

DESIGN 001

KIẾN THỨC CƠ BẢN VỀ THIẾT KẾ CƠ SỞ DỮ LIỆU

Một cơ sở dữ liệu được thiết kế đúng sẽ cung cấp cho bạn quyền truy nhập vào thông tin chính xác, mới nhất. Do một thiết kế phù hợp rất thiết yếu để bạn đạt được mục tiêu của mình khi làm việc với cơ sở dữ liệu nên việc dành thời gian cần thiết để tìm hiểu các nguyên tắc thiết kế tối ưu là vô cùng quan trọng. Cuối cùng, nhiều khả năng bạn sẽ tìm ra một cơ sở dữ liệu đáp ứng được nhu cầu của mình và có thể dễ dàng thay đổi.

Bài viết này sẽ cung cấp hướng dẫn để lên kế hoạch tạo cơ sở dữ liệu trên máy tính. Bạn sẽ biết cách để quyết định những thông tin mình cần, cách phân chia thông tin đó vào các bảng và cột thích hợp, cũng như mối quan hệ của chúng với nhau. Bạn cần đọc bài viết này trước khi tạo cơ sở dữ liệu trên máy tính lần đầu.

TRONG BÀI VIẾT NÀY CHÚNG TA SẼ HỌC

Một số điều khoản về cơ sở dữ liệu cần biết

Thiết kế cơ sở dữ liệu tốt là gì?

Quy trình thiết kế

Xác định mục đích của cơ sở dữ liệu

Tìm và sắp xếp thông tin cần thiết

Phân chia thông tin vào các bảng

Biến mục thông tin thành các cột

Chỉ định khóa chính

Tạo mối quan hệ cho bảng

Tinh chỉnh thiết kế

Áp dụng các quy tắc chuẩn hóa

MỘT SỐ ĐIỀU KHOẢN VỀ CƠ SỞ DỮ LIỆU CẦN BIẾT

Mysql sẽ sắp xếp thông tin của bạn vào **các bảng**: danh sách các hàng và cột gợi nhớ đến một bảng hoặc bảng tính của nhân viên kế toán. Trong cơ sở dữ liệu đơn giản, bạn có thể chỉ có một bảng. Với hầu hết cơ sở dữ liệu, bạn sẽ cần nhiều bảng. Ví dụ: bạn có thể có một bảng lưu trữ thông tin về sản phẩm, một bảng khác lưu trữ thông tin về đơn hàng và một bảng khác nữa chứa thông tin về khách hàng.

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> oc_address	★ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> oc_api	★ Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_general_ci	2.3 KiB	-
<input type="checkbox"/> oc_api_ip	★ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> oc_api_session	★ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> oc_attribute	★ Browse Structure Search Insert Empty Drop	11	MyISAM	utf8_general_ci	2.1 KiB	-
<input type="checkbox"/> oc_attribute_description	★ Browse Structure Search Insert Empty Drop	11	MyISAM	utf8_general_ci	2.2 KiB	-
<input type="checkbox"/> oc_attribute_group	★ Browse Structure Search Insert Empty Drop	4	MyISAM	utf8_general_ci	2.0 KiB	-
<input type="checkbox"/> oc_attribute_group_description	★ Browse Structure Search Insert Empty Drop	4	MyISAM	utf8_general_ci	2.1 KiB	-
<input type="checkbox"/> oc_banner	★ Browse Structure Search Insert Empty Drop	3	MyISAM	utf8_general_ci	2.1 KiB	-
<input type="checkbox"/> oc_banner_image	★ Browse Structure Search Insert Empty Drop	14	MyISAM	utf8_general_ci	3.0 KiB	-
<input type="checkbox"/> oc_cart	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	32.0 KiB	-
<input type="checkbox"/> oc_category	★ Browse Structure Search Insert Empty Drop	38	MyISAM	utf8_general_ci	4.2 KiB	-
<input type="checkbox"/> oc_category_description	★ Browse Structure Search Insert Empty Drop	38	MyISAM	utf8_general_ci	11.4 KiB	-
<input type="checkbox"/> oc_category_filter	★ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> oc_category_path	★ Browse Structure Search Insert Empty Drop	71	MyISAM	utf8_general_ci	2.9 KiB	-

Chính xác hơn, mỗi hàng sẽ được gọi là một **bản ghi**, còn mỗi cột sẽ được gọi là một **trường**. Bản ghi là một cách hữu dụng và nhất quán để tổng hợp thông tin về nội dung nào đó. Trường là một mục thông tin đơn — một loại mục xuất hiện trong mọi bản ghi. Ví dụ: trong bảng Sản phẩm, mỗi hàng hoặc bản ghi sẽ chứa thông tin về một sản phẩm. Mỗi cột hoặc trường sẽ chứa một số loại thông tin về sản phẩm đó, như tên hoặc giá cả.

Options		product_id	model	sku	upc	ean	jan	isbn	mpn	location	quantity	stock_status_id	image	manufacturer_id
<input type="checkbox"/>	Edit Copy Delete	28	Product 1								939	7	catalog/demo/htc_touch_hd_1.jpg	5
<input type="checkbox"/>	Edit Copy Delete	29	Product 2								999	6	catalog/demo/palm_treo_pro_1.jpg	6
<input type="checkbox"/>	Edit Copy Delete	30	Product 3								7	6	catalog/demo/canon_eos_5d_1.jpg	9
<input type="checkbox"/>	Edit Copy Delete	31	Product 4								1000	6	catalog/demo/nikon_d300_1.jpg	0
<input type="checkbox"/>	Edit Copy Delete	32	Product 5								999	6	catalog/demo/ipod_touch_1.jpg	8
<input type="checkbox"/>	Edit Copy Delete	33	Product 6								1000	6	catalog/demo/samsung_syncmaster_941bw.jpg	0
<input type="checkbox"/>	Edit Copy Delete	34	Product 7								1000	6	catalog/demo/ipod_shuffle_1.jpg	8
<input type="checkbox"/>	Edit Copy Delete	35	Product 8								1000	6	catalog/demo/ipod_shuffle_1.jpg	8

THIẾT KẾ CƠ SỞ DỮ LIỆU TỐT LÀ?

Một số nguyên tắc hướng dẫn quy trình thiết kế cơ sở dữ liệu. Nguyên tắc đầu tiên: thông tin trùng lặp (hay còn gọi là dữ liệu dự phòng) là có hại vì thông tin này gây lãng phí về dung lượng và làm tăng khả năng lỗi cũng như sự không nhất quán. Nguyên tắc thứ hai là sự chính xác và sự hoàn chỉnh của thông tin rất quan trọng. Nếu cơ sở dữ liệu của bạn có chứa thông tin không chính xác thì mọi báo cáo sử dụng thông tin từ cơ sở dữ liệu đó cũng sẽ chứa thông tin không chính xác. Kết quả là mọi quyết định bạn đưa ra dựa trên các báo cáo đó sẽ trở thành thông tin sai lệch.

Do đó, thiết kế cơ sở dữ liệu tốt là một thiết kế:

- Phân chia thông tin của bạn vào các bảng theo đối tượng để giảm thông tin thừa.
- Cung cấp cho Mysql thông tin cần thiết để kết hợp thông tin trong các bảng với nhau, khi cần.
- Giúp hỗ trợ và đảm bảo tính chính xác cũng như tính toàn vẹn cho thông tin của bạn.
- Điều chỉnh nhu cầu báo cáo và xử lý dữ liệu của bạn.

