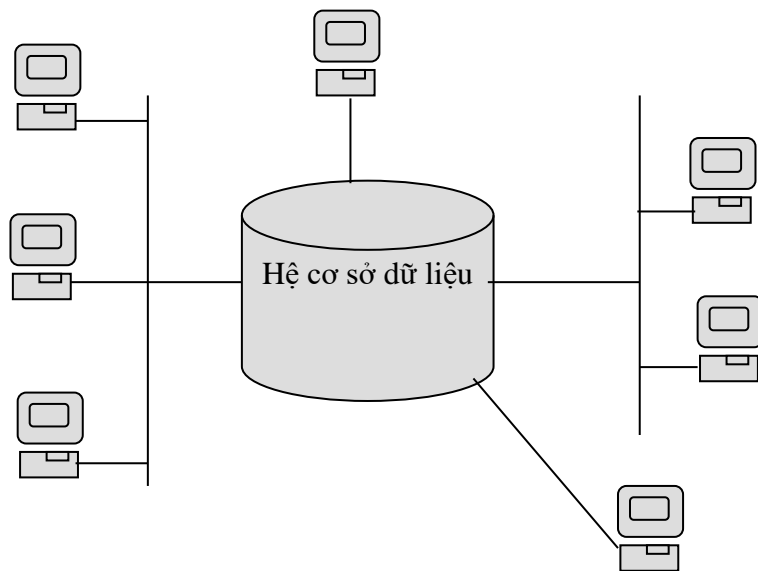


CHƯƠNG 1 : ĐẠI CƯƠNG VỀ CÁC HỆ CƠ SỞ DỮ LIỆU

1.1 Khái niệm về cơ sở dữ liệu (Database)

1.1.1 Khái niệm về cơ sở dữ liệu

Cơ sở dữ liệu là một tập hợp có cấu trúc của các dữ liệu được lưu trữ trên các thiết bị ghi nhớ và có thể truy xuất được bởi các chương trình máy tính, được gọi là chương trình quản trị cơ sở dữ liệu, để thỏa mãn đồng thời cho nhiều user sử dụng.



Hình 1.1 Hệ cơ sở dữ liệu

1.1.2 Mục đích của các hệ cơ sở dữ liệu

Giả sử ta xem xét một phần việc lưu trữ thông tin ở ngân hàng tiết kiệm về các khách hàng và các tài khoản mà được lưu trong các files hệ thống thường trực. Hơn nữa, hệ thống này chứa một số các chương trình ứng dụng cho phép người sử dụng thao tác các files này, bao gồm các chương trình:

- ☐ Chương trình ghi nợ hoặc gửi tiền vào một tài khoản
- ☐ Chương trình thêm một tài khoản mới
- ☐ Chương trình quyết toán
- ☐ Chương trình phát sinh bảng thống kê hàng tháng

Các chương trình này được viết bởi các lập trình viên hệ thống để đáp ứng các nhu cầu của tổ chức ngân hàng. Các chương trình ứng dụng mới được thêm vào hệ thống khi có nhu cầu phát sinh.

Giả sử các điều lệ mới của chính phủ cho phép ngân hàng cung cấp các thông tin kiểm tra các tài khoản. Như thế một số các files thường trực mới sẽ được tạo ra để chứa các thông tin về tất cả các kiểm tra tài khoản hiện có trong ngân hàng và các

chương trình ứng dụng mới cần được viết ra. Vì thế theo thời gian nhiều files, nhiều chương trình ứng dụng được thêm vào hệ thống.

Hệ thống xử lý file (file-processing system) mô tả ở trên được hỗ trợ bởi một hệ điều hành. Nhiều mẫu tin thường trực được lưu trữ trong nhiều files và một số các chương trình ứng dụng khác nhau cũng được viết ra để trích hoặc thêm các mẫu tin vào các files thích hợp. Lược đồ này có một số bất lợi chính:

- Sự dư thừa dữ liệu và sự mâu thuẫn dữ liệu (Data redundancy and inconsistency): Khi các files và các chương trình ứng dụng được tạo ra bởi các lập trình viên khác nhau qua một thời gian dài thì các files này có các định dạng khác nhau và các chương trình được viết bằng nhiều ngôn ngữ lập trình. Hơn nữa, cùng một mẫu thông tin có thể được nhân bản ở nhiều nơi (files). Ví dụ địa chỉ và số phone của một khách hàng có thể xuất hiện trong file chứa các mẫu tin tài khoản tiết kiệm và trong file chứa các mẫu tin kiểm tra tài khoản. Sự dư thừa này dẫn đến kho lưu trữ và chi phí truy xuất phải lớn hơn. Hơn nữa, nó có thể dẫn đến sự mâu thuẫn dữ liệu, do đó nhiều bản copies của cùng một dữ liệu sẽ không phù hợp lâu dài. Ví dụ một sự thay đổi địa chỉ khách hàng có thể chỉ được phản ánh trong các mẫu tin tài khoản tiết kiệm mà không ở nơi khác trong hệ thống. Kết quả dẫn đến sự mâu thuẫn dữ liệu.

- Sự khó khăn trong việc truy xuất dữ liệu.

