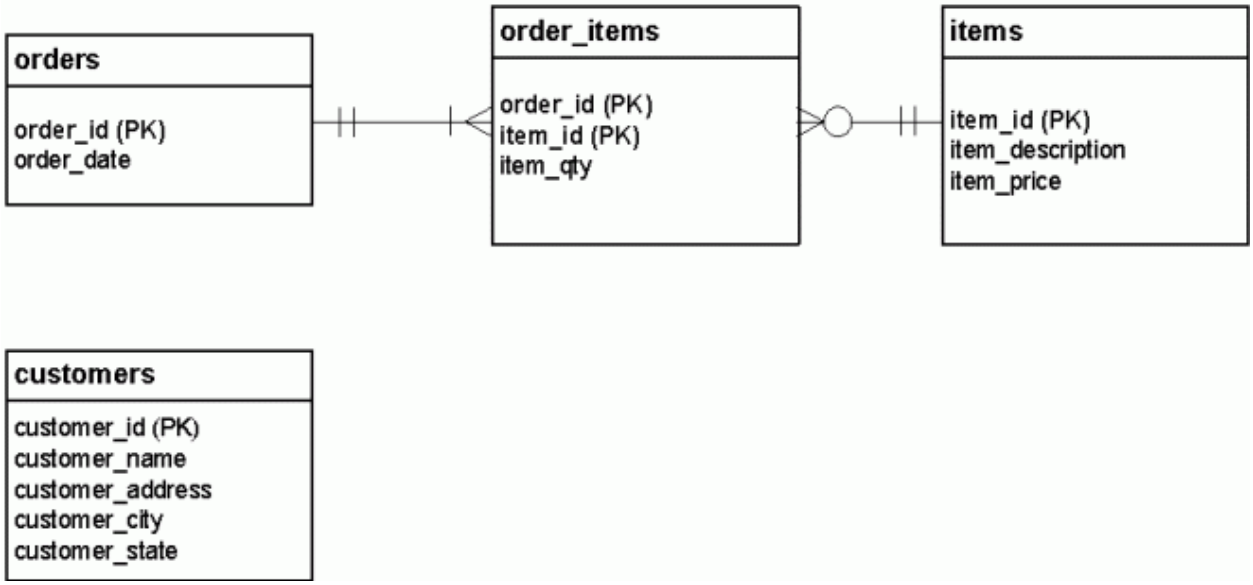
Cuối cùng, chúng ta trở lại vấn đề liên quan đến thông tin về Khách hàng. Với CSDL hiện tại, nếu một khách hàng có hơn một order, chúng ta sẽ phải nhập thông tin về khách hàng đó nhiều lần. Xảy ra hiện tượng này là do trong bảng **order** có tồn tại các thuộc tính phụ thuộc vào một thuộc tính không khóa.

Để hiểu rõ hơn khái niệm này, xem xét thuộc tính **order_date**. Nó có thể tồn tại độc lập với thuộc tính **order_id**? *Không*: một "order date" sẽ là vô nghĩa nếu không có order. Khi đó, **order_date** được gọi là phụ thuộc vào thuộc tính khóa (vì order_id là một thuộc tính khóa).

Còn thuộc tính **customer_name** thì sao— liệu nó có thể tồn tại một mình bên ngoài bảng **orders**? *Có*. Vẫn có nghĩa khi nói về một khách hàng mà không đề cập tới yêu cầu mua hàng hay hóa đơn. Tương tự với các thuộc tính **customer_address**, **customer_city**, và **customer_state**. Bốn thuộc tính này chỉ phụ thuộc vào **customer_id** – một thuộc tính không khóa.

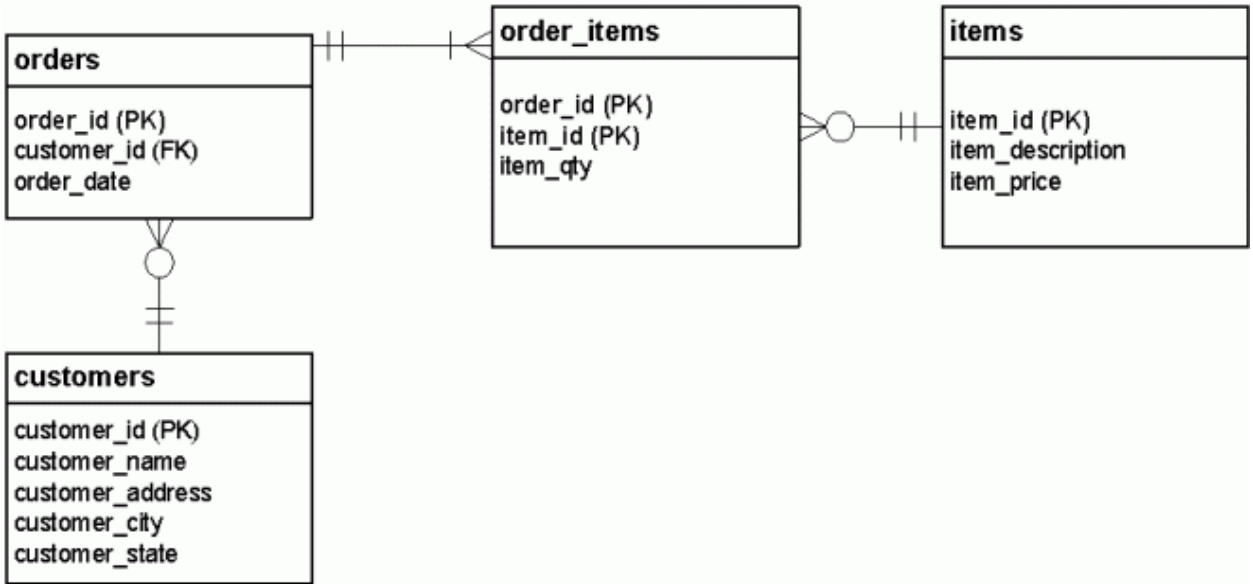
Các trường này sẽ thuộc về một bảng khác, của riêng chúng, với **customer_id** làm khóa chính (xem Hình I).