QUY TRÌNH THIẾT KẾ

Quy trình thiết kế bao gồm những bước sau:

- **Xác định mục đích của cơ sở dữ liệu**

Việc này giúp bạn chuẩn bị cho các bước tiếp theo.

- **Tìm và sắp xếp thông tin cần thiết**

Thu thập mọi loại thông tin bạn có thể muốn ghi vào cơ sở dữ liệu, như tên sản phẩm và số đơn hàng.

- **Phân chia thông tin vào các bảng**

Phân chia mục thông tin vào các thực thể hoặc đối tượng chính, như Sản phẩm hoặc Đơn hàng. Mỗi đối tượng sau đó sẽ trở thành một bảng.

- **Biến mục thông tin thành các cột**

Quyết định loại thông tin bạn muốn lưu trữ trong từng bảng. Mỗi mục sẽ trở thành một trường và được hiển thị dưới dạng cột trong bảng. Ví dụ: một bảng Nhân viên có thể chứa các trường như Họ và Ngày Tuyển dụng.

- **Chỉ định khóa chính**

Chọn khóa chính cho mỗi bảng. Khóa chính là một cột, dùng để xác định từng hàng riêng. Ví dụ có thể là ID Sản phẩm hoặc ID Đơn hàng.

- **Thiết lập mối quan hệ cho bảng**

Xem bảng và quyết định mối quan hệ của dữ liệu trong bảng đó với dữ liệu ở các bảng khác. Thêm các trường vào bảng hoặc tạo bảng mới để làm rõ mối quan hệ, nếu cần.

- **Tinh chỉnh thiết kế của bạn**

Phân tích thiết kế để tìm lỗi. Tạo bảng và thêm một số bản ghi dữ liệu mẫu. Xem liệu bạn có thể nhận được kết quả mong muốn từ bảng không. Tiến hành điều chỉnh thiết kế, nếu cần.

- **Áp dụng các quy tắc chuẩn hóa**

Áp dụng các quy tắc chuẩn hóa dữ liệu để xem liệu bảng của bạn đã được cấu trúc phù hợp chưa. Tiến hành điều chỉnh bảng, nếu cần.

XÁC ĐỊNH MỤC ĐÍCH CỦA CƠ SỞ DỮ LIỆU

Tốt nhất, bạn nên ghi lại mục đích của cơ sở dữ liệu ra giấy — mục đích của cơ sở dữ liệu, bạn dự định sử dụng cơ sở dữ liệu như thế nào và ai sẽ sử dụng cơ sở dữ liệu. Đối với cơ sở dữ liệu nhỏ dành cho doanh nghiệp gia đình, ví dụ: bạn có thể ghi đơn giản như "Cơ sở dữ liệu khách hàng lưu danh sách thông tin khách hàng cho mục đích tạo quảng cáo qua thư và tạo báo cáo". Nếu cơ sở dữ liệu phức tạp hơn hoặc có nhiều người dùng, như thường xảy ra trong môi trường doanh nghiệp thì mục đích có thể đơn giản chỉ là một đoạn văn hoặc nhiều hơn và cần bao gồm thời gian, cũng như cách mỗi người sử dụng cơ sở dữ liệu. Tốt nhất, bạn nên có tuyên bố sứ mệnh được xây dựng tốt có thể được tham chiếu trong suốt quy trình thiết kế. Việc có tuyên bố này sẽ giúp bạn tập trung vào mục tiêu khi đưa ra quyết định.

TÌM VÀ SẮP XẾP THÔNG TIN CẦN THIẾT

Để tìm và sắp xếp thông tin cần thiết, hãy bắt đầu bằng thông tin hiện có. Ví dụ: bạn có thể ghi lại đơn hàng vào sổ cái hoặc giữ lại thông tin khách hàng trên biểu mẫu giấy trong tủ hồ sơ. Thu thập những tài liệu đó và liệt kê từng loại thông tin hiển thị (ví dụ: mỗi hộp bạn điền trên một biểu mẫu). Nếu bạn không có sẵn bất cứ biểu mẫu nào thì thay vào đó, hãy tưởng tượng bạn phải thiết kế một biểu mẫu để ghi lại thông tin khách hàng. Bạn muốn đưa thông tin nào vào biểu mẫu? Bạn sẽ tạo những ô nào để điền? Xác định và liệt kê từng mục trong sổ này. Ví dụ: giả sử bạn hiện đang giữ danh sách khách hàng trên thẻ chỉ mục. Việc kiểm tra các thẻ này có thể cho biết rằng mỗi thẻ lưu giữ một tên khách hàng, địa chỉ, thành phố, tiểu bang, mã bưu chính và số điện thoại. Mỗi mục trong sổ này đại diện cho một cột tiềm năng trong bảng.

Khi bạn chuẩn bị danh sách này, đừng lo lắng về việc phải tạo bảng hoàn hảo ngay từ đầu. Thay vào đó, hãy liệt kê từng mục bạn nghĩ ra. Nếu người khác sử dụng cơ sở dữ liệu, hãy hỏi cả ý kiến của họ. Bạn có thể tinh chỉnh danh sách sau đó.

Tiếp theo, hãy cân nhắc loại báo cáo hoặc quảng cáo qua thư bạn có thể muốn tạo ra từ cơ sở dữ liệu. Ví dụ: bạn có thể muốn báo cáo doanh số sản phẩm hiển thị doanh số theo khu vực hoặc báo cáo tóm tắt hàng tồn kho hiển thị mức tồn kho sản phẩm. Có thể bạn cũng muốn tạo thư biểu mẫu để gửi đến khách hàng, công bố một sự kiện bán hàng hoặc đưa ra phí trả thêm. Thiết kế báo cáo trong đầu và tưởng tượng báo cáo sẽ trông như ra sao. Bạn muốn đưa thông tin nào vào báo cáo? Liệt kê từng mục. Thực hiện tương tự cho thư biểu mẫu và cho mọi báo cáo khác bạn dự định tạo.

Thương gia Northwind

Hàng tồn kho Sản phẩm

<u>ID Sản phẩm</u>	<u>Tên</u>	<u>Số lượng Có sẵn</u>
1	Chai	36
2	Chang	17



Việc đưa ý tưởng vào báo cáo và thư tín bạn có thể muốn tạo sẽ giúp xác định được những mục cần thiết trong cơ sở dữ liệu. Ví dụ: giả sử bạn mang đến cho khách hàng cơ hội chọn tham gia (hoặc không tham gia) cập nhật email định kỳ và bạn muốn in một danh sách những người đã chọn tham gia đó. Để ghi lại thông tin đó, bạn cần thêm một cột “Gửi email” vào bảng khách hàng. Đối với mỗi khách hàng, bạn có thể đặt trường là Có hoặc Không.

Yêu cầu gửi thư email cho khách hàng sẽ đề xuất một mục khác để ghi lại. Khi bạn biết có khách hàng muốn nhận thư email, bạn cũng sẽ cần biết địa chỉ email để gửi thư cho họ. Do đó, bạn cần ghi lại địa chỉ email của từng khách hàng.

