



# **Thiết kế Cơ sở dữ liệu quan hệ**

**(tài liệu hướng dẫn)**

Mục lục	Lời mở đầu
Lời giới thiệu	Mô tả
Bài học	Cơ sở lý luận
Mục tiêu bài học	Mục tiêu
Nội dung	Điều kiện tiên quyết
Phần chữ	Phương pháp nghiên cứu
Phần minh họa	Quy ước
Bảng	
Thủ thuật	
Ghi chú	
Tóm tắt	
Phối hợp	
Byte kiến thức	
Lời chuyên gia	
Bài tập	
Thực hành	
Bài tập trên lớp	
Bài tập về nhà	
Đáp án bài tập	

Đánh giá khóa học

# Mục lục

Lời giới thiệu	
Lời mở đầu .....	
Mô tả.....	
Cơ sở lý luận .....	
Mục tiêu .....	
Điều kiện tiên quyết.....	
Phương pháp nghiên cứu .....	
Quy ước .....	
<b>Bài 1A: Giới thiệu về Hệ Quản trị CSDL quan hệ .....</b>	
Giới thiệu về Hệ Quản trị CSDL.....	
Người sử dụng.....	
Tại sao phải sử dụng Hệ Quản trị CSDL .....	
Lợi ích của phương pháp tiếp cận CSDL .....	
Cấu trúc của Hệ Quản trị CSDL .....	
Mức ngoài .....	
Mức định nghĩa .....	
Mức trong .....	
Các cấu kiện chức năng của Hệ Quản trị CSDL .....	
Người quản lý CSDL .....	
Người quản lý file.....	
Người quản lý ổ đĩa .....	
Lập đồ án và thiết kế CSDL.....	
Sự cần thiết phải lập đồ án CSDL .....	
Chu kỳ phát triển CSDL .....	
Hậu quả của việc lập đồ án và thiết kế CSDL sơ sài.....	
Tóm tắt.....	
<b>Bài 1B – Mô hình dữ liệu .....</b>	
Mô hình dữ liệu .....	
Mô hình logic dựa trên đối tượng.....	
Mô hình logic dựa trên bản tin.....	
Đại số quan hệ.....	

Các toán tử trong đại số quan hệ .....	
Ứng dụng của Hệ Quản trị CSDL quan hệ .....	
Tóm tắt.....	

## **Bài 1C – Phối hợp**

Byte kiến thức .....	
Mô hình phân cấp và mạng lưới.....	
Người quản trị CSDL .....	
Lời chuyên gia.....	
Luyện tập .....	
Thủ thuật .....	
Các câu hỏi thường gặp .....	
Bài tập.....	

## **Bài 1D – Thực hành**

Bài tập ở lớp.....	
Bài 1.....	
Bài 2.....	
Bài 3.....	
Bài 4.....	
Bài tập về nhà.....	

## **Bài 2A – Thiết kế CSDL logic**

Mô hình khái niệm .....	
Ánh xạ lược đồ tới bảng .....	
thủ thuật thiết kế CSDL logic .....	
Tóm tắt.....	

## **Bài 2B – Chuẩn hóa và phi chuẩn hóa dữ liệu**

Chuẩn hóa .....	
Phương pháp top-down và bottom-up .....	
Dư thừa dữ liệu.....	
Sự cần thiết phải chuẩn hóa .....	
Sự khác nhau giữa các dạng chuẩn và phi chuẩn hóa .....	
Các dạng chuẩn .....	
Phi chuẩn hóa .....	

Tóm tắt.....

## Bài 2C – Phối hợp

Byte kiến thức .....

    Miền.....

    Dạng chuẩn thứ tư .....

    Các dạng chuẩn khác .....

Lời chuyên gia.....

    Luyện tập .....

    Thủ thuật .....

    Các câu hỏi thường gặp .....

Bài tập.....

## Bài 2D – Thực hành

Bài tập ở lớp.....

    Bài 1.....

    Bài 2.....

    Bài 3.....

Bài tập về nhà.....

## Bài 3A – Thiết kế CSDL vật lý.....

Ngôn ngữ hỗ trợ mô hình quan hệ .....

    Các kiểu quan hệ .....

    Bảng cơ sở.....

    Kết quả truy vấn .....

    Hiển thị .....

Thực hiện hiển thị trên Hệ Quản trị CSDL quan hệ.....

    Thao tác DML trên hiển thị .....

    Các loại hiển thị.....

    Hiển thị dùng để làm gì .....

Toàn vẹn dữ liệu.....

Tóm tắt.....

## Bài 3B – Thực hiện giao dịch và bảo mật

Xử lý giao dịch CSDL .....

    Khôi phục giao dịch.....

Nhật ký giao dịch .....	
Các vấn đề xảy ra cùng lúc .....	
Khóa .....	
Các loại khóa.....	
Khóa chết .....	
Bảo mật dữ liệu .....	
Khái niệm về sơ đồ bảo mật dữ liệu .....	
Xây dựng sơ đồ bảo mật dữ liệu .....	
Tóm tắt.....	

## **Bài 3C – Phối hợp**

Byte kiến thức .....	
12 quy tắc của CODD.....	
Mục lục .....	
Khôi phục tư khóa chết.....	
Khôi phục CSDL.....	
Lời chuyên gia.....	
Luyện tập .....	
Thủ thuật .....	
Các câu hỏi thường gặp .....	
Bài tập.....	

## **Đáp án**

Đáp án cho bài 1C .....	
Đáp án cho bài 2C .....	
Đáp án chp bài 3C .....	

Lời giới thiệu

## Mô tả vắn tắt nội dung học tập

Giáo trình giới thiệu khái niệm, cấu trúc của hệ thống quản lí cơ sở dữ liệu (CSDL) và hệ thống quản lí CSDL hệ thức. Đồng thời, giáo trình bàn luận các vấn đề về thiết kế, phân tích trong quá trình thiết kế 1 CSDL và những vấn đề thiết yếu về ngôn ngữ truy nhập CSDL. Bên cạnh đó, giáo trình cũng đề cập đến những khái niệm trong giao dịch và bảo mật CSDL.

## Cơ sở lí luận

Sự phổ biến của các phần mềm quản lí CSDL giúp việc tạo ra các CSDL và ứng dụng của chúng một cách dễ dàng. Tuy nhiên, việc tạo CSDL mà không có một thiết kế chặt chẽ thường dẫn đến các lỗi CSDL hoặc sự cố hệ thống. Vì thế, kiến thức vững chắc về thiết kế CSDL cả về lý thuyết lẫn thực tiễn là yếu tố tối quan trọng. Giáo trình này cung cấp cho học viên những hiểu biết cần thiết cho việc tạo 1 CSDL chặt chẽ.

## Mục tiêu

Sau khi hoàn thành khóa học, học viên có thể:

- Định nghĩa hệ thống quản lí CSDL
- Phân biệt các thành phần của hệ thống CSDL
- Định nghĩa cấu trúc CSDL
- Định nghĩa chu kỳ đời sống và phương pháp luận phát triển của thiết kế CSDL
- Mô tả các loại hình CSDL
- Tạo giản đồ thực thể - quan hệ
- Liệt kê các hệ thức giữa các thực thể
- Định nghĩa hệ thống CSDL hệ thức
- Định nghĩa đại số quan hệ
- Tạo các CSDL thiết kế logic
- Định nghĩa chuẩn hóa và phi chuẩn hóa
- Tạo thiết kế CSDL vật lý
- Định nghĩa ngôn ngữ truy nhập CSDL (SQL)



- Thực thi hệ thức thông qua SQL
- Định nghĩa khung nhìn
- Giải thích những hạn chế trong xây dựng CSDL gây ra bởi SQL
- Định nghĩa, xác nhận, hủy các giao dịch CSDL
- Nhận diện các vấn đề tranh chấp trong xử lý CSDL nhiều người dùng
- Thực hiện chặn đối với dữ liệu
- Thực hiện bảo mật CSDL

## Điều kiện tiên quyết

Để hoàn thành chương trình học, học viên cần đáp ứng những yêu cầu sau:

Hiểu biết thực tiễn về hệ điều hành

# Phương pháp phối hợp hướng dẫn-tự học trong giảng dạy (Learning Architecture Based on Collaborative Constructivism (LACC))

## Giới thiệu

Nhìn chung, các phương pháp giảng dạy chủ yếu được thực hiện qua các hình thức sau:

- Đối thoại, trong đó giáo viên thuyết minh các kiến thức trong các bài giảng, giáo trình, học viên ghi chép, ghi nhớ để lập lại các kiến thức thu nhận được.
- Đối thoại, trong đó giáo viên truyền đạt kiến thức trong sách vở, học viên kết hợp xây dựng bài.
- Xây dựng tình huống, học viên tự tìm tòi, nghiên cứu để giải quyết tình huống, qua đó tích lũy kiến thức.

Bất kỳ hệ thống giảng dạy nào cũng đều có thể kết hợp tất cả các phương thức trên để tạo ra hiệu quả tốt nhất. Phương pháp giảng dạy truyền thống chú trọng vào hình thức đối thoại đầu tiên. Hai hình thức sau, cùng với những lý luận về nhận thức tình huống và phối hợp các phương pháp, đã đặt ra những thử thách to lớn cho việc xây dựng phương pháp luận mới trong việc dạy và học.

## Các giai đoạn trong học tập kinh nghiệm:

Theo David A Kolb, học tập kinh nghiệm gồm 4 giai đoạn:

- Concrete experiences (cụ thể/cảm tính)– CE: học viên tiếp cận bài học qua các ví dụ cụ thể
- Reflective observation (quan sát/phản hồi) – RO: học viên dựa trên kinh nghiệm tích lũy của mình để tìm hiểu ý nghĩa bài học
- Active experimentation (tích cực/hành động) – AE: học viên từ các ý nghĩa để đưa ra kết luận
- Abstract conceptualization (tư duy/ trừu tượng) – AC: học viên thử nghiệm với các trường hợp tương tự để có những kinh nghiệm mới

## Phân loại học viên



Dựa trên các giai đoạn học tập kinh nghiệm, có thể chia thành 4 kiểu người học với các phong cách:

- Học tập tách biệt (Divergers): người học có khả năng tưởng tượng tốt
- Học tập hòa đồng (Assimilators): người học có khả năng xác định, phân tích vấn đề
- Học tập hội tụ (Convergers) : người học có khả năng xác định và giải quyết vấn đề
- Học tập phối hợp (Accommodators): người học có khả năng lãnh đạo và chấp nhận rủi ro.

Phương pháp giảng dạy độc đáo LACC kết hợp nhiều phương thức giảng dạy xây dựng nên các phần học khác nhau thích ứng với khả năng và nhu cầu của từng học viên.

### Các phần trong chương trình:

Phương pháp giảng dạy LACC của NIIT gồm các phần sau:

- Cơ sở: phần mở đầu khóa học, dưới sự hướng dẫn của một chuyên viên, giới thiệu về các khái niệm mới, giúp học viên hình thành kiến thức nền tảng. Giảng viên đưa ra các bài thực hành cụ thể thông qua giải thích và minh họa, cung cấp cho học viên lượng kiến thức yêu cầu cho học phần sau.
- Phối hợp: phần giúp học viên làm quen với ngữ cảnh học tập qua việc cộng tác. Học viên được hướng dẫn và khuyến khích thực hành kỹ năng quan sát và phán đoán. Học viên có quyền lựa chọn nhiều công cụ học tập như nghiên cứu tình huống, phân tích giả thiết và vận dụng những kiến thức đã học trong các ngữ cảnh lớn hơn.

- Thực nghiệm: Tại đây, học viên được tìm hiểu cận kẽ vấn đề qua việc thực nghiệm ứng dụng. học viên thực nghiệm những hiểu biết và quan sát của mình để đưa ra kết luận, tạo được hiệu quả cao trong học tập.
- Vận dụng thực tiễn: phần này đưa học viên vào một môi trường thích hợp cho việc làm sáng tỏ các khái niệm. Tại đây học viên sẽ tự tìm tòi, nghiên cứu, so sánh, đối chiếu, vận dụng những hiểu biết của mình để đưa ra hướng giải quyết cho tình huống mình nghiên cứu. Học viên được làm quen với việc thực hiện các công trình nghiên cứu, tìm kiếm thông tin, thực hành kỹ năng phân tích (ISAS) và kỹ năng đối chiếu (RR).



### Trang thiết bị cần thiết

- Phòng học trực tuyến: phòng học được trang bị máy tính nối mạng.
- Môi trường thích hợp cho các nhóm phối hợp và vận dụng kỹ năng

## \*Tài liệu tham khảo

- Kolb, D. (1984a). *Experiential learning: Experience as the source of learning and development*. Eglewood Cliffs, NJ: Prentice Hall
- Kolb, D. (1984b). *Learning Styles Inventory*. Boston: McBer & Co.
- R Dangwal [NIIT] & Mitra [NIIT]. *Learning Styles and Perception of Self*, Giáo dục quốc tế, tạp chí điện tử, tập 4, số 4 (12/2000) ISN 1327-9548, Australia (2000)
- *Children and the Internet: Experiments with minimally invasive education in India*, Sugata Mitra [NIIT] và Vivek Rana [NIIT], Tạp chí Công nghệ giáo dục Anh Quốc, 32,2, trang 221-232 (2001)
- *Minimally Invasive education: A progress report on the “Hole-in-the-wall” experiments*, S. Mitra [NIIT], Tạp chí Công nghệ Giáo dục Anh Quốc, 34,3, trang 367-371 (2003)

## Chú thích:

	Cơ sở
	Mở đầu khái niệm mới
	Thông tin thêm
	Thủ thuật
Chữ kiểu Courier new	Mã hoặc cú pháp chương trình
<i>Chữ in nghiêng</i>	Giới thiệu thuật ngữ mới
<b>Chữ in đậm</b>	Tùy chọn menu, tùy chọn trong các hộp thoại, phím gõ
	Phối hợp
	Lời khuyên của chuyên gia
	Thực hành, bài tập thí nghiệm
	Bài tập về nhà

## Bài 1: 1A

# Giới thiệu hệ thống quản lý cơ sở dữ liệu hệ thức

## Mục tiêu bài học

Trong bài này, học viên sẽ tìm hiểu

- Định nghĩa hệ thống quản lý cơ sở dữ liệu (CSDL)
- Nhận diện các người dùng CSDL
- Nêu rõ sự cần thiết của hệ thống CSDL
- Định nghĩa cấu trúc của hệ thống CSDL dựa trên:
  - Mức khung nhìn
  - Mức khái niệm
  - Mức vật lý
  
- Phân biệt các cấu kiện chức năng của hệ quản trị CSDL
- Nêu sự cần thiết của việc quy hoạch CSDL
- Phân biệt các giai đoạn của chu kỳ phát triển CSDL (DDLC)
- Nhận biết các hậu quả của sự sơ sài trong quy hoạch và thiết kế CSDL

Chú ý: Tham khảo trên trang web [www.niitstudent.com](http://www.niitstudent.com) để tích lũy thêm kinh nghiệm học tập

Cơ sở dữ liệu là một tập hợp các bản ghi chép dữ liệu. Một trong những nhiệm vụ quan trọng của hệ thống máy tính là lưu trữ và quản lý các dữ liệu. Để hoàn tất nhiệm vụ này, học viên cần một phần mềm chuyên dụng, được gọi là Hệ quản trị CSDL (DBMS). Hệ quản trị CSDL được thiết kế để duy trì một lượng lớn dữ liệu. Công tác quản lý dữ liệu bao gồm:

- Xác định kết cấu của bộ nhớ
- Cung cấp các thiết bị cho xử lý dữ liệu
- Đảm bảo bảo mật dữ liệu trước các truy nhập trái phép

Hệ quản trị CSDL hiện nay có thể dùng được cho nhiều loại máy tính khác nhau, từ máy tính để bàn đến các máy tính trung tâm. Kích thước và khả năng của máy tính quyết định đến các công cụ của hệ thống, như bảo mật và lưu trữ.

Hệ máy tính đơn người dùng chỉ cho phép một đối tượng truy cập CSDL tại 1 thời điểm. Trong hệ máy tính nhiều người sử dụng, nhiều đối tượng có thể truy cập CSDL trong cùng 1 thời điểm.

Cùng với sự phát triển khả năng của máy tính để bàn, hệ thống CSDL nhiều người sử dụng có thể hỗ trợ 1 nhóm người sử dụng kết nối trực tiếp với nhau và tiếp cận đồng thời CSDL. Hệ thống CSDL này có thể nhân lên để có thể hỗ trợ hàng trăm, thậm chí hàng ngàn người sử dụng, tùy thuộc vào cấu hình của phần cứng mà hệ thống CSDL đang vận hành.

Mục đích chính của hệ quản trị CSDL:

- Cung cấp môi trường làm việc thuận tiện và hiệu quả, được dùng để lưu trữ và khôi phục dữ liệu từ CSDL
- Quản lý thông tin về người sử dụng hệ quản trị CSDL và những tác vụ mà người sử dụng có thể thực hiện đối với dữ liệu

## Đối với người sử dụng

Một người sử dụng hệ quản trị CSDL có thể thực hiện các tác vụ sau trên CSDL

- Thêm các tệp (file) vào CSDL
- Chèn dữ liệu vào 1 file hiện hành
- Khôi phục dữ liệu từ các file



- Cập nhật dữ liệu cho các file
- Xóa dữ liệu trong các file
- Gỡ bỏ file từ CSDL
- Tuân thủ quy định về bảo mật và toàn vẹn CSDL

Có 3 loại người sử dụng CSDL:

- Lập trình viên ứng dụng
- Người sử dụng trực tiếp
- Nhà quản trị CSDL (DBA)

Lập trình viên ứng dụng viết các chương trình hỗ trợ người dùng sử dụng CSDL. Các chương trình này thường được viết bằng các ngôn ngữ như C, C++, Visual Basic và Visual C++. Các chương trình này xử lý dữ liệu trong CSDL để khôi phục, chèn, xóa, hoặc sửa đổi dữ liệu.

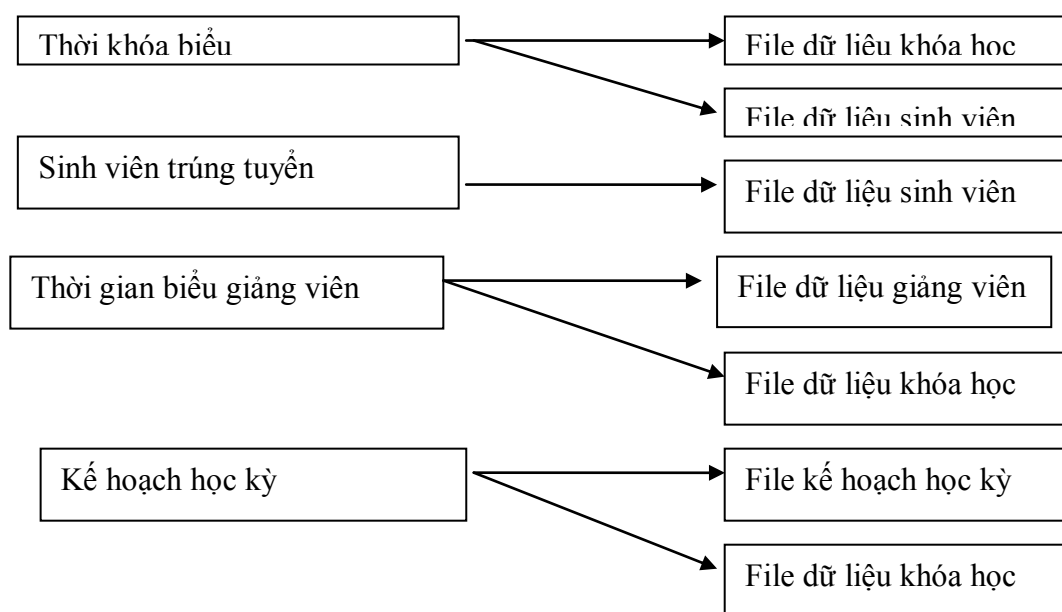
Người sử dụng tương tác với một hệ quản trị CSDL bằng cách yêu cầu chương trình ứng dụng hoặc bằng cách viết truy vấn trên ngôn ngữ truy nhập CSDL. Ngôn ngữ truy nhập cơ sở dữ liệu cho phép người sử dụng thực hiện các thao tác cơ bản, như khôi phục, xóa, chèn, và cập nhật dữ liệu.

Quản trị viên CSDL kết hợp chức năng của người thu thập thông tin về dữ liệu cần lưu trữ, thiết kế và duy trì CSDL và bảo mật. CSDL cần được thiết kế và duy trì để đưa ra thông tin chính xác cho người sử dụng.

## Tại sao cần sử dụng hệ thống quản lý cơ sở dữ liệu?

Trước khi hệ quản trị CSDL ra đời, phương pháp xử lý dữ liệu truyền thống được thực hiện như sau. Trong phương pháp truyền thống, một chương trình (hoặc một tập hợp các chương trình) được xây dựng cho từng ứng dụng, tạo ra hoặc một vài file dữ liệu cho một ứng dụng. nhiều dữ liệu xuất hiện nhiều lần trong các file khác nhau. Tuy nhiên, một ứng dụng có thể đòi hỏi file phải được sắp xếp theo từng phạm vi riêng biệt, trong khi ứng dụng khác cũng đòi hỏi dữ liệu đó. Một nhược điểm lớn của phương pháp truyền thống là kỹ thuật lưu trữ và truy cập gắn vào vào chương trình. Vì thế, khi cùng một dữ liệu được yêu cầu bởi hai ứng dụng, dữ liệu phải được lưu trữ ở hai nơi khác nhau vì mỗi ứng dụng đòi hỏi dữ liệu lưu trữ khác nhau.

Ví dụ, trường hợp của một trường đại học



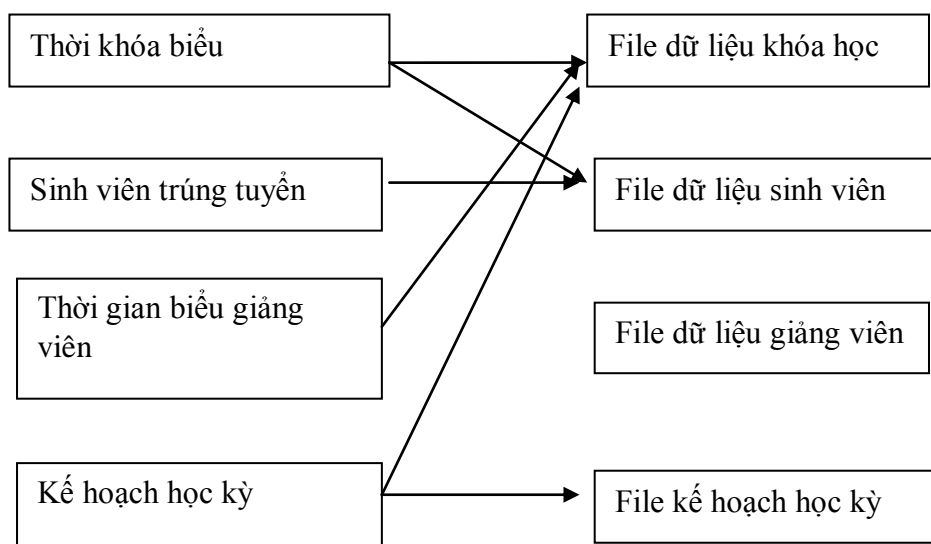
### *Liên kết giữa ứng dụng và file dữ liệu*

Một trường đại học thông thường cần ghi chép nhiều loại dữ liệu về chương trình học, sinh viên, học kỳ, giảng viên... khi bắt đầu 1 khóa học mới. Khi lập thời khóa biểu, trường cần dữ liệu từ file dữ liệu khóa học và file dữ liệu sinh viên. Dữ liệu trong file dữ liệu sinh viên được phân chia theo mã khóa học của từng sinh viên. Khi lập danh sách sinh viên trúng tuyển, dữ liệu trong file dữ liệu sinh viên được sắp xếp theo ngày đăng ký nhập học. Vì thế, mặc dù cùng 1 file dữ liệu được dùng cho các tác vụ khác nhau, nó cần được sao thành nhiều bản và lưu tại nhiều nơi thích hợp cho từng tác vụ.

Phương thức truyền thống trong việc xử lý dữ liệu có những nhược điểm sau:

- Dữ liệu sao chép nhiều lần: file dữ liệu khóa học chứa những thông tin về nhiều khóa học khác nhau. Thông tin bao gồm mã môn học, tên môn học, đặc thù môn học, etc. các thông tin này được sử dụng để lập thời gian biểu cho giảng viên và lập kế hoạch cho kỳ học. nói cách khác, cùng 1 dữ liệu được sử dụng với nhiều mục đích khác nhau. Thay vì lưu trữ thông tin trong 1 file, nhiều file được dùng cho những ứng dụng khác nhau. Điều này được xem như dư thừa dữ liệu
- Dữ liệu không nhất quán: Cùng 1 dữ liệu lưu trữ ở nhiều nơi khác nhau dẫn đến sự thiếu nhất quán. Ví dụ, danh sách sinh viên trúng tuyển thêm vào 10 sinh viên. Dữ liệu về những sinh viên mới này cần được cập nhật trong cả file dữ liệu dùng cho việc lập thời khóa biểu. Nếu sự thay đổi này không được cập nhật ở tất cả các bản sao của file, các bản sao sẽ không nhất quán.

Lợi ích của phương thức sử dụng CSDL



*Phương thức sử dụng CSDL*

Ưu điểm chủ yếu của phương thức sử dụng CSDL trong xử lý dữ liệu được là hệ quản trị CSDL có chế độ kiểm soát tập trung đối với dữ liệu.

Dưới đây là một số lợi ích của việc sử dụng CSDL:

- Giảm sự rườm rà: sử dụng cơ sở dữ liệu, các dữ liệu không nhất thiết phải lưu trữ một cách độc lập cho từng ứng dụng.
- Tránh tình trạng thiếu nhất quán: giảm được sự chông chéo trong lưu trữ dữ liệu đồng thời giúp tránh được tình trạng thiếu nhất quán.
- Dữ liệu được chia sẻ: nhiều tác vụ hiện thời có thể cùng sử dụng một dữ liệu, hoặc tác vụ mới có thể được thực hiện sử dụng chính dữ liệu đó.
- Các quy phạm được tuân thủ chặt chẽ: với việc quản lý dữ liệu một cách tập trung, quản trị viên đảm bảo tính chuẩn mực của dạng thức dữ liệu khi lưu trữ và xử lý, đặc biệt là trong trao đổi và dịch chuyển dữ liệu giữa hai hệ thống máy tính.
- Tăng tính bảo mật: chỉ những người sử dụng hợp pháp được phép truy cập vào CSDL. Quản trị viên có thể kiểm tra độ an toàn đối với từng công việc thực hiện trên dữ liệu. Ví dụ, một cá nhân có thể truy cập vào một file nhưng không có quyền xóa hoặc thay đổi file đó.
- Đảm bảo tính toàn vẹn: sự thiếu nhất quán trong việc nhập dữ liệu thường làm ảnh hưởng đến tính toàn vẹn dữ liệu. Ngay cả khi dữ liệu không quá rườm rà, CSDL vẫn có khả năng không nhất quán. Ví dụ, 1 sinh viên đăng ký 10 môn học, trong khi số lượng môn học tối đa được phép chọn là 7 môn. Trường hợp khác, sinh viên đăng ký môn học không trong chương trình của kỳ. Chúng ta có thể tránh được các tình trạng trên khi sử dụng hệ quản trị CSDL. Hệ quản trị CSDL sẽ tiến hành kiểm tra tính toàn vẹn ngay khi các thao tác cập nhật được tiến hành.

# Cấu trúc của Hệ thống Quản lý Cơ sở dữ liệu

Cấu trúc của hệ quản trị CSDL có thể chia thành 3 mức:

- Mức khung nhìn
- Mức khái niệm
- Mức vật lý

Mức khung nhìn 1   Mức khung nhìn 2   Mức khung nhìn 3

Mức khái niệm

Mức vật lý

*Ba mức độ của hệ quản trị CSDL*

---

## Mức khung nhìn

Đây là mức độ cao nhất, gần nhất đối với người sử dụng, còn được gọi là *user view*. *User view* khác với cách lưu trữ dữ liệu thông thường trong CSDL. Những gì hiển thị tại đây chỉ là một phần của CSDL thực tế. Bởi vì mỗi cá nhân không sử dụng hết toàn bộ CSDL, chỉ có những phần cần thiết cho người sử dụng được hiển thị. Ví dụ, người sử dụng và lập trình viên sử dụng những hiển thị bên ngoài khác nhau.

Ví dụ, một giảng viên muốn hiển thị dữ liệu dưới dạng tập hợp danh sách sinh viên và môn học của trường. Người quản lý hiển thị dữ liệu dưới dạng tập hợp các tài liệu, giáo trình của trường. Mỗi người chỉ cần phần dữ liệu liên quan đến tác vụ của mình. Những phần dữ liệu này được xem là hiển thị cho người sử dụng (*user view*), hay hiển thị bên ngoài.

Mỗi người sử dụng dùng một loại ngôn ngữ để tiến hành thao tác trên CSDL. Lập trình viên sử dụng cả ngôn ngữ thế hệ thứ ba, như COBOL hay C, hoặc ngôn ngữ thế hệ thứ tư thiết kế cho hệ quản trị CSDL như Visual FoxPro hay MS Access.

Người sử dụng dùng ngôn ngữ truy vấn để truy cập vào dữ liệu trong CSDL.

Ngôn ngữ truy vấn là ngôn ngữ kết hợp bởi ba loại ngôn ngữ thứ cấp:

- Ngôn ngữ định nghĩa dữ liệu (DDL)
- Ngôn ngữ thao tác dữ liệu (DML)
- Ngôn ngữ điều khiển dữ liệu (DCL)

Ngôn ngữ định nghĩa dữ liệu định hình và khai báo đối tượng trong CSDL, ngôn ngữ thao tác dữ liệu thực hiện thao tác cho các đối tượng này. Ngôn ngữ điều khiển dữ liệu điều khiển sự truy cập của người dùng đối với đối tượng trong CSDL.

---

## Mức định nghĩa

Đây là mức nằm giữa mức vật lý và mức khung nhìn của hệ quản trị CSDL. *Mức định nghĩa* trên tổng thể được coi như toàn bộ CSDL, được sử dụng bởi quản trị viên. Trong mức khái niệm, dữ liệu được hiển thị không bị giới hạn bởi ngôn ngữ lập trình như ở mức khung nhìn.

*Chú ý: mức định nghĩa mô tả dữ liệu lưu trữ trong CSDL và mối quan hệ của chúng với các dữ liệu khác.*

---

## Mức vật lý

Dữ liệu trong mức này giải quyết vấn đề với lưu trữ vật lý và là mức thấp nhất của cấu trúc. Mức vật lý mô tả kết quả vật lý của những bản lưu trữ.

Dưới đây là ví dụ minh họa cho ba mức:

## Ánh xạ

Ánh xạ đo mức độ tương thích giữa các mức. có 2 mức ánh xạ theo cấu trúc này. Thứ nhất là ánh xạ giữa mức khung nhìn và mức khái niệm. Thứ hai là ánh xạ giữa mức khái niệm và mức vật lý. Ánh xạ mức khung nhìn-mức khái niệm đo sự tương thích giữa hiển thị khái niệm và hiển thị bên ngoài. Nó cụ thể hóa cách người dùng hiển thị dữ liệu khái niệm. Ánh xạ mức khái niệm-mức vật lý đo sự tương thích giữa hiển thị khái niệm và hiển thị bên trong. Nó giúp làm rõ cách thức lưu trữ dữ liệu khái niệm.

Bước đầu tiên trong việc thiết kế 1 CSDL là khái quát mức định nghĩa. Mức định nghĩa sau đó được ánh xạ với mức khung nhìn, cả về hiển thị mức khung nhìn lẫn điều kiện của nó. Tiếp theo, ánh xạ mức khái niệm-mức vật lý được tiến hành. Cách lưu trữ dữ liệu phụ thuộc vào mức khái niệm. Cấu trúc ba mức của hệ quản trị CSDL góp phần tạo nên sự độc lập của dữ liệu.

## Các cấu kiện của hệ quản trị CSDL

Một hệ quản trị CSDL gồm có nhiều module với các chức năng khác nhau trong toàn bộ hệ thống. Các cấu kiện chức năng của hệ quản trị CSDL là:

- Quản trị CSDL
- Quản lý tệp tin
- Quản lý ổ đĩa

### Quản trị CSDL

Bộ phận Quản trị CSDL là một cấu kiện phần mềm của Hệ quản trị CSDL. Phần mềm có chức năng chuyển đổi truy vấn của người sử dụng thành các lệnh thích hợp, góp phần duy trì tính nhất quán và toàn vẹn của CSDL và tăng cường bảo mật dữ liệu. Nó cũng đồng bộ những thao tác của nhiều người sử dụng cùng lúc. Ngoài ra, phần mềm còn có khả năng tiến hành phòng bị và khôi phục các thao tác.

### Quản lý tệp tin

Bộ phận Quản lý tệp tin xử lý tất cả các yêu cầu đối với dữ liệu. Nó tìm kiếm những vùng chứa bản ghi được yêu cầu. những vùng này sau đó được yêu cầu từ các từ bộ phận quản lý ổ đĩa. Bộ phận quản lý tệp tin chọn lựa bản ghi từ vùng tìm được và chuyển qua bộ phận quản lý CSDL. Bộ phận quản lý tệp tin đồng thời quản lý dung lượng và cấu trúc bộ nhớ.

### Quản lý ổ đĩa

Bộ phận quản lý ổ đĩa điều khiển dữ liệu đầu vào và đầu ra vật lý. Nó tương tác với bộ phận quản lý tệp tin để đọc và ghi trên các thiết bị lưu trữ khi nhận được yêu cầu từ bộ phận quản lý tệp tin. Các hiển thị được xử lý như dữ liệu chưa xử lý. Các thao tác thực hiện bởi bộ phận quản lý ổ đĩa là:

- Truy hồi các khối dữ liệu
- Thay thế các khối dữ liệu
- Gỡ bỏ các khối dữ liệu



## Quy hoạch và thiết kế CSDL

Để đảm bảo chuyển giao hiệu quả từ lưu trữ dữ liệu đơn lẻ sang lưu trữ tập trung sử dụng Hệ quản trị CSDL, chúng ta cần có quy hoạch CSDL. Quy hoạch CSDL là quá trình chiến lược để đáp ứng nhu cầu thông tin cho giai đoạn nhất định trong tương lai. Đây là hoạt động đầu tiên trong mọi dự án vận hành, góp phần rất lớn trong việc thiết kế và thực thi CSDL.

### Nhu cầu quy hoạch CSDL

Việc quy hoạch CSDL xuất phát từ nhu cầu thông tin của tổ chức. Quy hoạch CSDL định rõ loại CSDL tổ chức cần dùng và sẵn có, cũng như tìm kiếm thông tin hiện có, theo dõi thông tin và tìm hiểu nhu cầu tương lai. Thêm vào đó, quy hoạch CSDL còn giúp người sử dụng hiểu rõ về sự chuyển giao thông tin trong tổ chức.

Lợi ích của quy hoạch CSDL:

- Định hình những hiểu biết về quản lý tài nguyên thông tin.
- Xác định yêu cầu của nguồn thông tin, đảm bảo tính khả dụng của tài nguyên.
- Sử dụng hiệu quả tài nguyên, bao gồm sự phối hợp chặt chẽ giữa các bộ phận trong tổ chức.
- Cụ thể hóa kế hoạch hành động để đạt được mục tiêu đề ra.
- Phát triển CSDL phù hợp với yêu cầu của tổ chức.

### Chu trình phát triển CSDL

Chúng ta cần theo sát 1 chu trình khi thiết kế và thực thi CSDL cho tổ chức. Chu trình này được gọi là chu trình phát triển CSDL (DDLC). DDLC gồm có các giai đoạn:

- Quy hoạch sơ bộ
- Khảo sát tính khả thi
- Xác định yêu cầu
- Thiết kế khái niệm
- Cài đặt
- Đánh giá và bảo trì CSDL

## Quy hoạch sơ bộ

Đây là bước đầu tiên đối với mọi dự án phát triển CSDL. Tại đây, tổ chức cần tìm câu trả lời cho các câu hỏi sau:

- Bao nhiêu trình ứng dụng được sử dụng?
- Các trình ứng dụng có những chức năng gì?
- Các file được kết hợp với các trình ứng dụng như thế nào?
- File và ứng dụng mới nào được tạo ra?

Những thông tin thu được sau đó được sử dụng để xác định mối quan hệ giữa các ứng dụng và công năng của các ứng dụng đó. Thông tin trên giúp tìm ra yêu cầu tương lai của hệ thống và kiểm tra nguồn lợi tài chính mà hệ CSDL mang lại.

## Khảo sát tính khả thi

Gồm có khảo sát tính thực thi của dự án về công nghệ, vận hành và kinh tế của CSDL. Khảo sát tính khả thi đòi hỏi một báo cáo dựa trên lời giải đáp cho các câu hỏi sau:

- Tính khả thi về công nghệ: tổ chức có đủ khả năng công nghệ để phát triển CSDL hay không?
- Tính khả thi cho vận hành: tổ chức có đủ nguồn nhân lực, ngân sách, chuyên gia trong ngành để thực thi CSDL.
- Tính khả thi về kinh tế: Xem xét lợi ích tạo ra là gì, có thể tính toán được lợi nhuận và chi phí hay không?

Tổ chức tập trung vào từng giai đoạn riêng như sau:

- Khảo sát tính thực thi về công nghệ: kiểm tra những phần cứng và phần mềm cần thiết sẵn có để đáp ứng nhu cầu thông tin, bao gồm kiểm tra tài nguyên và hệ thống máy tính có đủ hay cần nhập thêm. Thêm vào đó, cần xem xét về những kỹ năng cần đào tạo.
- Khảo sát vận hành: kiểm tra những kỹ năng và nguồn nhân lực cần thiết để thực thi hệ CSDL
- Khảo sát tính thực thi kinh tế: khảo sát về chi phí – lợi nhuận từ hệ CSDL đề suất. Vấn đề được giải quyết bởi các câu hỏi sau:

Lợi nhuận sẽ có được sau bao lâu?

Các ban các nhau có thể sử dụng cùng 1 dữ liệu một cách dễ dàng hay không?

Hệ CSDL được ứng dụng như thế nào?

Lợi ích của các ứng dụng này trong thực tế là gì?

Chiến lược của các công ty cạnh tranh là gì?

Hệ CSDL đóng vai trò thế nào trong kế hoạch phát triển của công ty?

Chi phí cho các phần mềm, phần cứng và những lập trình liên quan đến phát triển hệ CSDL cần được xác định. Thêm vào đó, cần tính toán các phí tổn ngầm từ những thay đổi bất ngờ. Ví dụ, các phần mềm cần được nâng cấp để có thể tiếp tục công việc.

## Xác định yêu cầu

Trong bước này, cần xác định những vấn đề sau:

- Lĩnh vực ứng dụng của hệ CSDL
- Những yêu cầu thông tin về vùng chức năng và vùng quản lý
- Những yêu cầu về phần cứng và phần mềm

Chúng ta có thể tìm thấy những yêu cầu thông tin bằng cách sử dụng phiếu điều tra, phỏng vấn với nhân viên, các báo cáo và biểu mẫu. Chúng ta cần mở rộng các mô hình thông tin tổng quát, tạo trong quá trình quy hoạch CSDL để xây dựng mô hình cho từng vùng chức năng. Các mô hình này sẽ làm nền tảng cho việc thiết kế chi tiết hệ CSDL.

Những nhiệm vụ của xác định yêu cầu:

- Phân tích yêu cầu thông tin để xác định lĩnh vực ứng dụng của hệ CSDL. Chúng ta phải xác định tạo CSDL tập trung hay phân tán. Viết 1 bản tường trình ngắn gọn về lĩnh vực ứng dụng của hệ thống.
- Thảo vấn bản về yêu cầu đối với người sử dụng ở mức vận hành và quản lý. Các tư liệu nên bao gồm mô hình thông tin tổng quát cho mỗi vùng chức năng và chỉ rõ các chương trình cần thiết để đáp ứng các yêu cầu. Các tư liệu cũng có thể kèm theo các tài liệu từ các biểu mẫu, báo cáo, điều tra...thống kê từ người sử dụng.

- Thiết lập các yêu cầu về phần cứng, phần mềm kèm theo hỗ trợ cho sự hoạt động. Chúng ta cần có thông tin về các vấn đề sau:

- Số lượng truy cập vào hệ thống
- Số lượng mua hệ thống thống kê hàng ngày
- Số lượng đặt mua

Các thông tin này góp phần xác định Hệ quản trị CSDL cần dùng. Đồng thời, chúng xác định yêu cầu về dung lượng ổ đĩa và hỗ trợ in ấn. Chúng ta có thể kèm theo thông tin những báo cáo tường trình bao gồm các biểu đồ về cấu hình phần cứng, phần mềm.

- Thảo một kế hoạch cho từng bước phát triển của 1 hệ CSDL. Đồng thời, nhận biết các ứng dụng ban đầu. các ứng dụng này cần phải nhỏ và không quá quan trọng để tránh bị ảnh hưởng từ sự thực thi CSDL.

## Thiết kế khái niệm

Tại đây, chúng ta có thể tạo sơ đồ khái niệm của hệ CSDL. Sơ đồ khái niệm gồm các bộ phận của dữ liệu, mối quan hệ và hạn chế của chúng. Sơ đồ cần thật cụ thể để có thể đưa vào thực tế.

Đầu tiên, tạo mô hình chi tiết của khung nhìn trong từng vùng chức năng. Tiếp theo, kết hợp các mô hình lại với nhau thành một mô hình dữ liệu khái niệm chi tiết. mô hình tạo ra sơ đồ mức CSDL khái niệm. Chúng ta sẽ đề cập đến mô hình dữ liệu khái niệm trong các bài học sau.

## Thực thi

Tại đây, chúng ta cần có Hệ quản trị CSDL. Mô hình khái niệm chi tiết được chuyển thành mô hình thực thi của Hệ quản trị CSDL. Tiếp theo, chúng ta lập từ điển dữ liệu, đưa CSDL vào sử dụng, phát triển các trình ứng dụng, viết hướng dẫn cho người sử dụng. Từ điển dữ liệu là bộ lưu trữ trung tâm của các định nghĩa cấu trúc dữ liệu trong hệ CSDL. Nó gồm các thông tin về nguyên tắc bảo mật, quyền truy cập và các kiểm soát liên quan. Từ điển dữ liệu đồng thời được dùng để duy trì chuẩn dữ liệu, giải quyết các vấn đề xảy ra khi chia sẻ dữ liệu cho các ứng dụng. Vì vậy, từ điển dữ liệu có vai trò như trung tâm điều khiển của hệ CSDL. Như vậy, xây dựng từ điển CSDL là bước quan trọng của việc thực thi.