Giả sử một nhân viên của ngân hàng cần tìm kiếm tên của các khách hàng sống tại thành phố có mã 78733. Nhân viên này yêu cầu phòng xử lý dữ liệu đưa ra một danh sách như thế. Bởi yêu cầu này không được dự định trước khi hệ thống được thiết kế, nên không có một chương trình ứng dụng nào đáp ứng yêu cầu đó. Tuy nhiên có một chương trình phát sinh danh sách của tất cả các khách hàng. Người nhân viên lúc này có hai chọn lựa: một là lấy danh sách của tất cả các khách hàng rồi trích các thông tin cần thiết bằng tay, hai là yêu cầu phòng xử lý dữ liệu viết một chương trình ứng dụng cần thiết. Cả hai lựa chọn đều không thoả mãn. Giả sử rằng một chương trình như thế được viết và, vài ngày sau đó, chính nhân viên đó cần lọc lại danh sách sao cho chỉ chứa những khách hàng nào có tài khoản lớn hơn hoặc bằng \$10000. Một chương trình phát sinh một danh sách như thế không có. Một lần nữa, nhân viên này có hai chọn lựa như trước mà không có cái nào thoả mãn.

Ở đây các môi trường xử lý file quy ước không cho phép dữ liệu cần thiết được tìm kiếm một cách hợp lý và hiệu quả. Các hệ thống tìm kiếm thông tin tốt hơn phải được phát triển cho việc sử dụng chung.

- Sự tách biệt dữ liệu: Bởi dữ liệu được rải ở nhiều files, các files có thể được định dạng khác nhau, nên gây khó khăn cho việc viết các chương trình ứng dụng mới để tìm kiếm các thông tin cần thiết.

- Sự bất thường trong truy xuất đồng thời (Concurrent access anomalies)

Để cải tiến sự thực thi của toàn bộ hệ thống và đạt được thời gian đáp ứng nhanh, nhiều hệ thống cho phép nhiều người sử dụng cập nhật dữ liệu đồng thời. Trong môi trường như thế, sự tương tác của các cập nhật đồng thời có thể đưa đến hậu quả là sự mâu thuẫn dữ liệu.

Giả sử một tài khoản A có \$500. Nếu có hai khách hàng rút tiền (\$50 và \$100 tương ứng) từ tài khoản A ở cùng thời điểm, kết quả của việc giải quyết đồng thời có thể gây ra một sự mâu thuẫn ở tài khoản này. Cụ thể, tài khoản này có thể chứa \$450 hoặc \$400 thay vì \$350. Để đề phòng khả năng này, việc giám sát hệ thống phải được duy trì. Bởi dữ liệu có thể được truy xuất bởi các chương trình ứng dụng khác nhau mà trước đây không có quan hệ với nhau, nên việc giám sát càng khó khăn hơn.

□ Các vấn đề an toàn (security problems).

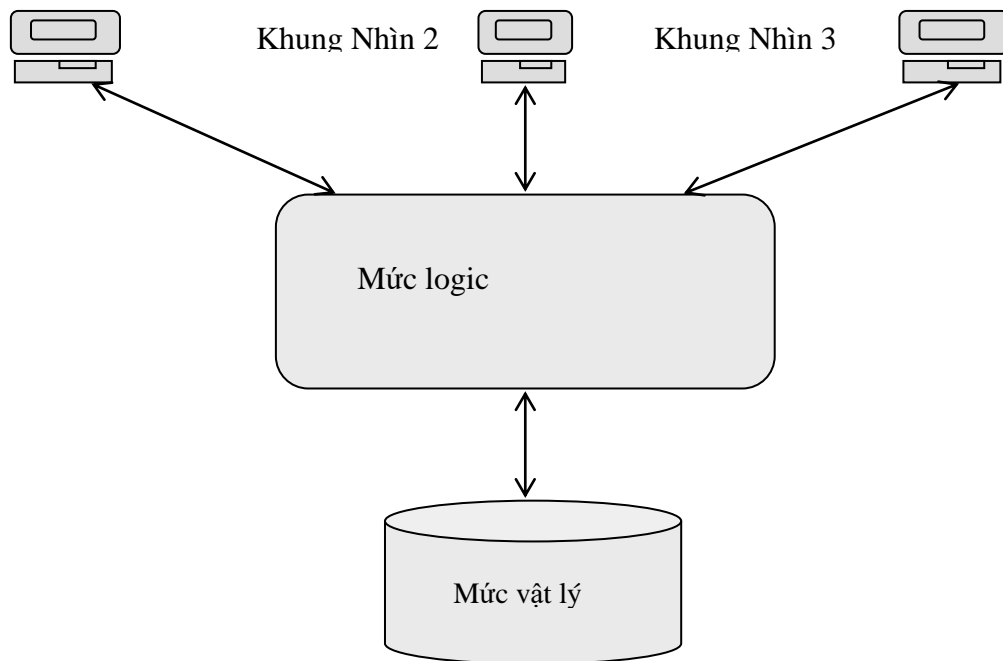
Không thể để mọi người sử dụng đều có khả năng truy xuất tất cả dữ liệu. Như trong hệ thống ngân hàng, bộ phận làm lương chỉ cần thấy một phần cơ sở dữ liệu là các thông tin về các nhân viên của ngân hàng. Họ không cần truy xuất thông tin tài khoản của khách hàng. Bởi các chương trình ứng dụng được thêm vào hệ thống theo một cách không dự tính trước nên nó gây khó khăn cho việc tuân thủ các ràng buộc an toàn này.

□ Các vấn đề toàn vẹn (Integrity problems).

Các giá trị dữ liệu lưu trữ trong cơ sở dữ liệu phải thoả mãn một số kiểu ràng buộc toàn vẹn. Ví dụ ngân khoản không bao giờ xuống thấp hơn một số (như \$25). Các ràng buộc này được đưa vào hệ thống bằng cách thêm những mã lệnh thích hợp, nó gây khó khăn khi thay đổi chương trình. Vấn đề sẽ phức tạp khi các ràng buộc bao gồm một số dữ liệu từ nhiều files khác nhau. Những khó khăn này, một số khác nữa, đã thúc đẩy sự phát triển các hệ quản trị cơ sở dữ liệu. Sau này, chúng ta sẽ thấy các quan niệm và các thuật toán mà đã được phát triển cho các hệ cơ sở dữ liệu để giải quyết các vấn đề đã bàn ở trên.