Tốt nhất, bạn nên xây dựng nguyên mẫu cho từng báo cáo hoặc danh sách đầu ra và cân nhắc những mục bạn cần tạo cho báo cáo. Ví dụ: khi bạn kiểm tra thư biểu mẫu, một vài ý tưởng có thể xuất hiện. Nếu bạn muốn đưa vào một lời chào phù hợp — ví dụ: từ “Ông”, “Bà” hoặc “Cô” để bắt đầu lời chào, bạn sẽ cần tạo một mục lời chào. Đồng thời, bạn thường có thể bắt đầu thư bằng cụm “Kính gửi ông Smith” thay vì “Kính gửi ông Sylvester Smith”. Điều này cho thấy bạn thường sẽ muốn lưu trữ họ tách biệt với tên.

Điểm chính cần ghi nhớ là bạn nên tách từng mẫu thông tin thành các phần nhỏ nhất, hữu dụng. Đối với tên, để giúp họ luôn sẵn dùng, bạn sẽ tách tên thành hai phần — Tên và Họ. Chẳng hạn để sắp xếp báo cáo theo tên, sẽ hữu ích nếu yêu cầu lưu trữ riêng họ của khách hàng. Nói chung, nếu bạn muốn sắp xếp, tìm kiếm, tính toán hoặc báo cáo dựa trên một mục thông tin thì bạn nên đặt mục đó vào một trường riêng.

Hãy nghĩ tới các câu hỏi bạn có thể muốn cơ sở dữ liệu trả lời. Ví dụ: bạn đã kết thúc bán hàng được cho bao nhiêu sản phẩm nổi bật tháng trước? Khách hàng thân thiết của bạn sống ở đâu? Nhà cung cấp sản phẩm bán chạy nhất của bạn là ai? Việc xem xét trước những câu hỏi này sẽ giúp bạn hướng tới các mục bổ sung cần ghi lại.

Sau khi thu thập thông tin này, bạn đã sẵn sàng cho bước tiếp theo.

PHÂN CHIA THÔNG TIN VÀO CÁC BẢNG

Để phân chia thông tin vào các bảng, hãy chọn các thực thể hoặc đối tượng chính. Ví dụ: sau khi tìm và sắp xếp thông tin về một cơ sở dữ liệu doanh số sản phẩm, danh sách sơ bộ sẽ trông như sau:

Khách hàng

Tên

Địa chỉ

Thành phố,

Tiểu bang, Mã Zip

Gửi Email

Lời chào

Địa chỉ Email

Nhà cung cấp

Tên Công ty

Tên Liên hệ

Địa chỉ

Thành phố,

Tiểu bang, Mã Zip

Sản phẩm

Tên Sản phẩm

Giá

Đơn vị trong Kho

Đơn vị trên Đơn hàng

Đơn hàng

Số Đơn hàng

Nhân viên bán hàng

Ngày Đặt hàng

Sản phẩm

Số lượng

Giá

Tổng cộng

Những thực thể chính được hiển thị ở đây là sản phẩm, nhà cung cấp, khách hàng và đơn hàng. Do đó, tốt nhất, bạn nên bắt đầu với bốn bảng sau: một bảng dữ kiện về sản phẩm, một bảng dữ kiện về nhà cung cấp, một bảng dữ kiện về khách hàng và một bảng dữ kiện về đơn hàng. Mặc dù bảng này không hoàn thiện danh sách nhưng đây là một điểm khởi đầu tốt. Bạn có thể tiếp tục tinh chỉnh danh sách này cho đến khi có được thiết kế tối ưu.

Khi lần đầu xem xét danh sách mục sơ bộ, bạn có thể muốn đặt tất cả chúng vào một bảng duy nhất, thay vì bốn bảng như hiển thị trong hình minh họa trước đó. Bạn sẽ biết lý do tại sao không nên đặt tất cả vào cùng một bảng ở đây. Hãy cân nhắc một chút, bảng hiển thị ở đây:

Options

	product_id	model	sku	upc	ean	jan	isbn	mpn	location	quantity	stock_status_id	image	manufacturer_id
<input type="checkbox"/> Edit Copy Delete	28	Product 1								939	7	catalog/demo/htc_touch_hd_1.jpg	5
<input type="checkbox"/> Edit Copy Delete	29	Product 2								999	6	catalog/demo/palm_treo_pro_1.jpg	6
<input type="checkbox"/> Edit Copy Delete	30	Product 3								7	6	catalog/demo/canon_eos_5d_1.jpg	9
<input type="checkbox"/> Edit Copy Delete	31	Product 4								1000	6	catalog/demo/nikon_d300_1.jpg	0
<input type="checkbox"/> Edit Copy Delete	32	Product 5								999	6	catalog/demo/ipod_touch_1.jpg	8
<input type="checkbox"/> Edit Copy Delete	33	Product 6								1000	6	catalog/demo/samsung_syncmaster_941bw.jpg	0
<input type="checkbox"/> Edit Copy Delete	34	Product 7								1000	6	catalog/demo/ipod_shuffle_1.jpg	8

Trong trường hợp này, mỗi hàng đều chứa thông tin về cả sản phẩm lẫn nhà cung cấp. Do bạn có thể có nhiều sản phẩm từ cùng một nhà cung cấp nên thông tin tên và địa chỉ nhà cung cấp sẽ bị lặp lại nhiều lần. Như vậy rất tốn dung lượng ổ đĩa. Việc chỉ ghi lại thông tin nhà cung cấp một lần trong bảng Nhà cung cấp riêng, rồi liên kết bảng đó với bảng Sản phẩm là giải pháp tốt hơn nhiều.

Vấn đề thứ hai của thiết kế này là khi bạn cần sửa đổi thông tin về nhà cung cấp. Ví dụ: giả sử bạn cần thay đổi địa chỉ của nhà cung cấp. Vì địa chỉ xuất hiện ở nhiều nơi nên bạn có thể vô tình thay đổi địa chỉ ở chỗ này nhưng quên thay đổi địa chỉ ở những chỗ khác. Việc ghi lại địa chỉ của nhà cung cấp ở một nơi duy nhất sẽ giải quyết được vấn đề.

Khi bạn thiết kế cơ sở dữ liệu của mình, hãy cố gắng chỉ ghi mỗi dữ kiện một lần. Nếu bạn thấy mình đang lặp lại cùng một thông tin ở nhiều chỗ, chẳng hạn như địa chỉ của một nhà cung cấp cụ thể, hãy đặt thông tin đó vào một bảng riêng.

Cuối cùng, giả sử chỉ có một sản phẩm được cung cấp bởi Coho Winery và bạn muốn xóa sản phẩm mà vẫn giữ lại thông tin tên và địa chỉ của nhà cung cấp. Bạn sẽ làm thế nào để xóa bản ghi sản phẩm mà không làm mất luôn thông tin nhà cung cấp? Không thể. Vì mỗi bản ghi sẽ chứa dữ kiện về một sản phẩm, cũng như dữ kiện về một nhà cung cấp nên bạn không thể xóa bảng này mà không xóa bảng kia. Để tách riêng các dữ kiện này, bạn phải phân tách một bảng thành hai: một bảng thông tin sản phẩm và một bảng khác chứa thông tin nhà cung cấp. Việc xóa bản ghi sản phẩm sẽ chỉ xóa dữ kiện về sản phẩm, chứ không xóa dữ kiện về nhà cung cấp.

Sau khi bạn chọn đối tượng mà bảng trình bày, các cột trong bảng đó sẽ chỉ lưu trữ dữ kiện về đối tượng. Ví dụ: bảng sản phẩm sẽ chỉ lưu trữ dữ kiện về sản phẩm. Vì địa chỉ nhà cung cấp là một dữ kiện về nhà cung cấp, không phải dữ kiện về sản phẩm nên dữ kiện này sẽ nằm trong bảng nhà cung cấp.