## Đánh giá và bảo trì CSDL

Đây là bước đánh giá chất lượng của CSDL đã đưa vào sử dụng. chúng ta có thể phỏng vấn người sử dụng về những điểm chưa đạt yêu cầu. Sau đó thực hiện những thay đổi cần thiết.

Bảo trì hệ thống là sự cải tiến, thêm vào các chương trình và bộ phận dữ liệu mới. những thay thế này kết hợp với sự thay đổi trong nhu cầu thương mại.

## Hậu quả của việc quy hoạch và thiết kế CSDL sơ sài

Nếu chúng ta không bỏ nhiều công sức cho thiết kế và quy hoạch CSDL, hệ CSDL chắc chắn sẽ thất bại. Nguyên nhân chủ yếu là:

- Đối tượng sử dụng chính bị loại ra khỏi quy trình quy hoạch và thiết kế
- Tập hợp quá nhiều hay quá ít thông tin
- CSDL không dễ dàng chỉnh sửa theo nhu cầu thương mại
- Khoảng cách thời gian giữa việc mua hệ CSDL và đưa vào sử dụng quá ít.
- Hệ CSDL không được kiểm tra trước khi đưa dữ liệu vào CSDL
- Hướng dẫn hạn chế hoặc không có hướng dẫn cho người sử dụng
- Quá trình quy hoạch bị gián đoạn khi có sự thay đổi về nhân sự
- Thiếu xác nhận khi phân bổ tài nguyên (thời gian và tiền bạc) cho dự án phát triển CSDL.

## Tóm tắt bài học

Qua bài học này, học viên đã học:

- Một Hệ quản trị CSDL là một tập hợp các dữ liệu liên kết và các chương trình khai thác dữ liệu đó.
- Mục đích chính của Hệ quản trị CSDL là:
  - Tạo môi trường thuận tiện và hiệu quả để lưu trữ và truy hồi dữ liệu từ CSDL
  - Quản lý các thông tin về người sử dụng Hệ quản trị CSDL và những tác vụ họ có thể làm đối với dữ liệu
- Dữ liệu trong CSDL có thể kết hợp và chia sẻ
- Người sử dụng Hệ quản trị CSDL có thể phân thành lập trình viên ứng dụng, người sử dụng trực tiếp và quản trị viên (DBA).
- Hệ quản trị CSDL có những ưu điểm sau:
  - Giảm bớt sự rườm rà
  - Nhất quán
  - Dữ liệu được chia sẻ
  - Duy trì chuẩn mực
  - Thắt chặt bảo mật
  - Toàn vẹn
- Cấu trúc CSDL bao gồm mức khung nhìn, mức khái niệm và mức vật lý.
- Các cấu kiện chức năng của Hệ quản trị CSDL:
  - Quản trị CSDL
  - Quản lý file
  - Quản lý ổ đĩa
- Quy hoạch CSDL là quy trình chiến lược trong việc tìm hiểu nhu cầu thông tin cho tổ chức trong khoảng thời gian nhất định trong tương lai, góp phần trong thiết kế và thực thi CSDL.
- Quy trình phát triển CSDL (DDLC) gồm 6 giai đoạn:
  - Quy hoạch sơ bộ
  - Khảo sát tính khả thi
  - Xác định yêu cầu
  - Thiết kế khái niệm
  - Thực thi
  - Đánh giá và bảo trì CSDL
- Nếu chúng ta không bỏ nhiều công sức cho việc quy hoạch và thiết kế CSDL, CSDL sẽ thất bại.

## **Bài 1B**

# Mô hình dữ liệu

### Mục tiêu bài học

Trong bài này, học viên sẽ tìm hiểu:

- Mô tả các loại mô hình dữ liệu
- Tạo mô hình thực thể - liên hệ
- Liệt kê các mối liên hệ giữa các thực thể
- Định nghĩa Hệ quản trị CSDL quan hệ
- Mô tả hệ điều hành hoạt động trên các quan hệ

## Mô hình dữ liệu

Mô hình dữ liệu là sự mô tả của 1 tổ chức dữ liệu trong CSDL. Cùng với dữ liệu, mô hình dữ liệu còn mô tả mối quan hệ giữa dữ liệu và các hạn chế xác định trên dữ liệu. Mô hình dữ liệu được chia thành 2 nhóm:

- Mô hình logic dựa trên đối tượng: chú trọng vào mô tả dữ liệu, quan hệ giữa các dữ liệu và các hạn chế.
- Mô hình logic dựa trên mẫu tin

## Mô hình logic dựa trên đối tượng

Mô hình thông dụng nhất là mô hình thực thể - quan hệ (mô hình E/R). đây được coi là mô hình hiệu quả nhất trong thiết kế CSDL. Peter Chen giới thiệu mô hình này vào năm 1976. Từ đó trở đi, rất nhiều người đã góp phần làm nên giá trị của nó.

## Mô hình thực thể - quan hệ

Mô hình thực thể - quan hệ dựa trên nhận thức thực tế gồm tổ hợp các đối tượng hay thực thể và mối quan hệ giữa chúng. Chen đưa ra không chỉ mô hình E/R mà còn đưa ra kỹ thuật vẽ giản đồ tương ứng với nó. Quan sát giản đồ sau:



Giản đồ gồm nhiều bộ phận mô tả mối liên hệ giữa các bộ phận đó. Nhiều bộ phận của giản đồ là các thực thể: course và semester, mối quan hệ: taught, thành phần: title, code, start-dt, end-dt và sem #.

## Thực thể

Chen định nghĩa *thực thể* là “một thứ rất dễ nhận biết”. Thực thể là bất cứ vật, người, hành động, vị trí nào mà dữ liệu được ghi lại. 1 số ví dụ về thực thể là sinh viên, môn học, điểm. Trong kỹ thuật vẽ giản đồ, thực thể được đặt tên và đặt trong một hộp chữ nhật.

SINH VIÊN

MÔN HỌC

ĐIỂM SỐ

*Thực thể*

Chúng ta cần phân biệt giữa *tập thực thể* và *thể hiện thực thể*. Kiểu thực thể là tập hợp những thứ có đặc điểm chung. Ví dụ, SINH VIÊN, MÔN HỌC và ĐIỂM SỐ. Tập thực thể thường được viết hoa. Một thể hiện thực thể là những cá thể. Ví dụ, Jack Ripper là một thể hiện thực thể của thực thể SINH VIÊN, Triết học là một thể hiện thực thể của thực thể MÔN HỌC, A là thể hiện thực thể của thực thể ĐIỂM SỐ.

Có hai kiểu thực thể là phụ thuộc và độc lập. Một thực thể mà sự tồn tại phụ thuộc vào sự tồn tại của một thực thể khác gọi là *thực thể phụ thuộc*. Ví dụ, tập thực thể ĐĂNG KÝ. ĐĂNG KÝ chỉ ra các môn học được giảng dạy trong khóa học. một môn với cùng mã số môn có thể được dạy ở nhiều kỳ khác nhau và các trường khác nhau. Sự tồn tại của ĐĂNG KÝ phụ thuộc vào sự tồn tại của thực thể MÔN HỌC.

Thực thể B là thực thể tồn tại phụ thuộc vào thực thể A nếu:

- Một vài thể hiện của A tồn tại trước khi B tồn tại
- Khi A ngừng hoạt động, kéo theo B cũng ngừng hoạt động

Ví dụ, trường cho phép mở KHÓA HỌC trên thiết kế CSDL trong học kỳ II. Trường có thể mở khóa học này chỉ trong trường hợp thể hiện thực thể “thiết kế CSDL” tồn tại. Nếu trường dừng thể hiện thực thể “thiết kế CSDL” việc mở khóa học cũng sẽ ngưng lại. Thực thể phụ thuộc được mô tả bởi một hộp chữ nhật 2 đường bao và 1 đường dẫn tới thực thể mà nó phụ thuộc.

Một thực thể độc lập không phụ thuộc sự tồn tại của thực thể khác. Thực thể phụ thuộc gọi là thực thể yếu, thực thể độc lập gọi là thực thể thông thường

## *Thực thể phụ thuộc*

*Chú ý: Thực thể thông thường và những thực thể yếu của nó được xem như một đối tượng trong CSDL bởi vì thực thể yếu phụ thuộc vào thực thể thông thường.*

## Mối quan hệ

Chen định nghĩa *mối quan hệ* là “sự liên kết giữa các thực thể”. Ví dụ, sinh viên và giảng viên có mối liên kết với nhau. Mối quan hệ này thể hiện rằng 1 giảng viên dạy nhiều sinh viên và 1 sinh viên học nhiều giảng viên. Mối quan hệ ở đây là GIẢNG DẠY. Cũng như thực thể, mối quan hệ cũng cần được phân biệt giữa tập và thể hiện.

*Tập hợp các mối quan hệ* là sự liên kết giữa các tập thực thể. Ví dụ, mối liên kết giữa SINH VIÊN và GIẢNG VIÊN khi SINH VIÊN và GIẢNG VIÊN đều là tập thực thể. *Thể hiện các mối quan hệ* là sự liên kết giữa các hiện bản thực thể, ví dụ, mối liên kết giữa sinh viên J.Reins và giảng viên là tiến sĩ Jack.

Mối quan hệ được mô tả là một hình quả trám với tên của tập hợp mối quan hệ.

GIẢNG VIÊN

GIẢNG DẠY

SINH VIÊN

*Các mối quan hệ*

Một mối quan hệ có thể liên kết thực thể với chính nó. Ví dụ, 1 giảng viên có thể kết hôn với 1 giảng viên khác. Vì vậy, mối quan hệ sau đây có khả năng xảy ra:

GIẢNG VIÊN

KẾT HÔN

*Liên kết 1 thực thể với chính nó*

Ví dụ khác về thực thể liên kết với chính nó là một sinh viên thuyết trình trước các sinh viên khác.

SINH VIÊN

CHUYÊN ĐỀ THẢO LUẬN

*Liên kết 1 thực thể với chính nó*

Liên kết đa phương cũng có thể tồn tại giữa các thực thể. Ví dụ:

## QUẢN LÝ

GIẢNG VIÊN

GIẢNG DẠY

SINH VIÊN

## ĐÁNH GIÁ

*Liên kết đa phương với cùng một thực thể*

Các loại hình quan hệ:

Có 3 loại hình quan hệ là:

- Một – một
- Một – nhiều
- Nhiều – nhiều

Đặt trường hợp 1 trường đại học. Đối với mỗi KHOA (như khoa KH xã hội), chỉ có 1 TRƯỞNG KHOA. Đây là ví dụ của quan hệ một – một.

KHOA

ĐIỀU HÀNH BỞI

TRƯỞNG KHOA

*Quan hệ một – một*

Một SINH VIÊN có thể CHUYÊN MÔN về 1 môn học, nhưng nhiều SINH VIÊN cũng có thể đăng ký cho 1 CHUYÊN MÔN. Đây là ví dụ cho quan hệ nhiều – một.

SINH VIÊN m ĐĂNG KÝ 1 CHUYÊN MÔN

*Quan hệ nhiều – một*

1 SINH VIÊN có thể đăng ký nhiều MÔN HỌC và nhiều SINH VIÊN có thể cùng đăng ký 1 MÔN HỌC. Đây là ví dụ của quan hệ nhiều – nhiều.

SINH VIÊN m THAM GIA m MÔN HỌC

*Quan hệ nhiều – nhiều*

Các loại hình quan hệ giữa 2 thực thể được biểu diễn bằng lược đồ E/R với những ký hiệu riêng. Một thực thể có thể kết hợp với 1 hoặc nhiều thực thể khác.

## Các thuộc tính

*Thuộc tính* là đặc điểm của một thực thể. Ví dụ, SỐ HỒ SƠ là 1 đặc điểm của thực thể SINH VIÊN. 1 thể hiện thuộc tính là 1 đặc điểm cụ thể của một thể hiện thực thể. Ví dụ, Peter là 1 thể hiện của thực thể SINH VIÊN và hồ sơ số 101 là thể hiện của thuộc tính SỐ HỒ SƠ, đồng thời là đặc điểm của thể hiện thực thể Peter. 1 kiểu thuộc tính là đặc điểm của kiểu thực thể. Ví dụ, “nam giới” là 1 thể hiện thuộc tính, trong khi “giới tính” là 1 kiểu thực thể. Thuộc tính được ký hiệu bằng hình elip, đánh dấu bằng tên đặc điểm. Các khóa được gạch chân. 1 khóa duy xác định thể hiện thực thể một cách duy nhất. Trong ví dụ dưới đây, 1 sinh viên dễ phân biệt bởi số hồ sơ hơn bằng tên vì có nhiều sinh viên trùng tên, còn số hồ sơ thì không bị trùng lặp. Quan hệ cũng có thuộc tính.

ĐỊA CHỈ

TÊN

SINH VIÊN

SỐ HỒ SƠ

ĐĂNG KÝ

*Thuộc tính*

## Tập thực thể con và tập thực thể cha

1 *tập thực thể con* là tập con của một thực thể khác. Ví dụ, thực thể MÔN HỌC gồm 2 nhóm – nhóm môn học theo kỳ và nhóm môn học tự chọn. Trong biểu đồ sau, MÔN HỌC là tập thực thể cha, còn MÔN THEO KỲ và MÔN TỰ CHỌN là tập thực thể con.

Một vài thuộc tính rất phổ biến đối với cả hai tập thực thể con, ví dụ như, “tên” và “nội dung”. MÔN TỰ CHỌN có vài thuộc tính như “mã môn học” không thuộc về tập thực thể con MÔN THEO KỲ. tập MÔN THEO KỲ có vài thuộc tính như “số thứ tự học kỳ” không thuộc về tập MÔN TỰ CHỌN.

	MÃ SỐ	
	MÔN HỌC	NỘI DUNG
		TÊN
MÔN TỰ CHỌN		MÔN THEO KỲ
THỜI HẠN		SỐ THỨ TỰ KỲ

*Tập thực thể con và tập thực thể cha*

Tập thực thể con luôn luôn phụ thuộc vào tập thực thể cha. Thuộc tính của tập thực thể cha là đúng đối với mọi tập thực thể con. Điều ngược lại là sai.

Tập thực thể con liên kết với tập thực thể cha trên mỗi quan hệ vô danh. Tập thực thể cha liên kết với các mối quan hệ bằng 1 đường và một thanh ngang, như trên biểu đồ. Tập thực thể cha được mô tả bởi thuộc tính của tất cả các tập con. Tập con được mô tả bằng thuộc tính của riêng nó.

## Mô hình logic dựa trên mẫu tin

Sau khi làm quen với mô hình E/R, điển hình của mô hình logic dựa trên đối tượng, chúng ta sẽ xét tiếp đến các mô hình dữ liệu khác.

Có 3 loại mô hình dựa trên mẫu tin:

- Mô hình phân cấp
- Mô hình mạng
- Mô hình quan hệ

Trong mô hình phân cấp, dữ liệu được diễn giải theo dạng cây. Dữ liệu trong mô hình phân cấp biểu hiện bởi tập hợp các mẫu tin và các mối quan hệ giữa các dữ liệu được thể hiện bằng các đường dẫn.

Một mô hình mạng tương tự với mô hình phân cấp ở chỗ dữ liệu và quan hệ giữa chúng được mô tả dưới dạng mẫu tin và đường dẫn. Tuy nhiên, các mẫu tin trong CSDL được mô tả bằng lược đồ.

Trong mô hình quan hệ, CSDL được xây dựng như những bản ghi có khuôn mẫu cố định. Mỗi loại bản ghi có một số thuộc tính và trường cố định, thường có độ dài cố định.

Trong 3 mô hình này, mô hình quan hệ là phổ biến nhất.

## Mô hình quan hệ

Tiến sĩ E.F. Codd bước đầu mô tả mô hình quan hệ vào năm 1970. Mô hình quan hệ là sự nỗ lực nhằm đơn giản hóa cấu trúc CSDL. Nó biểu diễn mọi dữ liệu dưới dạng bảng hàng – cột đơn giản.

### *Bảng danh sách giảng viên*

Hệ quản trị CSDL quan hệ là Hệ quản trị CSDL trong đó mọi dữ liệu hiển thị cho người sử dụng được sắp xếp chặt chẽ trong bảng số liệu, mọi thao tác dữ liệu đều được thực hiện trên các bảng này.

## Cấu trúc dữ liệu quan hệ

Nguyên tắc tổ chức của CSDL quan hệ là bảng, hệ thống hàng – cột dạng chữ nhật. Mỗi bảng trong CSDL là một tên bảng duy nhất quy định nội dung của nó. Bảng được gọi là *hệ thức*.

Hàng (bản ghi) trong bảng được gọi là *bộ*, còn cột (trường) được gọi là *thuộc tính*. Các cột trong bảng phải có tên riêng. Số các bộ được gọi là *bản số*, số các thuộc tính gọi là *bậc* của bảng.



Một đặc tính quan trọng nữa của một quan hệ là các hàng không được sắp xếp theo thứ tự. Một hàng không được nhận biết bằng vị trí của nó trong bảng. Tất cả các bảng phải có một vài cột hoặc tổ hợp các cột để nhận diện riêng biệt mỗi hàng trong bảng. Tại một thời điểm nhất định, không có hai hàng nào trong bảng chứa cùng giá trị trong 1 cột hoặc tổ hợp cột đó. Cột này được gọi là *khóa chính* của bảng. Ví dụ, trong bảng sinh viên, SỐ HỒ SƠ là khóa chính của phân biệt duy nhất mỗi sinh viên. Trong một bảng mỗi hàng khác với tất cả các hàng khác được gọi là hệ thức như trong toán học.

Một *miền* là một vùng chứa để các thuộc tính lấy giá trị cụ thể. Ví dụ, miền cho thuộc tính MÃ SỐ là tập hợp các mã số của nhà cung cấp hợp pháp. Một tập hợp giá trị xuất hiện trong một cột tại một thời điểm nhất định là một tập con của miền đó.

Bảng NHÀ CUNG CẤP

*Cấu trúc dữ liệu của một Hệ Quản trị CSDL quan hệ*

## Biểu diễn các thông tin khuyết thiếu

Vấn đề khuyết thiếu thông tin xảy ra rất phổ biến trong thực tế. Ví dụ, 1 người không biết ngày thảo luận khi chưa được công bố. Trong một Hệ Quản trị CSDL hệ thức, thông tin khuyết thiếu được biểu diễn bởi giá trị NULL trong bảng giá trị. Nếu 1 hàng có giá trị NULL trong cột tức là 1 thuộc tính cụ thể của hàng đó khuyết thiếu. NULL không đồng nghĩa với với trống hoặc 0.

## Biểu diễn các mối quan hệ trong một Hệ Quản trị CSDL quan hệ

Điểm khác nhau cơ bản trong mô hình quan hệ và các mô hình dữ liệu đề cập trước là các chỉ dẫn rõ ràng như quan hệ cha – con trong mô hình phân cấp bị bỏ trong mô hình quan hệ. Tuy nhiên, các quan hệ này vẫn tồn tại trong mô hình quan hệ. Mặc dù vậy, các mối quan hệ này được biểu diễn bằng các giá trị dữ liệu phổ biến lưu trữ trong hai hay nhiều bảng. Mọi quan hệ được biểu diễn theo cách thức này.

1 cột trong 1 bảng với các giá trị gắn với khóa chính trong các bảng khác được gọi là *khóa phụ*. 1 khóa chính và một khóa phụ tạo ra mối liên kết cha – con trong các bảng chứa chúng.

### *Quan hệ cha – con trong mô hình quan hệ*

Cũng như tổ hợp các cột được dùng như khóa chính trong bảng, khóa phụ cũng được coi như 1 tổ hợp các cột. Số của cột và kiểu dữ liệu của khóa nhất thiết phải giống nhau. 1 bảng có thể chứa hơn 1 khóa phụ nếu liên kết với hơn 1 bảng.

Bảng GIẢNG VIÊN

## Bảng PHỐI HỢP

### *Quan hệ cha – con trong mô hình quan hệ*

Trong bảng GIẢNG VIÊN, mã số giảng viên (I\_CODE) là khóa chính, trong bảng phối hợp, I\_CODE là khóa phụ. Chỉ có mã số giảng viên trong bảng giảng viên là có thể truy nhập vào bảng phối hợp.

## Đại số quan hệ

Mô hình quan hệ dựa trên nguyên lý của đại số quan hệ. Đại số quan hệ là tập hợp các toán tử thực hiện trên quan hệ. Mỗi toán tử sử dụng một hoặc hai quan hệ làm input (đầu vào) để tạo ra các quan hệ mới, gọi là output (đầu ra)

### Các toán tử sử dụng trong đại số quan hệ

8 toán tử quan hệ được định danh. Những biểu diễn tượng trưng của liệt kê tóm tắt của chúng được thu gọn trong các phần sau:

#### **PHÉP CHỌN**

Phép chọn tách các bộ dữ liệu cụ thể hay hàng ra khỏi một quan hệ cho trước dựa trên 1 điều kiện nào đó.

*Phép chọn tách các bộ dữ liệu hay hàng*

Ví dụ, 1 bảng sinh viên có thuộc tính ROLLNO (số hồ sơ), STUDNAME (tên sinh viên), AGE (tuổi) và GENDER (giới tính). Điều kiện để tách dữ liệu là những sinh viên tuổi trên 25. Kết quả như sau:

*Ví dụ của phép chọn*

## **PHÉP CHIẾU**

Phép chiếu tách dữ liệu những thuộc tính hoặc cột nhất định khỏi quan hệ cho trước.

*Phép chiếu tách thuộc tính hoặc cột*

Ví dụ, nếu chỉ tách tên và tuổi của sinh viên, kết quả sẽ như sau (coi như bảng sinh viên chỉ có 6 bộ dữ liệu)

Ví dụ của phép chiếu

## **PHÉP NHÂN**

Phép nhân xây dựng quan hệ mới dựa trên hai quan hệ nhất định. Nó bao gồm tất cả các tổ hợp có thể của các bộ dữ liệu, mỗi bộ từ một trong hai quan hệ.

*Ví dụ của phép nhân*

Cho tương xứng, hai bảng cần có những thuộc tính chung

Phép nhân tạo ra tích Cartesian giữa hai bảng. Ví dụ, theo dõi hai bảng sau:

Bảng GIẢNG VIÊN

Bảng PHỐI HỢP

Tích Cartesian của các bảng này sẽ là tất cả những tổ hợp khả năng của các bộ dữ liệu.

Tích Cartesian của bảng giảng viên và bảng phối hợp

## **PHÉP HỢP**

Phép hợp xây dựng quan hệ mới từ các bộ xuất hiện trong 1 trong 2 quan hệ cụ thể.

### *Phép hợp*

Để phép hợp tương xứng, hai bảng cần có chung tập thuộc tính (tập hợp các cột có cùng kiểu dữ liệu).

Xem xét hai bảng A và B. A gồm các số hồ sơ và tên của tất cả các sinh viên học chuyên ngành Toán. Các bảng này tương thích vì có thể áp dụng được phép hợp.

A

B

A – B

*Ví dụ của phép hợp*

## **PHÉP GIAO**

Phép giao xây dựng quan hệ gồm các bộ dữ liệu cùng xuất hiện trong 2 quan hệ.

*Phép giao*

Xem xét 2 bảng A và B. Nancy đang theo học cùng lúc 2 ngành. Vì thế, tên của cô ấy xuất hiện trong cả hai bảng. Phép giao của 2 bảng tách 1 hàng chung cho cả hai bảng. PHÉP GIAO có thể áp dụng khi các bảng tương thích nhau trong phép hợp.



*Ví dụ của PHÉP GIAO*

## **PHÉP TRỪ**

Phép trừ xây dựng quan hệ giữa các bộ dữ liệu xuất hiện trong quan hệ thứ nhất mà không xuất hiện trong quan hệ thứ hai.

### *PHÉP TRỪ*

Phép trừ áp dụng cho các bảng tương thích nhau trong phép hợp. trong trường hợp bảng A và B, toán tử “A TRỪ B” tách tất cả các hàng thuộc về A chứ không thuộc về B.

A-B

*Ví dụ cho PHÉP TRỪ*

## **PHÉP KẾT NỐI**

Phép kết nối xây dựng quan hệ từ hai quan hệ cụ thể. Quan hệ này gồm tất cả các tập hợp có thể của các bộ dữ liệu, mỗi bộ từ 1 quan hệ thỏa mãn điều kiện.

Ví dụ, cho 2 bảng X và Y. Bảng X gồm số hồ sơ của sinh viên và mã môn học họ đăng ký. Bảng Y gồm mã số của các giảng viên và mã môn học họ dạy. Nếu cần tổ hợp của của sinh viên học Vật lý và giảng viên môn Vật lý, chúng ta phải thực hiện phép kết nối. Phép kết nối đòi hỏi phải có các thuộc tính giống nhau.

## *PHÉP KẾT NỐI*

Biểu đồ sau đây minh họa cho phép kết nối giữa bảng X và Y.

X

Y

PHÉP KẾT NỐI

*Ví dụ của PHÉP KẾT NỐI*

## **PHÉP CHIA**

Phép chia sử dụng 2 quan hệ và xây dựng quan hệ mới gồm giá trị của 1 thuộc tính từ 1 quan hệ, gắn với mọi giá trị trong quan hệ kia. PHÉP CHIA ngược lại với PHÉP NHÂN

### *PHÉP CHIA*

Quan sát trường hợp sau:

Nhân công

Thành phố

Kết quả phép chia

Chia

*Ví dụ của PHÉP CHIA*

## Ứng dụng của Hệ Quản trị CSDL quan hệ

Hệ Quản trị CSDL quan hệ được sử dụng trong:

- Đặt vé máy bay hoặc tàu hỏa
- Ứng dụng trong ngành ngân hàng
- Công nghiệp sản xuất
- Xử lý đặt lệnh
- Hệ thống quản lý bệnh viện
- Hệ thống quản lý thư viện
- Hệ thống quản lý khách sạn

Những sản phẩm của Hệ Quản trị CSDL quan hệ phổ biến ngày nay:

- Sybase
- Oracle
- Microsoft SQL Server
- Ingress
- DB2

## Tóm tắt

Trong bài này, học viên nắm được:

- Mô hình quan hệ được chia thành:
  - Mô hình quan hệ dựa trên đối tượng
  - Mô hình quan hệ dựa trên mẫu tin
- Kỹ năng vẽ giản đồ thực thể - quan hệ
  - Thực thể ký hiệu bằng hình chữ nhật
  - Quan hệ ký hiệu bằng hình thoi
  - Thuộc tính ký hiệu bằng hình elip
- Quan hệ bao gồm nhiều-nhiều, một-nhiều, một-một đều được thể hiện bằng biểu trưng.
- Thực thể yếu biểu hiện bằng hình hộp hai đường
- Tập thực thể con nối với tập thực thể cha bằng 1 quan hệ cô danh, đánh dấu bằng thanh ngang trên đầu
- Trong mô hình quan hệ, dữ liệu thể hiện bằng bảng (quan hệ) của các hàng (bộ dữ liệu) và các cột (thuộc tính)
- Số các bộ gọi là bản số của bảng, số các thuộc tính gọi là bậc của bảng.
- 1 thuộc tính (hoặc tập thuộc tính) là duy nhất đối với bộ dữ liệu gọi là khóa chính
- Thông tin khiếm khuyết được thể hiện bằng NULL trong bảng
- Khóa phụ là một cột gắn với khóa chính của bảng khác.
- Mô hình quan hệ dựa trên nguyên lý đại số quan hệ
- 8 toán tử của quan hệ là, phép chọn, phép chiếu, phép hợp, phép giao, phép nhân, phép chia, phép trừ, phép kết nối.

Bài 1C:

# Phối hợp

## Byte kiến thức

Trong phần này, chúng ta sẽ nghiên cứu về:

- Mô hình mạng và mô hình phân cấp
- Quản trị dữ liệu

### Mô hình mạng và mô hình phân cấp

Trong mô hình mạng, dữ liệu thể hiện dưới dạng các bản ghi. Những bản ghi này liên hệ với nhau bằng các đường dẫn. Mỗi bản ghi gồm có các trường và mỗi trường gồm 1 giá trị dữ liệu. Mỗi bản ghi trong mô hình mạng tương tự như 1 thực thể trong mô hình E/R. Mỗi trường là một thuộc tính và mỗi đường dẫn giống như 1 quan hệ. 1 đường dẫn liên kết 2 bản ghi.

Ví dụ, tài khoản của một khách hàng trong CSDL ngân hàng được thể hiện trong mô hình mạng với 2 bản ghi sau:

- KHÁCH HÀNG

■	Tên
■	Địa chỉ
■	Thành phố

- TÀI KHOẢN

■	Số tài khoản
■	Kết toán

Các trường trong mỗi bản ghi KHÁCH HÀNG sẽ là tên, địa chỉ và thành phố. Tương tự, các trường trong mỗi bản ghi TÀI KHOẢN là số tài khoản và kết toán.

Giản đồ cấu trúc dữ liệu được dùng để thiết kế mô hình mạng. Lược đồ cấu trúc dữ liệu tương tự với lược đồ E/R. Nó được dùng để xác định cấu trúc logic của CSDL. Biểu đồ sau biểu diễn bản ghi KHÁCH HÀNG, bản ghi TÀI KHOẢN và mối liên hệ giữa chúng

#### *Giản đồ cấu trúc dữ liệu*

Biểu đồ trên thể hiện rằng khách hàng tên Mike có 568 trong ngân hàng.

Như mô hình mạng, mô hình phân cấp cũng gồm nhiều bản ghi. Mỗi bản ghi gồm các trường, hai bản ghi liên kết bằng đường dẫn. Tuy nhiên, trong mô hình phân cấp, các bản ghi được tổ chức bằng cấu trúc cây. Biểu đồ cấu trúc cây biểu diễn thiết kế của mô hình phân cấp, bao gồm các hộp và các đường

Đọc ví dụ trên với bản ghi KHÁCH HÀNG và TÀI KHOẢN. Biểu đồ sau thể hiện lược đồ cấu trúc cây gồm bản ghi KHÁCH HÀNG và TÀI KHOẢN và liên kết giữa chúng.

### *Lược đồ cấu trúc cây*

Biểu đồ trên biểu diễn khách hàng tên Mike có tài khoản 568, Tim có tài khoản 749 và 114, Mitchell có tài khoản 254.

Thấy rằng các bản ghi được tổ chức dưới dạng cây với 1 rễ. Rễ là một điểm phân nhánh hình thức. CSDL phân cấp là tập hợp các cây với các rễ, tập hợp này tạo thành rừng.

Chỉ có quan hệ một-một và quan hệ một-nhiều có thể tồn tại giữa nút cha và nút con.

## Quản trị CSDL

Nhà quản trị CSDL đóng vai trò như trung gian giữa dữ liệu lưu trong CSDL và truy vấn báo cáo với hệ CSDL. Nhà quản trị CSDL có các nhiệm vụ sau:

- Tương tác với người quản lý tệp tin: Nhà quản trị CSDL dịch ngôn ngữ quản trị CSDL sang các lệnh đối với hệ thống tệp tin cấp thấp. Điều này là cần thiết vì dữ liệu lưu trong ổ đĩa cứng sử dụng hệ thống tệp tin cung cấp bởi hệ điều hành. Vì vậy, nhà quản trị CSDL thao tác với lưu trữ, khôi phục và cập nhật dữ liệu



- Duy trì sự nhất quán: nhiều hạn chế được chỉ rõ cho toàn bộ CSDL để đảm bảo giá trị dữ liệu trong CSDL là hợp lệ. Ví dụ, trong CSDL của 1 trường đại học, sinh viên có thể đăng ký tối đa là 4 môn. Nếu số môn vượt quá 4, CSDL sẽ báo lỗi. Hạn chế này được định rõ bởi Nhà quản trị CSDL. Trách nhiệm của người quản lý CSDL là kiểm tra những hạn chế này có bị vi phạm hay không. Nếu có, người quản lý CSDL cần có những biện pháp thích hợp.
- Duy trì bảo mật: giới hạn bảo mật như hạn chế truy cập vào những vùng khác nhau của CSDL được chỉ rõ bởi nhà quản trị CSDL. Người quản lý CSDL cần đảm bảo tuân thủ những giới hạn này.
- Quy trình thi hành dự phòng và khôi phục: hệ CSDL có thể bị lỗi bất cứ khi nào do lỗi nguồn, lỗi phần mềm hoặc sự cố đĩa. Người quản lý CSDL chịu trách nhiệm tìm ra các lỗi và phục hồi CSDL tại nơi bị lỗi. Để làm được điều này, người quản lý CSDL cần tiến hành dự phòng và khôi phục.
- Kiểm tra đồng thời: CSDL thường được sử dụng bởi nhiều người cùng lúc. Điều này có thể gây ra việc dữ liệu thiếu nhất quán khi tất cả người sử dụng cùng cập nhật dữ liệu một lúc. Người quản lý CSDL cần tránh tình trạng này bằng cách kiểm soát sự tương tác giữa những người sử dụng.

## Lời của chuyên gia

Trong phần này, chúng ta học về cách luyện tập tối ưu để biểu thị các mối quan hệ trong các mối quan hệ, kỹ năng vẽ lược đồ E/R chứa hệ thức nhị phân, và những câu hỏi thường gặp về hệ CSDL và mô hình E/R.

## Luyện tập

### Biểu thị các mối quan hệ trong các mối quan hệ

Mô hình E/R có 1 nhược điểm là chúng ta không thể biểu thị các mối quan hệ trong các mối qua hệ. Ví dụ, với 1 CSDL chứa thông tin về nhân viên. Mỗi nhân viên làm một phần việc cụ thể sử dụng máy móc khác nhau. Lược đồ E/R trong trường hợp này như sau

*Lược đồ E/R mô tả các mối quan hệ trong các mối quan hệ*

Quan hệ giữa WORKS (công việc) và USE (sử dụng) có thể liên kết nhưng phải thay đổi cấu trúc logic của CSDL. Lược đồ E/R trước đó có thể thay đổi được nhờ phép gộp. Phép gộp có thể xem như sự rút gọn trong đó ta sử dụng mỗi quan hệ như thực thể bậc cao hơn. Vì thế, trong ví dụ trước, chúng ta có thể coi thực thể EMPLOYEE (nhân công) và PROJECT (dự án) cùng với thực thể WORKS như một thực thể bậc cao hơn. Những thực thể này cũng giống như các thực thể khác. Biểu đồ sau vẽ lược đồ E/R sử dụng phép gộp.

*Lược đồ E/R với phép gộp*

## Thủ thuật

### Sử dụng quan hệ tam phân

Chúng ta có thể đơn giản lược đồ E/R có chứa quan hệ nhị phân bằng cách sử dụng lược đồ E/R có chứa quan hệ tam phân. Quan hệ tam phân là mối quan hệ giữa 3 thực thể. Ví dụ, ngân hàng New Edge giữ CSDL về tất cả các khách hàng. Quản trị CSDL, Mary Peterson, muốn dựng lược đồ E/R cho trường hợp sau:

1 khách hàng có thể có nhiều tài khoản. các tài khoản này có thể ở những chi nhánh khác nhau của ngân hàng.

Tình huống có thể được mô tả bởi sử dụng 2 quan hệ nhị phân như trong lược đồ sau:

*Lược đồ E/R sử dụng quan hệ nhị phân*

Lược đồ E/R trên có thể chuyển thành lược đồ E/R sau sử dụng quan hệ tam phân

### *Lược đồ E/R sử dụng quan hệ tam phân*

Trong lược đồ này, quan hệ tam phân lấy tên từ chữ cái đầu của một thực thể. Ví dụ, C từ CUSTOMER (khách hàng), A từ ACCOUNT (kế toán), B từ BRANCH (chi nhánh) quan sát thấy lược đồ E/R trên đơn giản hơn so với lược đồ sử dụng quan hệ nhị phân.

## Các câu hỏi thường gặp

- Lược đồ E/R cho 1 doanh nghiệp có thể được thiết kế bằng nhiều cách. Những tiêu chí nào cần được xem xét khi đưa ra quyết định

- Sử dụng quan hệ nhị thức hay tam thức
- Sử dụng thuộc tính hay thực thể
- Sử dụng phép gộp
- Sử dụng thực thể thường hay thực thể yếu
- Sử dụng thực thể hay quan hệ

- Những nhược điểm của việc sử dụng hệ CSDL là gì?

Những nhược điểm của sử dụng hệ CSDL là:

- **Lỗi hệ thống:** Nếu máy tính lưu trữ CSDL bị hỏng, tất cả người sử dụng CSDL sẽ phải chờ cho đến khi máy tính hoạt động bình thường trở lại. Thêm nữa, nếu phần mềm ứng dụng có ảnh hưởng tới CSDL bị lỗi, CSDL có thể bị phá hủy lâu dài. Vì vậy, khi hệ CSDL được sử dụng trong một tổ chức, tất cả các phần mềm tác động trực tiếp lên CSDL cần được xác định rõ. Tiếp đó, cần thường xuyên bảo dưỡng để đảm bảo tất cả các phần mềm không có sai sót. Một tổ chức không sử dụng hệ CSDL thì không gặp phải các sự cố hệ thống vì các dữ liệu và phần mềm bị phân tán.
- **Xung đột trong tổ chức:** Sử dụng CSDL chung có thể dẫn đến xung đột trong tổ chức. Một số người sử dụng không muốn mất quyền kiểm soát đối với dữ liệu. Chia sẻ dữ liệu còn gây ra các vấn đề khác. Ví dụ, một nhóm người sử dụng có thể gây hại cho dữ liệu của nhóm khác.
- **Sự cố của kế hoạch phát triển:** một dự án phát triển phần mềm trong 1 tổ chức có thể gặp sự cố vì một số lý do. Ví dụ, lợi ích của hệ CSDL không thuyết phục được Ban quản trị hoặc thời gian hoàn thiện của dự án dài hơn cho phép.
- **Yêu cầu nguồn nhân lực trình độ cao:** việc đưa hệ CSDL vào hoạt động cần nhân viên trình độ cao để có thể kiểm soát CSDL. Chi phí chung cho nhân công sẽ bị đẩy cao vì chi phí cho ban quản trị CSDL là khoản chi phí lâu dài.
- **Chi phí chung :** Phí tổn cho việc sử dụng CSDL là rất lớn. nhu cầu về CSDL trong tổ chức đòi hỏi phải đầu tư cho cả phần mềm và phần cứng. Thêm nữa, chi phí vận hành cũng sẽ gia tăng. Ví dụ, công việc liên quan đến hệ quản trị CSDL diễn ra chậm hơn so với công việc không cần Hệ quản trị CSDL.

- Vai trò là gì và chúng được biểu diễn thế nào trong lược đồ E/R?

Vai trò là chức năng một thực thể thực hiện trong một mối quan hệ. Vai trò thường không được chỉ ra trên lược đồ E/R. Tuy nhiên, chúng cần được chỉ rõ khi cần làm sáng tỏ các quan hệ. Chúng ta có thể chỉ dẫn các vai trò trên lược đồ E/R bằng cách ghi tên trên các đường dẫn nối các thực thể và quan hệ. Ví dụ, trong lược đồ E/R dưới đây TEAM LEADER (trưởng nhóm) và TEAM MEMBER (thành viên nhóm) là những chỉ dẫn vai trò giữa thực thể EMPLOYEE (nhân viên) và quan hệ WORK-FOR (làm việc cho).

### *Lược đồ E/R chỉ dẫn vai trò*

- Vì sao mô hình quan hệ lại phổ biến hơn mô hình mạng lưới và mô hình phân cấp?

Mô hình quan hệ không sử dụng con trỏ hoặc đường dẫn như mô hình mạng lưới và mô hình phân cấp. Thay vào đó, mô hình quan hệ nhận dạng các bản ghi sử dụng giá trị lưu trong đó. Điều đó cho phép sử dụng các toán tử trong mô hình quan hệ.

- Những nhiệm vụ chính của nhà quản trị CSDL là gì?

Nhiệm vụ chính của nhà quản trị CSDL là:

- Định nghĩa sơ đồ: Nhà quản trị CSDL tạo ra các sơ đồ CSDL bằng cách viết định nghĩa cho cấu trúc logic của CSDL. Những định nghĩa này sau đó sẽ được chuyển thành các bảng trong CSDL.
- Định nghĩa cấu trúc bộ nhớ và phương pháp truy nhập: nhà quản trị CSDL tạo ra cấu trúc bộ nhớ và phương pháp truy nhập bằng cách viết định nghĩa cho chúng. Những định nghĩa này sau đó được chuyển thành cấu trúc bộ nhớ và phương pháp truy nhập.
- Chỉnh sửa sơ đồ và tổ chức vật lý: Nhà quản trị CSDL chỉnh sửa sơ đồ dữ liệu và bộ nhớ vật lý khi cần thiết. Việc này được thực hiện nhờ viết các định nghĩa sử dụng bởi trình biên dịch ngôn ngữ định nghĩa dữ liệu hay bộ nhớ dữ liệu và trình biên dịch ngôn ngữ định nghĩa để thay đổi.
- Cấp quyền truy nhập dữ liệu: Nhà quản trị CSDL cần chỉ rõ quyền hạn sử dụng cho mỗi người sử dụng. Điều đó giúp kiểm soát sự sử dụng CSDL, đồng thời hạn chế việc truy cập vào một số vùng của CSDL.
- Chỉ rõ những ràng buộc về tính nhất quán: cần chỉ rõ sự ràng buộc về nhất quán đối với CSDL. Những ràng buộc này được lưu trong 1 cấu trúc hệ thống đặc biệt.



## Luyện tập

1. Người sử dụng hệ quản trị CSDL nào tham gia thiết kế và bảo trì CSDL và bảo mật CSDL?
2. Mức cao nhất trong cấu trúc của 1 hệ quản trị CSDL là mức \_\_\_\_\_.
3. Thành phần nào của hệ quản trị CSDL xử lý thao tác nhập và xuất vật lý với dữ liệu?
4. Một \_\_\_\_\_ là một đặc điểm của một thực thể cho sẵn.
5. Toán tử quan hệ nào sau đây tách bộ dữ liệu hay hàng cụ thể ra khỏi quan hệ cho trước, dưới 1 điều kiện nào đó?
  - a. Phép chiếu
  - b. Phép chọn
  - c. Phép kết nối
  - d. Phép chia
6. \_\_\_\_\_ là sự rút gọn trong đó ta sử dụng mối quan hệ như thực thể bậc cao hơn.
7. Một khách hàng có thể có nhiều tài khoản trong một ngân hàng nhưng một tài khoản chỉ có thể dùng cho 1 khách hàng. Đây là quan hệ \_\_\_\_\_.
8. Đúng hay sai?