1.2 Kiến trúc của một hệ thống cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu là tập hợp các files có mối quan hệ và một tập chương trình cho phép những người sử dụng truy xuất và thay đổi các files này. Mục đích chính của hệ cơ sở dữ liệu là cung cấp cho những người sử dụng một cái nhìn trừu tượng (abstract view) về dữ liệu. Hệ thống sẽ dấu một số chi tiết phức tạp như làm thế nào dữ liệu được lưu trữ và duy trì. Tuy nhiên để cho hệ thống có thể dùng được, dữ liệu phải được tìm kiếm một cách có hiệu quả. Việc này đã dẫn đến việc thiết kế kiến trúc cơ sở dữ liệu phức tạp cho sự thể hiện dữ liệu trong cơ sở dữ liệu. Bởi nhiều người sử dụng hệ thống cơ sở dữ liệu không phải là chuyên gia máy tính nên sự phức tạp được dấu đi dưới một số mức.



Hình 1.2 Kiến trúc hệ cơ sở dữ liệu

☐ Mức vật lý (Physical level)

Đây là mức thấp nhất mô tả dữ liệu được lưu trữ thực sự như thế nào. Tại mức vật lý, các cấu trúc vật lý phức tạp được mô tả chi tiết.

☐ Mức logic

Mức logic là mức cao tiếp theo, nó mô tả một mô hình dữ liệu phản ánh thế giới thực mà ta cần lưu trữ trong cơ sở dữ liệu. Ở đây toàn bộ cơ sở dữ liệu được mô tả như là một số lược đồ quan hệ đơn giản. Mặc dù việc hiện thực các lược đồ quan hệ này ở mức logic có thể bao gồm nhiều cấu trúc phức tạp ở mức vật lý, người sử dụng ở mức logic không cần quan tâm đến chúng. Mức logic được dùng cho người quản trị cơ sở dữ liệu và các lập trình viên, họ phải quyết định những thông tin nào được giữ lại trong cơ sở dữ liệu và lập trình như thế nào.

☐ Mức khung nhìn (View level)

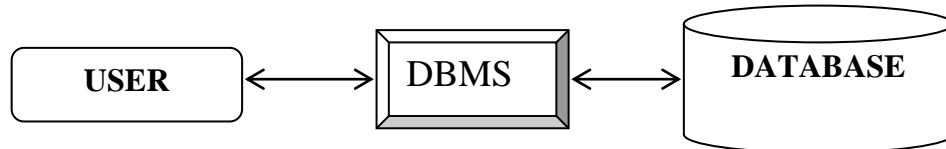
Đây là mức cao nhất mô tả chỉ một phần cơ sở dữ liệu. Thay vì sử dụng lược đồ đơn giản hơn ở mức logic, một số lược đồ phức tạp sẽ được giữ lại do kích thước dữ liệu lớn của cơ sở dữ liệu. Nhiều người sử dụng hệ thống cơ sở dữ liệu không quan tâm đến tất cả thông tin này mà chỉ một phần của cơ sở dữ liệu. Do đó để đơn giản hóa sự tương tác với hệ thống, mức khung nhìn được định nghĩa. Hệ thống có thể cung cấp nhiều khung nhìn trên cùng một cơ sở dữ liệu cho các đối tượng sử dụng khác nhau.

Mối quan hệ giữa ba mức được minh họa qua hình 1.2

1.3 Hệ quản trị cơ sở dữ liệu (Database Management System - DBMS)

1.3.1 Khái niệm

Hệ quản trị cơ sở dữ liệu là một phần mềm tức là một hệ thống các chương trình cho phép người sử dụng giao tiếp với cơ sở dữ liệu như minh họa ở hình 1.3.



Hình 1.3 Giao tiếp giữa người sử dụng với cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu cho phép ta tổ chức cơ sở dữ liệu, lưu trữ nó trên thiết bị ghi nhớ và cung cấp cho chúng ta các thủ tục để sửa đổi cấu trúc cơ sở dữ liệu, cập nhật dữ liệu và truy vấn trên các dữ liệu.

1.3.2 Các chức năng của hệ quản trị cơ sở dữ liệu

Các hệ quản trị cơ sở dữ liệu cung cấp cho chúng ta những chức năng sau:

1) Hỗ trợ một mô hình dữ liệu để tổ chức cơ sở dữ liệu nghĩa là một công cụ để trừu tượng hóa một cách toán học thế giới thực cần quản lý và thông qua đó người sử dụng có thể thấy được các dữ liệu của thế giới thực này.

Ví dụ: Tổ chức thông tin về Sinh viên gồm các thông tin:

Mã sinh viên
Họ tên
Địa chỉ
Năm sinh
Lớp

Mô hình dữ liệu quan hệ cho ta thiết lập một quan hệ :
Sinh viên (masv, họ tên, địa chỉ, năm sinh , lớp).