BIẾN MỤC THÔNG TIN THÀNH CÁC CỘT

Để xác định các cột trong bảng, hãy quyết định thông tin bạn cần theo dõi về đối tượng được ghi trong bảng. Ví dụ: với bảng Khách hàng, Tên, Địa chỉ, mã Zip Thành phố Tiểu bang, Gửi email, Lời chào và Địa chỉ email sẽ bao gồm một danh sách cột khởi đầu tốt. Mỗi bản ghi trong bảng chứa cùng một tập hợp cột để bạn có thể lưu trữ thông tin Tên.

Địa chỉ, mã Zip Thành phố-Tiểu bang, Gửi email, Lời chào và Địa chỉ email về từng bản ghi. Ví dụ: cột địa chỉ có chứa địa chỉ của khách hàng. Mỗi bản ghi sẽ chứa dữ liệu về một khách hàng và trường địa chỉ sẽ chứa địa chỉ của khách hàng đó.

Sau khi đã quyết định nhóm tập hợp cột ban đầu của từng bảng, bạn có thể tinh chỉnh thêm cho các cột. Ví dụ: bạn nên lưu trữ tên khách hàng thành hai cột riêng biệt: tên và họ để có thể sắp xếp, tìm kiếm và lập chỉ mục trên chính những cột đó. Tương tự, địa chỉ thực sự gồm năm thành phần riêng lẻ, địa chỉ, thành phố, tiểu bang, mã bưu điện và quốc gia/vùng lãnh thổ, bạn cũng nên lưu trữ chúng vào các cột riêng biệt. Nếu bạn muốn tiến hành tìm kiếm, lọc hoặc sắp xếp hoạt động, chẳng hạn như theo tiểu bang thì bạn cần lưu thông tin tiểu bang ở một cột riêng.

Bạn cũng cần cân nhắc xem liệu cơ sở dữ liệu chỉ giữ thông tin có nguồn gốc trong nước hay có nguồn gốc quốc tế. Ví dụ: nếu bạn dự định lưu trữ các địa chỉ quốc tế, bạn nên có cột Khu vực thay cho cột Tiểu bang vì cột này có thể phù hợp cho cả các tiểu bang trong nước và các khu vực thuộc quốc gia/vùng lãnh thổ khác. Tương tự, Mã Bưu điện sẽ hữu ích hơn Mã Zip nếu bạn chuẩn bị lưu trữ các địa chỉ quốc tế.

Danh sách sau trình bày một số mẹo cho việc quyết định số cột của bạn.

- **Đừng bao gồm dữ liệu đã tính toán**

Trong hầu hết trường hợp, bạn không nên lưu trữ kết quả tính toán trong bảng. Thay vào đó, bạn có thể yêu cầu mysql thực hiện tính toán khi muốn xem kết quả. Ví dụ: giả sử có một báo cáo Sản phẩm Đã Đặt hàng, hiển thị tổng phụ số đơn vị đã đặt cho từng danh mục sản phẩm trong cơ sở dữ liệu. Tuy nhiên, sẽ không có cột tổng phụ Sản phẩm Đã Đặt hàng trong bất cứ bảng nào. Thay vào đó, bảng Sản phẩm sẽ bao gồm cột Số đơn vị Đã Đặt hàng, lưu trữ số đơn vị đã đặt cho từng sản phẩm. mysql sẽ sử dụng dữ liệu đó để tính toán tổng phụ mỗi khi bạn in báo cáo. Không nên lưu tổng phụ trong bảng.

- **Lưu trữ thông tin trong các phần logic nhỏ nhất**

Bạn có thể muốn để trường đơn cho tên đầy đủ hoặc cho tên sản phẩm kèm theo mô tả sản phẩm. Nếu bạn kết hợp nhiều loại thông tin trong một trường thì sau đó sẽ rất khó để truy xuất dữ kiện riêng. Hãy tìm cách tách thông tin thành các phần logic, ví dụ: tạo các trường riêng cho tên và họ hoặc cho tên sản phẩm, danh mục và mô tả.

Khách hàng

Tên

Địa chỉ

Thành phố

Khu vực

Mã Bưu chính

Quốc gia

Gửi Email

Lời chào

Địa chỉ email

Nhà cung cấp

Tên Công ty

Tên Liên hệ

Địa chỉ

Thành phố

Khu vực

Mã Bưu chính

Quốc gia

Điện thoại

Sản phẩm

Tên Sản phẩm

Đơn Giá

Đơn vị trong Kho

Đơn vị trên

Đơn hàng

Số lượng cho mỗi

đơn vị

Đơn hàng

Số Đơn hàng

Nhân viên

bán hàng

Ngày Đặt hàng

Sản phẩm

Số lượng

Giá

Sau khi tinh chỉnh các cột dữ liệu trong từng bảng, bạn đã sẵn sàng chọn khóa chính cho từng bảng.

CHỈ ĐỊNH KHÓA CHÍNH

Mỗi bảng cần có một cột hoặc tập hợp cột riêng giúp xác định từng hàng được lưu trữ trong bảng. Đây thường là số nhận dạng duy nhất, chẳng hạn như số ID nhân viên hoặc số sê-ri. Trong thuật ngữ về cơ sở dữ liệu, thông tin này được gọi là **khóa chính** của bảng. mysql sử dụng các trường khóa chính để liên kết nhanh dữ liệu từ nhiều bảng và kết hợp dữ liệu với nhau giúp bạn.

Nếu bạn đã có mã định danh duy nhất cho bảng, như số sản phẩm để xác định riêng từng sản phẩm trong danh mục, bạn có thể sử dụng mã định danh đó làm khóa chính của bảng — nhưng chỉ khi các giá trị trong cột này sẽ luôn khác với từng bản ghi. Khóa chính của bạn không thể có giá trị trùng lặp. Ví dụ: không sử dụng tên người làm khóa chính vì tên không phải là duy nhất. Bạn rất dễ gặp tình huống hai người có cùng tên trong bảng.

Khóa chính sẽ luôn chỉ có một giá trị. Nếu một lúc nào đó, giá trị cột trở thành không xác định hoặc không rõ (giá trị bị thiếu), bạn sẽ không thể sử dụng giá trị đó làm thành phần trong khóa chính.

Bạn phải luôn chọn khóa chính có giá trị không đổi. Trong một cơ sở dữ liệu sử dụng nhiều bảng, bạn có thể sử dụng khóa chính của mình làm tham chiếu trong bảng khác. Nếu khóa chính thay đổi, thay đổi cũng cần được áp dụng ở những nơi khóa được tham chiếu. Việc sử dụng khóa chính sẽ không thay đổi việc giảm cơ hội khóa chính có thể trở nên không đồng bộ với các bảng khác tham chiếu đến khóa.

Thông thường, người ta sẽ sử dụng một số duy nhất bất kỳ làm khóa chính. Ví dụ: bạn có thể gán một số đơn hàng duy nhất cho từng đơn hàng. Mục đích duy nhất của số đơn hàng là để xác định đơn hàng. Sau khi gán, số đơn hàng

— sẽ không bao giờ thay đổi.

se không bao giờ thay đổi.