Hình I:



Tuy nhiên, để ý trong Hình I, chúng ta đã cắt đứt mối quan hệ giữa bảng Orders với các thông tin về khách hàng. Do vậy, chúng ta phải khôi phục mối quan hệ bằng cách tạo ra một khóa ngoại (**Foreign key - FK**) trong bảng Orders. Khóa ngoại về bản chất là một thuộc tính trỏ tới khóa chính của một bảng khác. **Hình J** là ERD hoàn thiện của chúng ta:

Hình J: ERD hoàn chỉnh

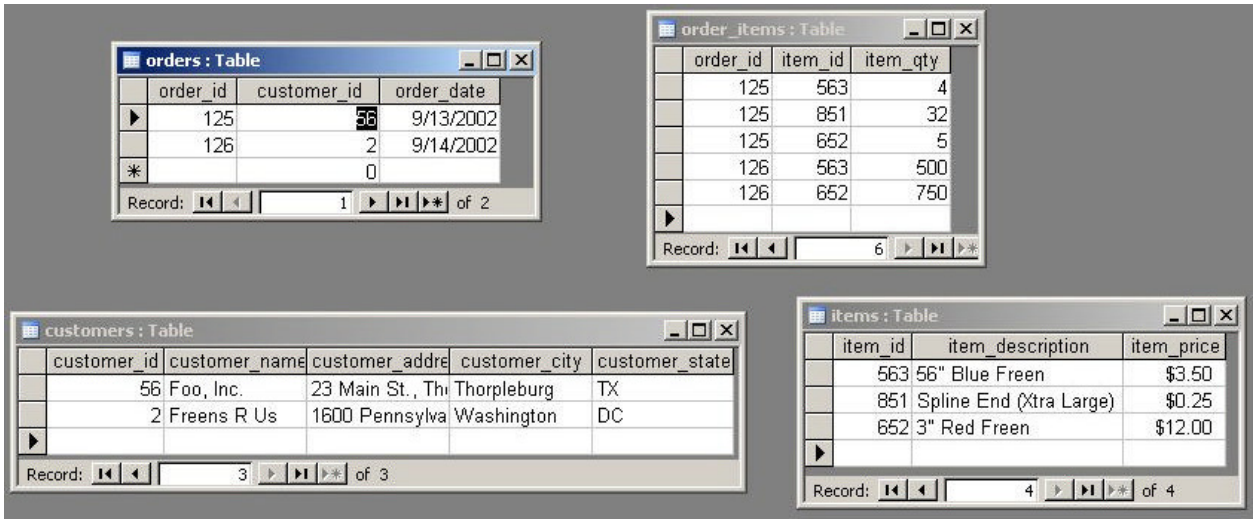


Quan hệ giữa **orders** và **customers** có thể được diễn giải như sau:

- *Một order được tạo bởi một và chỉ một customer;*
- *Một customer có thể có nhiều order hoặc không có order nào cả.*

Cuối cùng, đây là dữ liệu trong bốn bảng của chúng ta. Nhớ rằng, 3NF tách các cột, không phải tách hàng.

Hình K:



orders : Table

order_id	customer_id	order_date
125	56	9/13/2002
126	2	9/14/2002
	0	

Record: 1 of 2

order_items : Table

order_id	item_id	item_qty
125	563	4
125	851	32
125	652	5
126	563	500
126	652	750

Record: 6

customers : Table

customer_id	customer_name	customer_addre	customer_city	customer_state
56	Foo, Inc.	23 Main St., Th	Thorpleburg	TX
2	Freens R Us	1600 Pennsylva	Washington	DC

Record: 3 of 3

items : Table

item_id	item_description	item_price
563	56" Blue Freen	$3.50
851	Spline End (Xtra Large)	$0.25
652	3" Red Freen	$12.00

Record: 4 of 4

Tham khảo

Sau đây là một số tài liệu tham khảo hữu ích:

- The Art of Analysis, by Dr. Art Langer, devotes considerable space to normalization. Springer-Verlag Telos (January 15, 1997) ISBN: 0387949720
- Báo cáo khoa học năm 1969 của Dr. Codd's về chuẩn hóa CSDL: www.acm.org/classics/nov95
- The Wikipedia article on normalization bàn về 5 dạng chuẩn hóa: en.wikipedia.org/wiki/Database_normalization