Tương tự, để tổ chức thông tin về môn học gồm các thông tin:

mã môn học
Tên môn
số tiết

Ta có quan hệ sau: Môn học (mã môn học, tên môn học, sốtiết)

2) Hỗ trợ cho một vài ngôn ngữ lập trình cấp cao cho phép người sử dụng định nghĩa cấu trúc dữ liệu, truy xuất dữ liệu; ngoài ra còn cung cấp một ngôn ngữ để thao tác lên dữ liệu và truy vấn dữ liệu. Các ngôn ngữ đó được gọi là ngôn ngữ hỏi (Query Language), trong đó ngôn ngữ được sử dụng rộng rãi nhất là ngôn ngữ SQL (Structured Query Language)

Ví dụ: Giả sử ta có mô hình dữ liệu quan hệ gồm hai quan hệ sau:

Nhân viên (tên nhân viên ,Phòng)
Phòng ban (Phòng, Trưởng phòng)

Thông tin của hai quan hệ được mô tả ở hình 1.4

Nhân viên		Phòng Ban	
Tên nhân viên	Phòng	Phòng	Trưởng phòng
Lê Văn A	Đào Tạo	Kế hoạch	Phạm Văn F
Trần Thị B	Hành chánh	Kế toán	Nguyễn Thị G
Nguyễn Văn C	Kế toán	Đào Tạo	Lê Thị H
Lê Thị E	Kế hoạch	Hành chánh	Võ Văn T

Hình 1.4 Bảng thể hiện lược đồ quan hệ

* Ai là trưởng phòng của nhân viên Lê Văn A ?

Áp dụng câu lệnh của ngôn ngữ SQL ta trả lời câu hỏi trên như sau::

```
Select      trưởng phòng
From        nhân viên, phòng ban
Where       nhân viên.tên nhân viên = 'Lê Văn A ' and
            nhân viên. phòng = phòng ban.phòng
```

* Cho biết danh sách nhân viên của trưởng phòng Lê Thị H ?

Áp dụng câu lệnh của ngôn ngữ SQL ta trả lời câu hỏi trên như sau::

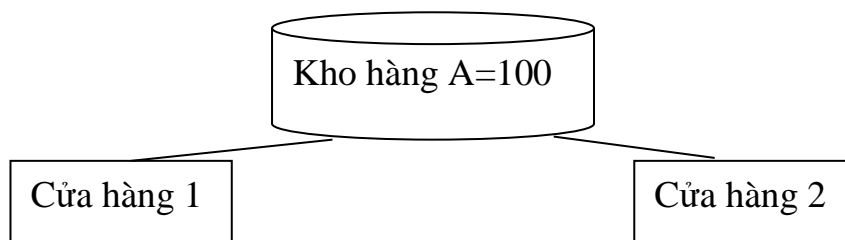
```
Select      Tên nhân viên
From        Nhân viên, phòng ban
Where       Phòng ban.trưởng phòng = 'Lê Thị H' and
            nhân viên.phòng = phòng ban.phòng
```

* Cho biết tên những người thuộc phòng kế toán

```
Select tên nhân viên
From      nhân viên
Where     nhân viên. phòng = 'kế toán'
```

3) Quản lý giao dịch (transaction)

Cứ một lần truy xuất cơ sở dữ liệu được gọi là một giao dịch. Hệ quản trị cơ sở dữ liệu cung cấp công cụ cho phép nhiều người sử dụng truy xuất đồng thời đến cơ sở dữ liệu.



Hình 1.5 Giao tác giữa cửa hàng với kho hàng

Ví dụ: Giả sử cùng lúc hai cửa hàng giao dịch với kho hàng và cần cung cấp 90 mặt hàng A cho cửa hàng 1 và 30 cho cửa hàng B. Nếu không quản lý chặt chẽ thì kho hàng có thể đồng ý cấp cho cả 2 cửa hàng. Do đó phải thực hiện xong giao dịch của một cửa hàng thì giao dịch của cửa hàng còn lại mới được thực hiện. Khi có một giao dịch thay đổi một dữ liệu thì hệ quản trị cơ sở dữ liệu sẽ ngăn cản mọi giao dịch khác truy xuất đến dữ liệu này cho đến khi giao dịch trước đó đã kết thúc.

4) Khả năng bảo vệ và phục hồi dữ liệu : Hệ quản trị cơ sở dữ liệu có khả năng bảo vệ và phục hồi dữ liệu từ các hệ thống bị hư hỏng do các tác nhân:

- Virus
- Chương trình không hoàn chỉnh, thiếu an toàn nên bị hỏng bởi chính người sử dụng.
- Đĩa hư

Cơ chế bảo vệ dữ liệu là backup, nén lại dữ liệu lại thường xuyên và cất chúng vào một thiết bị lưu trữ an toàn đồng thời lưu trữ mọi giao dịch vào một nhật ký. Ngoài ra khi hệ thống bị hỏng hệ quản trị cơ sở dữ liệu cho phép phục hồi lại phần nào các dữ liệu bị mất dựa vào nhật ký giao dịch đó.

5) Điều khiển truy xuất: Hệ quản trị cơ sở dữ liệu có khả năng giới hạn quyền truy xuất dữ liệu của người sử dụng và hơn nữa còn kiểm tra tính hợp lệ của dữ liệu khi đưa vào cơ sở dữ liệu

□ Quyền truy xuất: Hệ quản trị cơ sở dữ liệu cho phép cấp hoặc lấy đi các quyền tham nhập và truy xuất cơ sở dữ liệu cho những người sử dụng như các quyền:

- Không được xem dữ liệu
- Được xem nhưng không được sửa dữ liệu
- Được xem và được sửa dữ liệu

Mỗi người sử dụng được cấp cho một quyền truy xuất và quyền đó được lưu trữ trong 1 bảng phân quyền.