Nếu bạn không có sẵn cột hoặc tập hợp cột có thể làm khóa chính tốt, hãy cân nhắc sử dụng cột có kiểu dữ liệu Số Tự động. Khi bạn sử dụng kiểu dữ liệu Số Tự động, Mysql sẽ tự động gán một giá trị cho bạn. Mã định danh không chứa dữ kiện; mã định danh không chứa thông tin thực tế mô tả hàng mà mã đại diện. Mã định danh không chứa dữ kiện là lựa chọn lý tưởng để dùng làm khóa chính vì chúng sẽ không thay đổi. Một khóa chính sẽ chứa các dữ kiện về một hàng — số điện thoại hoặc tên khách hàng chẳng hạn — nhiều khả năng sẽ thay đổi vì tự bản thân thông tin thực tế có thể thay đổi.

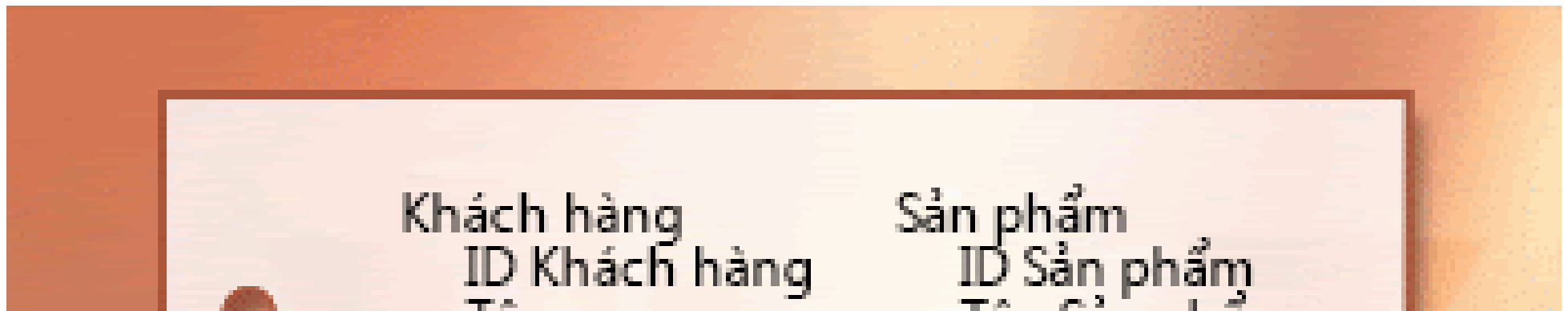
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	product_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More

1. Một cột được đặt là kiểu dữ liệu Số Tự động thường sẽ là một khóa chính tốt. Không có hai sản phẩm nào có ID giống nhau.

Trong một số trường hợp, bạn có thể muốn sử dụng hai trường trở lên cùng nhau, cung cấp khóa chính cho bảng. Ví dụ: một bảng Chi tiết Đơn hàng lưu trữ các khoản mục về đơn hàng sẽ sử dụng hai cột trong khóa chính: ID Đơn hàng và ID Sản phẩm. Khi khóa chính triển khai nhiều cột, khóa còn được gọi là khóa tổng hợp.

Với cơ sở dữ liệu doanh số sản phẩm, bạn có thể tạo ra cột Số Tự động cho từng bảng để làm khóa chính:

ID_Sản_phẩm cho bảng Sản phẩm, ID_Đơn_hàng cho bảng Đơn hàng, ID_Khách_hàng cho bảng khách hàng và ID_Nhà_cung_cấp cho bảng Nhà cung cấp.



Tên
Địa chỉ
Thành phố
Khu vực
Mã Bưu chính
Quốc gia
Gửi Email
Lời chào
Địa chỉ email

Nhà cung cấp
ID Nhà cung cấp
Tên Công ty
Tên Liên hệ
Địa chỉ
Thành phố
Khu vực
Mã Bưu chính
Quốc gia
Điện thoại

Tên sản phẩm
Đơn Giá
Đơn vị trong Kho
Đơn vị trên
Đơn hàng
Số lượng cho mỗi
đơn vị

Đơn hàng
ID Đơn hàng
Nhân viên
bán hàng
Ngày Đặt hàng
Sản phẩm
Số lượng
Giá

TẠO MỐI QUAN HỆ CHO BẢNG

Giờ bạn đã phân chia thông tin vào các bảng, bạn cần một cách để kết hợp thông tin lại với nhau theo các cách có ý nghĩa. Ví dụ: biểu mẫu sau đây bao gồm thông tin từ một vài bảng.

1

Đơn hàng

Gửi hóa đơn Đến: Alfreds Futterkiste

Obere Str. 57

Berlin

12209

Đức

Gửi hàng Đến: Alfreds Futterkiste

Obere Str. 57

Berlin

12209

Đức

Nhân viên bán hàng: Suyama, Michael

Gửi Qua:



Nhanh



Thông nhất



Liên bang

Mã Đơn hàng: 10643

Ngày Đặt hàng: Ngày 25 tháng 8 năm 1997

Ngày Yêu cầu: Ngày 22 tháng 9 năm 1997

4

Sản phẩm:

Đơn Giá:

Số lượng:

Chiết khấu:

Doanh thu ứng với Số lượng hàng:

Spegesild

\$12,00

2

25%

\$18,00

Chartreuse verte

\$18,00

21

25%

\$283,50

Rossle Sauerkraut

\$45,60

15

25%

\$513,00

0%

In Hóa đơn

Tổng phụ:

\$814,50

Cước phí

\$29,46

Tổng:

\$843,96

1. Các thông tin trong biểu mẫu này được lấy từ bảng Khách hàng...
2. ...bảng Nhân viên...
3. ...bảng Đơn hàng...
4. ...bảng Sản phẩm...
5. ...và bảng Chi tiết Đơn hàng.

mysql là một hệ thống quản lý cơ sở dữ liệu tương quan. Trong cơ sở dữ liệu tương quan, bạn phân chia thông tin vào các bảng riêng, dựa theo đối tượng. Sau đó, bạn sử dụng mối quan hệ giữa các bảng để kết hợp thông tin với nhau khi cần.

Tạo mối quan hệ một đối nhiều

Hãy cân nhắc ví dụ sau: bảng Nhà cung cấp và Sản phẩm trong cơ sở dữ liệu đơn hàng sản phẩm. Một nhà cung cấp có thể cung cấp bất cứ số sản phẩm nào. Kết quả là, với mọi nhà cung cấp được trình bày trong bảng Nhà cung cấp, có thể có nhiều sản phẩm được trình bày trong bảng Sản phẩm. Do đó, mối quan hệ giữa bảng Nhà cung cấp và bảng Sản phẩm là mối quan hệ một đối nhiều.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	product_id	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	model	varchar(64)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	3	sku	varchar(64)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	4	upc	varchar(12)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	5	ean	varchar(14)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	6	jan	varchar(13)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	7	isbn	varchar(17)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	8	mpn	varchar(64)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	9	location	varchar(128)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	10	quantity	int(4)		No	0			Change Drop More
<input type="checkbox"/>	11	stock_status_id	int(11)		No	None			Change Drop More
<input type="checkbox"/>	12	image	varchar(255)	utf8_general_ci	Yes	NULL			Change Drop More
<input type="checkbox"/>	13	manufacturer_id	int(11)		No	None			Change Drop More
<input type="checkbox"/>	14	shipping	tinyint(1)		No	1			Change Drop More
<input type="checkbox"/>	15	price	decimal(15,4)		No	0.0000			Change Drop More
<input type="checkbox"/>	16	points	int(8)		No	0			Change Drop More
<input type="checkbox"/>	17	tax_class_id	int(11)		No	None			Change Drop More
<input type="checkbox"/>	18	date_available	date		No	0000-00-00			Change Drop More
<input type="checkbox"/>	19	weight	decimal(15,8)		No	0.00000000			Change Drop More
<input type="checkbox"/>	20	weight_class_id	int(11)		No	0			Change Drop More
<input type="checkbox"/>	21	length	decimal(15,8)		No	0.00000000			Change Drop More
<input type="checkbox"/>	22	width	decimal(15,8)		No	0.00000000			Change Drop More
<input type="checkbox"/>	23	height	decimal(15,8)		No	0.00000000			Change Drop More
<input type="checkbox"/>	24	length_class_id	int(11)		No	0			Change Drop More
<input type="checkbox"/>	25	subtract	tinyint(1)		No	1			Change Drop More
<input type="checkbox"/>	26	minimum	int(11)		No	1			Change Drop More
<input type="checkbox"/>	27	sort_order	int(11)		No	0			Change Drop More