Tập thực thể con tồn tại ngay cả khi độc lập với tập thực thể cha.

9. Nối toán tử quan hệ với đúng mô tả:

Toán tử quan hệ	Mô tả
1. Phép chiếu	A. Tách các thuộc tính hay hàng cụ thể khỏi quan hệ cho trước
2. Phép chia	B. Sử dụng hai quan hệ để tạo quan hệ mới chứa giá trị của 1 quan hệ mà gắn với giá trị của quan hệ kia.
3. Phép kết nối	C. Dùng hai quan hệ để tạo quan hệ mới chứa tất cả các kết hợp có thể của các bộ, mỗi bộ từ một trong hai quan hệ.
4. Phép nhân	D. Sử dụng hai quan hệ để tạo quan hệ mới chứa tất cả các kết hợp có thể của các bộ, mỗi bộ từ 1 quan hệ sao cho thỏa mãn điều kiện cụ thể.

10. Trong mô hình nào dữ liệu được biểu diễn dưới dạng cây?

11. Vẽ lược đồ E/R cho trường hợp sau:

Một tổ chức có hai loại nhân viên, nhận lương hoặc nhận tiền công. Cả hai loại đều có những đặc tính chung là mã nhân viên, tên và địa chỉ. Tuy nhiên, nhân viên nhận lương có những đặc tính kèm theo là lương cơ bản, phụ cấp và trợ cấp y tế. Nhân viên làm công óc những đặc tính riêng là tiền công theo ngày và tiền công làm ngoài giờ.

**Bài 1D:**

# Thực hành

## Bài tập

### Bài 1:

New Era là một công ty chế tạo máy có hai nguồn cung ứng. Nhà cung ứng thứ nhất đồng ý thanh toán tín dụng, nhà cung ứng thứ hai chỉ cho phép chi trả bằng tiền mặt trước khi giao hàng. Công ty muốn lưu trữ thông tin về hai nhà cung ứng một cách riêng rẽ. Đối với nhà cung ứng thanh toán qua tín dụng, “credit period” (thời hạn tín dụng) và “credit limit” (mức tín dụng) cần được ghi lại. Đối với nhà cung ứng thanh toán tiền mặt, “date of payment” (ngày thanh toán) cần được ghi lại.

John là quản trị viên CSDL tại New Era. Anh cần vẽ lược đồ E/R thể hiện tình huống trên.

### Bài 2:

Ron Floyd là quản trị viên CSDL mới vào làm ở ngân hàng Standard. Ban quản trị của ngân hàng muốn vi tính hóa các hoạt động ngân hàng. Theo chính sách của ngân hàng, một khách hàng có thể có nhiều tài khoản nhưng một tài khoản không thể chung cho nhiều khách hàng. Bước đầu thiết kế CSDL, Ron được yêu cầu tạo lược đồ E/R thể hiện quan hệ giữa khách hàng và tài khoản của họ.

### Bài 3:

Red Sky Systems là một công ty IT mới thành lập. Công ty chuyên thực hiện các dự án phần mềm cho khách hàng. Công ty duy trì 1 CSDL gồm bản ghi về các nhân viên và dự án. Mọi thông tin liên quan đến nhân viên lưu trong 1 bảng có tên là Employee (nhân viên). Thông tin về dự án lưu trong bảng có tên Project (dự án). Một bảng khác có tên là Assignment (phân công) lưu trữ thông tin về dự án nào được giao cho nhân viên nào. Ba bảng đó như sau:

EMPLOYEE

## PROJECT

## ASSIGNMENT

Chris là quản trị viên CSDL tại Red Sky Systems. Anh muốn chỉ ra các điều sau:

1. Tất cả chi tiết về nhân viên trong bộ phận tài chính
2. Tên và tuổi của nhân viên.
3. Tên dự án, tên nhân viên và bộ phận làm việc của tất cả các nhân viên
4. Tên của các nhân viên dưới 25 tuổi và có mã nhân viên lớn hơn 100.

Chỉ ra cách để thực hiện các nhiệm vụ trên và kết quả của chúng.

## Bài 4:

Jack là chủ một cửa hàng bách hóa. Anh có những khách hàng thường xuyên mua rau quả, đồng thời cũng có những người cung cấp rau quả thường xuyên. Anh đặt mã khách hàng và mã cung cấp cho những khách hàng và nhà cung cấp thường xuyên của mình để đơn giản hóa việc kinh doanh. Jack tạo một bảng để lưu các bản ghi về khách hàng thường xuyên và mặt hàng họ mua. Anh cũng lập một bảng khác để lưu các bản ghi về những nguồn cung ứng thường xuyên và sản phẩm mà họ cung cấp. Jack cần tạo 1 bảng thứ 3, Customer-Supplier (khách hàng-nhà cung cấp) biểu diễn các bản ghi về khách hàng thường xuyên và sản phẩm họ mua cùng với nguồn cung cấp mặt hàng đó. Chỉ ra cách để Jack có thể tạo bảng thứ 3 và kết quả của nó.

Chú ý: Sử dụng bản ghi Customer (khách hàng) và bản ghi Supplier (Nhà cung cấp) để giải bài tập.

Bản ghi Customer

Bản ghi Supplier

## Bài tập về nhà

1. Khái niệm nào sau đây định nghĩa liên kết giữa các thực thể?
  - a. Lược đồ E/R
  - b. Tập thực thể cha
  - c. Quan hệ
  - d. Thuộc tính
2. Khái niệm nào dưới đây thể hiện đặc tính của 1 thực thể?
  - a. Tập thực thể con
  - b. Thuộc tính
  - c. Lược đồ E/R
  - d. Tập thực thể cha
3. Kiểu quan hệ nào sau đây được thể hiện nếu 1 sinh viên đăng ký 2 môn học cùng 1 lúc?
  - a. Một – một
  - b. Nhiều – một
  - c. Một – nhiều
  - d. Nhiều – nhiều
4. Trong mô hình nào dưới đây dữ liệu được biểu diễn dưới dạng cây?
  - a. Mô hình phân cấp
  - b. Mô hình quan hệ
  - c. Mô hình thực thể - quan hệ
  - d. Mô hình mạng lưới
5. Toán tử quan hệ nào được dùng để tách một cột ra khỏi quan hệ cho trước?
  - a. Phép chọn
  - b. Phép nhân
  - c. Phép chia
  - d. Phép chiếu

6. Ai là người thực hiện nhiệm vụ thay thế khối dữ liệu
  - a. Quản trị viên CSDL
  - b. Người quản lý file
  - c. Người quản lý ổ đĩa
  - d. Người quản lý CSDL
7. Giai đoạn cuối cùng của chu kỳ phát triển CSDL?
  - a. Xác định yêu cầu
  - b. Đánh giá và bảo trì CSDL
  - c. Thiết kế khái niệm
  - d. Thực thi
8. Nhược điểm của phương pháp truyền thống trong xử lý dữ liệu?
  - a. Dữ liệu nhất quán
  - b. Giữ vững tiêu chuẩn
  - c. Những hạn chế trong bảo mật
  - d. Gấp đôi số lượng dữ liệu cần thiết
9. Thực thể phụ thuộc là:
  - a. Thực thể yếu
  - b. Thực thể thường
  - c. Thực thể con
  - d. Thực thể cha
10. khái niệm nào sau đây thể hiện vùng chứa của dữ liệu từ đó 1 hoặc nhiều thuộc tính lấy giá trị thực của nó?
  - a. Quan hệ
  - b. Thực thể
  - c. Miền
  - d. Bảng



## Bài 2A:

# Thiết kế CSDL logic

Trong bài này, chúng ta nghiên cứu:

- Vẽ lược đồ E/R cho các bản trong quan hệ với các chi tiết:

Thực thể thường  
Thuộc tính  
Quan hệ  
Thực thể yếu  
Thực thể con và thực thể cha

- Phân biệt các khóa
- Định nghĩa Chuyên biệt hóa và Tổng quát hóa

# Mô hình khái niệm

Mô hình khái niệm phản ánh các thực thể và quan hệ của chúng dựa trên những yêu cầu về xử lý dữ liệu của tổ chức. Để phát triển một CSDL thỏa mãn nhu cầu thông tin cả trong hiện tại lẫn tương lai, trước hết chúng ta cần thiết kế một mô hình khái niệm cho CSDL.

Thiết kế mô hình khái niệm không liên quan tới giai đoạn thực thi và vận hành của CSDL. Mô hình khái niệm có thể ánh xạ tới mô hình quan hệ, phân cấp hay mạng lưới. Nó độc lập với từng ứng dụng đơn lẻ, hệ quản trị CSDL, phần cứng và bộ nhớ vật lý của dữ liệu.

Phân tích dữ liệu là bước đầu tiên trong thiết kế mô hình khái niệm, bắt đầu với thu thập thông tin về dữ liệu. Bước này thường bao gồm một bản điều tra hoặc một phương pháp tương tự để có được một danh sách dữ liệu mà tổ chức cần. Các biểu mẫu, hóa đơn, báo cáo là những điểm khởi đầu để thu thập dữ liệu. Bước tiếp theo là kiểm tra sự vận hành và xử lý dữ liệu và loại bỏ các dữ liệu trùng lặp hoặc không hợp lý.

Phân tích dữ liệu bao gồm nhận biết các thực thể, thuộc tính của chúng và mối quan hệ giữa các thực thể dựa trên dữ liệu thu thập được.

Sau khi hoàn thành phân tích dữ liệu, chúng ta vẽ lược đồ thực thể - quan hệ. Lược đồ tổng quan về thiết kế, dễ dàng truyền đạt ý tưởng cho người sử dụng. Ánh xạ của vật thực cho các biểu tượng của thực thể - quan hệ là chủ quan. Thông thường, những đối tượng ban đầu là thuộc tính, sau khi duyệt lại thiết kế sẽ trở thành thực thể. Vì vậy, không có quy luật rõ ràng để nhận biết một đối tượng là thực thể, thuộc tính hay quan hệ.

## Ánh xạ lược đồ thực thể - quan hệ tới bảng

CSDL phù hợp với lược đồ E/R có thể được biểu diễn bởi tập hợp các bảng trong hệ thống quan hệ. Hãy xem xét sự ánh xạ của lược đồ thực thể - quan hệ tới bảng trong quan hệ với:

- Thực thể thường
- Thuộc tính
- Quan hệ
- Thực thể yếu
- Thực thể cha và thực thể con

## Thực thể thường

Cần nhắc lại rằng thực thể thường không phụ thuộc. Chúng có thể tồn tại độc lập với các thực thể khác. Chúng là một “khối hợp nhất” của CSDL. Mỗi thực thể thường ánh xạ tới 1 bảng. Ví dụ, quan sát lược đồ E/R dưới đây:

### *Thực thể thường*

Thực thể thường STUDENT (sinh viên) và BOOKS (sách) ánh xạ tới 2 bảng riêng biệt.

## Thuộc tính

Mỗi đặc tính hoặc thuộc tính thể hiện trong lược đồ E/R ánh xạ tới một thuộc tính của bảng tương ứng. Đặc tính hoặc thuộc tính của STUDENT và BOOKS ánh xạ tới thuộc tính của bảng thích hợp.

### *Thuộc tính*

Khóa chính của bảng được xác định là khóa của lược đồ E/R, `ROLL_NO` (số hồ sơ) và `CODE` (mã số). Ghi nhớ rằng khóa phải xác định được duy nhất mỗi bộ dữ liệu trong bảng.

## Quan hệ

Ảnh xạ của quan hệ dựa vào kiểu quan hệ, được nói đến trong phần trước. Mỗi kiểu dữ liệu ảnh xạ tới các bảng theo một cách thức khác nhau trong hệ quản trị CSDL hệ thống.

Nguyên lý quan trọng nhất là tạo các bảng sao cho thông tin từ thế giới thực được lưu trữ và phục hồi theo cách tối ưu; theo đó, số lượng bảng ít nhất, số lượng thuộc tính ít nhất. trong hệ thống quan hệ, phép kết nối phục hồi tất cả thông tin bằng cách kết hợp hai hay nhiều bảng.

## Quan hệ một – một

Trong quan hệ một – một, mỗi thể hiện thực thể có thể liên kết với chỉ 1 thể hiện của thực thể liên quan.

Ví dụ, cho rằng một sinh viên chỉ làm một đề tài và không sinh viên nào khác làm cùng đề tài. Như vậy, đây là mối quan hệ một – một giữa sinh viên và đề tài.

Chúng ta biểu diễn quan hệ giữa hai thực thể **Student** (Sinh viên) và **Project** (đề tài) dưới dạng lược đồ E/R như sau:

### *Quan hệ một - một*

Kiểu quan hệ này có thể phân tích bằng nhiều cách. Một cách là phân tích như trong quan hệ một – nhiều bằng cách giữ khóa phụ trong một trong hai bảng. cách khác là hợp nhất hai bảng thành một bảng để truy cập nhanh hơn. Ví dụ, nếu truy vấn yêu cầu dữ liệu từ hai bảng **Student** và **Project**, cách tốt nhất là hợp nhất hai bảng để cải thiện hoạt động truy vấn.

## Quan hệ một – nhiều

Trong quan hệ một – nhiều, một thể hiện của một thực thể có thể liên kết hơn 1 thể hiện của thực thể liên quan.

Ví dụ, một phòng có nhiều hơn một nhân viên.

Chúng ta biểu diễn quan hệ giữa nhân viên và phòng theo dạng lược đồ E/R như sau:

*Ánh xạ quan hệ một – nhiều*

**Dept\_ID** (ký hiệu phòng) của một nhân viên có thể lưu trữ trong bảng **Employee** như trên. Ví thể, **Dept\_ID** trong bảng **Employee** là khóa phụ trong bảng **Employee**.

**Quan hệ nhiều – nhiều**

Quan sát lược đồ sau:

*Quan hệ nhiều – nhiều*

Một sinh viên có thể mượn nhiều quyển sách, một quyển sách có thể mượn bởi nhiều sinh viên. Quan hệ ISSUES (cho mượn) liên quan đến cả sinh viên và sách. ISSUES cũng có nhiều thuộc tính riêng biệt, ví dụ, ngày mượn sách và ngày trả sách. Vì thế, chúng ta cần hiểu rằng thuộc tính không chỉ thuộc về duy nhất một thực thể. Chúng còn thuộc về mối quan hệ. mỗi quan hệ ISSUES là quan hệ nhiều – nhiều.

Bảng **ISSUE** phải bao gồm khóa chính của cả hai bảng **STUDENT** và **BOOKS** (**ROLL\_NO** và **CODE**). Đây là khóa phụ của bảng **ISSUE**, giúp nối hai bảng **STUDENT** và **BOOKS**.

### *Ảnh xạ quan hệ*

Nhớ lại phép kết nối nói đến trong đại số quan hệ. Nếu một người sử dụng muốn biết cuốn sách nào được mượn bởi sinh viên nào và vào ngày nào, chúng ta có thể sử dụng khóa phụ trong bảng **ISSUE** để kết nối cả ba bảng. Kết quả thu được là bản mô tả hoàn chỉnh của việc cho mượn sách như sau:

### *Bảng kết nối*

Khóa chính của bảng **ISSUE** là gì? Có hai khả năng. Thứ nhất là kết hợp hai khóa phụ (**ROLL\_NO** và **CODE**) nếu sự kết hợp nhận dạng duy nhất được mọi hàng trong bảng. Khóa này được xem như khóa phức hợp, tạo bởi nhiều hơn một thuộc tính. Khả năng thứ hai là tạo một thuộc tính mới cho khóa chính, ví dụ **ISSUENUM**.

Chúng ta có thể biểu diễn kiểu quan hệ giữa hai thực thể trong lược đồ thực thể - quan hệ bằng những ký hiệu tượng trưng. Một thực thể có thể kết hợp với một, nhiều hoặc không kết hợp với thực thể khác.

## Thực thể yếu

Thực thể yếu là thực thể tồn tại phụ thuộc vào thực thể khác. Ví dụ, những phụ thuộc của một nhân viên.

*Thực thể yếu*



Trong ví dụ trên, xhungs ta có thực thể yếu là DEPENDENT (phụ thuộc) phụ thuộc vào thực thể EMPLOYEE (nhân viên). Thực thể DEPENDENT có thể được ánh xạ tới các bảng riêng biệt như sau

*Thực thể yếu ánh xạ tới bảng*

Tập thực thể con và tập thực thể cha

Tập thực thể con là tập con của một thực thể khác. Tập thực thể con luôn phụ thuộc vào tập thực thể cha để tồn tại.

*Tập thực thể con và tập thực thể cha*

Mỗi kiểu thực thể ( thực thể con và thực thể cha) ánh xạ tới một bảng riêng biệt. Ví dụ:

EMPLOYEE (nhân viên)

HOURLY EMPLOYEE

(nhân viên làm theo giờ)

SALARIED EMPLOYEE

(nhân viên làm hưởng lương)

### *Ánh xạ thực thể con và thực thể cha*

Khóa chính của thực thể cha là khóa phụ của thực thể con. Nó tạo ra một đường dẫn giữa hai thực thể. Khóa phụ của thực thể con cùng đồng thời là khóa chính của nó. Thuộc tính được mã hóa nằm trong bảng EMPLOYEE chỉ ra thực thể con. Ví dụ, “S” cho nhân viên làm hưởng lương, “H” cho nhân viên làm theo giờ.

HOURLY EMPLOYEE

SALARIED EMPLOYEE

### *Cấu trúc bảng với thuộc tính được mã hóa*

## Thủ thuật trong thiết kế CSDL logic

### Thuộc tính

Không tạo thêm những thuộc tính không cần thiết. Một thuộc tính phải đáp ứng ba mục đích:

- Nhận dạng thực thể chủ
- Rút ra thực thể khác
- Đơn giản hóa mô tả của một thực thể

SALES (bán hàng)

### *Thuộc tính của một bảng*

Nếu các thực thể có chung nhiều thuộc tính, tiến hành hợp nhất các thực thể. Lưu ý rằng mục tiêu chính khi thiết kế CSDL là tối thiểu số lượng bảng với tối thiểu số lượng thuộc tính.

### *Hợp nhất thực thể có thuộc tính chung*

## Khóa

Một Hệ Quản trị CSDL quan hệ sử dụng định chỉ kết hợp, nhận dạng và xác định vị trí các hàng bằng giá trị. Địa chỉ vật lý là vô hình đối với người sử dụng. Vì thế, hệ thống quan hệ yêu cầu các khóa phải nhận dạng duy nhất được các hàng trong bảng.

Có nhiều loại khóa khác nhau:

- Khóa chính
- Khóa phụ
- Khóa chọn
- Khóa thay đổi
- Khóa phức hợp

Bất cứ thuộc tính nào (hay tập hợp thuộc tính) xác định duy nhất một hàng trong bảng là một lựa chọn cho khóa chính của bảng. Thuộc tính như vậy gọi là *khóa chọn*. Một trong những khóa chọn được dùng làm khóa chính, dựa vào mức độ phổ biến, mức độ sử dụng. Thuộc tính nào là lựa chọn cho khóa chính nhưng không trở thành khóa chính được gọi là *khóa thay đổi*. Khi khóa xác định duy nhất các hàng trong bảng được tạo nên bởi nhiều thuộc tính gọi là *khóa phức hợp*. Khóa phụ luôn luôn biểu diễn quan hệ.

Cần phải hiểu rằng khóa chính là cách duy nhất để xác định các hàng trong bảng. Vì vậy, giá trị NULL không được phép làm khóa chính. Nếu làm như vậy, sẽ rất khó để xác định được duy nhất các hàng.

Quan sát bảng VEHICLE (phương tiện giao thông) sau:

#### *Khóa chọn, khóa chính và khóa thay đổi*

Trong bảng trên, **ENGINE#** và **REGN#** là duy nhất ở mỗi bộ dữ liệu. Vì thế, chúng đều là khóa chọn. Nếu **ENGINE#** được chọn là khóa chính, **REGN#** sẽ là khóa thay đổi.

Khóa có thể đơn giản hoặc phức hợp. Khóa đơn giản tạo nên bởi 1 thuộc tính. Khóa phức hợp, ngược lại, gồm hai hoặc nhiều thuộc tính.

Quan sát bảng **Accounts** (tài khoản)

Trong ví dụ trên, nếu chỉ có một giao dịch được tiến hành đối với một tài khoản tại một thời điểm, chúng ta có thể chọn AccountNumber (số tài khoản), TransactionDate (ngày giao dịch), TransactionTime (giờ giao dịch) làm khóa phức hợp.

## Thực thể

Một số thuộc tính đòi hỏi thuộc tính nâng cao để xác định tính chất trong quá trình thiết kế CSDL và trở thành thực thể. Chúng ta có thể tạo thực thể mới để biểu diễn các nhóm thuộc tính lặp lại nhiều lần. Ví dụ, thuộc tính “address” (địa chỉ) có thể lấy các thuộc tính nâng cao như “block” (khu nhà), “street” (đường), “city” (thành phố), “state” (bang) và “pin” (mã số cá nhân). Trong trường hợp này, thuộc tính “address” trở thành thực thể như trong ví dụ dưới đây:

*Thuộc tính có thể trở thành thực thể*

## Thực thể con

Xem ví dụ sau. Thực thể PLANT (phân xưởng) có các thuộc tính thể hiện trong lược đồ sau. Tuy nhiên, một vài thuộc tính không áp dụng được cho tất cả các phân xưởng. Ví dụ, thuộc tính “pressure” (áp suất) chỉ áp dụng cho phân xưởng hơi nước. Trong trường hợp này, tất cả các phân xưởng không sử dụng hơi nước có giá trị NULL trong cột “pressure”. Một cách hiệu quả là thay thế thuộc tính bằng thực thể con. Đây là phương thức *chuyên biệt hóa*.

Chuyên biệt hóa là kết quả của việc lấy tập con của một thực thể bậc cao hơn để tạo thành tập thực thể bậc thấp hơn. Trong ví dụ này, PLANT là tập thực thể bậc cao hơn, trong khi HYDRO (khí Hydro), STEAM (hơi nước) và NUCLEAR (hạt nhân) là các tập thực thể bậc thấp hơn.

### *Chuyên biệt hóa*

Quan sát sự đối lập của ví dụ trên. Hai thực thể , SAVING – ACCOUNT (tài khoản tiết kiệm) và CURRENT – ACCOUNT (tài khoản vãng lai), có những thuộc tính chung. Trong trường hợp này, lựa chọn tốt hơn là tạo một thực thể cha để đơn giản hóa đa quy chiếu. Điều đó gọi là *tổng quát hóa*.

Tổng quát hóa là kết quả của việc dùng tổ hợp của hai hay nhiều tập thực thể bậc thấp hơn để tạo ra tập thực thể bậc cao hơn.

Chú ý: Tổng quát hóa nêu bật sự giống nhau giữa tập thực thể bậc thấp hơn và che khuất sự khác biệt.

### *Tổng quát hóa*

Tổng quát hóa là đối lập của chuyên biệt hóa. Trong tổng quát hóa, tất cả thực thể bậc cao hơn đồng thời là thực thể bậc thấp hơn. Ví dụ, một tài khoản là thực thể bậc cao hơn, tài khoản tiết kiệm và tài khoản vãng lai là thực thể bậc thấp hơn. Chuyên biệt hóa không có ràng buộc này. Ví dụ, tất cả các tài khoản phải là tài khoản tiết kiệm hoặc tài khoản vãng lai.

Trong bài này, học viên cần nắm được:

- Mô hình khái niệm phản ánh thực thể và quan hệ của chúng. Phân tích thông tin giúp xác định thực thể và quan hệ. Mô hình khái niệm độc lập với hệ thống mà nó thực thi.
- Thực thể thường không phụ thuộc. chúng có thể tồn tại độc lập, riêng rẽ với thực thể khác.
- Mỗi thực thể ánh xạ tới một bảng.. Mỗi thuộc tính trong lược đồ E/R ánh xạ một thuộc tính trong bảng.
- Thực thể với thuộc tính chung nên hợp nhất với nhau. Các thuộc tính có thể lấy thuộc tính nâng cao và trở thành thực thể.
- Ánh xạ của quan hệ dựa vào kiểu quan hệ. Mỗi kiểu quan hệ ánh xạ tới bảng theo một cách riêng trong hệ Quản trị CSDL quan hệ.
- Trong quan hệ một – một, thể hiện của thuộc tính có thể liên kết với chỉ một thể hiện của thực thể liên quan.
- Trong quan hệ một - nhiều, một hiển thị của thực thể có liên kết với nhiều hơn một hiển thị của thực thể khác.
- Quan hệ nhiều – nhiều ánh xạ tới bảng. Quan hệ một – một không phổ biến và có thể ánh xạ tới khóa phụ của bảng.
- Thực thể yếu là thực thể tồn tại phụ thuộc vào thực thể khác.
- Thực thể con là tập con của thực thể khác. Thực thể con luôn tồn tại phụ thuộc vào thực thể cha.
- Khóa chính của thực thể cha là khóa phụ của thực thể con, chúng tại ra đường dẫn giữa hai thực thể.
- Khóa chọn là những lựa chọn cho khóa chính. Khóa thay đổi là những khóa chọn không được chọn là khóa chính.
- Những thuộc tính không bắt buộc nên thay bằng thực thể con. Đó là bước chuyên biệt hóa.
- Để đơn giản hóa việc đối chiếu đa chiều, cần tạo ra thực thể cha mới. Đây là bước tổng quát hóa.



Bài 2B:

# Chuẩn hóa và phi chuẩn hóa CSDL

Trong phần này, chúng ta nghiên cứu về:

- Mô tả phương pháp top-down và bottom-up
- Mô tả sự dư thừa dữ liệu
- Mô tả dạng chính tắc thứ nhất, thứ hai và thứ ba.
- Mô tả dạng chính tắc Boyce – Codd
- Nhận thức được yêu cầu phi chuẩn hóa.

Chú ý: Bạn đã đăng ký tại [www.nittstudent.com](http://www.nittstudent.com) chưa?

# Chuẩn hóa

## Phương pháp Top-Down và Bottom-Up.

Trong phần trước, chúng ta tìm hiểu về thiết kế CSDL logic sử dụng lược đồ E/R.. Có hai phương pháp tiếp cận thiết kế CSDL logic:

- Phương pháp top – down
- Phương pháp bottom – up

Kỹ thuật mô hình hóa thực thể quan hệ là phương pháp top – down, bao gồm nhận dạng thực thể, quan hệ và thuộc tính, vẽ lược đồ E/R, ánh xạ lược đồ tới bảng.

Trong phần này, chúng ta nghiên cứu về **Chuẩn hóa**, phương pháp bottom – up. Chuẩn hóa là sự phân giải từng các bản ghi phức tạp thành bản ghi đơn giản. Chuẩn hóa làm giảm dư thừa dữ liệu theo nguyên lý phân giải không gây mất mát. Phân giải không gây mất mát là sự chia nhỏ bảng thành các bảng nhỏ hơn mà không làm mất đi thông tin dữ liệu.

Thông thường, quy trình chuẩn hóa giống như quy trình vẽ lược đồ E/R. Tuy nhiên, tùy vào mức độ chính xác và chi tiết của lược đồ, quy trình chuẩn hóa có thể không cần thiết. Các bảng xây dựng từ lược đồ E/R có thể đã được chuẩn hóa. Trên thực tế, ít nhất chúng cũng ở dạng chuẩn thứ nhất.

Những người chuộng phương pháp bottom – up thường không qua bước mô hình hóa thực thể - dữ liệu. Sau khi hoàn thành lưu trữ dữ liệu tại CSDL, dữ liệu này được chuẩn hóa. Phương pháp top down là cách hữu hiệu nhất cho các thiết kế hiện tại.

## Dư thừa dữ liệu

*Dư thừa dữ liệu* nghĩa là sự lặp đi lặp lại cùng một dữ liệu. Dư thừa làm tăng thời gian cần thiết để cập nhật, thêm hoặc bớt dữ liệu, đồng thời tăng dung lượng đã sử dụng của ổ đĩa.

Ví dụ, quan sát cấu trúc của bảng **Student** (sinh viên):

STUDENT

Dữ liệu mẫu của bảng **Student** như sau:

Các thông tin chi tiết về sinh viên và điểm của họ được ghi trong bảng **Student**. Thông tin chi tiết về sinh viên như **StudentId** (mã số sinh viên), **StudentName** (tên), **StudentAddress** (địa chỉ) được lặp lại trong khi vào điểm của sinh viên qua từng học kỳ. Dữ liệu lặp là dư thừa. Thêm vào đó, nếu cần chỉnh sửa địa chỉ của một sinh viên, chúng ta sẽ phải chỉnh sửa trong nhiều hàng về sinh viên đó. Nếu không hoàn thành được, thông tin giữa các hàng sẽ bị sai lệch.

Nếu có 1000 sinh viên và thông tin về mỗi sinh viên chiếm 200 bytes dung lượng thì sẽ có 200000 bytes bị lặp. Như vậy, bộ nhớ sẽ bị sử dụng không hợp lý.

Dư thừa có thể gây ra:

- Sự bất thường trong cập nhật dữ liệu: chèn, chỉnh sửa, xóa dữ liệu có thể gây ra sai lệch
- Thiếu nhất quán: lỗi thường xảy ra khi dữ kiện bị lặp
- Sử dụng không hợp lý khoảng nhớ của ổ đĩa

Chúng ta có thể sử dụng kinh nghiệm hoặc hiểu biết chung để xây dựng CSDL. Tuy nhiên, chúng ta cũng có thể sử dụng phương pháp hệ thống như *chuẩn hóa* để giảm thiểu dư thừa và lặp lại dữ liệu.

## Sự cần thiết của chuẩn hóa

Chuẩn hóa là 1 phương pháp khoa học dùng để phân cấu trúc của bảng thành các cấu trúc bảng đơn giản hơn theo những quy tắc nhất định. Sử dụng phương pháp này, chúng ta có thể giảm thiểu dư thừa trong bảng và bớt những lỗi thuộc về thiếu nhất quán và sử dụng bộ nhớ. Chúng ta cũng được đảm bảo rằng thông tin không bị thất lạc.

Chuẩn hóa mang lại nhiều lợi ích. Nó đẩy nhanh quá trình phân loại dữ liệu và tạo mục lục, nhiều phân mục hơn, ít mục cho bảng, ít NULL, thu gọn CSDL. Chuẩn hóa đơn giản cấu trúc của bảng. việc thực hiện tác vụ được liên kết trực tiếp với thiết kế CSDL. Thiết kế sơ sài gây trở ngại cho hoạt động của hệ thống. thiết kế logic tạo nền móng cho CSDL tối ưu.

Những quy tắc cho một thiết kế CSDL đạt chuẩn:

- Mỗi bảng phải có một ký hiệu nhận dạng.
- Mỗi bảng chỉ nên lưu dữ liệu cho một kiểu thực thể.
- Nên tránh các cột có giá trị NULL
- Nên tránh lặp lại các cột hoặc các giá trị.

# Các dạng chuẩn khác nhau và phi chuẩn hóa

## Các dạng chuẩn

Kết quả của chuẩn hóa là sự tạo thành các bảng thỏa mãn những quy tắc nêu trên và thể hiện ở các dạng chuẩn hóa nhất định. Các *dạng chuẩn hóa* được dùng để đảm bảo các lỗi bất thường hoặc thiếu nhất quán không xảy ra đối với CSDL. Một cấu trúc bảng luôn ở một dạng chuẩn. một vài dạng chuẩn được đề cập đến ở đây. Những dạng chuẩn quan trọng và phổ biến nhất là:

- Dạng chuẩn thứ nhất (1 NF)
- Dạng chuẩn thứ hai (2 NF)
- Dạng chuẩn thứ ba (3 NF)
- Dạng chuẩn (chính tắc) Boyce – Codd. (BCNF)

## QUAN HỆ CHỨA CHUẨN HÓA

1 NF

2NF

3NF

BCNF

Các dạng chuẩn thứ nhất, thứ hai và thứ ba lần đầu được định nghĩa bởi tiến sĩ E.F Codd. Sau đó, Boyce và Codd giới thiệu một dạng chuẩn hóa khác là dạng chính tắc Boyce – Codd. Như trong hình vẽ, quan hệ ở dạng chuẩn thứ nhất cũng có thể ở dạng thứ hai hay thứ ba.

### Phụ thuộc hàm.

Khái niệm chuẩn hóa dựa trên khái niệm ban đầu về *phụ thuộc hàm*. Trước hết, hãy tìm hiểu về khái niệm *phụ thuộc hàm*.

Có một quan hệ ( nhắc lại rằng bảng cũng có thể gọi là quan hệ) R, thuộc tính A phụ thuộc hàm vào thuộc tính B nếu mỗi giá trị của A trong R liên kết chính xác với một giá trị của B.

Nói cách khác, thuộc tính A phụ thuộc hàm vào thuộc tính B khi và chỉ khi với mỗi giá trị của B, ta xác định được duy nhất một giá trị của A.

Thuộc tính B gọi là định thức.

Quan sát bảng Employee (nhân viên):

Employee

Cho một giá trị của **Code** (mã số), chỉ có duy nhất một giá trị thích hợp của **Name** (tên). Ví dụ, với **Code** E1 cho chính xác 1 giá trị **Name** là Mac. Do đó, **Name** phụ thuộc hàm **Code**. Tương tự, chỉ có 1 giá trị **City** cho mỗi giá trị của **Code**. Vì vậy, thuộc tính **City** phụ thuộc hàm vào thuộc tính **Code**. Thuộc tính **Code** là định thức. Chúng ta có thể nói rằng **Code** quyết định **City** và **Name**.

Chúng ta đã có những khái niệm cơ bản về phụ thuộc hàm. Trong ví dụ trên về thực thể EMPLOYEE, thuộc tính Code là duy nhất ở từng bộ. Do đó, đó là một khóa chọn. Mọi thuộc tính phụ thuộc hàm vào khóa.

Tuy nhiên, phụ thuộc hàm không đòi hỏi 1 thuộc tính phải là khóa mới có thể khiến các thuộc tính khác phụ thuộc hàm. Ví dụ sau giải thích vấn đề này. Giả sử chúng ta cần lưu thông tin về điểm của sinh viên trong chương trình đào tạo từ xa. Thuộc tính cần lưu trữ là:

ID : mã sinh viên để gửi phiếu điểm.

CITY : tên thành phố

C\_CODE : mã môn học

SCORE : tổng điểm của sinh viên

Chú ý rằng thành phố phiếu điểm được gửi đến cũng là nơi sinh viên đang ở. Vì thế, CITY phụ thuộc hàm ID. Nhưng ID không phải là khóa chọn. Khóa chọn trong trường hợp này là sự kết hợp giữa ID và C\_CODE. Thuộc tính ID và C\_CODE là khóa ngoài đối chiếu bảng giữ thông tin khách hàng và sản phẩm. Vì vậy, mặc dù ID không phải là khóa chọn, các thuộc tính khác vẫn phụ thuộc hàm vào nó.

Phụ thuộc hàm cũng có thể được định nghĩa như sau:

Cho quan hệ R, thuộc tính A phụ thuộc hàm B khi và chỉ khi hai bộ dữ liệu của R nhận giá trị của B cũng nhận cả giá trị của A.

Cho rằng thông tin về điểm lưu trữ trong bảng SCORES\_INFO (thông tin điểm) như trong bảng sau. Chú ý rằng với mỗi giá trị cụ thể của ID, giá trị của CITY là như nhau trong các bộ.

SCORES\_INFO

Phụ thuộc hàm là một phần của quá trình tìm hiểu ý nghĩa của dữ liệu. Ví dụ, trong quan hệ SCORES\_INFO, CITY phụ thuộc hàm ID nghĩa là mỗi sinh viên sống ở duy nhất 1 thành phố. Điều đó có nghĩa là:

- Có một ràng buộc trong thực thể mà CSDL mô phỏng – mỗi sinh viên chỉ sống ở 1 thành phố.
- Ràng buộc này phải thấy được trong CSDL.

Đồng thời, chú ý rằng phụ thuộc hàm phản ánh quan hệ nhiều – một. Phụ thuộc hàm cũng có nghĩa là nhiều sinh viên có thể cùng sống trong một thành phố nhưng một sinh viên chỉ có thể sống ở 1 thành phố mà thôi.

Các khái niệm về chuẩn hóa sẽ đưa ra những định nghĩa rất đơn giản về sự phụ thuộc hàm.

## Dạng chuẩn thứ nhất (1 NF)

Một bảng ở dạng chuẩn thứ nhất nếu mỗi ô trong bảng chỉ chứa duy nhất một giá trị.



Quan sát bảng **Project** (dự án):

Project:

Dữ liệu trong bảng chưa được chuẩn hóa vì các ô trong **ProjCode** (mã dự án) và **Hours** (Giờ) có nhiều hơn một giá trị.

Áp dụng định nghĩa 1 NF vào bảng **Project**, chúng ta có bảng sau:

Project

*Chú ý: Mô hình quan hệ không chấp nhận các bảng chưa được chuẩn hóa. Vì vậy, các bảng lấy từ lược đồ E/R ít nhất cũng phải ở dạng chuẩn thứ nhất.*

## Dạng chuẩn thứ hai (2 NF)

1 bảng ở dạng chuẩn thứ hai khi nó ở dạng chuẩn thứ nhất mà mọi thuộc tính trong hàng phụ thuộc hàm vào toàn bộ khóa chứ không chỉ một phần khóa.

Quan sát bảng Project:

Bảng có các hàng sau:

Trường hợp này đưa ra các vấn đề:

- Chèn vào:

Không thể ghi lại tên phòng của một nhân viên trước khi nhân viên đó nộp dự án.

- Cập nhật:

Đối với 1 nhân viên, mã nhân viên, tên phòng, trưởng phòng lặp lại rất nhiều lần. Do đó, nếu một nhân viên chuyển sang phòng khác, sự thay đổi cần ghi lại ở tất cả các hàng của bảng Employee liên quan đến nhân viên đó. Mọi sự bỏ sót đều dẫn đến thiếu nhất quán.

- Xóa:

Khi sinh viên hoàn thành dự án, ghi chép về nhân viên bị xóa, thông tin về phòng của nhân viên đó cũng sẽ bị mất.

Khóa chính ở đây là khóa phức hợp (**Ecode + ProjCode**)

Bảng thỏa mãn điều kiện của 1 NF. Việc chúng ta cần làm là kiểm tra sự thỏa mãn của 2 NF.

Trong bảng, với mỗi giá trị của **Ecode**, có nhiều hơn một giá trị của **Hours**. Ví dụ, cho **Ecode** E101, có 3 giá trị của **Hours** – 90, 101, 60. Vì thế, **Hours** không phụ thuộc hàm vào **Ecode**. Tương tự, cho mỗi giá trị của **ProjCode**, có nhiều hơn 1 giá trị của **Hours**. Ví dụ, với **ProjCode** P27, có 3 giá trị của **Hours**, 90, 10 và 72. Tuy nhiên, cho sự kết hợp của **Ecode** và **ProjCode**, chỉ xác định duy nhất 1 giá trị của **Hours**. Vì thế, **Hours** phụ thuộc hàm vào khóa tổng hợp **Ecode + ProjCode**.

Bây giờ, chúng ta kiểm tra xem **Dept** (phòng) có phụ thuộc hàm vào khóa tổng hợp **Ecode + ProjCode** hay không. Với mỗi giá trị của **Ecode**, chỉ xác định 1 giá trị của **Dept**. Ví dụ, với **Ecode** 101 chỉ xác định 1 giá trị. Vì vậy, **Dept** phụ thuộc hàm **Ecode**. Tuy nhiên, với mỗi giá trị của **ProjCode**, xác định được nhiều hơn 1 giá trị của **Dept**. Ví dụ, **ProjCode** P27 cho hai giá trị của **Dept**, Systems (hệ thống) và Finance (tài vụ). Vì vậy, **Dept** không phụ thuộc hàm **ProjCode**. **Dept** phụ thuộc hàm vào một phần khóa (**ECode**) và không phụ thuộc hàm vào khóa tổng hợp (**ECode + ProjCode**). Tương tự đối với thuộc tính **DeptHead** (trưởng phòng). Vì vậy, bảng **Project** không ở 2 NF. Để bảng ở 2 NF, thuộc tính không phải là khóa phải phụ thuộc hàm hoàn toàn vào khóa tổng hợp chứ không phải 1 phần của khóa.

### Hướng dẫn chuyển một bảng sang 2 NF.

- Tìm và gỡ bỏ các thuộc tính phụ thuộc hàm vào một phần hóa chứ không phải toàn bộ khóa. Đưa chúng vào bảng khác.
- Phân nhóm các thuộc tính còn lại.

Để chuyển bảng **Project** sang 2 NF, chúng ta phải gỡ bỏ những thuộc tính không phụ thuộc hàm vào khóa tổng hợp và đưa chúng sang một bảng khác với những thuộc tính mà chúng phụ thuộc hàm hoàn toàn. Trong ví dụ trên, vì **Dept** không phụ thuộc hàm hoàn toàn vào khóa tổng hợp **ECode + ProjCode**, chúng ta phải đưa **Dept** và **Ecode** sang bảng riêng gọi là **EmployeeDept** (phòng của nhân viên). Chúng ta cũng chuyển **DeptHead** sang bảng **EmployeeDept**.

Bây giờ, bảng **Project** chứa **ECode**, **ProjCode** và **Hours**.

EmployeeDept

Project

### Dạng chuẩn thứ ba (3 NF)

Một quan hệ ở dạng chuẩn thứ ba khi ở 2 NF và tất cả các thuộc tính không phải là khóa chỉ phụ thuộc hàm vào khóa chính.

Quan sát bảng **Employee**.

EMPLOYEE

Các vấn đề:

- Chèn

Trưởng phòng của một phòng mới không có nhân viên không thể nhập vào cột **DeptHead** bởi vì khóa chính chưa xác định.

- Cập nhật

Đối với một phòng, mã của trưởng phòng (**DeptHead**) được lặp lại nhiều lần. Do đó, nếu trưởng phòng chuyển sang phòng khác, sự thay đổi sẽ gây sai sót ở toàn bộ bảng.