Kiểm tra tính hợp lệ của dữ liệu: hệ quản trị cơ sở dữ liệu cho phép ràng buộc các dữ liệu nhất là các dữ liệu nhập để thể hiện tính toàn vẹn của dữ liệu.

1.3.3 Khái niệm về sự độc lập dữ liệu và chương trình

Trong phần 1.2 chúng ta đã định nghĩa kiến trúc của hệ cơ sở dữ liệu. Kiến trúc ba lớp này cho phép thay đổi cấu trúc ở một lớp mà không ảnh hưởng đến lớp cao hơn kế nó. Điều này được gọi là độc lập dữ liệu. Độc lập dữ liệu và chương trình là cấu trúc dữ liệu dù có thay đổi nhưng chương trình vẫn không thay đổi.

Ta có 2 loại độc lập dữ liệu:

* Độc lập dữ liệu vật lý: là trường hợp sơ đồ vật lý bị thay đổi (nghĩa là đường địa chỉ thư mục dữ liệu bị thay đổi, các cấu trúc tập tin bị thay đổi nhưng sơ đồ ý niệm không thay đổi và như vậy các chương trình ứng dụng cũng không phải thay đổi. Để làm được điều này ta phải thay đổi các phép biến đổi từ sơ đồ vật lý

* Độc lập dữ liệu luận lý: là khi sơ đồ ý niệm thay đổi nhưng sơ đồ ngoài không thay đổi nghĩa là các chương trình ứng dụng không cần phải viết lại, từ đó ta phải thay đổi các phép biến đổi từ sơ đồ ngoài đến sơ đồ ý niệm.

1.3.4 Bộ quản lý cơ sở dữ liệu

Bộ quản lý cơ sở dữ liệu là một module chương trình cung cấp sự giao tiếp giữa dữ liệu ở mức thấp được lưu trữ trong cơ sở dữ liệu với các chương trình ứng dụng. Bộ quản lý cơ sở dữ liệu có nhiệm vụ thực hiện các chức năng được trình bày ở phần 1.3.2

1.3.5 Người quản trị cơ sở dữ liệu (Admin)

Bởi vì hệ thống cơ sở dữ liệu là một hệ thống dữ liệu lớn và được sử dụng thường xuyên và lâu dài do đó phải cần có một người quản lý tập trung cả hệ thống. Người đó được gọi là người quản trị cơ sở dữ liệu. Các chức năng của người quản trị cơ sở dữ liệu bao gồm:

- ✓ Xác định lược đồ cơ sở dữ liệu. Lược đồ cơ sở dữ liệu được tạo ra cho hệ thống sẽ được lưu trữ thường trực trong tự điển dữ liệu.
- ✓ Xác định được cấu trúc lưu trữ dữ liệu và phương thức truy xuất cơ sở dữ liệu.
- ✓ Thay đổi lược đồ và tổ chức vật lý. Các thay đổi lược đồ cơ sở dữ liệu hoặc các tổ chức lưu trữ vật lý mặc dù rất hiếm phải được thực hiện bởi người quản trị cơ sở dữ liệu thông qua ngôn ngữ định nghĩa dữ liệu.
- ✓ Cấp quyền truy xuất dữ liệu cho người sử dụng.
- ✓ Đặc tả các ràng buộc toàn vẹn. Các ràng buộc toàn vẹn được giữ trong một cấu trúc hệ thống đặc biệt và được tham khảo bởi người quản trị cơ sở dữ liệu khi có một sự thay đổi trong hệ thống.

1.3.6 Những người sử dụng cơ sở dữ liệu

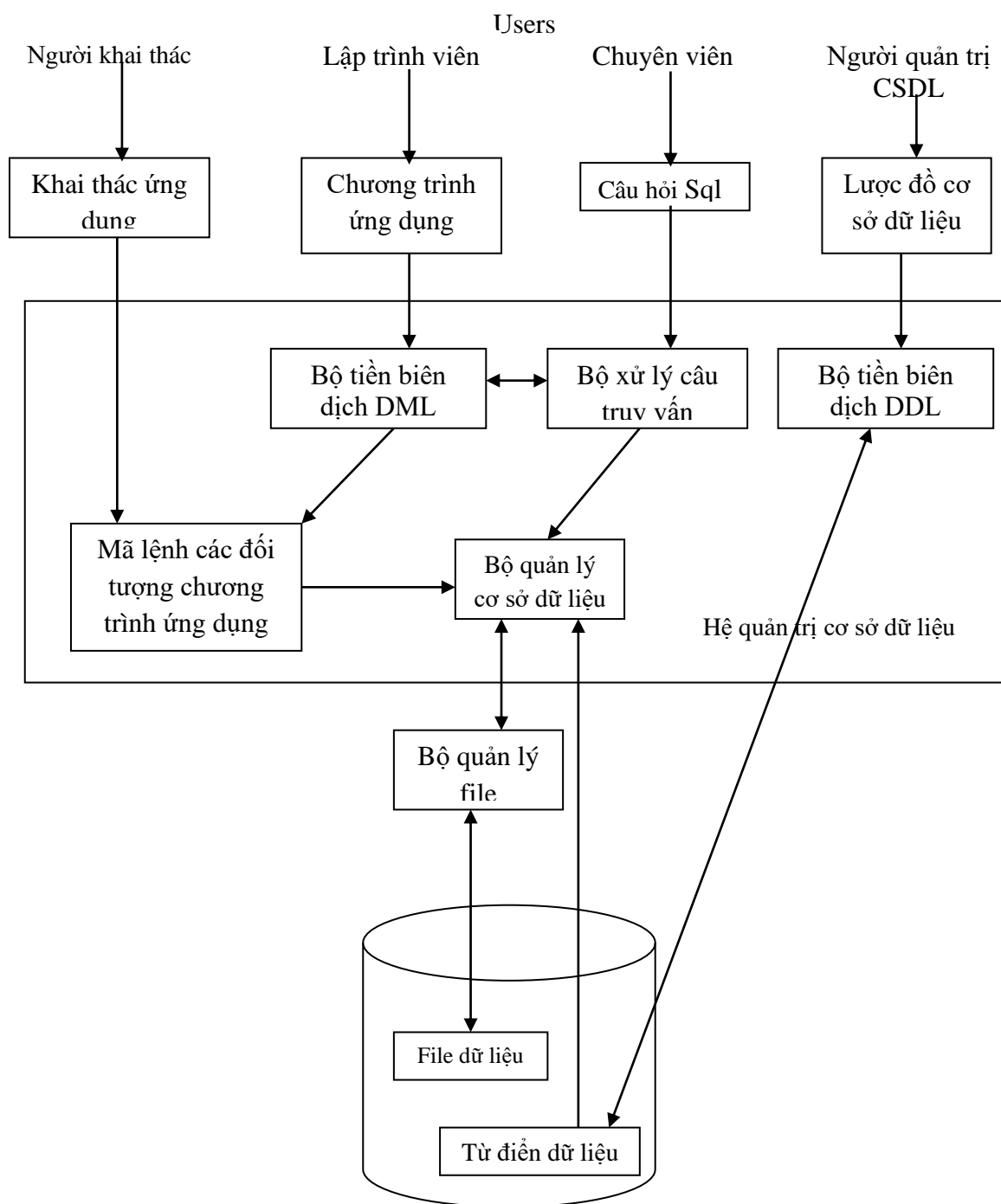
Mục đích chính của hệ thống cơ sở dữ liệu là cung cấp một môi trường cho việc tìm kiếm thông tin cho nhiều người sử dụng. Những người sử dụng cơ sở dữ liệu được chia thành 4 loại sau:

- ✓ Lập trình viên: người viết ra các chương trình ứng dụng cho cơ sở dữ liệu.
- ✓ Các chuyên viên: Những người này không dùng các chương trình ứng dụng để truy cập vào cơ sở dữ liệu. Họ sử dụng ngôn ngữ hỏi để truy cập trực tiếp vào cơ sở dữ liệu.
- ✓ Người khai thác: Người khai thác là người giao tiếp với hệ thống thông qua các chương trình ứng dụng.
- ✓ Người quản trị cơ sở dữ liệu

1.3.7 Cấu trúc tổng quát của hệ thống quản trị cơ sở dữ liệu

- ✓ Hệ quản lý tập tin (File manager): Bộ quản lý tập tin quản lý sự định vị các không gian lưu trữ và các cấu trúc dữ liệu dùng để thể hiện các thông tin lưu trữ trên đĩa.
- ✓ Bộ quản lý cơ sở dữ liệu (Database manager): Cung cấp sự giao tiếp giữa dữ liệu được lưu trữ ở mức thấp trong cơ sở dữ liệu với các chương trình ứng dụng.
- ✓ Bộ xử lý câu truy vấn (Query processor): Bộ này biên dịch các phát biểu trong ngôn ngữ hỏi thành các câu lệnh ở mức thấp sao cho bộ quản lý cơ sở dữ liệu có thể hiểu được. Hơn nữa nó còn tối ưu hoá câu truy vấn sao cho việc thực thi câu hỏi được nhanh hơn.
- ✓ Bộ tiền biên dịch DML (DML precompiler): Bộ tiền biên dịch DML dịch các phát biểu của ngôn ngữ hỏi được nhúng trong ngôn ngữ chủ thành các mã lệnh thích hợp.
- ✓ Bộ biên dịch DDL (DDL compiler): Biên dịch các phát biểu DDL thành một tập các bảng.
- ✓ Các files dữ liệu: lưu trữ chính cơ sở dữ liệu.
- ✓ Tự điển dữ liệu: Chứa các dữ liệu định nghĩa dữ liệu tức toàn bộ các định nghĩa của cơ sở dữ liệu.

Toàn bộ hệ thống quản trị cơ sở dữ liệu được mô tả trong hình 1.6.



Hình 1.6 Cấu trúc hệ thống quản trị cơ sở dữ

CHƯƠNG 2 CÁC MÔ HÌNH DỮ LIỆU

2.1 Mô hình dữ liệu (data model)

2.1.1 Các mô hình dữ liệu (Data models)

Cấu trúc cơ sở của cơ sở dữ liệu là quan niệm về mô hình dữ liệu. Mô hình dữ liệu có một tập các công cụ quan niệm cho việc mô tả dữ liệu, mô tả các mối quan hệ dữ liệu, các ngữ nghĩa dữ liệu và các ràng buộc nhất quán. Nhiều mô hình dữ liệu được đưa ra, chúng được phân thành ba nhóm: các mô hình logic dựa trên đối tượng, các mô hình logic dựa trên mẫu tin và các mô hình dữ liệu vật lý.

2.1.2 Các mô hình logic dựa trên đối tượng (Object-Based logical models)

Các mô hình logic dựa trên đối tượng được sử dụng để mô tả các mức quan niệm và mức tầm nhìn. Chúng được đặc trưng bởi các khả năng cấu trúc hóa linh động và cho phép đặc tả một cách rõ ràng các ràng buộc dữ liệu. Có nhiều loại mô hình khác nhau. Một số các mô hình được biết đến rộng rãi là:

- ☐ Mô hình thực thể liên kết
- ☐ Mô hình hướng đối tượng
- ☐ Mô hình nhị phân
- ☐ Mô hình dữ liệu ngữ nghĩa
- ☐ Mô hình dữ liệu chức năng

Ở đây chúng ta chỉ nghiên cứu mô hình thực thể liên kết và mô hình hướng đối tượng như là các thể hiện của lớp mô hình logic dựa trên đối tượng.