Để thể hiện mối quan hệ một đối nhiều trong thiết kế cơ sở dữ liệu của bạn, hãy lấy khóa chính ở bên "một" của mỗi quan hệ rồi thêm khóa đó dưới dạng một cột hay nhiều cột bổ sung vào bảng ở bên "nhiều" của mỗi quan hệ. Ví dụ: trong trường hợp này, bạn thêm cột ID Nhà cung cấp từ bảng Nhà cung cấp vào bảng Sản phẩm. Khi đó, Mysql có thể sử dụng số ID nhà cung cấp trong bảng Sản phẩm để xác định đúng nhà cung cấp cho từng sản phẩm.

Cột ID Nhà cung cấp trong bảng Sản phẩm được gọi là khóa ngoại. Khóa ngoại là một khóa chính khác của bảng. Cột ID Nhà cung cấp trong bảng Sản phẩm là một khóa ngoại vì cột này cũng là khóa chính trong bảng Nhà cung cấp.

Khách hàng
ID Khách hàng
Tên
Địa chỉ
Thành phố
Khu vực
Mã Bưu chính
Quốc gia
Gửi Email
Lời chào
Địa chỉ email

Nhà cung cấp

Sản phẩm
ID Sản phẩm
Tên Sản phẩm
Đơn Giá
Đơn vị trong Kho
Đơn vị trên
Đơn hàng
Số lượng cho mỗi
đơn vị
ID Nhà cung cấp

Đơn hàng

ID Đơn hàng

ID Nhà cung cấp
Tên Công ty
Tên Liên hệ
Địa chỉ
Thành phố
Khu vực
Mã Bưu chính
Quốc gia
Điện thoại

ID Đơn hàng
Nhân viên
bán hàng
Ngày Đặt hàng
Sản phẩm
Số lượng
Giá

Bạn cung cấp cơ sở để liên kết các bảng liên quan thông qua thiết lập ghép đôi khóa chính và khóa ngoại. Nếu bạn không chắc các bảng nào có chung cột, việc xác định mối quan hệ một đối nhiều sẽ đảm bảo hai bảng có liên quan thực sự sẽ yêu cầu cột chung.

Tạo mối quan hệ nhiều đối nhiều

Hãy cân nhắc mối quan hệ giữa bảng Sản phẩm và bảng Đơn hàng.

Mỗi một đơn hàng có thể bao gồm nhiều sản phẩm. Mặt khác, mỗi một sản phẩm có thể xuất hiện trên nhiều đơn hàng. Vì vậy, đối với mỗi bản ghi trong bảng Đơn hàng, có thể có nhiều bản ghi trong bảng Sản phẩm. Và đối với mỗi bản ghi trong bảng Sản phẩm, có thể sẽ có nhiều bản ghi trong bảng Đơn hàng. Mối quan hệ loại này được gọi là mối quan hệ nhiều đối nhiều vì với sản phẩm bất kỳ, có thể có nhiều đơn hàng; và với đơn hàng bất kỳ, có thể có nhiều sản phẩm. Lưu ý rằng để phát hiện mối quan hệ nhiều đối nhiều giữa các bảng, điều quan trọng là bạn hãy cân nhắc cả hai phía của mối quan hệ.

Đối tượng của hai bảng — đơn hàng và sản phẩm — có mối quan hệ nhiều đối nhiều. Vấn đề xuất hiện ở đây. Để hiểu rõ vấn đề, hãy tưởng tượng điều gì xảy ra nếu bạn cố gắng tạo mối quan hệ giữa hai bảng bằng cách thêm trường ID Sản phẩm vào bảng Đơn hàng. Để có nhiều sản phẩm trên mỗi đơn hàng, bạn cần nhiều bản ghi trong bảng Đơn hàng cho mỗi đơn hàng. Bạn sẽ lặp lại thông tin đơn hàng cho từng hàng liên quan đến một đơn hàng duy nhất — kết quả là thiết kế không hiệu quả, dẫn đến dữ liệu không chính xác. Bạn sẽ gặp phải vấn đề tương tự nếu bạn đặt trường ID Đơn hàng vào trong bảng Sản phẩm — bạn sẽ có nhiều bản ghi trong bảng Sản phẩm cho từng sản phẩm. Bạn giải quyết vấn đề này như thế nào?

Câu trả lời là tạo một bảng thứ ba, thường gọi là bảng nối, tách mối quan hệ nhiều đối nhiều thành hai mối quan hệ một đối nhiều. Bạn chèn khóa chính từ một trong hai bảng vào bảng thứ ba. Kết quả là, bảng thứ ba sẽ ghi lại từng lần xuất hiện hoặc trường hợp mối quan hệ.

ĐƠN HÀNG

ID Đơn hàng	ID Khách hàng
10248	WILMK
10311	DUMON

SẢN PHẨM

ID Sản phẩm	Tên Sản phẩm
11	Queso Cabrales
42	Singaporean Hokkien Fried Mee
69	Gudbrandsdalsost
72	Mozzarella di Giovanni

CHI TIẾT ĐƠN HÀNG

ID Đơn hàng	ID Sản phẩm	Đơn Giá	Số lượng
10248	11	21,00	12
10248	42	14,00	10
10248	72	34,80	5
10311	42	14,00	6
10311	69	28,80	7

Mỗi bản ghi trong bảng Chi tiết Đơn hàng đại diện cho một khoản mục trên đơn hàng. Khóa chính của bảng Chi tiết Đơn hàng bao gồm hai trường — khóa ngoại từ bảng Đơn hàng và bảng Sản phẩm. Việc sử dụng trường ID Đơn hàng riêng không hiệu quả như khóa chính cho bảng này vì một đơn hàng có thể có nhiều khoản mục. ID Đơn hàng được lặp lại cho từng khoản mục trên đơn hàng để trường không chứa các giá trị duy nhất. Việc sử dụng trường ID Sản phẩm riêng cũng không hiệu quả vì một sản phẩm có thể xuất hiện trên nhiều đơn hàng khác nhau. Nhưng kết hợp lại, hai trường sẽ luôn tạo ra một giá trị duy nhất cho từng bản ghi.