- Xóa

Nếu ghi chép về một nhân viên bị xóa, thông tin liên quan đến trưởng phòng cũng đồng thời bị xóa. Do đó, thông tin bị thất lạc.

Chúng ta cần kiểm tra xem bảng có ở 3 NF không. Vì mỗi ô trong bảng có 1 giá trị, bảng ở 1 NF.

Khóa chính của bảng **Employee** là **ECode**. Với mỗi giá trị của **ECode**, chỉ có một giá trị của **Dept**. Do đó, thuộc tính **Dept** phụ thuộc hàm vào khóa chính **ECode**. Tương tự, với mỗi giá trị của **ECode**, chỉ xác định 1 giá trị của **DeptHead**. Vì vậy, **DeptHead** phụ thuộc hàm vào khóa chính **ECode**. Do đó, mọi thuộc tính phụ thuộc hàm hoàn toàn vào khóa tổng hợp, **ECode**. Như vậy bảng ở 2 NF.

Tuy nhiên, thuộc tính **DeptHead** cũng phụ thuộc vào thuộc tính **Dept**. Để trở thành 3 NF, tất cả các thuộc tính không phải khóa phải phụ thuộc hàm vào khóa chính. Bảng không ở khóa chính vì **DeptHead** phụ thuộc hàm vào **Dept**, không phải khóa chính.

### Hướng dẫn chuyển một bảng sang 3 NF.

- Tìm và gỡ bỏ thuộc tính không phải khóa phụ thuộc hàm vào thuộc tính không phải là khóa chính. Đưa chúng vào bảng riêng.
- Nhóm các thuộc tính còn lại.

Để chuyển bảng **Employee** sang 3NF, chúng ta phải gỡ bỏ cột **DeptHead** vì nó không phụ thuộc hàm vào khóa chính **ECode**, đưa nó vào bảng khác gọi là **Department** (phòng) cùng với thuộc tính **Dept** mà nó phụ thuộc hàm.

Employee

Department

## Dạng chuẩn Boyce – Codd (BCNF)

Định nghĩa ban đầu của 3 NF chưa đầy đủ trong một số trường hợp, không thỏa mãn đối với các bảng:

- Có nhiều khóa chọn
- Các khóa chọn đa hợp
- Các khóa chọn lồng nhau (có ít nhất một thuộc tính chung)

Vì vậy, một dạng chuẩn khác, *dạng chuẩn Boyce-Codd* được sử dụng. Chúng ta phải hiểu rằng trong các bảng khi 3 điều kiện trên không xuất hiện, chúng ta có thể dừng lại ở dạng chuẩn thứ ba. Trong những trường hợp như vậy, dạng chuẩn thứ ba cũng giống với dạng chuẩn Boyce-Codd.

Quan hệ ở dạng chuẩn Boyce-Codd (BCNF) khi và chỉ khi mọi định thức đều là khóa chọn.

Quan sát bảng **Project**:

Project

Bảng trên dư thừa dữ liệu. Nếu tên của nhân viên phải sửa, ta sẽ phải sửa toàn bộ bảng để tránh sự thiếu nhất quán.

**ECode + ProjCode** là khóa chính. Chú ý rằng **Name + ProjCode** có thể chọn làm khóa chính nên đó là một khóa chọn.

- **Hours** phụ thuộc hàm vào **ECode + ProjCode**
- **Hours** phụ thuộc hàm vào **Name + ProjCode**
- **Name** phụ thuộc hàm vào **ECode**
- **ECode** phụ thuộc hàm vào **Name**

Chúng ta dễ dàng nhận thấy:

- Có nhiều khóa chọn, **ECode + ProjCode** và **Name + ProjCode**
- Các khóa chọn đa hợp
- Khóa chọn lồng vào nhau vì thuộc tính **ProjCode** là chung

Đây là trường hợp đòi hỏi chuyển sang dạng chuẩn Boyce-Codd. Bảng đang ở dạng chuẩn thứ ba. Thuộc tính không phải khóa duy nhất là **Hours**, phụ thuộc vào khóa tổng hợp **ECode + ProjCode** hoặc **Name + ProjCode**.

**Ecode** và **Name** là các định thức vì chúng phụ thuộc hàm vào nhau. Tuy nhiên chúng không phải khóa chọn. Ở BCNF, định thức phải là khóa chọn.

## Hướng dẫn chuyển một bảng sang BCNF

- Tìm và gỡ bỏ các khóa chọn lồng nhau. Đưa phần của khóa chọn và thuộc tính nó phụ thuộc hàm vào một bảng riêng.
- Nhóm các thuộc tính còn lại vào một bảng

Như vậy, gỡ bỏ Name và ECode và đưa chúng vào bảng riêng. Chúng ta thu được bảng sau:

## Phi chuẩn hóa

Sản phẩm cuối cùng của chuẩn hóa là một tập các bảng liên kết với nhau chứa CSDL. Trong một số trường hợp, để có kết quả đơn giản hơn, chúng ta cần nối các bảng lại. Điều này ảnh hưởng đến hoạt động truy vấn. Khi đó, chúng ta nên chấp nhận dư thừa bằng cách thêm vào cột hay bảng.

### Định nghĩa phi chuẩn hóa.

Sự đưa thêm vào bảng dư thừa dữ liệu để cải thiện hoạt động gọi là *phi chuẩn hóa*.

Ví dụ, quan sát bảng sau, gồm thông tin về sản phẩm và đơn đặt hàng.

Orders (đơn đặt hàng)

Products (sản phẩm)



Nếu chúng ta phải tính toán tổng chi phí cho mỗi đơn đặt hàng gồm chi phí mua sản phẩm cùng thuế bằng 10% chi phí sản phẩm, truy vấn để tính tổng doanh số như sau:

Nếu có hàng ngàn hàng, máy chủ sẽ mất rất nhiều thời gian để xử lý truy vấn và đưa ra kết quả gồm sự kết hợp và tính toán. Do đó, để đẩy nhanh sự xử lý truy vấn, chúng ta phải lưu dữ liệu chi phí của mỗi đơn hàng cùng với thuế như sau:

Orders

Bây giờ, để tìm ra tổng doanh số, chúng ta đưa ra một truy vấn đơn giản:

Cấu trúc bảng đã đơn giản hóa truy vấn và đẩy nhanh quá trình xử lý truy vấn. bằng việc lưu thêm bảng, chúng ta gây ra dư thừa nhưng cải thiện hoạt động truy vấn.

Quyết định phi chuẩn hóa rõ ràng gây ra sự cân nhắc giữa hiệu quả hoạt động và dữ liệu nhất quán. Phi chuẩn hóa làm tăng dung lượng bộ nhớ cần sử dụng.

## Tóm tắt

Trong bài học này, chúng ta đã tìm hiểu:

- Chuẩn hóa có chức năng đơn giản hóa cấu trúc của bảng.
- Chuẩn hóa tạo ra các bảng mới thỏa mãn điều kiện cho trước, hiển thị ở một số dạng chuẩn. Các dạng chuẩn đảm bảo các hoạt động bất thường và sự thiếu nhất quán không xảy ra đối với CSDL. Cấu trúc bảng luôn ở một dạng chuẩn nhất định. Một số dạng chuẩn đã được giới thiệu trong bài.
- Những dạng chuẩn quan trọng và phổ biến nhất là:

Dạng chuẩn thứ nhất (1 NF)

Dạng chuẩn thứ hai (2 NF)

Dạng chuẩn thứ ba (3 NF)

Dạng chuẩn Boyce – Codd (BCNF)

- Sự sử dụng tính chất dư thừa có chủ ý trong bảng để cải thiện hiệu quả hoạt động được gọi là phi chuẩn hóa.
- Quyết định thực hiện phi chuẩn hóa gây ra sự cân nhắc giữa hiệu quả hoạt động và việc bảo toàn dữ liệu.
- Phi chuẩn hóa làm tăng dung lượng bộ nhớ cần sử dụng.

**Bài 2C:**

# Phối hợp

## Byte kiến thức

Trong phần này, chúng ta sẽ nghiên cứu về:

- Miền
- Dạng chuẩn thứ tư
- Các dạng chuẩn khác

### Miền

Miền là tập giá trị dữ liệu cùng kiểu. Nói cách khác, miền là vùng chứa giá trị từ đó giá trị thực của thuộc tính được lấy.

Ví dụ, quan sát miền CITIES (thành phố) chứa tên của các thành phố. CUSTOMER (khách hàng) và PRODUCT (sản phẩm) là hai bảng cùng có thuộc tính là tên của thành phố. Thuộc tính CITY trong bảng CUSTOMER ghi tên các thành phố mà các khách hàng đang ở. Thuộc tính CITY trong bảng PRODUCT ghi tên các thành phố tại đó sản phẩm được sản xuất. Cả hai thuộc tính này ở hai bảng đều lấy giá trị từ miền CITIES.

CUSTOMER

Miền: CITY

PRODUCT

### *Miền và thuộc tính*

Mọi giá trị của thuộc tính phải được lấy từ miền nằm dưới. Tuy nhiên, không phải mọi giá trị trong miền đều xuất hiện trong bảng. Trong hình minh họa, một số thành phố trong miền không tồn tại trên thuộc tính. Tuy nhiên, sau đó, mọi giá trị đều xuất hiện trên trong thuộc tính vì chúng là giá trị thực. Miền là một khái niệm và không có Hệ Quản trị CSDL quan hệ nào định nghĩa được hoàn hảo một khái niệm.

Một quan hệ là tập hợp các miền, gồm có hai phần, tựa đề và phần chính. Phần tựa đề gồm một tập cố định các thuộc tính. Phần chính gồm các hàng có thể thay đổi theo thời gian. Thuộc tính của quan hệ là không đổi. Nếu thêm hoặc bớt một thuộc tính vào quan hệ, quan hệ sẽ thay đổi. Những thay đổi như vậy chỉ ra rằng hoặc một miền mới được thêm vào tập hợp các miền có sẵn hoặc một miền xác định ban đầu một quan hệ không còn tồn tại nữa.

## Dạng chuẩn thứ tư

Theo định nghĩa 1 NF, một quan hệ không thể có thuộc tính đa trị. Điều đó có nghĩa là cho một giá trị của thuộc tính, chúng ta không thể xác định nhiều giá trị của thuộc tính khác. Tuy nhiên, trong thực tế, có những trường hợp đòi hỏi thuộc tính phải đa trị. Ví dụ, trong một trường học, sinh viên của khoa được giao nhiều nhiệm vụ. Hoặc, trong hệ thống bảo hiểm y tế, một nhân viên có rất nhiều khách hàng.

Khi chúng ta mô hình hóa các trường hợp này trong CSDL quan hệ, chúng ta có thể gặp phải sự dư thừa hoặc giá trị null. Ví dụ, đối với CSDL của trường học, hình minh họa sau mô tả bảng FACULTY (khoa) sử dụng các khái niệm trên.

FACULTY

### *Bảng với tính phụ thuộc đa trị*

Điều kiện đòi hỏi phải sao lưu các giá trị thì mới đảm bảo tính độc lập lẫn nhau của các thuộc tính đa trị gọi là tính phụ thuộc đa trị. Tính phụ thuộc đa trị là sự ràng buộc đối với bảng như phụ thuộc hàm.

Rõ ràng là tính phụ thuộc đa trị đòi hỏi sự sao lưu các giá trị dữ liệu. Vì vậy, khi chuẩn hóa, chúng ta phải gỡ bỏ các phụ thuộc đa trị. Để làm được điều này, chúng ta cần dùng dạng chuẩn thứ tư (4 NF). Một bảng ở 4 NF nếu nó ở 3 NF và không có các phụ thuộc đa trị. Để áp dụng 4 NF, chúng ta cần đưa tất cả các thuộc tính đa trị vào một bảng chứa khác của thuộc tính. Bảng trên có thể được chuẩn hóa như sau:

FAC-COURSE (môn học)

FAC-COMMITTEE (hội đồng)

#### *Bảng 4 NF*

Hai bảng trên ở 4 NF vì các thuộc tính đa trị như COURSE và COMMITTEE đang ở trong bảng của riêng chúng. Khóa chính của mỗi bảng trên gồm cả hai thuộc tính trong bảng. Điều này có nghĩa là khóa chính của bảng FAC-COURSE gồm cả FACULTY-NAME và COURSE. Khóa chính của bảng FAC-COMMITTEE gồm FACULTY-NAME và COMMITTEE.

### Các dạng chuẩn khác

Các dạng chuẩn 2 NF, 3 NF và 4 NF là kết quả của phụ thuộc hàm và phụ thuộc đa trị. Ngoài ra còn có những ràng buộc khác như quy tắc kinh doanh đưa ra nhu cầu về dạng chuẩn thứ năm (5 NF). Để một bảng ở 5 NF, bảng phải ở dạng 4 NF và tuân thủ một số quy tắc kinh doanh. Mục đích của 5 NF là tạo ra các bảng mà không thể khai triển thêm được nữa. Bây giờ chúng ta sẽ tìm hiểu bằng một ví dụ.

Quan sát bảng sau:

Bảng trên chỉ ra rằng khoa Khoa học Máy tính đề xuất 3 môn học CS150, CS103 và CS104. Các môn học này được đăng ký bởi nhiều sinh viên. Tại đây quy tắc kinh doanh là một môn học không thể được đăng ký bởi tất cả các sinh viên một sinh viên không thể đăng ký tất cả các môn học. Bảng trên không minh họa tính phụ thuộc đa trị vì các cột SUBJECT (môn học) và STUDENT (sinh viên) không độc lập. Cả hai cột không liên quan đến nhau và chứa những thông tin quan trọng. Vì vậy, bảng không thể khai triển thành hai bảng sau mà không sai lạc thông tin về môn học được đăng ký bởi sinh viên.

DEPT-SUB (DEPARTMENT, SUBJECT)

DEPT-STUD (DEPARTMENT, STUDENT)

Tuy nhiên, chúng ta có thể khai triển bảng trên thành ba bảng mà không làm mất thông tin.

DEPT-SUB (khoa, môn học)

Department	Subject
Toán	
Khoa học máy tính	
Khoa học máy tính	
Khoa học máy tính	
Hóa học	
Vật lý	

DEPT-STUD (khoa, sinh viên)

Department	Student
Toán	
Khoa học máy tính	
Khoa học máy tính	
Khoa học máy tính	
Hóa học	
Vật lý	

SUB-STUD (môn học, sinh viên)

Subject	Student

Nếu nguyên tắc kinh doanh không tồn tại, chúng ta không cần đến 5 NF.



Ngoài 5 NF, có một dạng chuẩn khác nữa là dạng chuẩn khóa – miền (DKNF) được gợi ý bởi Fagin. Dạng chuẩn này dựa vào định nghĩa khóa và miền thuộc tính. Bảng ở dạng DKNF nếu mọi ràng buộc trong bảng là kết quả của định nghĩa về miền và khóa.

## Lời của chuyên gia

Trong phần này, chúng ta sẽ tìm hiểu về cách tốt nhất để sử dụng khóa chính, thủ thuật về tổng quát hóa và một vài câu hỏi thường gặp về mô hình dữ liệu, lược đồ E/R và chuẩn hóa.

## Luyện tập

### Khóa chính

Cách tốt nhất để sử dụng khóa chính như sau:

- Khóa chính nên ở dạng số: bởi vì như vậy chúng ta có thể tự động lấy số cho mỗi bản ghi bằng mệnh đề nhận dạng. Điều này làm giảm rủi ro khi sao lưu giá trị khóa chính thành hai bản ghi. Tuy nhiên, nếu chúng ta sử dụng dạng chữ-số, chúng ta phải lấy giá trị khóa chính cho mỗi bản ghi bằng tay. Vì vậy, chúng ta có thể sẽ lấy cùng một giá trị khóa chính cho hai bản ghi.
- Khóa chính chỉ nên có duy nhất 1 cột: mặc dù nhiều cột có thể tạo thành một khóa chính, khóa chính tạo bởi 1 cột vẫn hiệu quả hơn. Bởi vì nếu nhiều cột tạo thành một khóa, khoảng nhớ cần thiết để lưu trữ khóa sẽ nhiều hơn. Thêm vào đó, nếu cần phải nhận dạng một bản ghi, chúng ta phải đối chiếu cả một nhóm các cột.
- Khóa chính không đổi theo thời gian: đây là một yêu cầu vì thay đổi khóa chính sẽ khiến chúng ta khó sử dụng các dữ liệu lịch sử vì các đường dẫn đã bị phá hủy.
- Khóa chính nên vô nghĩa: khóa chính không nên có một ý nghĩa nhất định. Điều này đảm bảo rằng khóa chính sẽ không thay đổi theo thời gian. Ví dụ, chúng ta có thể đặt mã nhân viên theo vị trí của nhân viên đó, như C1256 là mã của nhân viên ở Chicago. Bây giờ, nếu nhân viên với mã số C1256 chuyển tới Dallas, chúng ta sẽ phải thay đổi giá trị khóa chính.

## Thủ thuật

### Chuyển một lược đồ E/R chứa tổng quát hóa thành bảng

Chúng ta có thể chuyển một lược đồ E/R chứa tổng quát hóa thành bảng. phương pháp là tạo một bảng cho một thực thể bậc cao hơn. Đồng thời, tạo một bảng cho một thực thể bậc thấp hơn. Mỗi bảng chứa các cột cho mọi thuộc tính của thực thể bậc thấp hơn. Các bảng cũng gồm các cột cho mỗi thuộc tính của khóa chính của thực thể bậc cao hơn. Để hiểu điều này, quan sát lược đồ E/R sử dụng tổng quát hóa.

#### *Lược đồ E/R sử dụng tổng quát hóa*

Tại đây, ACCOUNT (tài khoản) là thực thể bậc cao hơn với khóa chính ACC-NUMBER (số tài khoản), SAVING-ACCOUNT (tài khoản tiết kiệm) và CURRENT-ACCOUNT (tài khoản vãng lai) là các thực thể bậc thấp hơn. Trong lược đồ E/R này, chúng ta có ba bảng: ACCOUNT với cột ACC-NUMBER và BALANCE (kết toán), SAVING-ACCOUNT với cột ACC-NUMBER và INTEREST-RATE (lãi suất), CURRENT-ACCOUNT với cột ACC-NUMBER và OVERDRAFT-LIMIT (hạn mức thấu chi).

## Những câu hỏi thường gặp

- Tại sao chúng ta cần tạo mô hình quan niệm khi chúng ta có thể trực tiếp tạo mô hình quan hệ?

Số lượng các thực thể, quan hệ, phép gộp, chuyên biệt hóa càng tăng thì sự phức tạp của mô hình dữ liệu càng tăng. Khi tăng sự phức tạp của mô hình, chúng ta sẽ gặp khó khăn khi xây dựng mô hình dữ liệu. Tuy nhiên, một CSDL được dùng để cung cấp thông tin cho nhân viên trong tổ chức. Vì vậy, cấu trúc CSDL không được có lỗi. Kể cả khi mô hình dữ liệu không thể đảm bảo cấu trúc chính xác hoàn toàn, mô hình quan niệm sẽ dẫn đến ít lỗi hơn mô hình quan hệ. Đây là lý do chúng ta phải tạo mô hình quan niệm trước khi xây dựng mô hình quan hệ.

- Giữa 3NF và BCNF, chúng ta nên chọn dạng nào?

Lợi thế của việc dùng 3 NF là chúng ta có thể tạo 3 NF mà không gây ra mất mát phụ thuộc hàm. Bất lợi của 3 NF là sự lặp lại của thông tin. Tuy nhiên, nếu được chọn giữa 3 NF và BCNF, chúng ta thường chọn 3 NF bởi vì nếu không thể kiểm tra phụ thuộc hàm, hoạt động của hệ thống sẽ bị ảnh hưởng hoặc gây ra mất toàn vẹn dữ liệu. Vì vậy, để tránh rủi ro, chúng ta nên dùng 3 NF. Tuy nhiên, nếu không có các rủi ro trên, chúng ta có thể dùng BCNF.

- Sự khác nhau giữa thực thể yếu và thực thể con là gì?

Thực thể yếu phụ thuộc vào thực thể thường để tồn tại trong khi thực thể con là một phần của thực thể thường. Ví dụ, một thực thể có tên là Students được dùng để lưu trữ mọi thông tin về sinh viên. Nếu tất cả các sinh viên đều đăng ký một môn học và một số sinh viên đang nghỉ, không đăng ký một môn nào cả. Trong trường hợp này, chúng ta có thể có thực thể con là Break-Students (sinh viên đang nghỉ), lưu thông tin về tất cả các sinh viên đang nghỉ. Chú ý rằng thực thể con chứa tất cả các cột của thực thể cha mà nó phụ thuộc. Thực thể yếu, mặt khác, chứa các thuộc tính khác với những thuộc tính mà thực thể thường phụ thuộc.

- Sự khác nhau giữa tập thực thể con và tập thực thể cha là gì?

Chúng ta sẽ tìm hiểu qua một ví dụ. Một thực thể có tên là Student có hai thực thể con là Boarder (sinh viên nội trú) và Day-scholar (sinh viên ngoại trú). Thực thể Students lưu dữ liệu về sinh viên như tên, tuổi, môn học, lớp, khó chính là mã số sinh viên. Thực thể con Boarder có thuộc tính riêng là Room-no (số phòng), trong khi thực thể Day-scholar có thuộc tính riêng là Locker-no (mã số tủ). ngoài các thuộc tính riêng này, thực thể con chứa khóa chính của thực thể cha.

- Một bản ghi được chuẩn hóa hoàn toàn chứa những gì?

Một bản ghi chuẩn hóa hoàn toàn chứa các thông số:

■ Khóa chính để nhận diện thực thể  
Tập thuộc tính mô tả thực thể

## Luyện tập

1. Mệnh đề sau được lấy ra từ trường hợp của một xí nghiệp muốn bảo toàn dữ liệu của mình: “Một nhà cung ứng chuyên từng phần nhất định”. Nêu các thực thể có trong mệnh đề này và quan hệ của chúng. Vẽ lược đồ E/R mô tả quan hệ.
2. Bạn vừa nhận được một cấu trúc bản dự thảo cho bảng Position (vị trí làm việc). Sau khi kiểm tra cấu trúc bảng với các dữ liệu, bạn phát hiện ra lỗi trong quá trình chèn, xóa và sửa chữa dữ liệu. Bạn nhận thấy cấu trúc bảng này sẽ dẫn đến thiếu nhất quán dữ liệu và sẽ chiếm nhiều dung lượng bộ nhớ. Sửa chữa cấu trúc bảng sau để tối ưu hóa việc lưu trữ dữ liệu.

Cấu trúc bảng như sau:

<i>Position</i>
cPositionCode (mã vị trí)
vDescription (Mô tả)
iBudgetedStrength (khả năng dự toán)
siYear(năm)
iCurrentStrength (Khả năng hiện tại)
vSkill (kỹ năng)

Bảng mẫu cho bảng **Position** như sau:

cPositionCode	vDescription	iBudgetedStrength	iCurrentStrength	vSkill
	Giám đốc kinh doanh			Giao tiếp
	Giám đốc tiếp thị			Thuyết trình
	Nhân viên phân tích tài chính			Lãnh đạo nhóm
	Trợ lý giảng dạy			Giao tiếp
	Nhân viên phân tích dữ liệu			Thuyết trình
	Kế toán			Thuyết phục
	Giám đốc kiểm toán			Lập kế hoạch
	Nhân viên trực điện thoại			Giao tiếp
	Phụ trách văn phòng			Lãnh đạo nhóm
	Cố vấn pháp luật			Tin học văn phòng
	Trợ lý giám đốc			Liên hệ
	Chiêu đãi viên			Giao tiếp

	cấp cao			
	Cố vấn			Lãnh đạo nhóm
	Nhân viên kỹ thuật			Thuyết trình
	Lễ tân			Thuyết phục

3. Mệnh đề sau đúng hay sai:

Thuộc tính có thể lấy thuộc tính mở rộng và trở thành thực thể.

4. Mỗi khi phiếu lương của nhân viên được hoàn thành, tiền thưởng (nếu có) sẽ được tính và in trong phiếu lương. Ba bảng sau được dùng cho truy vấn dữ liệu – **MonthlySalary** (lương tháng), **Employee** (nhân viên) và **EmployeeReferrals** (tiền thưởng). Cấu trúc các bảng như sau:

<i>Employee</i>	<i>MonthlySalary</i>	<i>EmployeeReferral</i>
cEmployeeCode (mã nhân viên)	cEmployeeCode	cEmployeeReferralNo (số nhận thưởng)
vFirstname (tên)	mMonthlySalary (lương tháng)	cEmployeeCode
vLastName (họ)	dPayDate (ngày trả lương)	cCandidateCode (mã số dự thưởng)
cCandidateCode		
vAddress (địa chỉ)		
cCity (thành phố)		
cZip (hòm thư)		
cCountryCode (mã vùng)		
cPhone (số điện thoại)		
vQualification (bằng cấp)		
dBirthDate (ngày sinh)		
cSex (giới tính)		
cCurrentPosition (vị trí hiện tại)		
cDesignation (người nhận)		
cEmailId (địa chỉ email)		
cDepartmentCode (mã phòng)		
cRegion (vùng)		
imPhoto (Ảnh)		
vSkill (kỹ năng)		
dJoiningDate (ngày vào làm)		
dResignationDate (ngày thôi làm)		
cSocialSecurityNo (số thẻ bảo hiểm xã hội)		

Tuy nhiên, vì các cấu trúc bảng quá lớn, chúng ta cần tăng khả năng truy vấn bằng cách sửa cấu trúc bảng. Chỉ ra các làm tăng xử lý truy vấn.

5. Hậu quả của dư thừa là gì?
  - a. Cập nhật bất thường
  - b. Tăng khả năng xử lý dữ liệu
  - c. Thiều nhất quán
  - d. Tiêu tốn không cần thiết dung lượng bộ nhớ

6. Mệnh đề sau đúng hay sai:

Việc thiết kế mô hình khái niệm có liên quan tới giai đoạn đưa vào thực thi và hoạt động của của CSDL.

7. Quy tắc nào sau đây cần tuân theo để tạo một thiết kế CSDL hiệu quả?

- a. Mỗi bảng phải có một ký hiệu nhận dạng
- b. Phải tránh các cột chứa giá trị NULL
- c. Mỗi bảng phải lưu dữ liệu cho nhiều kiểu thực thể
- d. Phải tránh sự trùng lặp giá trị hoặc cột

8. Nối các thuật ngữ sau với đúng mô tả.

1. Chuyên biệt hóa	A. Là một phương pháp khoa học dùng để chia nhỏ các cấu trúc bảng phức tạp ra làm các cấu trúc bảng đơn giản hơn sử dụng các quy tắc nhất định
2. Chuẩn hóa	B. Tạo ra dư thừa dữ liệu có chủ ý để cải tăng khả năng xử lý
3. Tổng quát hóa	C. Kết quả của việc lấy tập con của thực thể bậc cao hơn đạo thực thể bậc thấp hơn.
4. Phi chuẩn hóa	D. Kết quả của việc lấy hợp của hai hay nhiều thực thể bậc thấp hơn để tạo thực thể bậc cao hơn.

9. Quan hệ ở dạng chuẩn thứ ba và không có phụ thuộc đa trị thì ở \_\_\_\_\_

10. Đối với một xí nghiệp sản xuất:

“Một nhà cung ứng cung cấp những phần nhất định. Một phần riêng biệt không nhất thiết phải cung cấp bởi một nhà cung ứng duy nhất. Không nhà cung ứng nào chỉ cung cấp một phần nào đó.”

Đây là kiểu quan hệ gì? Vẽ lược đồ E/R để mô tả quan hệ.



**Bài 2D:**

# Thực hành

## Bài tập

### Bài 1:

Shopping Spree là một cửa hàng tạp phẩm hàng đầu New Jersey. Cửa hàng có một lượng khách thường xuyên mua hàng khối lượng lớn. Cửa hàng thường xuyên lấy phản hồi của khách hàng để điều tra sự hài lòng của họ về sản phẩm. Jim Lewis, nhân viên điều tra thị trường của Shopping Spree cần dựng lược đồ E/R thể hiện trường hợp trên và ánh xạ lược đồ tới bảng.

### Bài 2:

Steve Irving là nhân viên thiết kế CSDL của Thư viện New Publications. Thư viện có ba phòng: Học thuật, Báo chí, Văn học. Thực thể bậc cao hơn được biểu diễn như sau:

Chủ	Thư viện	Vị trí
	Sách vật lý	Truyền trình thám
	Tạp chí	

Steve cần vẽ lược đồ chuyên biệt hóa cho các thực thể nói trên.

### Bài 3:

Don Allen là phụ trách kinh doanh của một cửa hàng tạp phẩm. Don cần phải kiểm tra ghi chép về việc kinh doanh của cửa hàng hàng tuần. Nhân viên kỹ thuật đã tạo một bảng gồm các trường hợp lý. Don cần chuẩn hóa các bảng và tạo các bảng độc lập truyền tải kiểu thông tin riêng biệt.

Chú ý: dùng bảng SALES DATA (dữ liệu bán hàng) để giải bài tập.

SALES DATA

<i>C_ID</i> (mã số)	<i>C_Name</i> (tên)	<i>Order_No</i> (số thứ tự)	<i>Item</i> (tên hàng)	<i>Qty</i> (số lượng)	<i>Rate</i> (đơn h giá)	<i>Bill_No</i> (số hóa đơn)	<i>Bill_Amt</i> (số lượng hóa đơn)	<i>Bill_Dt</i>

## Bài tập về nhà:

1. Quan hệ nào hình thành khi có nhiều sinh viên cùng đăng ký một môn học?
  - a. Một – một
  - b. Một – nhiều
  - c. Nhiều – một
  - d. Nhiều – nhiều
2. Kiểu quan hệ nào cho phép chúng ta ghép nối hai bảng cho truy cập nhanh hơn?
  - a. Một – một
  - b. Nhiều – một
  - c. Một – nhiều
  - d. Nhiều – nhiều
3. Khóa nào được tạo thành bởi nhiều hơn một thuộc tính?
  - a. Khóa chính
  - b. Khóa phụ
  - c. Khóa thay thế
  - d. Khóa phức hợp
4. Khóa nào là lựa chọn cho khóa chính nhưng không được chọn làm khóa chính?
  - a. Khóa thay thế
  - b. Khóa phụ
  - c. Khóa chọn
  - d. Khóa phức hợp
5. Đây là kết quả của việc hợp hai hay nhiều tập thực thể bậc thấp hơn để tạo ra tập thực thể bậc cao hơn?
  - a. Quan hệ
  - b. Chuyên biệt hóa
  - c. Thực thể cha
  - d. Tổng quát hóa

6. Chuẩn hóa làm giảm dư thừa sử dụng:
  - a. Lược đồ E/R
  - b. Khóa chính và khóa phụ
  - c. Phân giải bảo toàn dữ liệu
  - d. Phương pháp top-down
7. Bất thường trong cập nhật gây ra bởi:
  - a. Chuẩn hóa
  - b. Quan hệ thực thể
  - c. Phụ thuộc hàm
  - d. Dư thừa
8. Thuộc tính A trong một quan hệ phụ thuộc hàm vào thuộc tính B, thuộc tính B gọi là:
  - a. Định thức
  - b. Khóa chọn
  - c. Khóa phức hợp
  - d. Khóa chính
9. Trong dạng chuẩn nào sau đây, mọi định thức trong quan hệ là khóa chọn?
  - a. 1 NF
  - b. 2 NF
  - c. 3 NF
  - d. BCNF
10. Trong dạng chuẩn nào sau đây mọi thuộc tính trong hàng phụ thuộc hàm vào khóa tổng hợp của quan hệ, không phải một phần khóa?
  - a. 1 NF
  - b. 2 NF
  - c. 3 NF
  - d. BCNF

## Bài 3A:

# Thiết kế CSDL vật lý

Trong bài này, chúng ta nghiên cứu về:

- Phân biệt các kiểu quan hệ:

■ Bảng cơ sở  
■ Kết quả vấn tin  
■ Hiển thị

- Tạo bảng
- Thay đổi bảng
- Gỡ bỏ bảng
- Bảng vấn tin và sử dụng
- Định nghĩa hiển thị
- Sử dụng ngôn ngữ quản lý dữ liệu (DML) trên hiển thị
- Phân biệt các kiểu hiển thị khác nhau
- Phân biệt các ràng buộc nhất quán dữ liệu

## Ngôn ngữ sử dụng trong mô hình quan hệ

Phần lớn các hệ CSDL quan hệ đều sử dụng Ngôn ngữ truy vấn theo cấu trúc (SQL). Cũng như các ngôn ngữ truy vấn khác, SQL là sự kết hợp giữa 3 ngôn ngữ thứ cấp là:

- Ngôn ngữ định nghĩa dữ liệu
- Ngôn ngữ khai thác dữ liệu
- Ngôn ngữ điều khiển dữ liệu

Câu lệnh DDL gồm các thao tác để tạo và xóa các bảng và mục lục. Câu lệnh DML dùng để nhập dữ liệu vào các bảng vừa tạo ra. DML cũng dùng để cập nhật, xóa dữ liệu và thực hiện các truy vấn phức tạp trên bảng. Câu lệnh DCL dùng để điều khiển sự truy cập của người dùng vào các bảng

### Các kiểu quan hệ:

Hệ thống quan hệ gồm nhiều kiểu quan hệ khác nhau. Tuy nhiên, ba kiểu quan hệ quan trọng nhất là:

- Bảng cơ sở
- Kết quả truy vấn
- Hiển thị

### Bảng cơ sở

Bảng cơ sở là một bảng đã được định danh tồn tại vật lý trong CSDL. Chúng ta có thể tạo vật lý một bảng sau khi đã ánh xạ lược đồ thực thể - quan hệ tới bảng và chuẩn hóa bảng. Câu lệnh DDL được sử dụng để tạo bảng.

### Tạo bảng

Mệnh đề SQL tạo bảng là CREATE TABLE (tạo bảng). Ví dụ, câu lệnh sau tạo bảng về khách hàng:

Câu lệnh CREATE TABLE theo sau bởi tên bảng, tức khách hàng trong trường hợp này. Chú ý rằng mọi thứ sau tên bảng đều phải ở trong ngoặc kép. Thông tin trong ngoặc kép gồm có tên cột hay thuộc tính của bảng khách hàng. Các thuộc tính trong bảng này là cust-no (số khách hàng), name (tên), phone-number (số điện thoại) và city (thành phố). CHAR (ký tự), có nghĩa là cột cust-no sẽ chứa kiểu dữ liệu ký tự. Kích thước của cột viết trong ngoặc kép sau tên kiểu dữ liệu.

Khóa chính cũng được xác định trong câu lệnh trên sử dụng mệnh đề PRIMARY KEY (khóa chính). Mệnh đề có nghĩa là thuộc tính cust-no là khóa chính. Khóa chính không chứa giá trị NULL. Vì thế, mệnh đề NOT NULL (không phải NULL) được chỉ rõ cùng thuộc tính cust-no. Mệnh đề NOT NULL có thể dùng để xác định bất cứ thuộc tính nào không chứa NULL bất kể đó có phải là khóa chính hay không. Ví dụ, nếu chúng ta không muốn có giá trị NULL trong tên cột, câu lệnh CREATE TABLE có thể sửa thành:

Bây giờ chúng ta sẽ tạo bảng có thuộc tính tham chiếu tới bảng khác. Ví dụ, câu lệnh SQL để tạo bảng sale (bán hàng) là:

Nếu khóa chính được hình thành bởi nhiều hơn một thuộc tính, thì các thuộc tính được đặt tên theo mệnh đề PRIMARY KEY và tách biệt bởi dấu phẩy. Trong ví dụ trên, khóa chính được tạo thành bởi hai thuộc tính, cust-no và prod-no (số hiệu sản phẩm). Mệnh đề FOREIGN KEY (khóa phụ) theo sau bởi tên của thuộc tính. Mệnh đề REFERENCES (tham chiếu) theo sau bởi tên của bảng mà thuộc tính tham chiếu.



## Sửa đổi bảng

Chúng ta có thể sửa đổi bảng hiện có bằng câu lệnh ALTER TABLE (sửa bảng). Ví dụ, câu lệnh SQL chứa cột address (địa chỉ) trong bảng customer như sau:

Câu lệnh trên thêm vào một cột thứ năm, cột địa chỉ trong bảng customer và đặt giá trị NULL cho tất cả các giá trị của cột. Chúng ta không thể xác định mệnh đề NOT NULL cùng với câu lệnh ALTER TABLE.

## Gỡ bỏ bảng

Chúng ta cần gỡ bỏ bảng bằng câu lệnh DROP TABLE (bỏ bảng). Ví dụ, câu lệnh SQL để gỡ bỏ bảng customer là:

Khi tạo bảng, mô tả của bảng được lưu lại trong danh mục hệ thống. Danh mục hệ thống lưu trữ dữ liệu tên của bảng, thuộc tính, kiểu dữ liệu của thuộc tính. Danh mục hệ thống cũng lưu trữ thông tin như người dùng của bảng. Vì vậy, khi viết câu lệnh DROP TABLE, mô tả của bảng sẽ bị gỡ bỏ khỏi danh mục hệ thống.

## Kết quả vấn tin

Kết quả truy vấn tạo cho bảng cũng là bảng khác. Ví dụ, truy vấn để liệt kê tên của mọi khách hàng trong bảng customer sẽ cho ra bảng sau:

### *Kết quả truy vấn*

Chúng ta sẽ tìm hiểu thêm về câu lệnh DML trong SQL được dùng để truy vấn bảng và sử dụng chúng. Các câu lệnh là: SELECT (chọn), INSERT (chèn), UPDATE (cập nhật), DELETE (xóa).

## Câu lệnh SELECT

SELECT là câu lệnh DML có hiệu quả nhất trong SQL. Mọi toán tử quan hệ đều được thực hiện qua câu lệnh SELECT.

### Truy vấn đơn giản

Nếu chúng ta cần tên và số điện thoại của tất cả khách hàng, câu lệnh SQL cho truy vấn này có thể là:

Câu lệnh đưa ra bảng sau:

#### *Kết quả truy vấn*

Câu lệnh thể hiện phép chiếu trong đại số quan hệ, chỉ lấy ra các cột name và phone-number từ bảng customer. Mệnh đề SELECT xác định các cột và mệnh đề FROM xác định bảng mà các cột được lấy. Một truy vấn tương tự như sau:

Trong câu lệnh trên, cột tên được tiếp đầu bằng tên của bảng mà nó phụ thuộc. Những tên cột như vậy được gọi là tên cột định tính.

Bây giờ, nếu chúng ta cần các cột trong bảng customer chỉ cho các khách hàng sống ở Boston, câu lệnh SQL như sau:

Câu lệnh thể hiện phép chọn trong toán tử quan hệ. Dấu hoa thị theo sau mệnh đề SELECT có nghĩa là mọi cột trong bảng sẽ được thể hiện.

Tuy nhiên, sự thể hiện của các hàng sẽ được chọn sao cho thỏa mãn điều kiện (city = "Boton")

Kết quả truy vấn như sau:

### *Kết quả truy vấn*

Quan sát truy vấn sau:

Kết quả là

### *Kết quả truy vấn*

Bảng trên đã nhân đôi các hàng. SQL không thể tự động xóa các hàng bị lặp trong truy vấn. Tuy nhiên, chúng ta có thể bỏ các hàng lặp bằng mệnh đề DISTINCT (phân biệt).

### *Kết quả truy vấn*

Chú ý rằng kết quả truy vấn không có các hàng lặp lại.

## Khôi phục theo thứ tự

Các hàng trong quan hệ không có thứ tự nào cả. Tuy nhiên, SQL có thể tạo nột thứ tự trong kết quả truy vấn bằng mệnh đề `ORDER BY` (thứ tự trong). Ví dụ, nếu muốn tìm tên của các mẫu sản phẩm giá dưới \$200, và thể hiện kết quả truy vấn theo thứ tự tăng dần của tên thành phố, câu lệnh SQL là:

### *Kết quả truy vấn*

Thứ tự mặc định là tăng dần. Tuy nhiên, chúng ta cũng có thể tạo thứ tự giảm dần bằng mệnh đề `DESC` như sau:

## Truy vấn kết hợp.

Truy vấn trong đó dữ liệu được khôi phục từ hơn một bảng gọi là truy vấn kết hợp. Khả năng kết hợp hai hay nhiều bảng là tính chất quan trọng nhất của SQL.

## Đăng hợp

Đăng hợp còn được gọi là kết hợp tự nhiên. Ví dụ, nếu chúng ta cần kết hợp giữa các bản ghi về khách hàng sống trong cùng một thành phố, nơi một sản phẩm nào đó được sản xuất, câu lệnh SQL như sau:

Chú ý rằng sự kết nối giữa bảng customer và product (sản phẩm) được xác định sử dụng điều kiện hợp `customer.city = product.city`. Đồng thời, cột names được định tính bằng cách phân biệt cột của một bảng với bảng khác. Điều này thể hiện phép kết hợp trong đại số quan hệ.

Kết quả truy vấn như sau:

### *Kết quả truy vấn*

Với phép kết hợp, chúng ta cũng có thể xác định riêng các cột để phục hồi. Ví dụ,

Kết quả truy vấn chỉ có tên của các khách hàng và sản phẩm ở trong cùng một thành phố.

Chúng ta cũng có thể xác định điều kiện bổ sung của phép kết hợp. Ví dụ,

## Tự kết hợp

Tự kết hợp là kết hợp một bảng với chính nó. Tự kết hợp phản ánh hiệu quả của SQL trong các truy vấn phức tạp. Ví dụ, nếu chúng ta cần tên theo cặp của các khách hàng sống trong cùng một thành phố, truy vấn như sau:

Truy vấn bao gồm tự kết hợp, kết hợp giữa các thành phố theo cặp với nhau. Để tiến hành truy vấn, chúng ta cần hai bản sao của bảng customer. Chúng ta phải kiểm tra tất cả các cặp khách hàng, từ bản sao thứ nhất của bảng customer và bản thứ hai.

Để phân biệt giữa hai bản sao, hai biến được đưa vào, thứ nhất và thứ hai. Biến thứ nhất thể hiện bản sao thứ nhất của bảng customer và biến thứ hai từ bản sao thứ hai của bảng. Truy vấn được tiến hành bằng cách kiểm tra tất cả tất cả các cặp giá trị khả năng của biến thứ nhất và thứ hai. Trong mỗi rường hợp, điều kiện WHERE (ở) được kiểm tra. Điều kiện AND (và) `first.cust-no < second.cust-no` loại bỏ những cặp giá trị (x,x). Đồng thời, nó đảm bảo cặp giá trị (x,y) và (y,x) không xuất hiện cùng nhau.