☐ **Mô hình thực thể mối liên kết (Entity-Relationship Model)**

Mô hình thực thể mối liên kết (ER) dựa trên sự nhận thức về thế giới thực bao gồm một tập các đối tượng cơ bản được gọi là thực thể (entity) và các mối liên kết (relationship) giữa các đối tượng này. Một thực thể là một đối tượng mà phân biệt được với những đối tượng khác bởi một tập thuộc tính đặc tả. Ví dụ các thuộc tính *number* và *balance* mô tả một tài khoản cụ thể trong ngân hàng và số dư của nó.

Một mối liên kết là mối quan hệ giữa các thực thể. Ví dụ mối liên kết *CustAcct* liên kết khách hàng với mỗi tài khoản của họ.

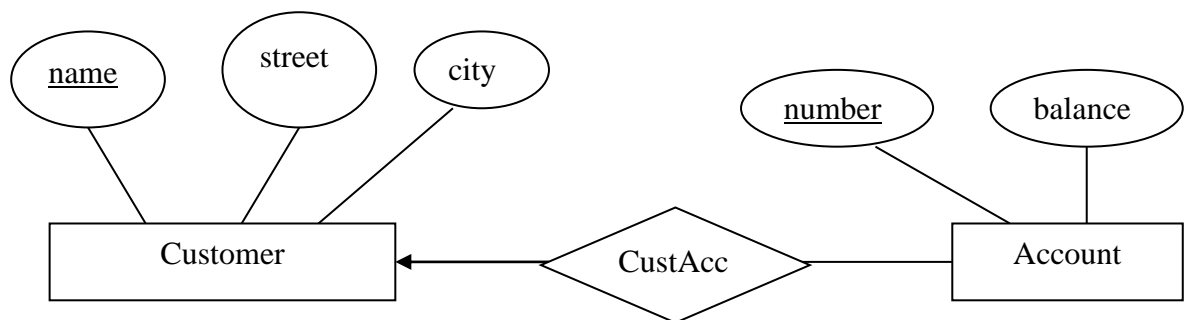
Tập tất cả các thực thể cùng kiểu và tập các mối liên kết có cùng kiểu được gọi là tập thực thể và tập mối liên kết tương ứng.

Bổ sung cho các thực thể và các liên kết, mô hình thực thể mối liên kết ER thể hiện một vài ràng buộc mà nội dung dữ liệu trong cơ sở dữ liệu phải tuân theo. Một trong các ràng buộc quan trọng là ràng buộc số phần tử ánh xạ (mapping cardinalities), nó biểu diễn số lượng các thực thể của tập thực thể này có thể liên kết số thực thể của tập thực thể kia.

Cấu trúc logic của toàn bộ cơ sở dữ liệu có thể được biểu diễn bằng một sơ đồ thực thể mối liên kết ER (ER diagram) bao gồm các thành phần sau:

- ✓ Hình chữ nhật thể hiện tập thực thể.
- ✓ Hình elip thể hiện các thuộc tính.
- ✓ Hình thoi thể hiện mối liên kết giữa các tập thực thể.
- ✓ Đường nối liên kết các thuộc tính với tập thực thể và tập thực thể với mối liên kết.

Mỗi thành phần đều có tên mà thực thể và mối liên kết thể hiện chúng. Để minh họa, ta xét một phần hệ thống cơ sở dữ liệu ngân hàng bao gồm các khách hàng và các tài khoản của họ. Sơ đồ ER tương ứng được minh họa ở hình 2.1



Hình 2.1 Sơ đồ ER

□ Mô hình hướng đối tượng (*The Object-Oriented Model*)

Tương tự mô hình ER, mô hình hướng đối tượng cũng dựa trên một tập các đối tượng. Một đối tượng chứa các giá trị được lưu trong các biến riêng (instance variables) bên trong đối tượng. Không như các mô hình hướng mẫu tin, các giá trị này tự nó là các đối tượng. Do đó, các đối tượng chứa các đối tượng đến một độ sâu lồng vào nhau tùy ý. Một đối tượng cũng chứa phần mã lệnh để điều hành đối tượng. Các phần mã lệnh này được gọi là phương thức (method).

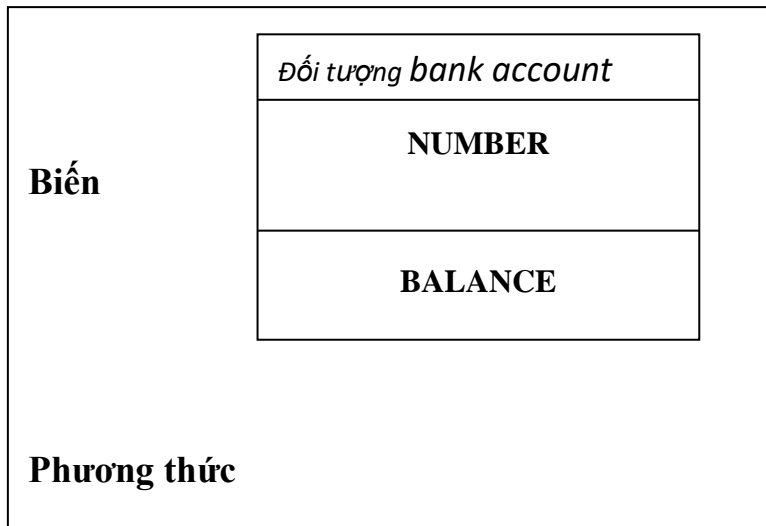
Các đối tượng chứa cùng kiểu giá trị và cùng kiểu phương thức được gom lại trong một lớp (class). Một lớp có thể được xem như một định nghĩa kiểu cho các đối tượng.