Trong cơ sở dữ liệu doanh số sản phẩm, bảng Đơn hàng và bảng Sản phẩm không liên quan trực tiếp đến nhau. Thay vào đó, chúng liên quan gián tiếp qua bảng Chi tiết Đơn hàng. Mỗi quan hệ nhiều đối nhiều giữa đơn hàng và sản phẩm được thể hiện trong cơ sở dữ liệu bằng cách sử dụng hai mối quan hệ một đối nhiều:

- Bảng Đơn hàng và bảng Chi tiết đơn hàng có mối quan hệ một đối nhiều. Mỗi đơn hàng có thể có nhiều khoản mục nhưng mỗi khoản mục chỉ được kết nối với một đơn hàng.
- Bảng Sản phẩm và bảng Chi tiết Đơn hàng có mối quan hệ một đối nhiều. Mỗi sản phẩm có thể có nhiều khoản mục liên kết nhưng mỗi khoản mục chỉ tham chiếu đến một sản phẩm.

Từ bảng Chi tiết Đơn hàng, bạn có thể xác định mọi sản phẩm trên một đơn hàng cụ thể. Bạn cũng có thể xác định mọi đơn hàng về một sản phẩm cụ thể.

Sau khi kết hợp bảng Chi tiết Đơn hàng, danh sách bảng và trường sẽ trông như sau:

Khách hàng		Sản phẩm	
ID Khách hàng		ID Sản phẩm	
Tên		Tên Sản phẩm	
Địa chỉ		Đơn Giá	
Thành phố		Đơn vị tính/Kho	

thanh phố
Khu vực
Mã Bưu chính
Quốc gia
Gửi Email
Lời chào
Địa chỉ email

Nhà cung cấp
ID Nhà cung cấp
Tên Công ty
Tên Liên hệ
Địa chỉ
Thành phố
Khu vực
Mã Bưu chính
Quốc gia
Điện thoại

Đơn vị trong kho
Đơn vị trên Đơn hàng
Số lượng cho mỗi đơn vị
ID Nhà cung cấp

Đơn hàng
ID Đơn hàng
Nhân viên bán hàng
Ngày Đặt hàng
ID Khách hàng
Tên người nơi Vận chuyển
Địa chỉ nơi Vận chuyển
Thành phố nơi Vận chuyển
Khu vực nơi Vận chuyển
Mã Bưu chính nơi Vận chuyển
Quốc gia nơi Vận chuyển

Chi tiết Đơn hàng
ID Đơn hàng
ID Sản phẩm
Đơn Giá
Số lượng

Tạo mối quan hệ một đối một

Một kiểu mối quan hệ khác là mối quan hệ một đối một. Ví dụ: giả sử bạn cần ghi lại một số thông tin bổ sung về sản phẩm đặc biệt mà bạn sẽ hiếm khi cần hoặc chỉ áp dụng cho một vài sản phẩm. Vì bạn không cần thông tin này thường xuyên và vì việc lưu trữ thông tin trong bảng Sản phẩm sẽ tạo ra dung lượng trống cho tất cả sản phẩm áp dụng nên bạn cần đặt thông tin vào bảng riêng. Giống như bảng Sản phẩm, bạn sẽ sử dụng ID Sản phẩm làm khóa chính. Mối quan hệ giữa bảng bổ sung này và bảng Sản phẩm là mối quan hệ một đối một. Với mỗi bản ghi trong bảng Sản phẩm, sẽ có sẵn một bản ghi khớp duy nhất trong bảng bổ sung. Khi bạn xác định mối quan hệ này, cả hai bảng phải có chung trường.

Khi bạn phát hiện có nhu cầu về mối quan hệ một đối một trong cơ sở dữ liệu, hãy cân nhắc xem liệu bạn có thể kết hợp thông tin từ hai bảng vào trong một bảng không. Nếu vì lý do nào đó mà bạn không muốn kết hợp, có lẽ vì việc đó có thể dẫn đến nhiều dung lượng trống, danh sách sau đây sẽ cho biết cách bạn sẽ trình bày mối quan hệ trong thiết kế:

- Nếu hai bảng có cùng đối tượng, có lẽ bạn có thể thiết lập mối quan hệ bằng cách sử dụng cùng khóa chính trong cả hai bảng.
- Nếu hai bảng có đối tượng khác nhau với khóa chính khác nhau, hãy chọn một trong hai bảng (bảng này hoặc bảng kia) và chèn khóa chính của bảng vào bảng còn lại làm khóa ngoại.

Việc xác định mối quan hệ giữa các bảng sẽ giúp bạn đảm bảo rằng bạn có các bảng và cột đúng. Khi tồn tại mối quan hệ một đối một hoặc một đối nhiều, các bảng liên quan sẽ cần có chung một hoặc nhiều cột. Khi tồn tại mối quan hệ nhiều đối nhiều thì sẽ cần có một bảng thứ ba để thể hiện mối quan hệ.

TINH CHỈNH THIẾT KẾ

Sau khi bạn có các bảng, trường và mối quan hệ cần thiết, bạn nên tạo và điền dữ liệu mẫu vào các bảng, đồng thời thử làm việc với thông tin: tạo các truy vấn, thêm các bản ghi mới và v.v.. Việc này sẽ giúp tô sáng các vấn đề tiềm ẩn — ví dụ: bạn có thể cần thêm một cột bạn đã quên chèn trong giai đoạn thiết kế hoặc bạn có thể có một bảng cần phân tách thành hai bảng để loại bỏ trùng lặp.

Xem liệu bạn có thể sử dụng cơ sở dữ liệu để nhận được câu trả lời mong muốn không. Tạo bản thảo thô cho các biểu mẫu và báo cáo và xem liệu chúng có hiển thị dữ liệu như dự kiến không. Tìm dữ liệu trùng lặp không cần thiết khi bạn tìm thấy bất cứ trùng lặp nào, hãy thay đổi thiết kế để loại bỏ trùng lặp.

Khi dùng thử cơ sở dữ liệu ban đầu, bạn sẽ có thể phát hiện ra cơ hội để cải thiện. Sau đây là một vài nội dung cần kiểm tra:

- Bạn có quên bất cứ cột nào không? Nếu có, thông tin có thuộc về bảng hiện có không? Nếu đó là thông tin về vấn đề khác, bạn có thể cần tạo ra một bảng khác. Tạo cột cho mọi mục thông tin bạn cần theo dõi. Nếu không thể tính toán thông tin từ các cột khác thì có thể bạn sẽ cần tạo cột mới cho thông tin.
- Có cột nào là dư thừa vì thông tin có thể được tính toán từ các trường hiện tại không? Nếu một mục thông tin có thể được tính toán từ các cột hiện có khác — giá chiết khấu được tính từ giá bán lẻ chẳng hạn — tốt hơn bạn nên tính luôn, tránh tạo cột mới.
- Bạn có nhập lại thông tin trùng lặp vào một trong các bảng của mình không? Nếu vậy, có lẽ bạn cần phân chia bảng thành hai bảng có mối quan hệ một đối nhiều.
- Bạn có bảng với nhiều trường, số bản ghi hạn chế và nhiều trường trống trong bản ghi riêng? Nếu vậy, hãy cân nhắc thiết kế lại bảng để bảng có ít trường và có nhiều bản ghi hơn.
- Mỗi mục thông tin đã được tách thành nhiều phần hữu ích nhỏ nhất? Nếu bạn cần báo cáo, sắp xếp, tìm kiếm hoặc tính toán trên một mục thông tin, hãy đặt mục vào chính cột thông tin đó.
- Mỗi cột có chứa dữ kiện về đối tượng của bảng không? Nếu cột không chứa thông tin về đối tượng của bảng thì cột sẽ thuộc về một bảng khác.
- Tất cả mối quan hệ giữa các bảng đã được trình bày bằng trường chung hay bằng bảng thứ ba chưa? Các mối quan hệ một đối một và một đối nhiều yêu cầu cột chung. Mối quan hệ nhiều đối nhiều yêu cầu bảng thứ ba.