## Hàm tập hợp

Có những truy vấn không thể trả lời bằng cách tạo câu lệnh SELECT. Ví dụ như:

- Đã bán được bao nhiêu sản phẩm?
- Sản phẩm nào là đắt nhất?
- Tổng số đã được bán là bao nhiêu?
- Giá trung bình của sản phẩm là bao nhiêu?
- Sản phẩm nào là rẻ nhất?
- Tổng số lượt bán hàng là bao nhiêu?

Tất cả các truy vấn trên có thể giải quyết bằng cách sử dụng hàm tập hợp của SQL như SUM (tổng), AVG (trung bình) và COUNT (đếm). Câu trả lời cho các truy vấn đó như sau:

- Đã bán được bao nhiêu sản phẩm?

Kết quả truy vấn là 5.

- Sản phẩm nào là đắt nhất?

Kết quả của truy vấn là:

- Tổng số đã được bán là bao nhiêu?

Kết quả truy vấn là 1027.

- Giá trung bình của sản phẩm là bao nhiêu?

Kết quả của truy vấn là 7397.67.

- Sản phẩm nào là rẻ nhất?

Kết quả truy vấn là:

- Tổng số lượt bán hàng là bao nhiêu?

Kết quả truy vấn là 7.

Tất cả các hàm tập hợp, trừ hàm COUNT (\*), cho phép sử dụng mệnh đề DISTINCT và bỏ qua giá trị NULL.

Nếu chúng ta cần tổng số lượng đã được bán, truy vấn sẽ như sau:

Kết quả truy vấn là:

### *Kết quả truy vấn*

Mệnh đề GROUP BY (nhóm bằng) tạo nhóm dựa trên tên cột xác định được bằng mệnh đề này. Trong một nhóm, tất cả các hàng có cùng giá trị cho cột GROUP BY. Trong ví dụ trên, bảng sale được nhóm sao cho một nhóm có tất cả các hàng đều có chung số hiệu sản phẩm. Sau đó, mệnh đề SELECT được dùng để cho mỗi nhóm thay vì cho mỗi hàng trong bảng. Biểu thức trong mệnh đề SELECT là đơn trị trong nhóm. Chỉ cột GROUP BY và hàm SUM có thể giảm giá trị của các hàng xuống thành đơn trị. Mệnh đề GROUP BY hoàn toàn khác với mệnh đề ORDER BY. GROUP BY không sắp xếp lại kết quả hiển thị theo một thứ tự nào cả.

Nếu cần phải hiển thị mã sản phẩm cho tất cả các hàng hóa bán cho nhiều hơn một khách hàng, truy vấn sẽ là:

Chú ý rằng mệnh đề mới HAVING (có) được sử dụng trong truy vấn. Mệnh đề này giống với mệnh đề WHERE. Điểm khác biệt duy nhất là mệnh đề WHERE làm việc với các hàng còn mệnh đề HAVING làm việc với các nhóm. Biểu thức trong mệnh đề HAVING cũng phải đơn trị trên từng nhóm.

## Những đặc tính nâng cao của SQL

### Xử lý giá trị NULL

NULL nghĩa là các thông tin không xác định được hoặc đã bị mất. Không có giá trị nào được xác định là bằng NULL. Ví dụ, câu lệnh sau là không chính xác.

SQL đưa ra mệnh đề IS NULL (bằng NULL) (hoặc IS NOT NULL) để tìm giá trị NULL. Vì thế, câu lệnh trên cần được sửa thành:



Kết quả truy vấn như sau:

### *Kết quả truy vấn*

## Mẫu vấn tin con

Vấn tin nằm trong một vấn tin khác gọi là vấn tin con. Ví dụ, nếu cần tên của khách hàng đã mua sản phẩm P5690, truy vấn sẽ là:

Đầu tiên, truy vấn con trong ngoặc kép được đánh giá. Tất cả các giá trị của cust-no được prod-no P5690 lấy ra trong bảng sale. Tập con được tả lại. Sau đó, truy vấn bậc cao hơn được đánh giá. Tất cả các tên của khách hàng lấy ra từ bảng customer cho giá trị của tập con.

Mệnh đề IN (trong) dùng cho một tập giá trị. Nếu tập chỉ có một giá trị, IN đồng nghĩa với “=”.

Kết quả cho truy vấn trên giống như kết quả của phép kết hợp:

Như vậy, có hai dạng của cùng một truy vấn. Một dạng gồm truy vấn con và dạng khác gồm phép kết hợp. Điều này áp dụng cho hầu hết các truy vấn.

Chúng ta có thể xen kẽ các vấn tin con ở mức độ sâu hơn. Ví dụ, nếu cần tên của khách hàng đã mua một loại TV nào đó, truy vấn như sau:

## Truy vấn gồm phép hợp.

Phép hợp trong đại số quan hệ được thể hiện bằng mệnh đề UNION (hợp) trong SQL. Ví dụ, nếu cần mã của tất cả các sản phẩm là tủ lạnh hoặc đã được mua bởi khách hàng C4171, truy vấn như sau:

Kết quả của truy vấn trên là hợp của hai tập đã xác định. Một tập là tủ lạnh và tập khác là sản phẩm mua bởi khách hàng C4171.

Chúng ta đã biết tất cả các mệnh đề sử dụng với câu lệnh SELECT. Cú pháp hoàn chỉnh cho câu lệnh SELECT là:

## Câu lệnh INSERT (chèn)

Sau khi tạo bảng, chúng ta có thể đưa dữ liệu vào bằng câu lệnh INSERT. Câu lệnh INSERT có thể được dùng để chèn một hàng mới vào bảng. Ví dụ, nếu muốn chèn hàng vào bảng product (sản phẩm), câu lệnh SQL là:

Câu lệnh thêm một hàng vào bảng product với giá trị P9980, Amstrong, VCR, và 12990 lần lượt trong các cột prod-no, name, desc và price.

## Câu lệnh UPDATE (cập nhật)

SQL sử dụng câu lệnh UPDATE để cập nhật dữ liệu. Ví dụ, nếu chúng ta cần thay đổi giá của sản phẩm P4721 thành \$900, câu lệnh SQL là:

Nếu có hơn một hàng thỏa mãn mệnh đề WHERE, giá sẽ được đặt là \$900 cho tất cả các hàng đó. Nếu bỏ qua mệnh đề WHERE, tất cả các hàng trong bảng product sẽ cập nhật giá mới là \$900.

## Câu lệnh DELETE (xóa)

SQL sử dụng câu lệnh DELETE để xóa các hàng. Ví dụ, nếu cần xóa các hàng về khách hàng đang sống tại New York, câu lệnh SQL là:

## Hiện thị

Hiện thị là một bảng ảo, không tồn tại một cách vật lý. Nó khác bảng cơ sở, bởi bảng thực, tồn tại trong bộ nhớ vật lý. Chúng ta có thể coi hiện thị là một truy vấn SQL mà được lưu trữ lâu dài trong CSDL và được đặt tên. Tuy nhiên đối với người sử dụng, truy cập vào hiện thị cũng giống như truy cập vào bảng cơ sở. Hệ Quản trị CSDL tạo một ảo ảnh về bảng bằng cách gán tên cho hiện thị và lưu trữ nó trong CSDL. Các bảng nguồn của dữ liệu có thể được nhìn thấy qua hiện thị, và được tham chiếu như là bảng nguồn. Bảng sau thể hiện hai bảng nguồn và hiện thị tạo bởi các nguồn đó.

Bảng Employee (nhân viên)

Bảng Department (phòng)

Hiển thị Name – Dept (tên – phòng)

*Hiển thị với hai bảng nguồn*

### Định nghĩa một hiển thị

Hiển thị được định nghĩa sử dụng câu lệnh SQL là CREATE VIEW. Ví dụ, câu lệnh sau tạo hiển thị name – dept như trên:

Dữ liệu trong hiển thị name – dept lấy từ bảng Employee và Department. Như vậy, hai bảng này là bảng nguồn của hiển thị.

Khi câu lệnh CREATE VIEW được thực hiện, truy vấn theo sau mệnh đề Á (như) chưa được thực hiện. Thay vào đó, nó được lưu lại trong danh mục. Đối với người sử dụng, nó được nhìn thấy như là một bảng thực có tên là Name- Dept với các hàng và các cột. Trên thực tế, name –dept là một cửa sổ của bảng thực, bảng Employee và Department. Bất cứ thay đổi nào cho các bảng này cũng được quan sát qua hiển thị name-dept.

Nếu tên của các cột không được xác định khi định nghĩa một hiển thị, hiển thị sẽ lấy cột của bảng nguồn. Tuy nhiên, chúng ta cũng cần xác định tên của cột. Ví dụ, trong câu lệnh sau ko xác định tên cho cột thứ hai từ bảng nguồn, vì cột này xuất phát từ một hàm. Vì vậy, tên cột phải được xác định rõ ràng.

Khi đã định nghĩa một hiển thị, chúng ta có thể dùng nó trong câu lệnh SELECT như một bảng thực. Ví dụ,

Kết quả của truy vấn là:

### *Kết quả truy vấn*

Tên của hiển thị kèm theo mệnh đề FROM (từ) như là tên của bảng. Các cột của hiển thị đồng thời được tham chiếu như là cột của bảng.

## Hệ Quản trị CSDL xử lý hiển thị như thế nào?

Khi người sử dụng tham chiếu đến một hiển thị, hệ Quản trị CSDL tìm kiếm định nghĩa của hiển thị lưu trong CSDL. Sau đó, chuyển các yêu cầu của người sử dụng thành yêu cầu tương đương đối với bảng nguồn của hiển thị. Bằng cách này, Hệ Quản trị CSDL duy trì ảo ảnh của hiển thị. Ví dụ, truy vấn sau:

Được dịch bằng hệ Quản trị CSDL như sau:

Hệ Quản trị CSDL dịch truy đối với hiển thị name-dept sang truy vấn đối với bảng nguồn Employee.

Đối với những hiển thị phức tạp hơn, hệ Quản trị CSDL phải vật hóa hiển thị. Điều đó có nghĩa là Hệ Quản trị CSDL thực hiện truy vấn và lưu trữ kết quả của nó trong một bảng tạm thời. Bảng này bị xóa bỏ khi không còn cần thiết. Tuy nhiên, toàn bộ quá trình này không được nhìn thấy bởi người sử dụng. Một nhược điểm của nó là làm chậm quá trình xử lý.

## Thực thi hiển thị trong hệ Quản trị CSDL quan hệ

### Thao tác DML trên hiển thị

Quá trình dịch thực hiện bởi hệ Quản trị CSDL quan hệ là khá đơn giản đối với câu lệnh SELECT. Ví dụ, phức tạp gia tăng với các câu lệnh UPDATE, INSERT, và DELETE. Đây là do mọi hiển thị không thể cập nhật được. Ví dụ, quan sát hai hiển thị sau với cùng bảng cơ sở là Employee:

*Hiển thị A*

*Hiển thị B*

Nếu chèn thêm một hàng mới (1294, T1019) vào hiển thị A, hàng tương ứng được chèn vào bảng employee là (1294, NULL, T1019, NULL, NULL). Các cột name, basic, age không có trong hiển thị và giá trị của nó không thể đưa vào từ hiển thị. Theo mặc định, giá trị NULL sẽ được gán cho các cột này. Thao tác INSERT sẽ thành công nếu các cột name, basic, age không phải là NOT NULL khi tạo bảng employee.

Bây giờ, nếu chúng ta chèn hàng mới (T1019, 29) vào hiển thị B, hàng tương ứng chèn vào bảng employee là (NULL, NULL, T1019, NULL, 29). Trong trường hợp này, các cột emp-code, name và basic sẽ gán giá trị NULL. Thao tác INSERT trên hiển thị B sẽ thất bại vì khóa chính emp-code bị gán giá trị NULL. Khi bảng employee được tạo, khóa chính sẽ được xác định là NOT NULL.

Bây giờ, quan sát thao tác DELETE trên hiển thị A và B. Nếu hàng (1172, T1008) bị xóa khỏi hiển thị A, hàng thực tế bị xóa khỏi bảng employee là (1172, Larry Williams, T1008, 1900, 22). Nếu hàng (T1042, 29) bị xóa khỏi hiển thị B, hệ Quản trị CSDL sẽ xóa hàng tương ứng từ bảng employee. Tuy nhiên, Hệ Quản trị CSDL không có cách nào biết được hàng nào sẽ bị xóa không có khóa chính để nhận dạng các hàng. Điều tương tự xảy ra với thao tác UPDATE.

Vì thế, hiển thị, trên lý thuyết, có thể hoặc không thể cập nhật được. Trong các ví dụ trên, hiển thị A cập nhật được, trong khi hiển thị B không cập nhật được. Thuật ngữ cập nhật được trên lý thuyết được sử dụng vì một số hiển thị có thể cập nhật được trên giấy nhưng lại không cập nhật được trên thực tế. Hầu hết các sản phẩm Hệ Quản trị CSDL quan hệ không cho phép cập nhật từ hiển thị.

## Các loại hiển thị

Các loại hiển thị khác nhau là:

- Tập con theo cột
- Tập con theo hàng
- Tập con hàng – cột
- Nhóm
- Kết hợp



## Tập con theo cột

Hiện thị tập con theo cột gồm các hàng và một vài cột từ bảng nguồn. Ví dụ, câu lệnh CREATE VIEW tạo hiện thị tập con theo cột.

Hiện thị aa gồm cột emp-code và dept của bảng employee. Chú ý rằng định nghĩa các hiện thị không gồm mệnh đề WHERE. Vì thế, chúng gồm tất cả các hàng trong bảng.

## Tập con theo hàng

Hiện thị tập con theo hàng gồm tất cả các cột và một số hàng của bảng nguồn. một tập con theo hàng chia bảng theo chiều ngang. Vì thế, mỗi người sử dụng có thể nhìn một bảng kết hợp bằng những hàng cần thiết với mình. Ví dụ, câu lệnh CREATE VIEW tạo hiện thị tập con theo hàng.

Chú ý rằng mọi cột trong bảng employee đã bao gồm trong định nghĩa. Mệnh đề WHERE giới hạn các hàng sao cho tuổi của nhân viên nhỏ hơn 25.

## Tập con hàng – cột

Hiện thị tập con hàng cột gồm một số hàng và một số cột của bảng nguồn. Ví dụ, câu lệnh CREATE VIEW tạo hiện thị tập con hàng – cột

Chỉ có các cột được đặt tên rõ ràng trong danh sách SELECT và các gàng thỏa mãn điều kiện WHERE là có thể nhìn thấy qua hiện thị.

Tất cả các hiện thị ở đây (aa, ab, ac) là cập nhật được trên lý thuyết vì chúng có khóa chính là emp-code.

## Hiển thị theo nhóm

Hiển thị theo nhóm hiển thị dữ liệu là kết quả của một truy vấn gộp nhóm. Truy vấn này xác định định nghĩa hiển thị trong mệnh đề GROUP BY. Hiển thị gộp nhóm là dạng tóm tắt của bảng nguồn. Ví dụ, câu lệnh sau tạo hiển thị gộp nhóm:

Một hiển thị gộp nhóm không tương tác một – một với bảng nguồn của nó. Nó có thể dùng trong truy vấn như các hiển thị khác. Một hiển thị gộp nhóm không thể cập nhật được bởi vì mỗi hàng của hiển thị tương ứng với một nhóm hàng trong bảng nguồn. Hệ Quản trị CSDL không xác định được hàng nào để cập nhật.

## Hiển thị kết hợp

Hiển thị kết hợp được hình thành bằng hai hay ba truy vấn bảng trong định nghĩa hiển thị. Hiển thị kết hợp lấy dữ liệu từ hai hoặc ba bảng và thể hiện kết quả truy vấn bằng một bảng ảo duy nhất. Ví dụ, câu lệnh CREATE VIEW tạo hiển thị kết hợp.

Người sử dụng có thể trực tiếp truy vấn trên hiển thị. Tuy nhiên, hiển thị không thể cập nhật được bởi vì không có khóa chính của tất cả các bảng nguồn. Hệ Quản trị CSDL không thể nhận biết hàng nào để chèn, xóa hay cập nhật.

## Hiển thị dùng để làm gì?

Hiển thị là phần hết sức quan trọng của Hệ Quản trị CSDL bởi vì:

- Thông tin có giá trị: Hiển thị cho phép nhiều người dùng xem CSDL từ những góc độ khác nhau. Chỉ những phần thích hợp với người sử dụng là có thể nhìn thấy được.
- Hạn chế truy cập: Hiển thị hạn chế truy cập tới CSDL. Những người sử dụng khác nhau chỉ thấy những hàng hay cột nhất định của bảng. Ví dụ, chỉ những nhân viên được phép mới thấy được thông tin về trả lương của toàn bộ nhân viên trong công ty.
- Đơn giản hóa truy cập: Hiển thị đơn giản hóa việc truy cập. Một ví dụ là việc kết hợp ba bảng khi một người sử dụng không yêu cầu tất cả dữ liệu từ ba bảng này.

- Đơn giản hóa cấu trúc: Cấu trúc của dữ liệu được thể hiện đối với người sử dụng theo cách được đơn giản hóa. Điều này đơn giản hóa truy vấn của người sử dụng.
- Độc lập dữ liệu: Hiện thị giúp củng cố độc lập dữ liệu.

## Độc lập dữ liệu

Hệ thống có sự độc lập dữ liệu nếu người sử dụng và phần mềm sử dụng độc lập với cấu trúc logic của CSDL.

Ví dụ, một CSDL cần thiết lập lại cấu trúc. Một bảng trong CSDL phải được chia theo chiều dọc sao cho các cột thương được truy vấn được lưu ở những ổ nhanh hơn và các cột ít truy vấn lưu ở những ổ chậm hơn.

Ví dụ, nếu bảng Employee như bảng dưới phải chia thành hai bảng cơ sở:

E-X

*Bảng Employee được chia thành hai bảng*

Chú ý rằng bảng Employee là sự kết hợp giữa hai bảng e-x và e-y. chúng ta có thể tạo hiển thị sau chính xác như sự kết hợp:

Bất cứ phần mềm hay người sử dụng nào tham chiếu đến bảng nguồn đều có thể tham chiếu đến hiển thị Employee. Mọi thao tác SELECT đều sử dụng như trước.

Hiển thị Employee cập nhật được trên lý thuyết miễn là tồn tại tương tác một – một giữa e-x và e-y. Đối với mỗi giá trị của emp-code trong e-x, chỉ nên có một giá trị của emp-code trong e-y. Mọi thao tác cập nhật trên hiển thị employee phải được định nghĩa rõ ràng theo e-x và e-y.

## Bất lợi của hiển thị

Mặc dù mang lại nhiều lợi ích, hiển thị cũng có một số bất cập sau:

- **Hoạt động:** Hiển thị tạo ra hình dạng bên ngoài cho bảng. Hệ Quản trị CSDL vẫn phải dịch truy vấn cho hiển thị thành truy vấn cho bảng nguồn. Nếu hiển thị là một kết hợp phức tạp, truy vấn đơn giản cho hiển thị phải chuyển thành kết hợp phức tạp và sẽ tốn nhiều thời gian để hoàn thành. Vì thế, hiển thị làm ảnh hưởng đến hoạt động của Hệ Quản trị CSDL.
- **Hạn chế cập nhật:** hiển thị có thể hoặc không thể cập nhật được.

## Toàn vẹn dữ liệu

Toàn vẹn dữ liệu nghĩa là tính chính xác và hoàn thiện của dữ liệu trong CSDL. Khi nội dung của CSDL được sửa chữa, tính toàn vẹn dữ liệu có thể bị mất. Ví dụ,

- Thêm vào CSDL dữ liệu vô giá trị, như việc bán một sản phẩm không tồn tại.
- Sự thay đổi chỉ đượ thực hiện trên một bảng, nhưng chưa thực hiện trên các bảng khác.

Vai trò quan trọng của Hệ Quản trị CSDL quan hệ là đảm bảo tính toàn vẹn của dữ liệu. Để đảm bảo sự chính xác cũng như nhất quán của dữ liệu, Hệ Quản trị CSDL đưa ra một số ràng buộc về toàn vẹn dữ liệu. ràng buộc toàn vẹn dữ liệu hạn chế giá trị dữ liệu có thể chèn hoặc cập nhật vào CSDL. Một vài ràng buộc thường thấy là:

- Dữ liệu yêu cầu
- Kiểm tra tính hợp lệ
- Toàn vẹn thực thể
- Toàn vẹn tham chiếu
- Quy tắc công việc

Ràng buộc toàn vẹn là sự kiểm tra một cách hiệu quả sau khi thực hiện câu lệnh SQL. Nếu ràng buộc toàn vẹn không được thỏa mãn, Hệ Quản trị CSDL sẽ đưa ra thông báo lỗi.

### Dữ liệu yêu cầu

Đây là ràng buộc toàn vẹn dữ liệu đơn giant nhất, đòi hỏi cột chứa một giá trị không NULL. Trong quá trình tạo bảng, cột không chứa giá trị NULL gọi là NOT NULL. Xác định điều kiện NOT NULL cho cột buộc Hệ Quản trị CSDL thực hiện các ràng buộc sau:

- Mọi câu lệnh UPDATE hay INSERT phải gán giá trị không NULL.
- Chèn thêm hoặc cập nhật cột không NULL với giá trị NULL sẽ dẫn đến lỗi.

Một vấn đề với ràng buộc NOT NULL là nó chỉ có thể được xác định khi tạo bảng. Giá trị NULL sẽ không thể không được thừa nhận đối với cột của bảng tạo trước đó. Đây không phải là vấn đề lớn vì khi bảng được tạo, chúng ta phải xác định rõ ràng cột nào nhận giá trị NULL.

## Kiểm tra tính hợp lệ

Hệ quản trị CSDL kiểm tra xem dữ liệu bổ sung có phù hợp với miền của dữ liệu trong cột. Hệ quản trị CSDL đảm bảo chỉ dữ liệu thuộc kiểu đã xác định là có thể nhập được vào cột. Điều đó ngăn chặn việc nhập vào các dữ liệu và giá trị khác với kiểu dữ liệu và giá trị đã chọn. Ví dụ, nếu cột amount (số lượng) có kiểu dữ liệu là số thập phân, chúng ta sẽ không thể nhập được dữ liệu là xâu ký tự.

Hầu hết các sản phẩm thương mại đều không chỉ ra cách để hạn chế một cột với những giá trị dữ liệu cụ thể. Ví dụ, ngày thuê nhân viên 25/12 vẫn được chấp nhận mặc dù công ty nghỉ vào Giáng sinh. Tuy nhiên, một số sản phẩm Quản trị CSDL chỉ đưa ra kiểm tra tính hợp lệ như là một phần của gói sản phẩm, chứ không để trợ giúp cho CSDL. Qua những tính năng này, sự hợp lệ của dữ liệu nhập vào có thể được kiểm tra.

Để thay thế, chu trình kiểm tra tính hợp lệ trong chương trình viết bởi người sử dụng có thể kiểm tra tính hợp lệ của giá trị dữ liệu. Một số sản phẩm của Hệ Quản trị CSDL quan hệ cung cấp khả năng hợp lệ cho phép người sử dụng tạo ra các quy tắc quy định dữ liệu nào được phép nhập vào một cột nhất định. Hệ Quản trị CSDL kiểm tra mỗi lần thực hiện INSERT hoặc UPDATE. Nếu dữ liệu không thỏa mãn những quy tắc đó, hệ Quản trị CSDL sẽ loại bỏ thao tác đó.

## Toàn vẹn thực thể

Khóa chính là ký hiệu nhận dạng của hàng. Vì thế, khóa chính phải là duy nhất. Hệ Quản trị CSDL kiểm tra tính duy nhất của khóa chính trong từng câu lệnh INSERT hoặc UPDATE. Việc chèn hoặc cập nhật một hàng với khóa chính trùng lặp sẽ thất bại.

Phần lớn các sản phẩm thương mại xác định ràng buộc về tính duy nhất trong câu lệnh CREATE INDEX (tạo chỉ số). Sau khi một chỉ số duy nhất được tạo trên khóa chính, Hệ Quản trị CSDL đảm bảo các cột không chứa giá trị trùng lặp.

Có hai quy tắc ràng buộc thực thể đối với mô hình quan hệ:

- Quy tắc ràng buộc thực thể
- Quy tắc ràng buộc tham chiếu

Quy tắc toàn vẹn thực thể quy định rằng

Không có thành phần nào của khóa chính của quan hệ cơ sở được phép nhận giá trị NULL.

Quan hệ cơ sở tương tác với thực thể trong thực tế. Các thực thể này có thể nhận biết được theo nhiều cách. Khóa chính đảm nhiệm chức năng nhận biết duy nhất trong mô hình quan hệ. Nếu khóa chính là NULL, nghĩa là giá trị của cột đó không xác định được. Hơn nữa, điều đó có nghĩa là không có thông tin gì về hiển thị thực thể. Một thực thể không nhận dạng được thì không tồn tại.

## Toàn vẹn tham chiếu

Chúng ta đã học về khóa phụ và quan hệ cha-con mà nó tạo ra.

Bảng Offices (văn phòng)

Bảng Salesrep

### *Quan hệ cha-con tạo bởi khóa phụ*

Bảng Offices chứa thông tin về doanh số mục tiêu và thực tế của các chi nhánh công ty. Bảng Salesrep chứa thông tin về nhân viên trong phòng kinh doanh. Cột office tạo quan hệ cha-con giữa hai bảng. Ở đây, offices là bảng cha và salesrep là bảng con. Quan sát câu lệnh INSERT sau:

Thao tác chèn được thực hiện hoàn hảo trừ việc số office 10065 không tồn tại. Vì vậy, hàng mới chèn phá vỡ quan hệ cha-con giữa các bảng. Mọi giá trị nhập vào cột office trong bảng salesrep phải có giá trị tương ứng trong bảng offices. Đây là ràng buộc toàn vẹn tham chiếu.

Quy tắc toàn vẹn tham chiếu quy định rằng:

CSDL không được chứa những giá trị khóa phụ không khớp nhau.

Mọi giá trị hợp lệ của khóa phụ phải gắn với giá trị xuất hiện trong khóa chính. Toàn vẹn tham chiếu là phần quan trọng của mô hình quan hệ. Có 4 loại cập nhật CSDL có thể làm hỏng toàn vẹn tham chiếu của quan hệ cha-con trong CSDL:

- Chèn một hàng con mới: khi hàng mới chèn vào bảng con (salesrep), giá trị khóa phụ của nó (office) phải gắn với giá trị khóa chính (office) trong bảng cha (offices). Hàng con không thể tồn tại tách rời hàng cha. Mặt khác, chèn một hàng cha mới không gây vấn đề gì.
- Cập nhật khóa phụ trong hàng con: cũng giống với vấn đề nêu trên. Nếu giá trị cột office trong bảng salesrep được cập nhật, khóa phụ được sửa phải thích hợp với khóa chính.
- Xóa một hàng cha: nếu một hàng trong bảng cha có hơn một giá trị con bị xóa, hàng trong bảng con sẽ trở thành “dòng mồ côi”. Nói cách khác, giá trị khóa phụ sẽ không gắn với giá trị khóa chính trong bảng cha. Tuy nhiên, xóa một hàng trong bảng con không gây vấn đề gì.
- Cập nhật khóa chính trong bảng cha: tương tự như vấn đề trên. Khóa chính của bảng offices bị thay đổi, tất cả các hàng con hiện thời trở thành “mồ côi” vì khóa phụ không còn gắn với khóa chính.

Đặc tính toàn vẹn tham chiếu của Hệ Quản trị CSDL xử lý các vấn đề này. Trong trường hợp chèn một hàng con, Hệ Quản trị CSDL kiểm tra giá trị của cột khóa phụ với giá trị khóa chính. Nếu không thấy khớp nhau, Hệ Quản trị CSDL sẽ hủy bỏ câu lệnh INSERT và báo lỗi.

Cập nhật khóa phụ trong hàng con cũng được xử lý tương tự. Trong ví dụ, đại lý kinh doanh có thể được giao cho một cơ sở khác chỉ khi cơ sở đó tồn tại.



Khi cập nhật khóa chính trong bảng cha, nếu có hàng con nào gắn với giá trị khóa phụ, Hệ Quản trị CSDL sẽ hủy bỏ câu lệnh UPDATE, nghĩa là một số hiệu cơ sở không thể thay đổi trong bảng offices nếu một đại lý nào đó đang sử dụng cơ sở đó. Điều này ít khi xảy ra trong thực tế vì khóa chính ít khi thay đổi.

Nếu chúng ta đóng cửa cơ sở ở New York. Có ba cách giải quyết vấn đề:

- Ngăn việc xóa tên cơ sở cho đến khi các nhân viên được điều đến các cơ sở khác.
- Tự động xóa các bản ghi về nhân viên làm việc ở cơ sở New York.
- Đặt cột office trong bảng salesrep về NULL, chỉ ra rằng cơ sở đó không xác định.

Hầu hết các hệ thống sử dụng lựa chọn thứ nhất. Tuy vậy, có những hệ thống sử dụng cả ba lựa chọn qua khái niệm quy tắc xóa.

Đối với mỗi quan hệ cha-con tạo bởi khóa phụ, một quy tắc xóa tương ứng được xác định. Quy tắc xóa chỉ ra cách xử lý khi người sử dụng muốn xóa một hàng. Ba quy tắc xóa là:

- Quy tắc CHỌN: ngăn chặn việc xóa hàng cha nếu có hàng con tương thích. Xem lại ví dụ trước, quy tắc này ngăn chặn việc xóa một cơ sở khi vẫn còn nhân viên làm việc ở đó.
- Quy tắc XẾP TẦNG: Hệ Quản trị CSDL tự động xóa các hàng con nếu hàng cha tương ứng bị xóa. Như vậy, hàng con xếp bị xóa xếp tầng theo hàng cha. Trong ví dụ trên, mọi bản ghi về các đại lý sẽ bị xóa nếu cơ sở họ đang làm việc bị xóa.
- Quy tắc VÔ HIỆU HÓA: Hệ Quản trị CSDL tự động đặt khóa phụ về NULL trong hàng con nếu hàng cha bị xóa. Nếu cơ sở bị xóa, cơ sở của các đại lý bị xóa cơ sở được hiểu là không xác định.

Nếu không đặt ra các quy tắc xóa, CHỌN là quy tắc mặc định vì có ít nguy cơ phá hủy hệ thống nhất. Câu hỏi đặt ra là quy tắc xóa nào cần được áp dụng trên thực tế.

## Khóa phụ và giá trị NULL

Khác với khóa chính, khóa phụ được phép nhận giá trị NULL. Khóa phụ chứa giá trị NULL được hiểu là thỏa mãn ràng buộc toàn vẹn tham chiếu. Khi hàng con được cập nhật, giá trị sẽ được kiểm tra lại ở hàng cha.

Nếu áp dụng quy tắc VÔ HIỆU HÓA, khóa phụ sẽ nhận giá trị NULL. Trong một vài trường hợp, chúng ta không thể sử dụng giá trị NULL. Khi đó, các cột cần được đặt là NOT NULL khi tạo bảng.

## Quy tắc công việc

Nhiều toàn vẹn dữ liệu trong thực tế liên quan đến quy tắc hay luật lệ của công ty. Ví dụ, công ty không cho phép khách hàng đặt hàng quá hạn muracs tín dụng.

Hệ Quản trị CSDL chịu trách nhiệm lưu trữ và tổ chức dữ liệu. Tuy nhiên, để duy trì quy tắc công việc là nhiệm vụ của phần mềm ứng dụng truy cậpj vào CSDL. Có một số bất cập khi duy trì quy tắc công việc qua chương trình ứng dụng:

- Tồn nhiều công sức: nếu có nhiều chương trình xử lý việc cập nhật bảng salesrep, mỗi chương trình lại phải duy trì quy tắc liên quan đến cập nhật.
- Thiếu nhất quán: các chương trình viết bởi lập trình khác nhau thì duy trì quy tắc khác nhau.
- Vấn đề bảo trì: nếu quy tắc thay đổi, lập trình viên phải nhận diện chương trình nào xử lý thay đổi đó và sửa chữa cho thích hợp
- Phức tạp: nhiều quy tắc áp dụng cho việc cập nhật một bảng nên chỉ cập nhật đơn giản cũng có thể dẫn đến phức tạp.

Yêu cầu về duy trì quy tắc công việc không phải là hiếm gặp. Chương trình ứng dụng luôn phải quan tâm đến các ràng buộc này. Năm 1986, Sybase giới thiệu khái niệm kích hoạt coi quy tắc công việc như một phần của Hệ Quản trị CSDL quan hệ.

Đối với các thao tác gây ra thay đổi nội dung của bảng, người sử dụng có thể đưa ra một hành động tương ứng mà Hệ Quản trị CSDL có thể làm theo. Hành động này gọi là kích hoạt.

*Chú ý: để thiết kế kích hoạt, chúng ta cần xác định những điều kiện kích hoạt. Sau đó, chúng ta cần tạo hành động để thực hiện kích hoạt.*

Ba thao tác có thể kích hoạt là:

- Chèn một hàng
- Xóa một hàng
- Cập nhật một hàng

Ví dụ, khi đại lý bán hàng, cột sales sẽ tăng theo số lượng. số lượng hàng còn lại sẽ phải giảm dần trong một bảng khác.

Nếu một kích hoạt được xác định, kích hoạt phải được gọi ra ngay khi câu lệnh INSERT được thực hiện. Kích hoạt sau đó sẽ cập nhật các bảng đã nhận diện.

Kích hoạt đưa ra cách thay thế để thực thi ràng buộc toàn vẹn tham chiếu bởi khóa phụ và khóa chính.

# Tóm tắt

Trong bài học này, chúng ta đã học về:

- Ba kiểu quan hệ quan trọng nhất là:
  - Bảng cơ sở
  - Kết quả truy vấn
  - Hiển thị
- Câu lệnh SQL để tạo bảng là CREATE TABLE
- Có thể thay đổi bảng đang có bằng câu lệnh ALTER TABLE
- Có thể xóa bỏ một bảng bằng câu lệnh DROP TABLE
- Câu lệnh SQL để truy vấn bảng và sử dụng chúng là SELECT, INSERT, UPDATE và DELETE.
- Hiển thị là bảng ảo không tồn tại vật lý.
- Hiển thị định nghĩa bởi câu lệnh SQL CREATE VIEW
- Các loại hiển thị khác nhau là:
  - Tập con theo cột
  - Tập con theo hàng
  - Tập con hàng-cột
  - Nhóm
  - Kết hợp
- Hiển thị chỉ ra thông tin hợp lệ, giới hạn và đơn giản hóa truy cập, đơn giản hóa cấu trúc, độc lập dữ liệu logic.
- Toàn vẹn dữ liệu chỉ ra sự chính xác và hoàn chỉnh của dữ liệu trong CSDL
- Những ràng buộc toàn vẹn cơ bản là:
  - Dữ liệu yêu cầu
  - Kiểm tra sự hợp lệ
  - Toàn vẹn thực thể
  - Toàn vẹn tham chiếu
  - Quy tắc công việc.

## Bài 3B

# Thực hiện giao dịch và bảo mật

Trong phần này, chúng ta học về:

- Mô tả chu trình giao dịch
- Mô tả khôi phục giao dịch
- Định nghĩa nhật ký giao dịch
- Nhận diện vấn đề đồng thời
- Phân biệt các kiểu khóa
- Định nghĩa khóa chết
- Mô tả các khái niệm đồ án bảo mật
- Xây dựng đồ án bảo mật

## Xử lý giao dịch CSDL

Giao dịch là chuỗi gồm một hoặc nhiều câu lệnh SQL cùng nhau tạo nên một đơn vị logic của công việc. Mỗi câu lệnh trong giao dịch góp một phần trong nhiệm vụ và tất cả các câu lệnh cùng nhau hoàn thành nhiệm vụ. Giao dịch xảy ra khi dữ liệu được sửa chữa. Tất cả các câu lệnh trong giao dịch phải được thực hiện để đảm bảo tính nhất quán của CSDL.

Sau đây là ví dụ của một giao dịch thường gặp. Một sinh viên đặt mượn sách. Quy trình đặt mượn sách như sau:

Truy vấn bảng chứa thông tin về sách để kiểm tra xem sách còn sẵn có hay không.

Chèn thông tin đăng ký mượn sách vào bảng chứa thông tin mượn.

Cập nhật bảng chứa thông tin sách để trừ đi số lượng sách sẵn có bằng với số lượng sách được mượn.

Ba thao tác trên tạo nên một giao dịch logic. Khái niệm xử lý giao dịch là cần thiết cho chương trình cập nhật CSDL vì nó đảm bảo tính toàn vẹn cho CSDL.

Như một quy tắc, câu lệnh trong giao dịch được thực hiện như một đơn vị công việc trong CSDL. Khi đó, có thể là tất cả các câu lệnh đều thực hiện thành công hoặc không có câu lệnh nào được thực hiện cả.

Hệ Quản trị CSDL chịu trách nhiệm cho tính nhất quán của CSDL. Điều này là đúng kể cả khi chương trình ứng dụng bị hủy bỏ hoặc lỗi phần cứng xảy ra giữa chừng. Hệ Quản trị CSDL không bao giờ thực hiện giao dịch không hoàn toàn.

Theo ví dụ trên, nếu chương trình quản lý việc mượn sách bị gián đoạn ở bước thứ hai, CSDL sẽ phản ánh một giao dịch không hoàn toàn và thiếu nhất quán. Để đảm bảo tính nhất quán, Hệ Quản trị CSDL hủy bỏ toàn bộ các thay đổi vừa thực hiện. Như vậy, Hệ Quản trị CSDL đảm bảo rằng khi đang thực hiện cập nhật trong quá trình giao dịch mà xảy ra lỗi trước khi hoàn thành, các cập nhật sẽ bị hủy bỏ.

CSDL trước giao dịch

SELECT	SELECT	SELECT
UPDATE	UPDATE	UPDATE

Giao dịch

UPDATE	UPDATE	Lỗi phần cứng
DELETE	DELETE	
INSERT	Lỗi chương trình	

CSDL sau giao dịch

Hệ Quản trị CSDL	Hệ Quản trị CSDL
hủy bỏ mọi thay đổi	hủy bỏ mọi thay đổi

### *Khái niệm giao dịch SQL*

## Khôi phục giao dịch

Hai thao tác SQL hỗ trợ xử lý giao dịch là:

- Thao tác xác nhận

Điều này có nghĩa là một kết thúc giao dịch thành công. Nó khẳng định rằng Hệ Quản trị CSDL mà đơn vị logic công việc được sử dụng đã kết thúc thành công. CSDL được nhất quán và mọi cập nhật bởi quá trình giao dịch đã được xác nhận và có tác dụng lâu dài.

- Thao tác hủy bỏ

Điều này báo hiệu một kết thúc giao dịch không thành công, có nghĩa là một lỗi nào đó đã xảy ra và CSDL không nhất quán. Mọi cập nhật bởi giao dịch cho đến khi lỗi xảy ra sẽ bị hủy bỏ.

## Nhật ký giao dịch

Để hủy bỏ các thay đổi đối với CSDL, Hệ Quản trị CSDL duy trì một nhật ký giao dịch. Nhật ký giao dịch theo dấu mọi giao dịch thực hiện.

Khi người sử dụng thực hiện câu lệnh SQL để thay đổi CSDL, Hệ Quản trị CSDL ghi lại một bản ghi trong nhật ký giao dịch. Nhật ký giao dịch cho thấy hai bản sao của mỗi hàng chịu tác động của câu lệnh. Một bản sao chỉ ra hàng trước khi câu lệnh thực hiện và bản sao còn lại chỉ ra hàng với các thay đổi. Hệ Quản trị CSDL chỉ thay đổi các hàng sau khi nhật ký đã được viết. Nếu người sử dụng đưa ra một lệnh xác nhận, sự kết thúc giao dịch sẽ ghi lại trong nhật ký. Nếu người sử dụng đưa ra một lệnh hủy bỏ, Hệ Quản trị CSDL sẽ kiểm tra nhật ký để tìm ra bản sao của hàng trước khi thay đổi. Bằng cách này, Hệ Quản trị CSDL khôi phục các CSDL về trạng thái ban đầu, hủy bỏ các thay đổi trong quá trình giao dịch. Hình sau minh họa quá trình làm việc của nhật ký giao dịch trên Hệ Quản trị CSDL quan hệ.



Chuỗi câu lệnh SQL

Thời gian: 12:01

UPDATE

Nhật ký giao dịch

Vị trí hàng:

Trước:

Sau: (rỗng)

Thời gian: 12:04

DELETE

Nhật ký giao dịch

Vị trí hàng:

Trước:

Sau: (rỗng)

Thời gian: 12:05

INSERT

Nhật ký giao dịch

Vị trí hàng:

Trước:

Sau: (rỗng)

Thời gian: 12: 06

COMMIT

Nhật ký giao dịch

Vị trí hàng:

Trước:

Sau:

Giao dịch được xác nhận

*Nhật ký giao dịch*

## Vấn đề xảy ra cùng lúc

Xử lý giao dịch rất phức tạp đối với hệ thống nhiều người sử dụng. Hệ Quản trị CSDL cần được phục hồi sau những hư hỏng hoặc lỗi và đảm bảo các thao tác của người sử dụng không ảnh hưởng lẫn nhau. Mỗi người sử dụng phải có khả năng truy cập CSDL mà không làm ảnh hưởng đến công việc của người khác. Có ba vấn đề có thể xảy ra đối với môi trường nhiều người sử dụng:

- Cập nhật mất dữ liệu
- Vấn đề đọc dữ liệu chưa hoàn tất
- Dữ liệu không nhất quán

## Vấn đề cập nhật mất dữ liệu

Xảy ra khi một giao dịch cập nhật dữ liệu vào cơ sở dữ liệu thành công, nhưng lại ghi đè lên dữ liệu của giao dịch khác.

Xảy ra khi: 2 giao dịch đang đọc mẫu tin dữ liệu, sau đó giao dịch 1 ghi dữ liệu của bản ghi, giao dịch 2 cũng ghi kết quả chỉ có giao dịch 2 được cập nhật.

## Vấn đề đọc dữ liệu chưa hoàn tất

Ví dụ, nếu có 200 cuốn sách. Tại 1200 giờ, A nhận đặt 150 cuốn. A kiểm tra bảng sách, thấy còn 200 cuốn, và cập nhật còn lại 50 cuốn. Giao dịch không được xác nhận. Tại 1205 giờ, B nhận đặt 100 cuốn sách. B kiểm tra bảng sách và thấy chỉ còn 50 cuốn. Đơn đặt mượn không được chấp nhận. Tại 1210 giờ, A hủy bỏ giao dịch vì khách hàng đã hủy đơn đặt mượn.

Khi B được phép xem dữ liệu chưa hoàn tất, đơn đặt hàng từ khách hàng thứ hai đã bị từ chối. Trong trường hợp khách hàng của B quyết định lấy 50 cuốn sách còn lại, B cập nhật dữ liệu là còn lại 0 cuốn. Tuy nhiên, khi A ra lệnh hủy bỏ, số sách còn lại sẽ đưa về 200 cuốn, mặc dù 50 cuốn đã được xác nhận cho khách hàng của B.

Như vậy, 1 giao dịch đọc DL của giao dịch khác chưa được hoàn thành mà DL đó có thể bị thay đổi không chính xác hoặc không tồn tại.

## Vấn đề dữ liệu không nhất quán.

Coi rằng A sử dụng chương trình báo cáo về đơn đặt mượn sách. Nếu chương trình lấy đơn số 100066, in ra và bổ sung giá trị của đơn đặt mượn tổng số là \$100. Trong khi đó, khách hàng hủy đơn đặt trước và thay bằng đơn mới trị giá \$120. thay đổi được thực hiện thành công. Vài giây sau, chương trình nhận được đơn mới, in ra và bổ sung tổng số \$120.

Vì thế, chương trình báo cáo sẽ gồm cả hai đơn đặt hàng. Điều này khác với vấn đề đọc dữ liệu chưa hoàn tất. Ở đây, chương trình báo cáo đã xử lý cả hai dữ liệu trước và sau khi thay đổi., gây ra kết quả sai.

Theo ví dụ trên, khi nhiều người cùng sử dụng CSDL sẽ dẫn đến sai lệch dữ liệu. Vì vậy, hai giao dịch sẽ được thực hiện đồng thời, chúng ta phải có phương pháp để Hệ Quản trị CSDL đảm bảo kết quả là như nhau bất kể ai truy cập CSDL trước.

Phương pháp phổ biến nhất là sử dụng khóa.

## Khóa

Khi giao dịch cần được đảm bảo rằng đối tượng trong CSDL không thay đổi bất ngờ, chúng ta cần đến khóa. Tác dụng của khóa là ngăn tất cả các giao dịch khác thay đổi đối tượng CSDL. Khi đó, giao dịch thứ nhất sẽ xử lý đối tượng sao cho đối tượng ổn định chừng nào còn được yêu cầu.

Giả sử giao dịch A truy cập vào CSDL. Hệ Quản trị CSDL khóa tất cả các phần dữ liệu mà giao dịch đã thay đổi. Giao dịch B cũng bắt đầu xử lý cùng một lúc. Hệ Quản trị CSDL cũng khóa các phần dữ liệu mà nó truy cập. Nếu giao dịch B truy cập vào phần CSDL khóa cho giao dịch A, Hệ Quản trị CSDL sẽ chặn truy cập và giao dịch B phải chờ đến khi dữ liệu được bỏ khóa. Hệ Quản trị CSDL sẽ bỏ khóa giao dịch A khi lệnh COMMIT hoặc ROLL BACK được thực hiện.

Sau đó, Hệ Quản trị CSDL sẽ cho phép giao dịch B xử lý công việc. Giao dịch B cũng tương tự khóa một phần CSDL, bảo vệ chúng trước các giao dịch khác.

## Các chế độ khóa

Có chế độ khóa cơ bản là:

- Khóa chia sẻ
- Khóa độc quyền

Khóa chia sẻ là loại khóa chia sẻ tài nguyên cho phép đọc dữ liệu, không cho phép thay đổi bất kỳ thuộc tính nào của tài nguyên.

Khóa độc quyền không tương thích với các loại khóa khác. Khóa này ngăn ngừa 2 người sử dụng cùng cập nhật, xóa, thêm dữ liệu.

Nếu giao dịch A giữ khóa độc quyền với một số dữ liệu, giao dịch B yêu cầu một trong hai loại khóa với cùng dữ liệu đó, giao dịch B phải chờ cho đến khi khóa được giải phóng.

Nếu giao dịch A giữ khóa chia sẻ với dữ liệu, giao dịch B yêu cầu khóa độc quyền với cùng dữ liệu đó, giao dịch B phải chờ đến khi giao dịch A được giải phóng.

Nếu giao dịch A giữ khóa chia sẻ và giao dịch B yêu cầu khóa chia sẻ, yêu cầu của giao dịch B sẽ được thông qua.

Như vậy, kỹ thuật sử dụng khóa giải quyết cả ba vấn đề xảy ra đồng thời: ngăn chặn cập nhật mất dữ liệu, đọc dữ liệu chưa hoàn tất và dữ liệu thiếu nhất quán. Tuy nhiên khóa cũng đưa ra một vấn đề khác, một giao dịch yêu cầu tài nguyên, nếu tài nguyên không sẵn dùng tại thời điểm đó, giao dịch đi vào trạng thái chờ. Quá trình chờ có thể không bao giờ chuyển trạng thái trở lại vì tài nguyên chúng yêu cầu bị giữ bởi những giao dịch đang chờ khác.

## Khóa chết

Khóa chết là trường hợp hai hay nhiều giao dịch cùng ở trạng thái chờ, mỗi giao dịch phải chờ lần lượt các giao dịch khác giải phóng rồi mới được xử lý.

Xem xét trường hợp chương trình A cập nhật bảng đặt hàng, khóa một phần của bảng. Chương trình B cập nhật bảng sản phẩm, khóa một phần bảng đó. Nếu bây giờ A cần cập nhật bảng sản phẩm, B cần cập nhật bảng đặt hàng. Mỗi chương trình phải muốn cập nhật phần bảng không bị khóa bởi chương trình kia thì phải chờ cho đến khi chương trình kia xác nhận giao dịch và giải phóng dữ liệu.

Đó là ví dụ giữa hai giao dịch. Nếu có 4 hoặc 5 giao dịch liên quan đến khóa chết, sự phức tạp còn hơn thế nhiều.

Để xử lý giao dịch trong khóa chết, Hệ Quản trị CSDL tuân tự kiểm tra các giao dịch. Khi phát hiện khóa chết, Hệ Quản trị CSDL chọn giao dịch mà khóa chết mất tác dụng và hủy bỏ giao dịch đó. Nó sẽ giải phóng dữ liệu và cho phép các giao dịch tiếp tục được xử lý.

Giao dịch thất bại nhận tin báo lỗi, thông báo rằng khóa chết đã mất và giao dịch bị hủy bỏ. Phương pháp này ngăn việc xảy ra khóa chết vĩnh viễn hoặc sai lệch dữ liệu. Nếu sự mất khóa chết xảy ra trong cSQL tương tác, câu lệnh SQL sẽ được gỡ lại. Nếu khóa chết xảy ra ở chương trình ứng dụng, chương trình ứng dụng sẽ xử lý tình huống.

## Bảo mật CSDL

Sữ liệu của một công ty là tài sản quan trọng nhất. Vậy nên duy trì bảo mật dữ liệu là mối quan tâm hàng đầu đối với mỗi công ty. Yêu cầu bảo mật CSDL rất đa dạng. Ví dụ:

- Dữ liệu chỉ cho phép cập nhật bởi một số người dùng nhất định.
- Người dùng được chọn được phép cập nhật dữ liệu trong khi người khác chỉ được phép đọc và truy hồi dữ liệu.
- Đối với một số bảng, truy cập nên được giới hạn bởi nguyên tắc cột – cột.
- Người sử dụng được chọn được phép truy cập vào bảng qua các chương trình ứng dụng.

Duy trì các ràng buộc bảo mật là trách nhiệm của Hệ Quản trị CSDL.

### Khái niệm trong sơ đồ bảo mật.

Sơ đồ bảo mật Hệ Quản trị CSDL gồm ba khái niệm:

- Người sử dụng
- Đối tượng CSDL
- Quyền ưu tiên.

**Người sử dụng** truy cập vào CSDL. Mỗi khi Hệ Quản trị CSDL chèn, cập nhật, xóa, chọn một hàng, hoặc thực hiện các thao tác khác là thay cho quyền của người sử dụng. Hệ Quản trị CSDL cho phép hay ngăn chặn các tác vụ trên CSDL, tùy thuộc vào ai đang đặt yêu cầu. Như đã nêu ở trên, một số người sử dụng không có quyền thực hiện một số thao tác.

**Đối tượng CSDL** là các phần mà bảo mật được áp dụng. Các ràng buộc bảo mật thường áp dụng cho bảng và hiển thị. Tuy nhiên, bảo mật có thể áp dụng cho các đối tượng khác như chương trình ứng dụng.

**Quyền ưu tiên** dành cho những người được đặc biệt được thao tác trên phần đối tượng CSDL cụ thể.

## Người sử dụng

Người sử dụng được cấp một mã sử dụng để nhận diện với Hệ Quản trị CSDL. Mã sử dụng được cấp bởi Quản trị CSDL. Mỗi mã sử dụng quyết định câu lệnh được thực hiện hay chặc bởi Hệ Quản trị CSDL. Nếu mã sử dụng được phép thực hiện thao tác, Hệ Quản trị CSDL sẽ cho phép người sử dụng thực hiện thao tác, nếu không thì nó sẽ chặn lại.

Trong hầu hết các Hệ Quản trị CSDL, cả mã người sử dụng và mật khẩu đều phải chỉ rõ. Hệ Quản trị CSDL kiểm tra mật khẩu để xác định quyền sử dụng của người dùng. Đây là cách xác nhận.

Dựa vào tình huống:

- Mỗi người sử dụng được cấp một mã sử dụng riêng
- Một mã sử dụng chia sẻ bởi hai hay nhiều người.

Trong một CSDL lớn, có những nhóm người với cùng nhu cầu. Trong mỗi nhóm, tất cả những người sử dụng có nhu cầu như nhau về truy cập CSDL. Để tiện dụng, một mã sử dụng sẽ được cấp cho tất cả nhóm. Ví dụ, tất cả phòng xử lý đơn đặt hàng là một nhóm, cùng cần truy cập một dữ liệu. Vì vậy, họ chia sẻ một mã sử dụng và có quyền truy cập như nhau vào đối tượng CSDL.

## Đối tượng CSDL

Ràng buộc bảo mật áp dụng cho các đối tượng cụ thể trong CSDL. Các đối tượng đó là:

- Bảng
- Hiển thị

Mỗi bảng hoặc hiển thị có thể được bảo vệ đơn lẻ. Truy cập vào bảng hay hiển thị có thể cho phép hay ngăn chặn tùy từng đối tượng.

## Quyền ưu tiên

Quyền ưu tiên là tập các hành động mà người sử dụng có thể thực hiện trên đối tượng CSDL. Bốn quyền ưu tiên cho bảng và hiển thị là:

- Quyền SELECT
- Quyền INSERT
- Quyền DELETE
- Quyền UPDATE

Quyền SELECT cho phép truy hồi dữ liệu từ các bảng hoặc hiển thị. Nếu có quyền này, chúng ta có thể xác định tên của bảng hoặc hiển thị với mệnh đề FROM hoặc câu lệnh SELECT.

Quyền INSERT cho phép chèn dữ liệu vào bảng hoặc hiển thị. Nếu có quyền này, chúng ta có thể xác định tên bảng hoặc hiển thị với mệnh đề INTO và câu lệnh INSERT.

Quyền DELETE cho phép xóa dữ liệu trong bảng hoặc hiển thị. Nếu có quyền này, chúng ta có thể xác định tên bảng hoặc hiển thị với mệnh đề FROM và câu lệnh DELETE.

Quyền UPDATE cho phép thay đổi hàng trong bảng hoặc hiển thị. Nếu có quyền này, chúng ta có thể xác định tên bảng hoặc hiển thị theo bảng mục tiêu trong câu lệnh UPDATE.

Chủ của một bảng có thể là người tạo bảng, có tất cả các quyền với bảng, có quyền cấp phép truy cập cho người khác. Chủ của hiển thị, chỉ có quyền SELECT. Để có các quyền khác, phải được cấp phép sử dụng quyền trên bảng nguồn dữ liệu.

## Xây dựng sơ đồ bảo mật

Sơ đồ bảo mật CSDL được xây dựng nhờ các câu lệnh SQL:

- GRANT (cấp phép)
- REVOKE (hủy bỏ)

Để thực hiện thao tác trên CSDL, một người sử dụng phải có quyền với thao tác đó. Ví dụ, ngay cả câu lệnh đơn giản như sau:

Người sử dụng có quyền SELECT trên bảng X.

Câu lệnh GRANT được dùng để cấp quyền cho người sử dụng. Quyền có thể bị hủy bỏ qua câu lệnh REVOKE.



## Cấp quyền ưu tiên.

Câu lệnh GRANT được dùng để cấp quyền cho người sử dụng, xác định các điều sau:

- Một danh sách cụ thể các quyền được cấp
- Tên của bảng để áp dụng quyền ưu tiên
- Mã người dùng mà quyền ưu tiên cấp cho.

Ví dụ,

Trong ví dụ này, quyền được cấp là INSERT hoặc DELETE. Đối tượng là Product. Người sử dụng được cấp quyền là Nancy. Như vậy, tất cả các vấn đề của bảo mật – người sử dụng, đối tượng CSDL, quyền ưu tiên được xác định với câu lệnh GRANT.

Tất cả các quyền sử dụng có thể cấp cho một người qua câu lệnh:

Thay vì cung cấp cho từng người một, có thể sửa câu lệnh như sau

Từ PUBLIC (công chúng) được dùng để cấp quyền cho tất cả các người sử dụng CSDL. Câu lệnh SQL nói trên cấp quyền SELECT cho bảng Product cho tất cả các người sử dụng có phép.

Đôi khi, đối với bảng , chúng ta không muốn người dùng truy cập tất cả các cột. Quyền ưu tiên có thể được cấp cho những cột riêng của bảng hoặc hiển thị. Ví dụ, câu lệnh SQL sau:

Nancy có chỉ quyền cập nhật cho cột Price trong bảng Product.

## Thông qua quyền ưu tiên

Nếu đã tạo được đối tượng CSDL, bạn trở thành chủ, là người duy nhất được cấp quyền ưu tiên cho người khác. Người sử dụng mà bạn thông qua quyền ưu tiên được sử dụng đối tượng CSDL. Tuy nhiên, học không thể thông qua quyền ưu tiên này cho người khác. Như vậy bạn có thể kiểm soát được đối tượng của mình.

Bạn có thể cấp quyền ưu tiên cho người khác và cho phép người đó cấp quyền cho người khác nữa. Trong SQL, điều này thực hiện qua WITH GRANT OPTION của câu lệnh GRANT. Chỉ những quyền trong câu lệnh GRANT là được phép thông qua.

Sarah có thể cấp quyền cập nhật cho người khác với bảng Product.

### *Thông qua quyền ưu tiên*

Bây giờ, Sarah có thể đưa ra câu lệnh sau:

## Hủy bỏ quyền ưu tiên

Quyền ưu tiên bị hủy bỏ qua câu lệnh REVOKE:

Câu lệnh hủy bỏ hoặc cập nhật bảng Product cho Nancy hoặc Ivring. Như mọi quyền có thể cấp bởi một câu lệnh, tất cả các quyền có thể bị hủy bỏ chỉ bằng một câu lệnh. Tương tự, quyền có thể hủy bỏ từ tất cả các người sử dụng bằng một câu lệnh. Câu lệnh SQL sau minh họa tất cả các trường hợp:

Nhắc lại ví dụ trước, nếu quyền của Sarah bị hủy bỏ, quyền của Ivring cũng tự động bị hủy bỏ.

### Hiển thị và bảo mật

Hiển thị đóng vai trò quan trọng trong bảo mật. Hiển thị cho phép kiểm soát đối với những dữ liệu nhìn thấy được đối với người sử dụng. Nếu hiển thị là một vài cột, người sử dụng chỉ truy cập được vào bảng với những cột trong hiển thị.

Xem hình minh họa sau đây:

*Hiển thị và bảo mật*

*Chú ý: để đảm bảo kiểm soát truy cập, chúng ta có thể tạo các hiển thị khác nhau cho từng người dùng khác nhau.*

## Tóm tắt

Trong bài này, chúng ta đã học:

- Giao dịch là chuỗi các câu lệnh SQL cùng nhau tạo thành một đơn vị logic công việc
- Tất cả các câu lệnh đều phục vụ cho một giao dịch phải được thực hiện thống nhất trong CSDL
- Theo quy tắc, câu lệnh trong giao dịch phải được thực hiện như một đơn vị duy nhất của công việc
- Hệ Quản trị CSDL chịu trách nhiệm đảm bảo nhất quán cho CSDL. Điều đó là đúng kể cả khi chương trình ứng dụng hoặc phần cứng bị lỗi giữa chừng giao dịch.
- Để đảm bảo nhất quán, Hệ Quản trị CSDL hủy bỏ mọi thay đổi thực hiện bởi giao dịch chưa hoàn thành
- Hai thao tác SQL hỗ trợ xử lý giao dịch là:
  - Thao tác COMMIT nghĩa là kết thúc giao dịch thành công
  - Thao tác ROLLBACK nghĩa là kết thúc giao dịch thất bại.
- Để hủy bỏ các thay đổi, Hệ Quản trị CSDL duy trì nhật ký giao dịch. Nhật ký giao dịch theo dõi tất cả các giao dịch thực hiện.
- Xử lý giao dịch rất phức tạp nếu có nhiều người sử dụng. Có 3 vấn đề có thể xảy ra là:
  - Cập nhật mất dữ liệu
  - Vấn đề đọc dữ liệu chưa hoàn tất
  - Vấn đề dữ liệu thiếu nhất quán
- Cách phổ biến nhất để kiểm soát các vấn đề trên là sử dụng khóa
- Để đảm bảo rằng đối tượng CSDL nhất định không thay đổi bất ngờ, giao dịch cần đến khóa.
- Kỹ thuật sử dụng khóa giải quyết cả ba vấn đề. Tuy nhiên, nó cũng buộc giao dịch phải chờ lâu khi phần dữ liệu nó cần bị khóa bởi giao dịch khác.
- Khóa chết là trường hợp hai hay nhiều giao dịch cùng ở trạng thái chờ, mỗi giao dịch chờ giao dịch khác giải phóng khóa để xử lý.

- Yêu cầu bảo mật của CSDL là rất đa dạng. Một số yêu cầu bảo mật là:
  - Dữ liệu chỉ được cập nhật bởi một số người dùng chỉ định
  - Người dùng chỉ định được phép cập nhật dữ liệu khi người khác chỉ được xem và truy hồi dữ liệu
  - Đối với một số bảng, truy cập bị giới hạn bởi nguyên tắc cột – cột
  - Người dùng chỉ định chỉ được phép truy cập bảng qua chương trình ứng dụng
  
- Sơ đồ bảo mật cho Hệ Quản trị CSDL gồm có:
  - Người sử dụng
  - Đối tượng CSDL
  - Quyền ưu tiên
- Các đối tượng có thể áp dụng bảo mật là:
  - Bảng
  - Hiển thị
  
- Bốn quyền ưu tiên có thể xác định với bảng và hiển thị là:
  - Quyền SELECT
  - Quyền INSERT
  - Quyền DELETE
  - Quyền UPDATE
  
- Sơ đồ bảo mật cho CSDL có thể được xây dựng nhờ câu lệnh SQL:
  - GRANT
  - REVOKE
  
- Câu lệnh GRANT dùng để cấp quyền sử dụng cho người dùng. Quyền sử dụng có thể bị hủy bỏ qua câu lệnh REVOKE.

Bài 3C:

# Phối hợp

Chú ý: đăng ký trên [www.niitstudent.com](http://www.niitstudent.com) là bắt buộc.

## Byte kiến thức

Trong phần này, chúng ta nghiên cứu về:

- 12 quy tắc của CODD
- Mục lục
- Phục hồi từ khóa chết
- Phục hồi CSDL

12 quy tắc của CODD:

Tiến sĩ E.F.Codd đã đưa ra 12 quy tắc mà CSDL phải tuân theo nếu được xem là có quan hệ. Các quy tắc này xuất phát từ nghiên cứu của Codd về mô hình quan hệ. Không có Hệ Quản trị CSDL quan hệ nào thỏa mãn hoàn toàn các quy tắc của Codd.

Các quy tắc phát triển từ một quy tắc cơ sở- quy tắc 0:

Để một hệ thống có QUAN HỆ, CSDL, hệ QUẢN TRỊ phải sử dụng những đực trung QUAN HỆ để QUẢN LÝ CSDL.

12 quy tắc như sau:

1. Quy tắc thông tin: bắt buộc mọi thông tin trong CSDLQH phải được trình bày ở dạng tập hợp giá trị được lưu trong bảng.
2. Quy tắc truy cập bảo đảm: bắt buộc mỗi loại dữ liệu trong CSDLQH phải được truy cập bằng tên của bảng, khóa chính và tên của cột. Nói đơn giản là mọi dữ liệu được lưu ở dạng bảng và nếu bạn biết được tên bảng, khóa chính và tên cột có chứa dữ liệu bạn cần thì có thể lấy thông tin cần thiết.
3. Xử lý hệ thống các giá trị NUULL: Hệ Quản trị CSDL phải hỗ trợ giá trị NULL để thể hiện thông tin mất mát hoặc không sử dụng được. Chúng phải rõ ràng từ 0 đến vô cùng. Giá trị NULL phải độc lập dữ liệu. Ví dụ, giá trị NULL cho tất cả các kiểu dữ liệu là giống nhau.
4. Danh mục trực tuyến đối với mô hình quan hệ: Trong phần trước, chúng ta đã bàn về bảng hệ thống hoặc danh mục hệ thống. Danh mục hệ thống là tập hợp của các bảng Hệ Quản trị CSDL duy trì để sử dụng riêng. Các bảng này lưu những cấu trúc mô tả của CSDL. Các bảng này đồng thời được tạo, làm chủ và duy trì bởi Hệ Quản trị CSDL. Chúng có thể truy cập bởi người sử dụng theo cách của bảng thông thường, phụ thuộc vào quyền ưu tiên của người sử dụng. bảng hệ thống chỉ có thể đọc.



5. Quy tắc ngôn ngữ dữ liệu thay thế toàn diện: quy tắc chỉ ra rằng hệ thống phải hỗ trợ ít nhất các hàm sau:
  - a. Định nghĩa dữ liệu
  - b. Định nghĩa hiển thị
  - c. Thao tác xử lý dữ liệu
  - d. Ràng buộc toàn vẹn và bảo mật
  - e. Thao tác quản lý giao dịch
6. Quy tắc cập nhật hiển thị: mọi hiển thị có thể cập nhật trên lý thuyết phải cập nhật được trên hệ thống
7. Chèn, cập nhật, xóa bậc cao: quy tắc chỉ ra rằng các hàng phải được xử lý như tập hợp trong các thao tác chèn, cập nhật, xóa. Cũng giống như SELECT có thể xử lý tập hợp các hàng, các thao tác khác cũng cần sửa chữa dữ liệu với tập hợp chứa không phải chỉ một hàng.

Quy tắc ngăn không cho Hệ Quản trị CSDL hỗ trợ thay đổi một hàng trong một thời điểm trên CSDL. Vì thế, Hệ Quản trị CSDL phải cho phép nhiều hàng cùng được cập nhật.

8. Độc lập dữ liệu vật lý: chương trình ứng dụng phải giữ không suy giảm khi thay đổi tiến hành trên bộ nhớ hoặc phương pháp truy cập.
9. Độc lập dữ liệu logic: thay đổi không làm ảnh hưởng đến khả năng làm việc với CSDL
10. Độc lập toàn vẹn: ràng buộc toàn vẹn phải được lưu trong danh mục hệ thống. Khái niệm toàn vẹn dữ liệu không yêu cầu giải thích gì thêm.
11. Độc lập phân phối: CSDL phải xử lý các dữ liệu phân phối trên các hệ máy tính khác.
12. Quy tắc không có phiên bản thay thế: các bậc khác nhau của ngôn ngữ không bỏ qua các quy tắc và ràng buộc toàn vẹn. Hệ Quản trị CSDL đảm bảo không có bậc khác có thể bỏ qua tính toàn vẹn của CSDL.

## Mục lục

Truy vấn CSDL chỉ tham chiếu một phần của bản ghi trong một tập tin. Ví dụ, truy vấn “Tìm tất cả các khách hàng ở Boston” tham chiếu đến phần nhỏ của tập tin chứa dữ liệu khách hàng. Nếu hệ thống phải kiểm tra điều kiện cho mỗi bản ghi, tính hiệu quả của của hệ thống sẽ bị giảm xuống. Vì vậy, chúng ta cần có phương pháp để truy cập dữ liệu nhanh hơn. Để trực tiếp truy cập dữ liệu, chúng ta cần thiết kế bổ sung cấu trúc có thể dẫn tới tập tin. Cấu trúc này là mục lục, giống như danh mục trong thư viện.

Có hai loại mục lục:

- Bảng mục lục chính: chứa các chỉ dẫn trực tiếp đến tập tin
- Bảng mục lục thứ cấp: không chỉ dẫn trực tiếp đến tập tin. Thay vào đó, chỉ đến các vùng chứa chứa chỉ dẫn trực tiếp đến tập tin. Mục lục thứ cấp cải thiện hoạt động của truy vấn nhiều hơn là mục lục chính. Tuy nhiên nó cũng có những nhược điểm, làm phức tạp quá trình thay đổi CSDL.

Cũng như bảng cơ sở, mục lục được tạo và gỡ bỏ bằng câu lệnh SQL DDL. Trong hệ thống quan hệ, người sử dụng không quyết định được nên sử dụng mục lục hay không. Bộ xử lý truy vấn quyết định điều này.

Câu lệnh SQL tạo mục lục trên bảng customer:

Câu lệnh trên tạo mục lục x, dựa vào giá trị tăng dần của cột cust-no. Để tạo mục lục giảm dần, chúng ta thay ACS bằng DESC.

Chúng ta cũng có thể xác định lựa chọn duy nhất cho câu lệnh CREATE INDEX (tạo mục lục). Ví dụ,

Bằng xác định lựa chọn duy nhất, chúng ta đảm bảo rằng không có hai bản ghi nào của bảng cơ sở lấy cùng một giá trị trong cột thư mục hoặc tập hợp cột. Đây là caches tốt nhất để đảm bảo không lặp hàng trong bảng. Nếu một bảng đã vi phạm ràng buộc tính duy nhất, việc tạo mục lục sẽ thất bại.

Câu lệnh để xóa bỏ mục lục là:

Mô tả mục lục được lưu trong CSDL hệ thống. khi một mục lục bị xóa bỏ, mô tả của nó cũng bị gỡ bỏ khỏi danh mục hệ thống. Khi bảng bị xóa bỏ, tất cả các mục lục tạo trên bảng đó cũng bị xóa bỏ.

## Khôi phục từ khóa chết

Một khi khóa chết được xác định là tồn tại, hệ thống phải được phục hồi. Để làm được điều này, một vài giao dịch phải bị hủy bỏ để phá khóa chết. Khi tiến hành hủy bỏ, cần chú ý:

- Lựa chọn giao dịch: đầu tiên cần xác định giao dịch nào phải bị hủy bỏ. Giao dịch dẫn đến thiệt hại ít nhất nên được chọn.
  - Đã có bao nhiêu thao tác thực hiện bởi giao dịch và còn bao nhiêu để giao dịch hoàn thành?
  - Có bao nhiêu dữ liệu sử dụng bởi giao dịch?
  - Cần bao nhiêu dữ liệu nữa thì giao dịch hoàn thành?
  - Có bao nhiêu giao dịch phải được kèm theo trong hủy bỏ?

- Hủy bỏ: sau khi quyết định giao dịch nào bị hủy bỏ, chúng ta cần quyết định giao dịch bị hủy bỏ đến chừng nào. Câu trả lời đơn giản nhất cho vấn đề là hủy bỏ toàn bộ. Tuy nhiên, tốt nhất là chỉ hủy bỏ từ điểm khóa chết gây trở ngại.
- Thiếu: khi chọn được giao dịch để hủy bỏ, nhiều trường hợp chính giao dịch đó lại bị hủy bỏ khi khóa chết xuất hiện. Để tránh trường hợp này, giao dịch chỉ nên bị hủy bỏ vì khóa chết trong một số lần nhất định.

## Khôi phục CSDL

Chúng ta có thể mất thông tin lưu trong máy tính vì nhiều lý do:

- Sự cố hệ thống: hệ thống có thể ở trạng thái không bình thường, như là khóa chết. trạng thái này không cho chương trình hoàn thiện công việc một cách bình thường. Tuy nhiên, sự cố hệ thống không dẫn đến sự sai lạc dữ liệu.
- Lỗi logic: lỗi logic gồm dữ liệu mất mát ngăn cản khả năng hoàn thành công việc một cách bình thường.
- Lỗi phần cứng: lỗi cơ bản gồm lỗi ổ đĩa hoặc sai sót trong dung lượng truyền tải qua liên kết truyền.

Chúng ta phải có những biện pháp để khôi phục dữ liệu sau khi xảy ra sự cố hệ thống. rất khó có thể lấy lại CSDL như ban đầu nhưng cũng được một phần nào đó. Hệ Quản trị CSDL có thể hoàn thành các giao dịch với dữ liệu đã được phục hồi.

Để đảm bảo toàn vẹn dữ liệu, giao dịch cần ở trạng thái:

- Bị hủy: không nhất thiết giao dịch phải hoàn thành công việc của mình. Khi giao dịch chưa hoàn thành, nó sẽ bị hủy để không làm ảnh hưởng đến trạng thái thống nhất của CSDL. Sự khôi phục thực hiện bởi ROLLBACK.

- Đã xác nhận: khi giao dịch kết thúc thành công thì nó ở trạng thái đã xác nhận. Giao dịch đã xác nhận đưa CSDL đến trạng thái nhất quán mới.

Khôi phục CSDL gồm nhật ký giao dịch lưu toàn bộ lịch sử thay đổi ở CSDL và trạng thái của của mỗi giao dịch. Chúng ta có thể theo những cách sau để quyết định cách thức khôi phục:

- Nhật ký với các cập nhật bị trì hoãn: phương pháp này ghi lại toàn bộ thay đổi CSDL trong nhật ký. Tuy nhiên, chúng hoãn tất cả ghi chép về các thao tác thực hiện cho đến khi giao dịch được xác nhận một phần. Khi giao dịch được xác nhận một phần, thông tin trong nhật ký giao dịch được dùng để tiến hành những ghi chép bị trì hoãn. Sau đó, CSDL được cập nhật với kết quả ghi chép. Nếu hệ thống thất bại trước khi giao dịch hoàn thành, hoặc giao dịch thất bại, thông tin trong nhật ký sẽ bị bỏ qua.
- Nhật ký với cập nhật liên tục: mọi cập nhật sẽ được ghi lại ngay lập tức trong nhật ký. Khi hệ thống gặp sự cố, thông tin trong nhật ký sẽ lưu lại CSDL ở trạng thái nhất quán trước đó.

## Luyện tập

### Ngăn chặn khóa chết

Các cách để ngăn chặn khóa chết là:

- Mỗi giao dịch khóa dữ liệu của mình trước khi thao tác. Thêm vào đó, mọi dữ liệu phải được khóa tức thì hoặc không khóa dữ liệu nào cả. Cách này có một số bất cập: nhiều dữ liệu được khóa bởi giao dịch nhưng giao dịch lại không sử dụng nó trong thời gian dài, dẫn đến kém hiệu quả sử dụng dữ liệu, hoặc giao dịch phải chờ vĩnh viễn vì ít nhất một dữ liệu nó cần đang bị khóa bởi giao dịch khác.
- Đưa vào sự sắp xếp bộ phận cho dữ liệu và đảm bảo giao dịch khóa dữ liệu trong sắp xếp đó
- Sử dụng quyền thực hiện trước và hủy bỏ giao dịch. Đối với quyền thực hiện trước, chúng ta phải giữ nhả thời gian cho mỗi giao dịch. Nhả thời gian giúp cho việc quyết định giao dịch phải chờ dữ liệu hay hủy bỏ. Giao dịch với nhả thời gian ngắn nhất sẽ không bị hủy bỏ. Nếu giao dịch bị hủy bỏ, nó sẽ bắt đầu lại với nhả thời gian cũ.

### Thủ thuật

Khôi phục CSDL sử dụng nhật ký giao dịch. Thông thường, khôi phục dữ liệu gồm có quét nhật ký giao dịch để thấy giao dịch gần nhất. tuy nhiên, không có giới hạn cho việc quét nhật ký vì lỗi có thể xảy ra từ giao dịch đầu. Quá trình này rất tốn thời gian. Cách tốt hơn là tìm một điểm, gọi là điểm kiểm tra. Hệ Quản trị CSDL đảm bảo rằng mọi dữ liệu trước điểm này không có lỗi và đã được lưu cẩn thận. Sau đó, khôi phục CSDL sẽ bắt đầu từ điểm kiểm tra. Phương pháp này được xem là kiểm tra điểm. Khi Hệ Quản trị CSDL duy trì nhật ký giao dịch và thực hiện kiểm tra điểm với các hành động sau:

1. Hệ Quản trị CSDL tạm hoãn bắt đầu giao dịch mới cho đến khi mọi giao dịch đang tiến hành đã kết thúc hoặc bị hủy bỏ.
2. Hệ Quản trị CSDL tạo một bản sao dự phòng của CSDL
3. Hệ Quản trị CSDL ghi lại các nhật ký giao dịch trong bộ nhớ chính của ổ đĩa
4. Hệ Quản trị CSDL nối thêm một bản ghi vào cuốn nhật ký để chỉ ra điểm kiểm tra. Sau đó, Hệ Quản trị CSDL ghi dữ liệu vào bộ nhớ.

## Các câu hỏi thường gặp

- *Sau bao lâu thì chúng ta phải kiểm tra khóa chết?*

Điều đó phụ thuộc vào hai yếu tố:

- Tần suất của sự xuất hiện khóa chết
- Số các giao dịch bị ảnh hưởng bởi khóa chết

Chúng ta phải kiểm tra thường xuyên sự xuất hiện của khóa chết, chúng xảy ra khá thường xuyên. Bởi vì, các giao dịch khác không thể dùng dữ liệu của khóa chết cho đến khi khóa chết được giải phóng. Đôi khi, chúng ta phải kiểm tra khóa chết ngay khi thấy yêu cầu không được thực hiện ngay lập tức.

- *Có bao nhiêu cách để kiểm tra vấn tin con trả lại bao nhiêu hàng như kết quả?*

SQL đưa ra mệnh đề EXISTS để kiểm tra điều đó. Ví dụ câu lệnh SQL sau tìm tên của khách hàng có tài khoản tại chi nhánh Ridge

Tại đây mệnh đề EXISTS kiểm tra xem khách hàng có tài khoản ở chi nhánh Ridge hay không.

Chúng ta cũng có thể kiểm tra sự không tồn tại các hàng trong vấn tin con bằng mệnh đề NOT EXISTS. Ví dụ, câu lệnh SQL sau tìm tên của khách hàng không có tài khoản ở chi nhánh Ridge.

- *Ngoài UNION, chúng ta có thể thực hiện thao tác khác sử dụng câu lệnh SQL không?*

Cùng với UNION, chúng ta có thể thực hiện toán tử INTERSECT và MINUS qua mệnh đề INTERSECT và MINUS trong SQL. Ví dụ, câu lệnh sau tìm mã sản phẩm là tủ lạnh mua bởi khách hàng “C4171”

Câu lệnh SQL sau tìm mã của tất cả sản phẩm là tủ lạnh nhưng không phải mua bởi khách hàng “C4171”

- *SQL có gồm toán tử nào để so sánh giá trị không?*

SQL cung cấp toán tử so sánh BETWEEN. Toán tử này đơn giản hóa mệnh đề CLAUSE xác định điều kiện mà giá trị phải nhỏ hơn hoặc bằng giá trị nào đó và lớn hơn hoặc bằng giá trị nào đó. Ví dụ, nếu muốn tìm số tài khoản có kết toán trong vòng \$54000 đến \$66000. Có hai câu lệnh SQL cho truy vấn này. Một truy vấn là:

Truy vấn khác là:

Chú ý rằng câu lệnh SQL thứ hai dễ hiểu hơn. SQL cung cấp toán tử so sánh NOT BETWEEN.



## Bài tập

1. Nối đúng kiểu hiển thị với định nghĩa của nó:

1. Tập con theo hàng	A. Gồm một số cột và một số hàng từ bảng nguồn
2. Tập con theo hàng	B. Tạo thành bởi hai hay nhiều truy vấn bảng trong định nghĩa hiển thị
3. Kết hợp	C. Gồm tất cả các cột nhưng chỉ có một số hàng trong bảng nguồn
4. Tập con hàng – cột	D. Gồm tất cả các hàng nhưng chỉ có một vài cột trong bảng nguồn

2. Trong các bước sau:

- a. Giao dịch A truy hồi bản ghi R ở thời điểm T
- b. Giao dịch B truy hồi cùng bản ghi ở thời điểm T+1
- c. Giao dịch A cập nhật bản ghi ở thời điểm T+2
- d. Giao dịch B cập nhật cùng bản ghi đó vào thời điểm T=3

Kết quả của các bước trên là gì?

3. Vấn đề của câu lệnh SQL sau đây là gì?

- 1) Kiểu dữ liệu là DECIMAL không được xác định chính xác
- 2) Câu lệnh NOT NULL không được xác định trong thuộc tính a-no
- 3) Thuộc tính names không thể ở trong ngoặc đơn
- 4) Tên bảng không nên ở trong ngoặc đơn

4. Cho ví dụ sau:

Một sinh viên đi đăng ký học. Sinh viên phải đăng ký ít nhất 2 môn. Bạn đó đã chọn lịch sử Mĩ và xã hội học. Nhân viên trực ở phòng đăng ký thực hiện những thao tác sau:

- 1) Kiểm tra xem còn lớp học cho môn Lịch sử Mĩ không
- 2) Tìm được lớp thích hợp
- 3) Sinh viên nhận lớp
- 4) Kiểm tra xem còn lớp học cho môn Xã hội học không
- 5) Không tìm được lớp thích hợp
- 6) Từ chối sinh viên với môn này và đề nghị sinh viên học môn Tâm lý học
- 7) Đăng ký cho sinh viên hai môn học.

Nếu có sự cố xảy ra ở bước 4, Hệ Quản trị CSDL sẽ làm gì?

5. Câu sau đúng hay sai?

Các hàng trong quan hệ không có thứ tự nào cả.

6. Câu nào dưới đây là lợi ích của việc sử dụng hiển thị?

- 1) Dễ dàng cập nhật
- 2) Thông tin hợp lệ
- 3) Độc lập dữ liệu logic
- 4) Truy cập giới hạn

7. Bảng Item có các cột sau:

Item Code (khóa chính)

Item Name

Size

Rate

City

Hiển thị sau có cập nhật được không. Giải thích.

8. Câu lệnh SQL nào sau đây xây dựng sơ đồ bảo mật CSDL?

- a. GRANT
- b. CREATE
- c. REVOKE

d. DROP

9. Câu sau đúng hay sai:

Ràng buộc toàn vẹn thực thể chỉ ra rằng CSDL không được phép chứa giá trị khóa phụ không khớp.

10. Đây là khái niệm hình thành nguyên tắc sơ đồ bảo mật của Hệ Quản trị CSDL?

- a. Người sử dụng
- b. Đối tượng CSDL
- c. Quan hệ
- d. Quyền ưu tiên

# Đáp án

## Đáp án cho bài tập phần 1C

1. Nhà Quản trị CSDL
2. Mức ngoài
3. Người quản lý ổ đĩa
4. Thuộc tính
5. Phép chọn
6. Phép gộp
7. Một – nhiều
8. Sai
9. 1-A, 2-B, 3-D, 4-C
10. Mô hình phân cấp
- 11.

## Đáp án cho bài tập phần 2C

1. Thực thể: SUPPLIER, PÁT

Quan hệ: SHIP (hoặc SHIPMENT)

2. Dùng chuẩn hóa để tách thành bảng như sau:

3. Đúng

4. Phi chuẩn hóa bảng bằng cách thêm thuộc tính Referralbonus vào bảng MonthlySalary.



5. a, c, d
6. Sai
7. a, b, d
8. 1-C, 2-A, 3-D, 4-B
9. Dạng chuẩn thứ tư
10. Nhiều – nhiều



## Đáp án bài tập phần 3C

1. 1-D, 2-C, 3-B, 4-A
2. Cập nhật của Giao dịch A sẽ bị ghi đè bởi cập nhật của giao dịch B.
3. Lựa chọn 2
4. Hệ Quản trị CSDL sẽ hủy bỏ các thay đổi vừa thực hiện với CSDL
5. Đúng
6. 2, 3, 4
7. Không cập nhật được vì mọi thao tác sẽ thất bại khi Hệ Quản trị CSDL không xác định được hàng nào cần cập nhật nếu không có khóa chính.
8. 1, 3
9. Sai
10. 1, 2, 4

# Đánh giá về việc thực hiện mục tiêu bài học

## Thiết kế Cơ sở dữ liệu quan hệ

Tên \_\_\_\_\_ Lớp \_\_\_\_\_ Ngày \_\_\_\_\_

Các mục tiêu cho từng bài trong khóa học được liệt kê dưới đây, Hãy đánh dấu vào mục tiêu mà bạn đã đạt được. Cuối cùng tính phần trăm, nhớ điền đầy đủ họ tên và lớp, sau đó nộp cho người hướng dẫn của bạn.