Tinh chỉnh bảng Sản phẩm

Giả sử mỗi sản phẩm trong cơ sở dữ liệu doanh số sản phẩm đều thuộc về cùng một danh mục, chẳng hạn như đồ uống, gia vị hoặc hải sản. Bảng Sản phẩm có thể chứa một trường hiển thị danh mục cho từng sản phẩm.

Giả sử sau khi kiểm tra và tinh chỉnh thiết kế của cơ sở dữ liệu, bạn quyết định lưu trữ mô tả danh mục cùng với tên danh mục. Nếu bạn thêm trường Mô tả Danh mục vào bảng Sản phẩm, bạn phải lặp lại từng mô tả danh mục cho từng sản phẩm thuộc danh mục đó — đây không phải là giải pháp tốt.

Giải pháp tốt hơn là đưa Danh mục thành một đối tượng mới cho cơ sở dữ liệu để theo dõi, với bảng riêng và khóa chính riêng. Khi đó, bạn có thể thêm khóa chính từ bảng Danh mục vào bảng Sản phẩm làm khóa ngoại.

Các bảng Danh mục và Sản phẩm có mối quan hệ một đối nhiều: một danh mục có thể bao gồm nhiều sản phẩm nhưng một sản phẩm chỉ có thể nằm trong một danh mục.

Khi bạn xem lại cấu trúc bảng, hãy theo dõi các nhóm lặp lại. Chẳng hạn, xem xét bảng chứa những cột sau:

- ID Sản phẩm
- Tên
- ID Sản phẩm1
- Tên1
- ID Sản phẩm2
- Tên2
- ID Sản phẩm3
- Tên3

Ở đây, mỗi sản phẩm là một nhóm cột lặp lại khác với các cột khác chỉ khi thêm số vào cuối tên cột. Khi bạn thấy các cột được đánh số theo cách này, bạn cần truy nhập lại vào thiết kế.

Thiết kế này có một vài thiếu sót. Với người mới bắt đầu, thiết kế buộc bạn đặt giới hạn trên cho số sản phẩm. Ngay khi vượt quá giới hạn đó, bạn phải thêm một nhóm cột mới vào cấu trúc bảng, đây là một tác vụ quản trị chính.

Một vấn đề nữa là nếu những nhà cung cấp đó có số sản phẩm ít hơn mức tối đa thì sẽ làm lãng phí không gian, vì các cột bổ sung sẽ trống. Thiếu sót nghiêm trọng nhất với thiết kế này là thiết kế khiến bạn khó có thể thực hiện nhiều tác vụ, như sắp xếp hoặc lập chỉ mục bảng bằng ID hay tên sản phẩm.

Bất cứ khi nào bạn thấy nhóm lặp lại, hãy xem kỹ lại thiết kế, chú ý đến việc phân tách bảng thành hai. Trong ví dụ ở trên, tốt hơn bạn nên sử dụng hai bảng, một bảng cho nhà cung cấp và một bảng cho sản phẩm, được liên kết bằng ID nhà cung cấp.

ÁP DỤNG CÁC QUY TẮC CHUẨN HÓA

Bạn có thể áp dụng các quy tắc chuẩn hóa dữ liệu (đôi khi được gọi là các quy tắc chuẩn hóa) làm bước tiếp theo trong thiết kế. Bạn sử dụng những quy tắc này để xem liệu bảng của bạn đã được cấu trúc đúng chưa. Quy trình áp dụng quy tắc vào thiết kế cơ sở dữ liệu của bạn được gọi là chuẩn hóa cơ sở dữ liệu hoặc đơn giản là chuẩn hóa.

Chuẩn hóa sẽ hữu dụng nhất sau khi bạn trình bày mọi mục thông tin và đạt đến thiết kế sơ bộ. Ý tưởng này giúp bạn đảm bảo mình đã phân chia các mục thông tin vào bảng thích hợp. Chuẩn hóa không thể đảm bảo rằng bạn đã có tất cả các mục dữ liệu đúng để bắt đầu.

Bạn áp dụng các quy tắc này liên tục, ở mỗi bước để đảm bảo thiết kế của bạn đạt đến một trong những trạng thái được gọi là "biểu mẫu bình thường". Năm biểu mẫu bình thường được chấp nhận rộng rãi — biểu mẫu bình thường đầu tiên đến biểu mẫu bình thường thứ năm. Bài viết này cung cấp thêm chi tiết về ba biểu mẫu đầu tiên vì chúng là biểu mẫu bắt buộc cho phần lớn thiết kế cơ sở dữ liệu.

Biểu mẫu bình thường đầu tiên

Biểu mẫu bình thường đầu tiên nêu rõ mỗi giao điểm giữa hàng và cột trong bảng đó tồn tại một giá trị duy nhất và không phải là danh sách giá trị. Ví dụ: bạn không thể đặt tên trường là Giá cả và đặt nhiều Giá cả trong đó. Nếu bạn quan niệm mỗi giao điểm của hàng và cột là một ô thì mỗi ô chỉ có thể nhận một giá trị.

Biểu mẫu bình thường thứ hai

Biểu mẫu bình thường thứ hai yêu cầu mỗi cột không khóa phải hoàn toàn phụ thuộc trên khóa chính, chứ không chỉ trên một phần khóa. Quy tắc này được áp dụng khi bạn có khóa chính chứa nhiều cột. Ví dụ: giả sử bạn có bảng chứa những cột sau, khi ID Đơn hàng và ID Sản phẩm hình thành khóa chính:

- ID Đơn hàng (khóa chính)
- ID Sản phẩm (khóa chính)
- Tên Sản phẩm

Thiết kế này vi phạm biểu mẫu bình thường thứ hai vì Tên Sản phẩm phụ thuộc vào ID Sản phẩm chứ không phải ID Đơn hàng nên thiết kế không phụ thuộc hoàn toàn vào khóa chính. Bạn phải loại bỏ Tên Sản phẩm khỏi bảng. Tên Sản phẩm thuộc một bảng khác (Sản phẩm).

Biểu mẫu bình thường thứ ba

Biểu mẫu bình thường thứ ba không chỉ yêu cầu tất cả cột không khóa phụ thuộc hoàn toàn vào khóa chính mà còn yêu cầu chúng độc lập với nhau.

Nói cách khác, mỗi cột không khóa phải phụ thuộc vào khóa chính và không gì ngoài khóa chính. Chẳng hạn, giả sử bạn có bảng chứa những cột sau:

- ID_Sản_phẩm (khóa chính)
- Tên
- SRP
- Chiết khấu

Giả sử Chiết khấu phụ thuộc vào giá bán lẻ đề xuất (SRP). Bảng này vi phạm biểu mẫu bình thường thứ ba vì cột Chiết khấu không khóa phụ thuộc vào một cột không khóa khác, SRP. Sự độc lập của cột nghĩa là bạn có thể thay đổi bất cứ cột không khóa nào mà không làm ảnh hưởng đến các cột khác. Nếu bạn thay đổi giá trị trong trường SRP, cột Chiết khấu sẽ thay đổi theo vì vi phạm quy tắc này. Trong trường hợp này, cột Chiết khấu sẽ được di chuyển sang bảng khác có khóa trên SRP.