- | STT | Mục tiêu   |
|-----|--|
| 1.  | Định nghĩa Hệ Quản trị CSDL  |
| 2.  | Phân biệt thành phần của Hệ Quản trị CSDL: dữ liệu và người sử dụng                          |
| 3.  | Nêu sự cần thiết của hệ CSDL   |
| 4.  | Định nghĩa cấu trúc của Hệ Quản trị CSDL theo: Mức ngoài, mức trong, mức khái niệm           |
| 5.  | Phân biệt thành phần của Hệ Quản trị CSDL  |
| 6.  | Nêu sự cần thiết của thiết kế đồ án CSDL   |
| 7.  | Phân biệt các giai đoạn trong DDLC   |
| 8.  | Nêu hậu quả của việc lập đồ án và thiết kế CSDL sơ sài                                       |
| 9.  | Nêu các kiểu mô hình dữ liệu   |
| 10. | Tạo mô hình thực thể - quan hệ   |
| 11. | Liệt kê các quan hệ giữa các thực thể  |
| 12. | Định nghĩa Hệ Quản trị CSDL quan hệ  |
| 13. | Nêu các toán tử quan hệ  |
| 14. | Ánh xạ lược đồ E/R tới bảng cho: thực thể thường, thuộc tính, thực thể yếu, quan hệ, tập con |
| 15. | Phân biệt các khóa   |

16. Định nghĩa chuyên biệt hóa và tổng quát hóa
17. Mô tả phương pháp top-down và bottom-up
18. Mô tả dư thừa dữ liệu
19. Mô tả dạng chuẩn thứ nhất, thứ hai và thứ ba.
20. Mô tả dạng chuẩn Byce-Codd
21. Nêu sự cần thiết của phi chuẩn hóa
22. Phân biệt các quan hệ sau: bảng cơ sở, kết quả truy vấn, hiển thị
23. Tạo bảng
24. Thay đổi bảng
25. Gỡ bỏ bảng
26. Bảng truy vấn và sử dụng
27. Định nghĩa hiển thị
28. Thực hiện thao tác DML trên hiển thị
29. Phân biệt các kiểu hiển thị
30. Nêu ràng buộc toàn vẹn
31. Mô tả quá trình khôi phục giao dịch
32. Định nghĩa nhật ký giao dịch
33. Phân biệt các vấn đề xảy ra cùng lúc
34. Phân biệt các loại khóa
35. Định nghĩa khóa chết
36. Mô tả khái niệm về sơ đồ bảo mật
37. Thiết lập sơ đồ bảo mật

**Phần trăm:**

# THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Designing)

## Phần I – MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Modal)

# Cơ Sở Dữ Liệu Là Gì ?

- Dữ liệu = các thông tin cần lưu trữ vào máy tính để có thể *truy xuất (access)* và *truy vấn (query)*
- CSDL = không gian chứa dữ liệu có tổ chức :
  - Có tương quan (*relational*)
  - Có cấu trúc : các mẫu tin (*record*), cột thuộc tính (*column*)

# Sự cần thiết của CSDL

- Cung cấp khả năng truy vấn dữ liệu
  - Tránh sự dư thừa, trùng lặp dữ liệu
  - Liên kết chặt chẽ giữa dữ liệu và chương trình ứng dụng khai thác.
- ➔ Có nhiều mô hình CSDL khác nhau

# Hệ Quản Trị CSDL (Database Management System)

- Là 1 phần mềm quản lý CSDL, cung cấp các dịch vụ xử lý (truy cập – truy vấn) CSDL cho
  - Nhà phát triển ứng dụng (lập trình viên)
    - Tập các hàm, thư viện mã lệnh truy xuất
  - Người dùng cuối (End-user)
    - Giao diện sử dụng để truy xuất

# Các Mô Hình CSDL

- Mô hình phân cấp
  - Mô hình mạng
  - Mô hình quan hệ
  - Mô hình đối tượng
  - ...
- ➔ Các mô hình CSDL khác nhau có các Hệ Quản Trị CSDL tương ứng.



# Mô Hình CSDL Quan Hệ

Do E.F.Codd đề xuất năm 1971, gồm:

- Dữ liệu được mô tả thành tập các bảng (table) (*dòng dữ liệu và cột thuộc tính*). Các bảng có thể mô tả nội dung của 1 *đối tượng/thực thể* (Entity) hoặc 1 *mối kết hợp* (Relationship)
- Một hệ thống các **ký hiệu, phép toán và khái niệm** (thuật ngữ) để mô tả dữ liệu.

*Trong phần trình bày tiếp theo, ta sẽ dựa vào ví dụ là CSDL Quản Lý Sinh Viên của 1 trường Đại Học*

### Các Khái Niệm : *Thuộc tính* (Attribute)

- Thuộc vào 1 kiểu dữ liệu đơn (Data Type) nhất định và có một miền giá trị.
- Trong 1 *Lược đồ quan hệ* không được có 2 thuộc tính trùng tên.
- Ký hiệu :  $A_1, A_2, \dots B, C, \dots$

Ví dụ : với đối tượng Học Sinh, ta có thể có các thuộc tính : Họ, Tên, Ngày sinh, Điểm trung bình, ...

# *Lược đồ quan hệ* (Relation Schema) (1)

- Tập các thuộc tính của 1 đối tượng hoặc 1 mối kết hợp (1 bảng !)
- Ký hiệu :
  - $Q$
  - $Q$  với tập thuộc tính :  $Q(A_1, A_2, \dots, A_n)$
  - Tập thuộc tính của  $Q$  :  $Q^+$

# *Lược đồ quan hệ* (Relation Schema) (2)

- *Tân từ* của *Lược đồ quan hệ* :
  - Là phát biểu mô tả các ý nghĩa, điều kiện ràng buộc (giữa các thuộc tính) của lược đồ quan hệ.
  - Được phát biểu ở dạng ngôn ngữ tự nhiên hoặc ngôn ngữ hình thức, mã giả, ...

### Ví dụ :

- Mỗi sinh viên chỉ có 1 mã sinh viên và không trùng với các sinh viên khác.
- Ngày kết thúc học phần phải lớn hơn ngày bắt đầu học phần.
- ...

# *Lược đồ quan hệ* (Relation Schema) (3)

- Tập tất cả các *Lược đồ quan hệ* có trong 1 CSDL được gọi là *Lược đồ CSDL*.
- Ký hiệu : **R**

Ví dụ : trong CSDL Quản Lý Sinh Viên, **R** =

Q1 = SINHVIEN(MaSV, Ho, Ten, NgaySinh, DiemTB)

Q2 = HOCPHAN(MaHP, TenHP, SoTinChi)

Q3 = DANGKY\_HOCPHAN(MaSV, MaHP)

...

## *Quan hệ* (Relation)

- Sự thể hiện của *Lược đồ quan hệ* tại một thời điểm nào đó được gọi là *Quan hệ*.
- Ký hiệu : q, r, s, t, ... tương ứng với các *Lược đồ quan hệ* Q, R, S, T, ...

Ví dụ : lược đồ Q = SINHVIEN(MaSV, Ho, Ten, NgaySinh, DiemTB) tại 1 thời điểm t nào đó có quan hệ q =

{  
    SV01, Nguyễn Văn, Tài, 12/10/1980, 6.9  
    SV02, Trần Thị, Chi, 19/12/1980, 7.5  
    SV03, Lý Văn, Sử, 27/8/1978, 8  
}

## *Bộ* (Record)

- Là 1 dòng / phần tử của *Quan hệ*
- Là 1 tập các giá trị tương ứng với các thuộc tính của quan hệ
- Ký hiệu : p, q, t, ...

Ví dụ : *Quan hệ* q =

{

SV01, Nguyễn Văn, Tài, 12/10/1980, 6.9

SV02, Trần Thị, Chi, 19/12/1980, 7.5

SV03, Lý Văn, Sử, 27/8/1978, 8

}

có 3 bộ

## Kết luận

- Bộ **Quan hệ**
- Thuộc tính **Lược đồ quan hệ** **Lược đồ CSDL**
- Tân từ mô tả điều kiện, ràng buộc của các Thuộc tính trong *Lược đồ quan hệ*
- *Quan hệ* là 1 thể hiện (*instance*) tại 1 thời điểm của *Lược đồ quan hệ*



## *Siêu Khóa* (Super Key)

Cho *Lược đồ quan hệ*  $Q$  có thể hiện là quan hệ  $q$ ,  $s$  là 1 tập thuộc tính thuộc  $Q^+$ .  $t_1, t_2$  là 2 bộ tùy ý thuộc  $q$

*Ta ký hiệu  $t_1.s$  là tập các giá trị của  $t_1$  trên tập thuộc tính  $s$*

$$s' = s + A_i$$

*$s$  được gọi là 1 siêu khóa nếu và chỉ nếu :*

$$\blacksquare, t_1, t_2 : t_1.s \blacksquare_{2.s}$$

## *Khóa / Khóa chỉ định*

- Siêu khóa có ít thuộc tính nhất được gọi là *Khóa* hay *Khóa chỉ định*.
- ➔ 1 lược đồ quan hệ có thể có nhiều *khóa*
- Trong lược đồ quan hệ, 1 *khóa* được chọn làm *khóa chính* (primary key).
- *Thuộc tính* có tham gia vào *khóa* được gọi là *thuộc tính khóa*, ngược lại, được gọi là *thuộc tính không khóa*.

## *Khóa ngoại* (foreign key) (1)

- 1 tập *thuộc tính* được gọi là ***Khóa ngoại*** của *Lược đồ quan hệ* Q nếu và chỉ nếu nó là tập *thuộc tính* của  $Q^+$  và là khóa của 1 *lược đồ quan hệ* khác.

Ví dụ : Cho lược đồ cơ sở dữ liệu Quản Lý Sinh Viên, các lược đồ quan hệ :

$Q_1 = \text{SINHVIEN}(\underline{\text{MaSV}}, \text{Ho}, \text{Tên}, \text{DiemTB})$

$Q_2 = \text{HOCPHAN}(\underline{\text{MaHP}}, \text{TênHP}, \text{SốTinChi})$

$Q_3 = \text{DANGKY\_HOCPHAN}(\underline{\text{MaSV}}, \underline{\text{MaHP}})$

## *Khóa ngoại* (foreign key) (2)

- {MaSV} là *khóa* của  $Q_1$
- {MaHP} là *khóa* của  $Q_2$
- {MaSV, MaHP} là *khóa* của  $Q_3$
- {MaSV}, {MaHP} là các *khóa ngoại* của  $Q_3$

*Trong thiết kế CSDL, nhà thiết kế thường dựa vào các **Tân từ** để xác định khóa cho **Lược đồ quan hệ**.*

# Các Phép Toán : Phép toán tập hợp

**Phép hợp** (Union) :  $\cup$

Cho lược đồ quan hệ  $Q$ , với các quan hệ thể hiện  $q_1, q_2$

Ta có :




$$q_1 \cup q_2 = q_3 = \{r : r \in q_1 \vee r \in q_2\}$$

## Các Phép Toán : Phép giao (Intersection)

Phép giao : 

Cho lược đồ quan hệ Q, với các quan hệ thể hiện  $q_1, q_2$

Ta có :

$$q_1 \text{  } q_2 = q_3 = \{ r : r \text{  } q_1 \text{  } q_2 \}$$

## Các Phép Toán : Phép trừ (Minus, difference)

**Phép trừ : -**

Cho lược đồ quan hệ Q, với các quan hệ thể  
hiện  $q_1, q_2$

Ta có :

$$q_1 - q_2 = q_3 = \{r : r \text{ thuộc } q_1 \text{ nhưng không thuộc } q_2\}$$

## Tích Descartes

Cho lược đồ quan hệ  $Q_1(A_1, A_2, \dots, A_n)$  có quan hệ thể hiện là  $q_1$ ;  $Q_2(B_1, B_2, \dots, B_m)$  có quan hệ thể hiện là  $q_2$

$q_1 \times q_2 = q_3$ , ta có :

$q_3$  là 1 quan hệ của  $Q_3$  với  $Q_3^+ = Q_1^+ + Q_2^+$

$q_3 = \{r : r.(Q_1^+) \blacksquare \blacksquare .(Q_2^+) \blacksquare \blacksquare \}$



## Tích Descartes – Ví dụ

Cho lược đồ quan hệ  $Q_1(\text{MaSV}, \text{MaMH}, \text{DiemThi})$ ,  
 $Q_2(\text{MaMH}, \text{TenMH})$  có các quan hệ  $q_1, q_2$  :

$q_1 =$

MaSV	MaMH	DiemThi
99001	CSDL	5.0
99002	CTDL	2.0
99003	MANG	8.0

$q_2 =$

MaMH	TenMH
CSDL	Cơ sở dữ liệu
FOX	Foxpro

$\rightarrow q_3 =$

MaSV	MaMH	DiemThi	MaMH	TenMH
99001	CSDL	5.0	CSDL	Cơ sở dữ liệu
99001	CSDL	5.0	FOX	Foxpro
99002	CTDL	2.0	CSDL	Cơ sở dữ liệu
99002	CTDL	2.0	FOX	Foxpro
99003	MANG	8.0	CSDL	Cơ sở dữ liệu
99003	MANG	8.0	FOX	Foxpro

# Các Phép Toán : Phép toán quan hệ

- **Phép chiếu** (Projection)

\_ Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$  với quan hệ thể hiện là  $q$ ,  $X$  là 1 tập con của  $Q^+$

\_ **Phép chiếu của  $Q$  lên tập  $X$**  (ký hiệu là  $Q[X]$ ) tạo thành lược đồ quan hệ  $Q' = Q[X]$  có  $Q'^+ = X$ .

\_  $t$  là 1 bộ nào đó của  $q$ , **Phép chiếu của  $t$  lên tập  $X$**  (ký hiệu là  $t[X]$  hay  $t.X$ ) là 1 tập con của  $t$  chứa các giá trị ứng với các thuộc tính trong  $X$

\_ **Phép chiếu của  $q$  lên tập  $X$**  (ký hiệu là  $q[X]$  hay  $q.X$ ) tạo thành quan hệ  $q'$  :

$$q' = q[X] = q.X = \{t' : \blacksquare, t.X = t'\}$$

## Phép chiếu – Ví dụ

$q =$

MaSV	MaMH	DiemThi
99001	CSDL	5.0
99002	CTDL	2.0
99003	MANG	8.0

$q.\{\text{MaMH}\} = q[\{\text{MaMH}\}] =$

MaMH
CSDL
CTDL
MANG

## Phép chọn (Selection)

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$  với quan hệ thể hiện là  $q$ ,  $E$  là 1 *biểu thức điều kiện trên các bộ của  $q$* ,  $t$  là 1 bộ nào đó của  $q$ .

\_ Nếu  $t$  thỏa điều kiện  $E$ , ta ký hiệu là  $t(E)$  hay  $t:E$

\_ ***Phép chọn  $q$  trên (với) điều kiện  $E$***  (ký hiệu là  $q(E)$  hay  $q:E$ ) tạo thành  $q'$  là tập con các bộ của  $q$  :

$$q' = q(E) = q:E = \{t : t \text{ (bộ của } q) \text{ (E)}\}$$

## Phép chọn – Ví dụ

$q =$

MaSV	MaMH	DiemThi
99001	CSDL	5.0
99002	CTDL	2.0
99003	MANG	8.0

$E : q.DiemThi \geq 5$

$q(E) = q:E =$

MaSV	MaMH	DiemThi
99001	CSDL	5.0
99003	MANG	8.0

## Phép kết (Join)

Cho lược đồ quan hệ  $Q_1(A_1, A_2, \dots, A_n)$  có quan hệ thể hiện là  $q_1$ ;  $Q_2(B_1, B_2, \dots, B_m)$  có quan hệ thể hiện là  $q_2$ ;  $A_i$  và  $B_j$  là 2 thuộc tính có cùng miền giá trị, giả sử tồn tại 1 phép so sánh  $\blacksquare$  trên miền giá trị của  $A_i$  và  $B_j \Rightarrow$

Gọi  $q'_3 = q_1 \bowtie q_2$  và  $Q_3$  là lược đồ quan hệ của  $q'_3$

$\rightarrow A_i \blacksquare B_j$  là 1 biểu thức điều kiện trên  $q'_3$

# Phép kết (t.t)

- *Phép kết giữa  $q_1$  và  $q_2$  trên điều kiện  $A_i \bowtie B_i$  (ký hiệu là  $A_i \theta B_i$ ) được xác định như sau :*

$$q_3 = q_1 \bowtie_{A_i \theta B_i} q_2 = \{ t : \begin{array}{l} t.Q_1^+ = t_1 \\ t.Q_2^+ = t_2 \\ t_1.A_i \bowtie t_2.B_j \end{array} \}$$

**Hoặc** :  $q_3 = q_1 \bowtie_{A_i \theta B_i} q_2 = q'_3(A_i \bowtie B_j)$

# Phép kết – Ví dụ

$q_1 =$

MaSV	MaMH	DiemThi
99001	CSDL	5.0
99002	CSDL	2.0
99003	MANG	6.0

$q_2 =$

MaMH	DiemDau
CSDL	4.0
MANG	7.0

$A_i = \text{DiemThi} ; B_j = \text{DiemDau} ; \blacksquare \text{ '>=}'$

$q_3 = q_1 \overset{A.\theta B.}{\blacksquare} q_2 =$

MaSV	MaMH	DiemThi	MaMH	DiemDau
99001	CSDL	5.0	CSDL	4.0
99003	MANG	6.0	CSDL	4.0



## Phép kết – Kết luận

- Phép kết thực chất là gộp của *Tích Descartes* và phép chọn
- Trình tự thực hiện phép kết :
1. Tạo tích Descartes
  2. Thực hiện phép chọn với điều kiện  $E = A_I \blacksquare B_J$

## Phép kết – Ứng dụng

- Phép kết có rất nhiều ứng dụng trong thiết kế CSDL, đặc biệt khi thực hiện các truy vấn (query)
- Ở slide 29, ta nhận thấy phép kết  $\bowtie$  chưa thực hiện đúng ý đồ của nhà thiết kế là tìm ra các sinh viên đạt điểm đậu các môn học tương ứng (bộ thứ 2 của  $q_3$  không đúng). Trong trường hợp này, ta dùng phép kết

*kép* :

$$q_3 = q_1 \overset{A_i \theta B_j}{\bowtie} \overset{C_k \theta D_l}{\bowtie} q_2$$

trong đó :  $C_k = q_1.MaMH$  ,  $D_l = q_2.MaMH$  và  $\bowtie$  : ‘=’

## Phép kết tự nhiên (Natural Join)

- Trong trường hợp  $\blacksquare = '='$ , phép kết được gọi là *Phép kết bằng*
- Trong trường hợp  $A_I = B_J$ , và  $\blacksquare = '='$ , phép kết được gọi là *Phép kết tự nhiên*
- ➔ Ví dụ ở slide 31, phép kết giữa  $C_k$ ,  $D_l$  và  $\blacksquare$  được gọi là *kết tự nhiên*.
- Phép kết tự nhiên thường được sử dụng nhất trong thực tế

## Phép chia (Division)

Cho lược đồ quan hệ  $Q_1(A_1, A_2, \dots, A_n)$  có quan hệ thể hiện là  $q_1$ ;  $Q_2(B_1, B_2, \dots, B_m)$  có quan hệ thể hiện là  $q_2$ ; và  $n > m$ . Phép chia  $q_1$  cho  $q_2$  tạo thành  $q_3$  với lược đồ quan hệ  $Q_3$  được định nghĩa như sau :

$$Q_3^+ = \{A_1, A_2, \dots, A_{n-m}\}$$

$$q_3 = q_1 \div q_2 = \{t : t \in q_1, \exists t_2 \in q_2, t = t_1 \cdot Q_3^+ \cdot t_2\}$$
$$t_2 = t_1 \cdot \{A_{n-m+1}, \dots, A_n\}$$

## Phép chia – Ví dụ

q <sub>1</sub>				
A	A	A	A	A
1	2	3	4	5
a	b	d	c	g
a	b	d	e	f
b	c	e	e	f
e	g	c	c	g
e	g	c	e	f
a	b	e	g	c
a	b	d	t	l

q <sub>2</sub>	
B <sub>1</sub>	B <sub>2</sub>
c	g
e	f

$$q_3 = q_1 \blacksquare q_2 =$$

q <sub>3</sub>		
A	A	A
1	2	3
a	b	d
e	g	c

## Phép chia – Ví dụ (t.t)

Đáp số là kết quả của phép chia quan hệ  
DANGKY\_HOCPHAN cho quan hệ  
HOCPHAN.MAHP

DANGKY_HOCPHAN ■ HOCPHAN.MAHP
MaSV
SV01
SV03

# Các tính chất của Đại số quan hệ

Giáo trình trang 9

# THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Designing)

## Phần II – NGÔN NGỮ TRUY VẤN SQL

(Structured Query Language = ngôn ngữ truy vấn có cấu trúc)



# SQL = Structured Query Language

- Là ngôn ngữ dùng để truy vấn dữ liệu
- Ngôn ngữ = cú pháp (cấu trúc ngữ pháp) + các từ khóa (từ vựng) + hàm lập sẵn.
- Là 1 công cụ giao tiếp của Hệ Quản Trị CSDL
- Là cầu nối giữa :
  - Nhà phát triển (Lập trình viên ) và Hệ quản trị CSDL
  - Người dùng cuối (End-user) và Hệ quản trị CSDL

# SQL = Structured Query Language

- Ngôn ngữ SQL là một chuẩn chung tương đối giữa các Hệ quản trị CSDL khác nhau.
- 1 trong các cú pháp của SQL :

*SELECT* <tên các thuộc tính>

*FROM* <tên các quan hệ>

*WHERE* <điều kiện chọn>

...

# Cú pháp SQL – Kiểu Dữ liệu

(data type)


- Chuỗi (String) : được đặt trong dấu nháy kép hoặc đơn.

– Ví dụ :

```
SELECT *
```

```
FROM SINHVIEN
```

```
WHERE MASV = "SV01"
```



dữ liệu  
chuỗi

# Cú pháp SQL – Kiểu Dữ liệu (t.t)

(data type)

- **Số** (number)
  - Ví dụ : 1024 ; 4.5 ; ...
- **Ngày tháng** (date/time) : được đặt trong cặp dấu #, giữa ngày – tháng – năm là dấu phân cách “-” hoặc “/”, tên tháng có thể là số (1-12) hoặc viết tắt 3 chữ cái đầu.
  - Ví dụ : #12/2/2001# ; #1-Jan-94# ; ...

# Cú pháp SQL – Các toán tử số học

(Arithmetic Operations)

Toán tử	Ý nghĩa	Ví dụ	Kết quả
+	Cộng	$5 + 2$ #28/08/01# + 4	7 #01/09/01#
-	Trừ	#02/09/01# - 3	#30/08/01#
*	Nhân	$5 * 2$	10
/	Chia	$5 / 2$	2.5
\	Chia nguyên	$5 \setminus 2$	2
^	Lũy thừa	$5 ^ 2$	25
<i>Mod</i>	Chia dư	5 Mod 2	1

# Cú pháp SQL – Các toán tử so sánh

(Comparative Operations)

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<	Nhỏ hơn	$3 < 5$	True
<=	Nhỏ hơn hay bằng	$2 <= 5$	True
>	Lớn hơn	$2 > 5$	False
>=	Lớn hơn hay bằng	$2 >= 5$	False
=	Bằng nhau	$2 = 5$	False
<>	Khác nhau	$2 <> 5$	True

# Cú pháp SQL – Các toán tử luận lý

(Logical Operations)

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<i>Not</i>	Luật phủ định	Not (5 > 2) Not (2 > 5)	False True
<i>And</i>	Luật và	(5 > 2) And (2 > 5) (5 > 2) And (5 > 4)	False True
<i>Or</i>	Luật hay	(5 > 2) Or (2 > 5) (2 > 5) Or (4 > 5)	True False

### Ví dụ

- 1. SELECT*      HO,TEN  
*FROM*            SINHVIEN  
*WHERE*           NOT(MASV = 'SV01')
- 2. SELECT*      MASV,HO,TEN  
*FROM*            SINHVIEN  
*WHERE*           (DIEMTB >= 5) *AND*  
                    (DIEMTB <= 6.5)



# Cú pháp SQL – Các toán tử *BETWEEN...AND*

Cú pháp :

value1 *Between* value2 *and* value3

Ý nghĩa : trả về True nếu value1 nằm giữa value2 và value3  $\Leftrightarrow$  value2  $\leq$  value1  $\leq$  value3

Ví dụ :

```
SELECT      *  
FROM        SINHVIEN  
WHERE       DIEMTB BETWEEN 5 AND 6.5
```

# Cú pháp SQL – Các toán tử *LIKE*

Cú pháp :

value1 *LIKE* <khuôn mẫu giá trị>

Ý nghĩa :

Trả về các value1 có dạng thức giống như <khuôn mẫu giá trị>

Các ký tự đại diện dùng trong khuôn mẫu :

\* : đại diện cho tất cả ký tự bất kỳ

? : đại diện cho một ký tự bất kỳ

# : đại diện cho 1 ký tự số

[A<sub>1</sub>,A<sub>2</sub>,...] : đại diện cho 1 ký tự thuộc tập {A<sub>1</sub>, A<sub>2</sub>, ...}

[A<sub>1</sub> – A<sub>2</sub>] : đại diện cho 1 ký tự thuộc khoảng ký tự từ A<sub>1</sub> đến A<sub>2</sub>

### Các toán tử *LIKE* – Ví dụ

*SELECT* \*  
*FROM* SINHVIEN  
*WHERE* TEN *LIKE* '\*Hoa'

Chọn tất cả  
các cột có  
trong quan hệ

Ý nghĩa : tìm tất cả sinh viên có từ Hoa trong phần cuối của tên, ví dụ : 'Ngọc Thoa', 'Đào Hoa', ...

### Các toán tử *LIKE* – Ví dụ (t.t)

```
SELECT      *  
FROM        SINHVIEN  
WHERE       MASV LIKE 'SV0[1-4]'
```

Ý nghĩa : tìm tất cả sinh viên có mã sinh viên là SV01, SV02, SV03 hoặc SV04

➔ Toán tử *LIKE* được sử dụng nhiều trong các truy vấn tìm kiếm dữ liệu

# Cú pháp SQL – Các hàm lập sẵn

Cú pháp chung : <tênHàm>(Danh sách đối số)

Hàm **Iif**

Cú pháp : **Iif**(điều kiện, giá trị 1, giá trị 2)

Ý nghĩa : Trả về giá trị 1 nếu điều kiện đúng, ngược lại, trả về giá trị 2.

Ví dụ :

```
SELECT *
```

```
FROM SINHVIEN
```

```
WHERE DIEMTB >= IIF(GIOITINH='Nam',6.5,6)
```

### Hàm Date

Cú pháp : **Date()**

Ý nghĩa : Trả về giá trị ngày giờ hiện tại của hệ thống.

Ví dụ :

```
SELECT      *  
FROM HOADON  
WHERE NGAYLAP >= (DATE()-5)
```

# Hàm Sum

Cú pháp : **Sum**(<tên thuộc tính>)

Ý nghĩa : Trả về tổng của các giá trị tương ứng với <tên thuộc tính> của tất cả các bộ có trong quan hệ thỏa điều kiện WHERE.

Ví dụ :

```
SELECT          Sum(GIATRI)
FROM           HOADON
WHERE          NGAYLAP >= (DATE()-5)
```

Ý nghĩa : Trả về tổng giá trị của các hóa đơn có ngày lập trong vòng 6 ngày gần đây.

# Hàm Max

Cú pháp : **Max**(<tên thuộc tính>)

Ý nghĩa : Trả về giá trị lớn nhất trong các giá trị tương ứng với <tên thuộc tính> của các bộ có trong quan hệ thỏa điều kiện WHERE.

Ví dụ :

```
SELECT           Max(GIATRI)
FROM            HOADON
WHERE           NGÀYLAP >= (DATE()-5)
```

Ý nghĩa : Trả về giá trị lớn nhất trong các hóa đơn có ngày lập trong vòng 6 ngày gần đây.



# Một số hàm khác

- **Day**(<biểu thức ngày>) : trả về chỉ số của ngày trong <biểu thức ngày>.
  - Ví dụ : **Day**(#12/2/2005#) → 12
- **Month**(<biểu thức ngày>) : trả về chỉ số của tháng trong <biểu thức ngày>.
- **Year**(<biểu thức ngày>) : trả về chỉ số của năm trong <biểu thức ngày>.
- **Len**(<giá trị chuỗi>) : trả về độ dài của chuỗi

# Một số hàm khác (t.t)

– Ví dụ :

```
SELECT          *  
FROM           SINHVIEN  
WHERE LEN(TEN) > 4
```

- ***Chr***(<mã ASCII>) : trả về ký tự có mã ASCII tương ứng.
- ***InStr***(start,s1,s2) : trả về vị trí của chuỗi s2 trong chuỗi s1 kể từ vị trí start.
- ***LCase***(s) : trả về giá trị chuỗi in thường của chuỗi s
- ***UCase***(s) : trả về giá trị chuỗi in hoa của chuỗi s

# Một số hàm khác (t.t)

- ***Left***(s,n) : trả về chuỗi gồm n ký tự bên trái của chuỗi s.
- ***Right***(s,n) : trả về chuỗi gồm n ký tự bên phải của chuỗi s.
- ***Mid***(s,i,n) : trả về chuỗi con của chuỗi s gồm n ký tự kể từ vị trí i.
- ***Nz***(v1,v2) : trả về giá trị v1 nếu v1 khác Null, ngược lại trả về giá trị v2.

# Một số hàm khác (t.t)

- ***Min***(<tên thuộc tính>) : trả về giá trị nhỏ nhất trong các giá trị tương ứng với <tên thuộc tính> của các bộ thỏa điều kiện WHERE có trong quan hệ.
- ***Avg***(<tên thuộc tính>) : trả về giá trị trung bình cộng của các giá trị tương ứng với <tên thuộc tính> của các bộ thỏa điều kiện WHERE có trong quan hệ.
- ***Count***(<tên thuộc tính>) : trả về số lượng các giá trị tương ứng với <tên thuộc tính> của các bộ thỏa điều kiện WHERE và khác Null có trong quan hệ.

# Các Loại Truy Vấn SQL

## 1. Truy vấn chọn (Select query)

- Là các truy vấn bắt đầu bằng từ khóa *SELECT*
- Trả về 1 giá trị hoặc 1 tập các bộ

## 2. Truy vấn định nghĩa dữ liệu (Data Definition Query)

- Là các truy vấn bắt đầu bằng từ khóa *CREATE, DELETE, INSERT, ALTER, ...*
- Sử dụng để tạo, thêm, xóa, sửa các bảng (quan hệ), bộ, ràng buộc, ... trong CSDL

## 3. Truy vấn cập nhật dữ liệu (Data Modification Query)

# Truy vấn định nghĩa dữ liệu – Tạo lược đồ quan hệ

Ví dụ 1 :

```
CREATE TABLE SINHVIEN(  
MASV Text(10) CONSTRAINT k1 PRIMARY KEY,  
HOTEN Text(30), NGAYSINH Date, MALOP Text(10),  
DIEMTB Double )
```

Ghi chú :    \_ Từ in nghiêng là từ khóa của SQL  
              \_ *Text, Date, Double, ...* : các kiểu dữ liệu  
              (của thuộc tính)  
              \_ *Text(10)* : kiểu dữ liệu Text, có độ dài 10 ký  
tự

## Tạo lược đồ quan hệ (t.t)

*\_MASV Text(10) CONSTRAINT k1 PRIMARY KEY :*

Khai báo thuộc tính MASV là khóa chính với *quy tắc ràng buộc* tên là k1

Ví dụ 2 :

```
CREATE TABLE BANGDIEM(  
MASV Text(10), MAMH Text(10), DIEM Double,  
CONSTRAINT k2 PRIMARY KEY (MASV, MAMH)  
)
```

# Thêm, xóa, sửa thuộc tính (cột)

## **Thêm thuộc tính và quan hệ**

Ví dụ :

```
ALTER TABLE          SINHVIEN ADD COLUMN GIOITINH  
TEXT(10)
```

## **Sửa kiểu dữ liệu của thuộc tính :**

```
ALTER TABLE          SINHVIEN ALTER COLUMN GIOITINH  
BOOLEAN
```

## **Xóa thuộc tính**

Ví dụ :

```
ALTER TABLE          SINHVIEN DROP COLUMN GIOITINH
```



# Xóa, thêm các ràng buộc

## Xóa ràng buộc khóa chính

Ví dụ :

```
ALTER TABLE      SINHVIEN DROP CONSTRAINT k1
```

## Thêm ràng buộc khóa chính

Ví dụ :

```
ALTER TABLE      SINHVIEN ADD CONSTRAINT k1  
PRIMARY KEY (MASV)
```

## Thêm ràng buộc miền giá trị lên thuộc tính

Ví dụ :

```
ALTER TABLE      SINHVIEN ADD CONSTRAINT k3  
CHECK (DIEMTB >= 0 AND DIEMTB <= 10)
```

# Truy vấn chọn

Ví dụ 1 : Chọn tất cả sinh viên có điểm trung bình  $\geq 6.5$

*SELECT \* FROM SINHVIEN WHERE DIEMTB  $\geq 6.5$ ;*

Ví dụ 2 : Chọn 10 sinh viên có điểm trung bình cao nhất

*SELECT TOP 10 FROM SINHVIEN;*

Ví dụ 3 : Chọn 10% sinh viên có điểm trung bình cao nhất

*SELECT TOP 10% FROM SINHVIEN;*

Ví dụ 4 : Chọn có loại bỏ các bộ trùng : chọn các mức điểm khác nhau mà các sinh viên đã đạt được

*SELECT DISTINCT DIEMTB FROM SINHVIEN;*

Lưu ý : Dấu ; cho biết đã kết thúc câu lệnh SQL

# Truy vấn chọn từ nhiều bảng

Ví dụ 1 : Tìm tất cả các tên học phần mà sinh viên mang mã số SV01 đã đăng ký.

```
SELECT      HOCPHAN.TENHP
FROM        SINHVIEN,DANGKY_HOCPHAN,HOCPHAN
WHERE       SINHVIEN.MASV = 'SV01' AND
              SINHVIEN.MASV = DANGKY_HOCPHAN.MASV
              AND
              DANGKY_HOCPHAN.MAHP = HOCPHAN.MAHP;
```

Lưu ý :  $FROM Q_1, Q_2, \dots, Q_n \Leftrightarrow FROM Q_1 \times Q_2 \times \dots \times Q_n$  (Tích Descartes)

# Truy vấn chọn có kết

Ví dụ 1 : Tìm tất cả các tên học phần mà sinh viên mang mã số SV01 đã đăng ký.

```
SELECT      HOCPHAN.TENHP
FROM        (SINHVIEN INNER JOIN DANGKY_HOCPHAN ON
              SINHVIEN.MASV = DANGKY_HOCPHAN.MASV)
              INNER JOIN HOCPHAN ON
              DANGKY_HOCPHAN.MAHP = HOCPHAN.MAHP
WHERE        MASV = 'SV01';
```

Lưu ý : Phép kết chính là phép chọn có điều kiện từ tích Descartes.

## Truy vấn chọn dữ liệu (p.4)

---

# Truy vấn chọn có sắp thứ tự kết quả trả về

Ví dụ 1 : Tìm tất cả các tên sinh viên đã đăng ký học phần có mã là CSDL, sắp thứ tự kết quả trả về theo tên tăng dần, họ tăng dần và mã sinh viên giảm dần.

```
SELECT          MASV,HO,TEN
FROM (SINHVIEN INNER JOIN DANGKY_HOCPHAN ON
        SINHVIEN.MASV = DANGKY_HOCPHAN.MASV
WHERE           MAHP = 'CSDL'
ORDER BY       TEN ASC, HO ASC, MASV DESC;
```

Lưu ý : Khi thuộc tính giữa các bảng được truy vấn sau từ khóa From không trùng tên thì ta có thể ghi tường minh tên thuộc tính, mà không cần phải ghi :

<Tên bảng>.<Tên thuộc tính>

## Truy vấn chọn dữ liệu (p.5)

---

# Truy vấn chọn có sắp các kết quả trả về theo nhóm (group by)

Ví dụ 1 : Tìm tất cả các tên sinh viên đã đăng ký học phần ít nhất 3 học phần trở lên.

```
SELECT    SINHVIEN.MASV, SINHVIEN.HOTEN
FROM      DANGKY_HOCPHAN INNER JOIN SINHVIEN ON
            DANGKY_HOCPHAN.MASV=SINHVIEN.MASV
GROUP BY   SINHVIEN.MASV,SINHVIEN.HOTEN
HAVING     COUNT(DANGKY_HOCPHAN.MAHP)>=3
```

## Truy vấn chọn lồng nhau (nested/sub query)

- Là câu lệnh truy vấn khi mà trong biểu thức điều kiện của WHERE hoặc HAVING là một câu truy vấn khác.

Ví dụ : Lấy về thông tin của sinh viên có điểm trung bình cao nhất.

*SELECT*           MASV,HOTEN

*FROM*             SINHVIEN

*WHERE*           DIEMTB >=

*ALL(SELECT DIEMTB FROM SINHVIEN)*

# Các từ khóa trong truy vấn lồng nhau

- ***ANY, SOME*** : Kết quả các bộ trả về của query cha so sánh với 1 trong (bất kỳ) các bộ của query con.
- ***ALL*** : Kết quả các bộ trả về của query cha so sánh với tất cả các bộ của query con.
- ***IN*** : Kết quả các bộ trả về của query cha bằng với 1 trong (bất kỳ) các bộ của query con.
- ***NOT IN*** : Kết quả các bộ trả về của query cha không bằng với bất kỳ bộ nào của query con.
- ***EXISTS / NOT EXISTS*** : Kết quả các bộ trả về của query cha được thỏa khi query con có tồn tại ít nhất 1 bộ / không tồn tại bộ nào.



## Truy vấn lồng nhau – Ví dụ

Ví dụ : Lấy về thông tin của các sinh viên có đăng ký môn học CSDL.

*SELECT*            *MASV,HOTEN*

*FROM*             *SINHVIEN*

*WHERE*            *MASV IN*

*(SELECT MASV FROM DANGKY\_HOCPHAN  
WHERE MAHP='CSDL')*

## Truy vấn lồng nhau – Ví dụ

Ví dụ : Lấy về thông tin của các sinh viên không có đăng ký môn học CSDL.

*SELECT*            *MASV,HOTEN*

*FROM*             *SINHVIEN*

*WHERE*            *MASV NOT IN*

*(SELECT MASV FROM DANGKY\_HOCPHAN  
WHERE MAHP='CSDL')*

## Truy vấn lồng nhau – Ví dụ

Ví dụ : Trả về điểm trung bình cộng của các sinh viên nếu như có ít nhất 1 sinh viên có điểm trung bình  $\geq 5$ .

```
SELECT  AVG(DIEMTB)  
FROM    SINHVIEN  
WHERE    EXISTS(SELECT DIEMTB FROM  
            SINHVIEN WHERE DIEMTB $\geq$ 5)
```

# Truy vấn cập nhật dữ liệu – Cập nhật các bộ

Cú pháp :

*UPDATE* <TÊN BẢNG> *SET*

<TÊN THUỘC TÍNH 1> = <GIÁ TRỊ 1> ,

<TÊN THUỘC TÍNH 2> = <GIÁ TRỊ 2> ,

...

<TÊN THUỘC TÍNH n> = <GIÁ TRỊ n>

*WHERE* <ĐIỀU KIỆN>

# Cập nhật các bộ (t.t)

Ví dụ : Cộng thêm 1 điểm cho các sinh viên có điểm trung bình  $\geq 4$  và  $< 5$

```
UPDATE  SINHVIEN SETDIEMTB=DIEMTB+1  
WHERE    DIEMTB $\geq$ 4 AND DIEMTB $<$ 5
```

## Xóa các bộ

Cú pháp :

*DELETE FROM* <TÊN BẢNG>

*WHERE* <ĐIỀU KIỆN>

Ví dụ : Xóa các học sinh không có điểm trung bình

*DELETE FROM*      SINHVIEN

*WHERE*              DIEMTB = Null

## Thêm các bộ vào quan hệ (bảng)

Cú pháp :

```
INSERT INTO <TÊN BẢNG>(
<TÊN THUỘC TÍNH1>, <TÊN THUỘC TÍNH2>, ...)
VALUES(<GIÁ TRỊ 1>, <GIÁ TRỊ 2>, ...)
```

Lưu ý : Các giá trị trong *VALUES*(...) phải tương ứng với các thuộc tính trong <TÊN BẢNG>( ... )

Nếu có thuộc tính nào trong lược đồ quan hệ <TÊN BẢNG> không được khai báo trong <TÊN BẢNG>( ... ) và *VALUES*( ... ) thì giá trị của bộ mới được thêm vào ứng với thuộc tính đó sẽ được đặt bằng Null

# THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Designing)

## Phần III – RÀNG BUỘC TOÀN VỆ (entegrity constraint)



# Ràng buộc toàn vẹn – Khái niệm

- Là những điều kiện bất biến mà tất cả các bộ trong các quan hệ có liên quan(đến ràng buộc) phải thỏa tại mọi thời điểm.
- ➔ Ràng buộc toàn vẹn rất quan trọng vì nó qui định ràng buộc trên dữ liệu nhập/xuất trong CSDL.
- Ràng buộc toàn vẹn thường được mô tả bằng các *Tân từ* (xem phần I), do nhà thiết kế CSDL tìm và phát hiện ra trong quá trình phân tích CSDL.

# Các yếu tố của Ràng buộc toàn vẹn

1. **Điều kiện** : là điều kiện ràng buộc (nội dung chính của Ràng buộc toàn vẹn), thường được mô tả bằng ngôn ngữ đặc tả hình thức.

Ví dụ :

Ràng buộc  $R_1$  :

■■■■,  $t_2$  ■■■■ INHVIEN,  $t_1$ .MaSV ■■■■ $t_2$ .MaSV

Ràng buộc  $R_2$  :

■■■■ ■■■■ INHVIEN,  
■■■■ ■■■■ ĐANGKY\_HOCPHAN,  $t_2$ .MAHP = 'CSDL' ■■■■  
 $t_2$ .MaSV =  $t_1$ .MaSV

Ý nghĩa ràng buộc  $R_2$ : mọi sinh viên đều phải đăng ký học phần CSDL

# Các yếu tố của Ràng buộc toàn vẹn (t.t)

**2. Bối cảnh** : là các (lược đồ) quan hệ có liên quan đến ràng buộc toàn vẹn.

Như trong ví dụ của yếu tố *Điều kiện*, ràng buộc  $R_1$  có bối cảnh là quan hệ SINHVIEN, ràng buộc  $R_2$  có bối cảnh là quan hệ SINHVIEN và DANGKY\_HOCPHAN

# Các yếu tố của Ràng buộc toàn vẹn (t.t)

**3. Tầm ảnh hưởng** : các thao tác cập nhật dữ liệu (thêm / xóa / sửa) – có tác động lên các quan hệ trong bối cảnh của ràng buộc toàn vẹn – cần phải được kiểm tra lại điều kiện ràng buộc.

Ta thường xác định yếu tố *Tầm ảnh hưởng* bằng cách xây dựng *Bảng Tầm ảnh hưởng* của ràng buộc toàn vẹn.

# Bảng tầm ảnh hưởng

<Tên RBTV>	Thêm	Xóa	Sửa
<quan hệ 1>	+ hoặc -	+ hoặc -	+ hoặc -
<quan hệ 2>	+ hoặc -	+ hoặc -	+ hoặc -
...	+ hoặc -	+ hoặc -	+ hoặc -
<quan hệ n>	+ hoặc -	+ hoặc -	+ hoặc -

- <quan hệ 1>, <quan hệ 2>, ... , <quan hệ n> là các quan hệ trong bối cảnh của RBTV
- Tại ô dòng i, cột j là dấu +  $\Leftrightarrow$  thao tác j xảy ra tại <quan hệ i> cần được kiểm tra lại điều kiện RBTV trên bộ liên quan đến thao tác; là dấu - : ngược lại.

# Bảng tầm ảnh hưởng – Ví dụ

Trong 2 ràng buộc toàn vẹn ở slide 3

R1	Thêm	Xóa	Sửa
SINHVIEN	+	-	+(MASV)

Giải thích :

Thao tác Thêm : Thêm 1 bộ mới vào quan hệ SINHVIEN (tức thêm SV mới), thì phải kiểm tra MaSV có đã bị trùng trong bảng chưa

Thao tác Xóa : Xóa 1 bộ (xóa 1 SV) trong quan hệ SINHVIEN thì không cần phải kiểm tra ràng buộc về MaSV

Thao tác Sửa : Khi sửa thông tin của 1 bộ đang tồn tại trong quan hệ SINHVIEN, thì yêu cầu kiểm tra ràng buộc chỉ bắt buộc khi thao tác Sửa xảy ra trên thuộc tính MaSV (sửa MaSV)

## Bảng tầm ảnh hưởng – Ví dụ

<b>R2</b>	<b>Thêm</b>	<b>Xóa</b>	<b>Sửa</b>
SINHVIEN	+	-	+(MASV)
DANGKY_HOCPHAN	-	+	+

# Phân loại Ràng buộc toàn vẹn

- Ràng buộc có bối cảnh là 1 quan hệ :
  - RBTV miền giá trị.
  - RBTV liên thuộc tính.
  - RBTV liên bộ.
- Ràng buộc có bối cảnh là nhiều quan hệ :
  - RBTV phụ thuộc tồn tại.
  - RBTV liên bộ - liên quan hệ.
  - RBTV liên thuộc tính – liên quan hệ.



# RBTV liên bộ

- Là sự ràng buộc giữa các bộ trong cùng 1 quan hệ.
- Một loại RBTV liên bộ phổ biến là ràng buộc toàn vẹn về khóa : *trong 1 quan hệ, 2 bộ bất kỳ không được trùng khóa.*
- Bảng tầm ảnh hưởng chung :

R	Thêm	Xóa	Sửa
<quan hệ>	+	-	+

# RBTV trên miền giá trị

- Là điều kiện áp đặt trên miền giá trị của các thuộc tính.
- Ví dụ : thuộc tính DIEMTB của quan hệ SINHVIEN phải trong khoảng  $[0,10]$
- Bảng tầm ảnh hưởng chung :

R	Thêm	Xóa	Sửa
<quan hệ>	+	-	+

# RBTV liên thuộc tính

- Là điều kiện ràng buộc giữa các thuộc tính trong 1 (lược đồ) quan hệ.
- Ví dụ : trong quan hệ SINHVIEN, thuộc tính NGAYSINH phải luôn luôn nhận giá trị nhỏ hơn thuộc tính NGAYVAODOAN :

**SINHVIEN,**

$t.NGAYSINH < t.NGAYVAODOAN$

- Bảng tầm ảnh hưởng chung :

<b>R</b>	<b>Thêm</b>	<b>Xóa</b>	<b>Sửa</b>
<quan hệ>	+	-	+

# RBTV phụ thuộc tồn tại

- Phổ biến nhất là ràng buộc khóa ngoại :

██████████ DANNGKY\_HOCPHAN,

██████████ t<sub>1</sub> ██████████ INHVIEN : t.MASV = t<sub>1</sub>.MASV ██████████

██████████ t<sub>2</sub> ██████████ HOCPHAN : t.MAHP = t<sub>2</sub>.MAHP

- Bảng tầm ảnh hưởng chung : R<sub>1</sub> chứa khóa ngoại, R<sub>2</sub> chứa khóa chính

Quan hệ	Thêm	Xóa	Sửa
<quan hệ R1>	+	-	+
<quan hệ R2>	-	+	+

# RBTV liên thuộc tính - liên quan hệ

- Là điều kiện ràng buộc giữa các thuộc tính trong nhiều (lược đồ) quan hệ.
- Ví dụ : Cho lược đồ CSDL Quản Lý Bán Hàng với quan hệ DatHang, HoaDonXuat, ta có RBTV :

■ ■ DatHang,  $t_2$  ■ HoaDonXuat :

$$t_1.MaHD = t_2.MaHD \rightarrow t_1.NgayDatHang \leq t_2.NgayXuat$$

# THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Designing)

## Phần IV – PHỤ THUỘC HÀM (Functional Dependency)

# Phụ thuộc hàm – Khái niệm

- Phụ thuộc hàm là công cụ để biểu diễn hình thức các RBTV phụ thuộc.
- Các lý thuyết về Phụ thuộc hàm ứng dụng nhiều trong bài toán Chuẩn Hóa CSDL.
- Ký hiệu :

$X \rightarrow Y$  :  $Y$  phụ thuộc hàm vào  $X$  hay  $X$  xác định  $Y$ .

với  $X, Y$  là các tập thuộc tính (trong 1 lược đồ quan hệ).

## Phụ thuộc hàm – Định nghĩa

Cho  $Q(A_1, A_2, \dots, A_n)$ ;  $X, Y$  là 2 tập con của  $Q^+$ ;  $q$  là 1 quan hệ trên  $Q$ ;  $t_1, t_2$  là 2 bộ bất kỳ của  $q$ .

Ta có  $X$  xác định  $Y$ , ký hiệu  $X \rightarrow Y$ , nghĩa là

$$(t_1.X=t_2.X \Rightarrow t_1.Y=t_2.Y)$$

*Nếu 2 bộ bất kỳ trong  $q$  giống nhau trên  $X$  thì phải giống nhau trên  $Y$ .*

$X \rightarrow Y$  là PTH của  $Q$ , khi  $X \rightarrow Y$  đúng với mọi  $q$  là quan hệ trên  $Q$

Hệ quả :  $\blacksquare$ ,  $\blacksquare$   $\blacksquare$   $Q^+$ ,  $X \rightarrow \blacksquare$



# Phụ thuộc hàm hiển nhiên (Trivial Dependencies)

Nếu  $X \twoheadrightarrow Y$  thì  $X \rightarrow Y$  luôn đúng

Trong trường hợp này ( $X \twoheadrightarrow Y$ ),  $X \rightarrow Y$  được gọi là *Phụ thuộc hàm hiển nhiên*.

Ví dụ :  $X \rightarrow X$

Khi chuẩn hóa CSDL, ta thường không quan tâm đến các PTH hiển nhiên.

# Thuật toán kiểm tra PTH : *Satisfies*

Input :     \_ Quan hệ  $q$ ,  
              \_ Tập thuộc tính  $X, Y$

Output :  
           \_ True nếu  $X \rightarrow Y$ , ngược lại, False

# Thuật toán kiểm tra PTH (t.t)

### Bước 1 :

Sắp lại các bộ trong  $q$  sao cho các bộ giống nhau trên  $X$  nằm kế nhau.

### Bước 2 :

Kiểm tra nếu tất cả các bộ giống nhau trên  $X$  cũng giống nhau trên  $Y$  thì trả về True, ngược lại, trả về False.

# Hệ luật dẫn Amstrong (Amstrong inference rule) -

## Một số định nghĩa

Ký hiệu  $F$  là tập các phụ thuộc hàm của lược đồ quan hệ  $Q$ ,  $F = \{f_1, f_2, \dots, f_n\}$ , quy ước ta không quan tâm đến các phụ thuộc hàm hiển nhiên.

Định nghĩa : **Phụ thuộc hàm được suy diễn logic từ  $F$ .**

Phụ thuộc hàm  $d = X \rightarrow Y$  được suy diễn logic từ  $F$  nếu với mọi  $q$  trên  $Q$  thỏa  $F$  thì cũng thỏa  $d$ , ký hiệu  $F \models X \rightarrow Y$ , hay  $F \models d$ .

Định nghĩa : **Bao đóng của  $F$  (Closure)**, ký hiệu  $F^+$ , là tập tất cả các phụ thuộc hàm được suy diễn logic từ  $F$

# Các tính chất của Bao đóng $F^+$

## 1. Tính phản xạ

- Với mọi  $F^+$ , ta luôn có  $F \subseteq F^+$

## 2. Tính đơn điệu

- Nếu  $F \subseteq G$  thì  $F^+ \subseteq G^+$

## 3. Tính lũy đẳng

- Với mọi  $F$ , ta luôn có  $(F^+)^+ = F^+$

Định nghĩa : **Phủ của  $F$** , ký hiệu  $F^- = G - F^+$ , với  $G$  là tập tất cả các phụ thuộc hàm có thể có của  $Q$

# Hệ luật dẫn Amstrong

Cho  $X, Y, Z, W$  là các tập con của  $Q^+$ ;  $q$  là 1 quan hệ bất kỳ trên  $Q$ .

1. **Luật phản xạ** (Reflexive rule) :  $X \rightarrow X$
2. **Luật thêm vào** (Augmentation rule) :  $X \rightarrow Y \Rightarrow XZ \rightarrow Y$
3. **Luật hợp** (Union rule) :  $X \rightarrow Y, X \rightarrow Z \Rightarrow X \rightarrow YZ$
4. **Luật phân rã** (Decomposition rule) :  $X \rightarrow YZ \Rightarrow X \rightarrow Y, X \rightarrow Z$
5. **Luật bắt cầu** (Transitive rule) :  $X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$
6. **Luật giả bắt cầu** (Pseudo transitive rule) :  
$$X \rightarrow Y, YZ \rightarrow W \Rightarrow XZ \rightarrow W$$

## Bao đóng của tập thuộc tính

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$  với quan hệ  $q$ ;  
 $F$  là tập phụ thuộc hàm trên  $Q$ ,  $X$  là 1 tập con của  $Q^+$ .

Định nghĩa : **Bao đóng của  $X$  trên  $F$** , ký hiệu  $X_F^+$ ,  
là tập các thuộc tính  $A_i$  :

$X_F^+ = \{A_i \mid X \rightarrow A_i \text{ được suy diễn từ } F \text{ nhờ hệ luật dẫn Amstrong.}\}$

*Khi không có nhầm lẫn về  $F$ , ta viết  $X^+$  thay vì  $X_F^+$*

## Các tính chất của Bao đóng

Hệ quả : Bao đóng của Q chính là  $Q^+ : Q_F^+ = Q^+$

- Tính phản xạ :  $X \subseteq X^+$
- Tính đơn điệu :  $X \subseteq Y \Rightarrow X^+ \subseteq Y^+$
- Tính lũy đẳng :  $(X^+)^+ = X^+$
- $(XY)^+ \subseteq X^+Y^+$
- $(X^+Y)^+ = (XY^+)^+ = (X^+Y^+)^+$
- $X \rightarrow Y \Leftrightarrow Y^+ \subseteq X^+$
- $X \rightarrow X^+, \quad X^+ \rightarrow X$
- $X^+ = Y^+ \Leftrightarrow X \rightarrow Y \text{ và } Y \rightarrow X$



## Thuật toán tìm Bao đóng

Input : lược đồ quan hệ  $Q$ , tập phụ thuộc hàm  $F$ , tập thuộc tính  $X$ .

Output :  $X^+$

**Bước 1** :  $i = 0, X_i = X$

**Bước 2** : Duyệt tập  $F$ ,

Nếu  $F_k = Y \rightarrow Z$  có  $Y \not\subseteq X_i$  thì

$X_{i+1} = X_i \cup Z$  và  $F = F \setminus F_k$

**Bước 3** : Nếu  $X_{i+1} \subseteq X_i$  :

$i = i+1$ , trở lại bước 2

Nếu  $X_{i+1} = X_i$  : kết thúc

## Thuật toán tìm Bao đóng – Ví dụ

Cho :  $Q(ABCDEFGH)$ ,  $F = \{$

$f_1: \quad B \quad \rightarrow \quad A$

$f_2: \quad DA \quad \rightarrow \quad CE$

$f_3: \quad D \quad \rightarrow \quad H$

$f_4: \quad GH \quad \rightarrow \quad C$

$f_5: \quad AC \quad \rightarrow \quad D$

$\}$

$X = \{AC\}$  , tìm  $X^+$  ?

## Tìm Bao đóng - Ví dụ (t.t)

Bước	Các giá trị của $i, X_i, F$
1	$i = 0; X_0 = X = \{AC\}$
2	$X_{i+1} = X_1 = \{ACD\}; F = F \setminus f_5 = \{f_1, f_2, f_3, f_4\}$
3	$X_1 \not\subseteq K_0 \Rightarrow i = i+1 = 1$ , quay lại bước 2
4	$X_{i+1} = X_2 = \{ACDH\}; F = F \setminus f_3 = \{f_1, f_2, f_4\}$
5	$X_2 \not\subseteq K_1 \Rightarrow i = i+1 = 2$ , quay lại bước 2
6	$X_{i+1} = X_3 = \{ACDHE\}; F = F \setminus f_2 = \{f_1, f_4\}$
7	$X_3 \not\subseteq K_2 \Rightarrow i = i+1 = 3$ , quay lại bước 2
8	Không có $f_k$ nào.
9	$X_4 = X_3 \Rightarrow$ Kết thúc giải thuật. $X^+ = \{ACDHE\}$

# Thuật toán Kiểm tra $F \models d$

Cho  $d = X \rightarrow Y$ , kiểm tra  $F \models d$  ?

Bước 1 : Tính  $X^+ = X_F^+$

Bước 2 :  $Y \subseteq X^+ \Leftrightarrow F \models d$

## Tính chất của các PTH $\square^+$

Xét  $d = X \rightarrow Y \square^+$ ,  $\Rightarrow F \models d$  hay  $F^+ \models d$

$F \models d, \Rightarrow Y \square_{F^+}$  (1)

Gọi TN là tập các thuộc tính có xuất hiện ở vế trái của ít nhất 1 PTH trong  $F$

Từ thuật toán tìm bao đóng của tập thuộc tính, ta có kết luận rằng : Nếu  $X \square_{TN} \Rightarrow X_{F^+} = X$  (2)

## Thuật toán tìm $F^+$

Input : Lược đồ  $Q$ , tập PTH  $F$

Output : Tập PTH  $F^+$

Bước 1 :  $F^+ = \blacksquare$  Tìm tất cả các tập con của TN

Bước 2 : Ứng với mỗi tập con  $X$  của TN:

2.1 Tính  $X_F^+$

2.2 Tìm tất cả các tập con  $Y$  của  $X_F^+$  và  $Y \blacksquare X$

2.3  $F^+ = F^+ + X \rightarrow Y$

# THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Designing)

Phần V – PHỦ (Cover) CỦA TẬP PHỤ  
THUỘC HÀM

# Một số định nghĩa

Cho  $F, G$  là 2 tập phụ thuộc hàm,

**$F$  và  $G$  gọi là tương đương** nếu và chỉ nếu

$$F^+ = G^+$$

Ký hiệu :  $F \equiv G$

**$F$  gọi là phủ  $G$**  nếu và chỉ nếu

$$F^+ \supseteq G^+$$



# Thuật toán kiểm tra $F \equiv G$

Bước 1 : Tính  $F^+$ ,  $G^+$

Bước 2 : Nếu  $F^+ = G^+$ ,  $\Rightarrow F \equiv G$

## Phủ tối thiểu của 1 tập phụ thuộc hàm (p.1)

---

# Phủ tối thiểu (minimal cover) – Tập Phụ thuộc hàm không đầy đủ

Cho lược đồ  $Q$ , tập PTH  $F$ ,  $Z \rightarrow Y$  [redacted].

$Z \rightarrow Y$  gọi là có vế trái dư thừa hay  $Y$  phụ thuộc không đầy đủ vào  $Z$  hay  $Z \rightarrow Y$  là phụ thuộc hàm không đầy đủ nếu và chỉ nếu :

$$\text{[redacted]} \setminus \{Z \rightarrow Y\} \text{ [redacted]} \{(Z-A) \rightarrow Y\}$$

Ngược lại,  $Z \rightarrow Y$  gọi là phụ thuộc hàm đầy đủ hay không có vế trái dư thừa.

$F$  được gọi (tắt) là có vế trái không dư thừa, nếu  $F$  không chứa PTH có vế trái dư thừa.

# Phụ thuộc hàm không đầy đủ - Ví dụ

Cho  $Q(ABC)$ ,  $F = \{AB \rightarrow C, B \rightarrow C\}$

Xét  $AB \rightarrow C$  :

$$F' = F - \{AB \rightarrow C\} = \{B \rightarrow C\}$$

$$(AB-A) \rightarrow C = \{B \rightarrow C\}$$

$$\Rightarrow F' = (F - \{AB \rightarrow C\}) \blacksquare (AB-A) \rightarrow C = \{B \rightarrow C\}$$

Tính  $(F')^+$ , ta có  $(F')^+ = \{AB \rightarrow C, B \rightarrow C\}$

Tính  $F^+ = F = \{AB \rightarrow C, B \rightarrow C\}$

$$\Rightarrow F^+ = (F')^+$$

$\Rightarrow AB \rightarrow C$  là phụ thuộc hàm có vế trái dư thừa

# Thuật toán loại khỏi F các PTH không đầy đủ

Bước 1 : Tính  $F^+$

Bước 2 : Duyệt tập F, với mọi  $d = X \rightarrow Y \in F$  :

Bước 2.1 : Duyệt các tập con  $X' \subsetneq X$  :

Nếu  $X' \rightarrow Y \in F^+$  : thay  $X = X'$ , lặp lại 2.1

# Tập phụ thuộc hàm có vẻ phải 1 thuộc tính

- Định nghĩa : F được gọi là **tập phụ thuộc hàm có vẻ phải 1 thuộc tính** nếu và chỉ nếu mọi phụ thuộc hàm trong F đều có vẻ phải chỉ 1 thuộc tính.
- Ví dụ :  $F = \{A \rightarrow BC, B \rightarrow C, AB \rightarrow D\}$ , ta tách các phụ thuộc hàm trong F để F thỏa tiêu chuẩn là tập phụ thuộc hàm có vẻ phải 1 thuộc tính :

$$F = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow D\}$$

## Tập phụ thuộc hàm không dư thừa

- Định nghĩa : F được gọi là tập phụ thuộc hàm không dư thừa  $\Leftrightarrow$

Không  $\exists F' \subsetneq F, F' \models F$

Ngược lại, F được gọi là tập phụ thuộc hàm dư thừa.

- Ví dụ :  $F = \{A \rightarrow BC, B \rightarrow D, AB \rightarrow D\}$

F dư thừa vì  $F \models F' = \{A \rightarrow BC, B \rightarrow D\}$

# Thuật toán loại khỏi F các PTH dư thừa

Duyệt từng PTH  $X \rightarrow Y$  thuộc F :

Nếu  $(F - \{X \rightarrow Y\}) \models X \rightarrow Y$  thì  $F = F - \{X \rightarrow Y\}$

Ví dụ :  $F = \{A \rightarrow BC, B \rightarrow D, AB \rightarrow D\}$

Xét  $A \rightarrow BC$  :  $\{B \rightarrow D, AB \rightarrow D\}$  không thể  $\models A \rightarrow BC$

Xét  $B \rightarrow D$  :  $\{A \rightarrow BC, AB \rightarrow D\}$  không thể  $\models B \rightarrow D$

Xét  $AB \rightarrow D$  :  $\{A \rightarrow BC, B \rightarrow D\} \models AB \rightarrow D$  vì :

$A \rightarrow BC \Rightarrow A \rightarrow B$ , do  $B \rightarrow D \Rightarrow A \rightarrow D \Rightarrow AB \rightarrow D$

Vậy  $AB \rightarrow D$  là dư thừa trong F,  $\Rightarrow F = \{A \rightarrow BC, B \rightarrow D\}$

## **Tập PTH tối thiểu**

Định nghĩa : F được gọi là một **tập PTH tối thiểu** (hay F là 1 **phủ tối thiểu**) nếu và chỉ nếu F thỏa 3 điều kiện sau :

1. F là tập PTH có vế trái không dư thừa.
2. F là tập PTH có vế phải 1 thuộc tính.
3. F là tập PTH không dư thừa.



# Thuật toán tìm Phủ tối thiểu

Bước 1 :

Loại khỏi F các PTH có vế trái dư thừa.

Bước 2 :

Tách các PTH có vế phải nhiều hơn 1 thuộc tính thành các PTH có vế phải 1 thuộc tính.

Bước 3 :

Loại khỏi F các PTH dư thừa.

Algoritmo luôn tìm được ít nhất 1 PTH tối thiểu của 1 tập PTH bất kỳ.

Algoritmo cũng có thể tìm được nhiều PTH tối thiểu của 1 tập PTH bất kỳ.

## Thuật toán tìm PTT – Ví dụ

Input :  $Q(ABCD)$ ,  $F = \{AB \rightarrow CD, B \rightarrow C, C \rightarrow D\}$

Output :  $F_m = \text{PTT của } F$

Bước 1 :

$AB \rightarrow CD$  là PTH không đầy đủ, vì A là dư thừa trong vế trái:  $B \rightarrow C, C \rightarrow D \Rightarrow B \rightarrow CD$ .

$\Rightarrow F = \{B \rightarrow CD, B \rightarrow C, C \rightarrow D\}$

Bước 2 :

$F = \{B \rightarrow C, B \rightarrow D, B \rightarrow C, C \rightarrow D\}$

Bước 3 :

$F = \{B \rightarrow C, C \rightarrow D\}$

# Khóa<sub>(Key)</sub> của lược đồ quan hệ

Cho  $Q(A_1, A_2, \dots, A_n)$ , tập PTH  $F$ ,  $K$  là 1 tập con của  $Q^+$

Định nghĩa :  $K$  là 1 siêu khóa của  $Q$  nếu

$$K_F^+ = Q^+$$

Định nghĩa :  $K$  là 1 khóa của  $Q$  nếu

- $K_F^+ = Q^+$
- Không tồn tại  $K' \subset K$ ,  $K'_F^+ = Q^+$

# Thuật toán tìm khóa của LDQH

Input : Lược đồ quan hệ  $Q$ , tập PTH  $F$

Output :  $K$  là 1 khóa của  $Q$

Bước 1 : gán  $K = Q^+$

Bước 2 : Duyệt các thuộc tính  $A$  trong  $K$ ,

– Tính  $K'^+$  với  $K' = K - A$

– Nếu  $K'^+ = Q^+$ , gán  $K = K'$

➔ Khóa  $K$  tìm được có thể không là khóa duy nhất của  $Q$

## Tính chất của khóa

Ký hiệu :

- **Tập nguồn (TN)** : chứa tất cả các thuộc tính chỉ xuất hiện ở vế trái của các PTH trong F.
- **Tập đích (TD)** : chứa tất cả các thuộc tính chỉ xuất hiện ở vế phải của các PTH trong F.
- **Tập trung gian (TG)**  $= Q^+ - TN - TD$

Tính chất : Nếu K là 1 khóa của Q, thì

$$TN \cap K \text{ và } TD \cap K = \emptyset$$

# Tính chất của khóa – Chứng minh

Chứng minh TN  $\square \subseteq K$  :

Giả sử TN không  $\square \subseteq K$ ,  $\Rightarrow$  tồn tại 1 PTH  $X \rightarrow Y \square$   
và  $X$  không  $\square$  và không tồn tại 1 PTH  $Z \rightarrow V$   
 $\square$  sao cho  $X \square Y$

Dựa trên thuật toán tìm bao đóng của tập thuộc tính  $K \Rightarrow X$  không xuất hiện trong  $K_i$  nào  $\Rightarrow$   
 $X \square^+_{\square} \Rightarrow$  trái với giả thiết ( $K$  là khóa, nên  
 $X \square^+_{\square}$ )

# Tính chất của khóa – Chứng minh (t.t)

Chứng minh TD  $K = K^+$

Giả sử TD  $K \neq K^+$ :  $A \in K^+ \setminus K$

$A \in K^+ \Rightarrow$  tồn tại 1 PTH  $X \rightarrow A$  (1)

$A \in K^+ \Rightarrow K^+ = (K - A)^+ \cup \{A\}$ ;

$X \in (K - A)^+ \Leftrightarrow X \in K^+ - \{A\}$ ;

$A \in K^+$  vì  $X \rightarrow A$  không là PTH hiển nhiên (xem slide 4 chương 4)  $\Rightarrow X \in (K - A)^+ \Rightarrow K - A \rightarrow X$  (2)

(1) và (2)  $\Rightarrow K - A \rightarrow A \Rightarrow (K - A)^+ = [(K - A) \cup \{A\}]^+ = K^+$   
 $\Rightarrow$  vô lý vì  $K$  là khóa.

# Thuật toán tìm tất cả các khóa

Bước 1 : Tạo tập TN, TG

Bước 2 : Nếu  $TG = \text{TN} \rightarrow Q$  chỉ có 1 khóa  $K = TN$ , kết thúc thuật toán.

Bước 3 : Tìm tất cả tập con  $X_i$  của TG, đặt  $S_i = TN \setminus X_i$ , tính  $S_i^+$ . Gọi L là tập tất cả các  $S_i$

Bước 4 : Duyệt tập  $S_i$ , nếu  $S_i^+ \neq Q^+$  thì bỏ  $S_i$  khỏi L.

Bước 5 : Với mọi  $S_k, S_l$ , nếu  $S_k \subseteq S_l$  thì bỏ  $S_k$  khỏi L.

→ Tập L còn lại chính là tập tất cả các khóa của Q.



# THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Designing)

## Phần VI – CHUẨN HÓA (Standardize / Normalize) CƠ SỞ DỮ LIỆU

# Dạng Chuẩn<sup>2</sup> (Normal Form) của Lược Đồ Quan hệ

Định nghĩa : **Dạng chuẩn** là tập các tiêu chuẩn để đánh giá độ tốt & xấu (của một lược đồ Quan hệ).

Phân loại : Có 4 mức dạng chuẩn :

1. Dạng chuẩn 1
2. Dạng chuẩn 2
3. Dạng chuẩn 3
4. Dạng chuẩn Boyce-Codd (BC) ■ Dạng chuẩn 4
5. Dạng chuẩn 5 : không được đề cập trong bài giảng này

# Dạng Chuẩn 1

Định nghĩa : 1 lược đồ quan hệ  $Q$  được gọi là **đạt dạng chuẩn 1** nếu toàn bộ các thuộc tính đều mang *giá trị đơn* (single value).

Lưu ý :

*Giá trị đơn* : Chỉ nhận 1 giá trị tại 1 thời điểm. Ví dụ : chuỗi, số, ngày tháng, ...

Không phải *Giá trị đơn* : mảng, tập hợp, ...

→ Đa số các lược đồ quan hệ đều đạt dạng chuẩn 1.

# Dạng Chuẩn 2

Định nghĩa : 1 lược đồ quan hệ Q được gọi là **đạt dạng chuẩn 2** nếu Q đạt dạng chuẩn 1 và mọi *thuộc tính không khóa* của Q đều *phụ thuộc đầy đủ* vào mọi *khóa*.

Lưu ý :

*Thuộc tính không khóa* : Xem slide chương 1

*Phụ thuộc đầy đủ* : K là 1 khóa của Q, X là 1 thuộc tính không khóa, X gọi là *Phụ thuộc đầy đủ* vào K nếu và chỉ nếu không tồn tại K' sao cho PTH  $K' \rightarrow X$

## Dạng Chuẩn 2 – Ví dụ

Cho  $Q(ABCD)$ ,  $F = \{AB \rightarrow C; B \rightarrow D; BC \rightarrow A\}$

Áp dụng thuật toán xác định tất cả các khóa, ta có khóa của  $Q$  là  $K_1=AB$  và  $K_2=BC$ .

Nhận thấy :

$\left\{ \begin{array}{l} D \text{ là 1 thuộc tính không khóa} \\ \{B\} \text{ không là } K_2 \\ B \rightarrow D \text{ không là } F^+ \end{array} \right.$

→  $Q$  không đạt dạng chuẩn 2

## Dạng Chuẩn 2 – Hệ quả

1. Nếu  $Q$  đạt dạng chuẩn 1, và không có thuộc tính không khóa thì  $Q$  đạt dạng chuẩn 2.
2. Nếu  $Q$  đạt dạng chuẩn 1, và tất cả các khóa của  $Q$  đều chỉ có 1 thuộc tính thì  $Q$  đạt dạng chuẩn 2.

## Dạng Chuẩn 2 – Ý nghĩa

Cho quan hệ Q, K là 1 khóa, K'  $\rightarrow$  K, K'  $\rightarrow$  B là 1 thuộc tính không khóa. Giả sử K'  $\rightarrow$  B (Q không đạt dạng chuẩn 2)

Do K là khóa,  $\Rightarrow$  K  $\rightarrow$  B

ta lại có K'  $\rightarrow$  B

$\Rightarrow$  K chỉ đóng vai trò là siêu khóa đối với B hay B không phụ thuộc đầy đủ vào khóa. Trong thực tế, các quan hệ có tồn tại những phụ thuộc không đầy đủ thường mang lại sự không tối ưu cho CSDL.

# Phụ thuộc bắc cầu

Cho lược đồ quan hệ  $Q$ ,  $X$  là tập thuộc tính thuộc  $Q^+$ ,  
 $A$  là 1 thuộc tính thuộc  $Q^+$ .

Định nghĩa :  $A$  được gọi là **phụ thuộc bắc cầu** vào  $X$   
nếu và chỉ nếu :

- 1. Tồn tại tập con  $Y$  thuộc  $Q^+$  sao cho :  
 $X \rightarrow Y$  và  $Y \rightarrow A$
- 2.  $Y \rightarrow X$
- 3.  $A \not\rightarrow Y$

Định nghĩa :  $A$  được gọi là **phụ thuộc trực tiếp** vào  
 $X$  nếu và chỉ nếu  $A$  phụ thuộc  $X$  và không phụ  
thuộc bắc cầu vào  $X$



## Dạng Chuẩn 3

Định nghĩa : 1 lược đồ quan hệ  $Q$  được gọi là **đạt dạng chuẩn 3** nếu mọi thuộc tính không khóa của  $Q$  đều phụ thuộc trực tiếp vào mọi khóa.

Ý nghĩa : Các phụ thuộc hàm bậc cầu của thuộc tính không khóa vào khóa gây khó khăn trong việc kiểm tra các ràng buộc toàn vẹn khi CSDL vận hành.

# Dạng Chuẩn 3 – Hệ quả

1. Nếu Q đạt dạng chuẩn 3 thì Q đạt dạng chuẩn 2.

CM : Giả sử Q đạt dạng chuẩn 3 và không đạt dạng chuẩn 2,  $\Rightarrow$  có thuộc tính A và khóa K sao cho :  
 $K' \rightarrow A$  với  $K' \not\rightarrow K \Rightarrow K \rightarrow K' \not\rightarrow A \not\rightarrow K' \Rightarrow$   
A phụ thuộc bắc cầu vào K  $\Rightarrow$  trái với giả thuyết.

2. Nếu Q không có thuộc tính không khóa thì Q đạt dạng chuẩn 3.

CM : hiển nhiên

## Dạng Chuẩn 3 – Định lý

Cho lược đồ quan hệ  $Q$ ,  $F$  là tập các phụ thuộc hàm có về phải chỉ 1 thuộc tính.

$Q$  đạt dạng chuẩn 3 nếu và chỉ nếu mọi PTH  $X \rightarrow A$  đều có :

$\left\{ \begin{array}{l} X \text{ là siêu khóa} \\ A \text{ là thuộc tính khóa} \end{array} \right.$

## Dạng Chuẩn 3 – Chứng minh định lý

Chiều thuận : Cho Q đạt dạng chuẩn 3, xét PTH  $X \rightarrow A$  bất kỳ, giả sử A không là thuộc tính khóa và X không là siêu khóa.

Gọi K là 1 khóa của Q,  $\Rightarrow K \rightarrow X$ , ta có :

$$\left\{ \begin{array}{l} K \rightarrow X, X \rightarrow A \\ X \text{ không } \rightarrow K \text{ (vì nếu } X \rightarrow K \text{ thì } X \text{ là 1 siêu khóa)} \\ A \blacksquare X \text{ (do } A \blacksquare \text{ và } A \blacksquare) \end{array} \right.$$

$\rightarrow$  A phụ thuộc bắt cầu vào K  $\rightarrow$  trái với giả thiết

## Dạng Chuẩn 3 – Chứng minh định lý

Chiều đảo : Cho lược đồ quan hệ  $Q$ ,  $F$  là tập PTH có vế phải 1 thuộc tính, với mọi  $X \rightarrow A$  ta đều có  $X$  là siêu khóa (1) hay  $A$  là thuộc tính khóa (2).

Giả sử  $Q$  không đạt dạng chuẩn 3  $\Rightarrow$  tồn tại 1 khóa  $K$ , tập thuộc tính  $Y$ , thuộc tính không khóa  $A$  :

$$K \rightarrow Y, Y \rightarrow A, Y \text{ không } \rightarrow K, A \not\rightarrow Y$$

Do  $A$  là thuộc tính không khóa  $\Rightarrow Y$  là siêu khóa  $\Rightarrow Y \rightarrow K \Rightarrow$  trái với giả thiết.

# Thuật toán kiểm tra Dạng chuẩn 3

Input : lược đồ  $Q$ , tập PTH  $F$ .

Output :  $Q$  đạt dạng chuẩn 3 hay không.

Bước 1 :

Tìm tất cả các khóa của  $Q$

Bước 2 :

Tách các PTH trong  $F$  sao cho  $F$  chỉ gồm toàn các PTH có vế phải 1 thuộc tính

Bước 3 :

Xét từng PTH  $X \rightarrow A$  trong  $F$ , nếu có  $A$  không là thuộc tính khóa và  $X$  không là siêu khóa thì kết luận  $Q$  không đạt dạng chuẩn 3.

# Dạng chuẩn Boyce-Codd / DC 4

Định nghĩa : 1 lược đồ quan hệ Q được gọi là **đạt dạng chuẩn BC** nếu mọi PTH  $X \rightarrow A$   $\blacksquare^+$  đều có X là siêu khóa.

Hệ quả :

1. Nếu Q đạt dạng chuẩn BC thì Q đạt dạng chuẩn 3.  
CM : dễ dàng từ định nghĩa.
2. Nếu Q chỉ có 2 thuộc tính thì Q đạt dạng chuẩn BC.  
CM : hiển nhiên.

## Dạng chuẩn BC – Định lý

Định lý : Q đạt dạng chuẩn BC nếu và chỉ nếu mọi PTH  $X \rightarrow A$  đều có X là siêu khóa.

Chứng minh : Chiều thuận là hiển nhiên vì  $F^{+}$

Chiều đảo : Giả sử mọi  $X \rightarrow A$  đều có X là siêu khóa, và Q không đạt dạng chuẩn BC,  $\Rightarrow$  tồn tại  $Y \rightarrow B$  ( $F^{+} - F$ ) và Y không là siêu khóa, gọi K là 1 khóa của Q,  $\Rightarrow K \rightarrow Y$  và  $Y \rightarrow B$  và  $Y \not\rightarrow K \Rightarrow B$  phụ thuộc bắc cầu vào K  $\Rightarrow$  Q không đạt dạng chuẩn 3  $\Rightarrow$  Trái với giả thiết (Vì mọi  $X \rightarrow A$  đều có X là siêu khóa nên Q phải đạt dạng chuẩn 3 ).



# Thuật toán kiểm tra dạng chuẩn BC

Bước 1 : Tìm tất cả các khóa của Q.

Bước 2 : Từ F, tạo tập phụ thuộc hàm F' tương đương có vế phải 1 thuộc tính.

Bước 3 : Duyệt các PTH  $X \rightarrow A$  trong F', nếu X không là siêu khóa : kết luận Q không đạt BC, dừng thuật toán.

Kết luận Q đạt BC.

# Thuật toán kiểm tra dạng chuẩn của Lược đồ quan hệ

Bước 1 : Tìm tất cả khóa của Q.

Bước 2 : Kiểm tra dạng chuẩn BC, nếu Q đạt BC : kết luận Q đạt BC và dừng thuật toán.

Bước 3 : Kiểm tra dạng chuẩn 3, nếu Q đạt DC3 : kết luận Q đạt DC3 và dừng thuật toán.

Bước 4 : Kiểm tra dạng chuẩn 2, nếu Q đạt DC2 : kết luận Q đạt DC2 và dừng thuật toán, ngược lại, Q đạt DC1 (Qui ước mọi lược đồ quan hệ đều đạt DC1)

# Phép tách kết nối

- Tách kết nối  $\Leftrightarrow$  Tách 1 lược đồ quan hệ thành 2 hay nhiều lược đồ quan hệ khác.
- Phép tách kết nối : ứng dụng để *chuẩn hóa* lược đồ quan hệ  $\blacksquare$  *chuẩn hóa* CSDL  $\blacksquare$  tách 1 lược đồ quan hệ ở dạng chuẩn thấp thành các lược đồ quan hệ ở dạng chuẩn cao hơn.
- Tiêu chí hàng đầu của phép tách kết nối là phải *bảo toàn* nghĩa là giữ được sự *không mất mát thông tin* (về dữ liệu / các ràng buộc  $\blacksquare$  các PTH)

# Phép tách kết nối bảo toàn thông tin (■ dữ liệu)

Định nghĩa : Cho  $Q, Q_1, Q_2$  có

$$Q_1 \overset{+}{\square} Q_2 \overset{+}{=} X \square \text{ và } Q_1 \overset{+}{\square} Q_2 \overset{+}{=} Q^+$$

Ta nói rằng  $Q$  được *tách thành 2 lược đồ con*. Và nếu :

$$r = r.Q_1 \overset{X}{><} r.Q_2 \quad \text{thì}$$

phép tách này là *phép tách bảo toàn thông tin*.

## Tính chất

Nếu  $Q, Q_1, Q_2$  thỏa :

$$\left\{ \begin{array}{l} Q_1^+ \blacksquare Q_2^+ = X \\ Q_1^+ \blacksquare Q_2^+ = Q^+ \\ X \rightarrow Q_2^+ \end{array} \right.$$

Thì :

$$r = r.Q_1^+ \overset{X}{|><} r.Q_2^+, \text{ với } r \text{ là 1 quan hệ trên } Q$$

# Tính chất – Chứng minh

Chứng minh :  $t \vdash Q_1^+ \mid \overset{X}{>} \langle \mid r.Q_2^+$

(Dễ dàng từ định nghĩa)

Đặt  $r_1 = r.Q_1^+$  ,  $r_2 = r.Q_2^+$

$t \vdash Q_1^+ \mid \overset{X}{>} \langle \mid r_1$  và  $t \vdash Q_2^+ \mid \overset{X}{>} \langle \mid r_2$  và  
 $t_1.X = t_2.X = t.X$

$\Rightarrow t \vdash Q_1^+ \mid \overset{X}{>} \langle \mid r.Q_2^+$  (Xem slide 28, chương 1)

# Tính chất – Chứng minh (t.t)

Chứng minh :  $t \blacksquare Q_1^+ |> \overset{X}{<} | r.Q_2^+ \Rightarrow t \blacksquare$

$$t \blacksquare Q_1^+ |> \overset{X}{<} | r.Q_2^+ \Rightarrow \blacksquare \blacksquare : t_1 = t.Q_1^+ \quad (1)$$

$$t_1 \blacksquare = r.Q_1^+ \Rightarrow \blacksquare \blacksquare : t_1 = t'.Q_1^+ \quad (2)$$

$$(1), (2) \Rightarrow t.Q_1^+ = t'.Q_1^+ \Rightarrow t.X = t'.X \Rightarrow t.Q_2^+ = t'.Q_2^+ \text{ do } X \rightarrow Q_2^+$$

$$\Rightarrow t = t' \Rightarrow t \blacksquare$$

$$\rightarrow r = r.Q_1^+ |> \overset{X}{<} | r.Q_2^+$$

# Tính chất – Ý nghĩa

- Điều kiện  $X \rightarrow Q_2^+$  trong phép tách  $Q$  thành 2 quan hệ  $Q_1, Q_2$  tương đương với “ $X$  là 1 siêu khóa của  $Q_2$ ” hay “ $X$  chứa khóa của  $Q_2$ ”. Trong thực tế,  $X$  thường là khóa của  $Q_2$ .
- Phát biểu ngắn gọn lại tính chất: “Phép tách lược đồ  $Q$  thành  $Q_1, Q_2$  trên thuộc tính chung  $X$  là khóa của  $Q_2$  là phép tách bảo toàn thông tin”. Phép tách này còn được đặt tên là “Phép tách trên khóa”



# Tính chất – Ví dụ

Cho  $Q(\text{MaHS}, \text{HoTen}, \text{DiemTB}, \text{MaLop}, \text{TenLop})$ ,  $F =$   
 $\{ \text{MaHS} \rightarrow \text{HoTen}, \text{DiemTB}, \text{MaLop}, \text{TenLop} ;$   
 $\text{MaLop} \rightarrow \text{TenLop} \}$

Nhận xét : MaHS là khóa chính của Q và Q không đạt dạng chuẩn 3 (có phụ thuộc bắc cầu  $\text{MaHS} \rightarrow \text{MaLop} \rightarrow \text{TenLop}$ ).

Tách Q thành  $Q_1(\text{MaHS}, \text{HoTen}, \text{DiemTB}, \text{MaLop})$  và  $Q_2(\text{MaLop}, \text{TenLop})$  với  $X = \{ \text{MaLop} \}$

Nhận xét :  $Q_1, Q_2$  đều đạt dạng chuẩn BC

# Tính chất – Ứng dụng

- Để chuẩn hóa lược đồ quan hệ về dạng chuẩn BC, ta có thể thực hiện tách các lược đồ quan hệ chưa đạt BC thành các quan hệ đạt BC :

$Q(A_1, A_2, \dots, A_n)$  chưa đạt BC, có  $F = \{X \rightarrow Y, \dots\}$

Tách  $Q$  thành  $Q_1$  có  $Q_1^+ = Q^+ - Y$ ,  $Q_2^+ = XY$

Nếu  $Q_1, Q_2$  chưa đạt BC, lại tiếp tục tách  $Q_1, Q_2$  thành  $Q_{11}, Q_{12}, Q_{21}, Q_{22} \dots$