Một cách duy nhất mà một đối tượng có thể truy xuất dữ liệu của một đối tượng khác là gọi một phương thức của đối tượng đó. Điều này được gọi là gửi thông báo đến đối tượng.

Để minh họa mô hình hướng đối tượng, ta xét một đối tượng thể hiện tài khoản ngân hàng (*bank account*). Đối tượng này chứa các biến *number* và *balance* thể hiện số tài khoản và ngân khoản của nó. Đối tượng này cũng chứa phương thức chi trả lãi (*pay-interest*) (xem hình 2.2).

Giả sử trước đây ngân hàng chi trả lãi 6% cho tất cả các tài khoản nhưng nay ngân hàng thay đổi cách chi trả lãi như sau: trả lãi 5% đối với các tài khoản có ngân

khoản nhỏ hơn \$1000 và 6% đối với các tài khoản có ngân khoản lớn hơn hay bằng \$1000. Đối với hầu hết các mô hình, sự thay đổi này dẫn đến việc thay đổi mã lệnh trong một hoặc nhiều chương trình ứng dụng. Nhưng trong mô hình hướng đối tượng thì điều này chỉ cần thay đổi trong phương thức *pay-interest* mà thôi.



Hình 2.2 Mô hình một đối tượng

2.1.3 Các mô hình dựa trên mẫu tin (Record-Based Logical Models)

Các mô hình dựa trên mẫu tin được sử dụng để mô tả dữ liệu ở các mức quan niệm và khung nhìn. Ngược lại với các mô hình dựa trên đối tượng, chúng được dùng cho việc đặc tả toàn bộ cấu trúc logic của cơ sở dữ liệu và cung cấp một mô tả mức cao hơn cho việc hiện thực cơ sở dữ liệu vật lý.

Các mô hình được gọi là dựa trên mẫu tin vì cơ sở dữ liệu được xây dựng theo dạng các mẫu tin xác định. Mỗi kiểu mẫu tin định nghĩa một số các trường hoặc các thuộc tính và mỗi trường thường có chiều dài cố định.

Các mô hình dựa trên mẫu tin không chứa một cơ chế cho việc thể hiện các mã lệnh trực tiếp trong cơ sở dữ liệu. Vì thế nó có hai ngôn ngữ riêng rẽ nhưng gắn liền với mô hình để biểu diễn các truy vấn cơ sở dữ liệu và cập nhật cơ sở dữ liệu.

Ba mô hình dữ liệu dựa trên mẫu tin được chấp nhận rộng rãi nhất là mô hình quan hệ, mô hình mạng và mô hình phân cấp. Mô hình quan hệ là mô hình dành được sự ưa chuộng hơn hai mô hình kia trong những năm gần đây, nó sẽ được nghiên cứu sâu trong giáo trình này. Mô hình mạng và mô hình phân cấp chỉ còn được sử dụng trong một số các cơ sở dữ liệu cũ.

□ Mô hình quan hệ

Mô hình quan hệ thể hiện dữ liệu và mối quan hệ giữa chúng bằng một tập các bảng. Mỗi bảng chứa một số cột với tên duy nhất.

Để biểu diễn mô hình dữ liệu quan hệ với người dùng bảng, trong đó:

1 cột thì tương ứng với một thuộc tính.

1 hàng thì tương ứng với 1 bộ (record).

1 bảng thì tương ứng với 1 quan hệ (file).

Bảng \Leftrightarrow quan hệ

Cột \Leftrightarrow thuộc tính

Ví dụ : Để minh họa cho mẫu cơ sở dữ liệu thể hiện khách hàng (*customer*) và tài khoản (*account*) của họ, ta có hai bảng thể hiện sau (hình 2.3):

<i>name</i>	<i>street</i>	<i>city</i>	<i>number</i>
Lowery	Mapple	Queens	900
Shiver	North	Bronx	556
Shiver	North	Bronx	647
Hodges	Sidehill	Brooklyn	801
Hodges	Sidehill	Brooklyn	647

<i>number</i>	<i>balance</i>
900	55
556	100000
647	105366
801	10533

Hình 2.3 Một mẫu cơ sở dữ liệu quan hệ

2.2 Mô hình thực thể mối liên kết (Entity Relationship Model)

Mô hình thực thể là mô hình ban đầu để từ đó người ta biến đổi nó thành một trong 3 mô hình dữ liệu, nó cho phép mô tả sơ đồ ý niệm của thế giới thực mà không quan tâm đến hiệu quả hoặc thiết kế cơ sở dữ liệu vật lý. Sơ đồ thực thể mối liên kết (Entity Relationship Mode Diagram) sau này sẽ được biến đổi thành một sơ đồ quan niệm của một trong ba mô hình dữ liệu dựa theo mẫu tin ở trên. Trong giáo trình này ta chỉ nghiên cứu tới việc biến đổi mô hình thực thể mối liên kết thành mô hình quan hệ.

Mô hình thực thể mối liên kết gồm có 2 thành phần : thực thể và liên kết

2.2.1 Thực thể (Entity)

Thực thể là 1 sự vật hay sự việc mà ta có thể phân biệt được, tồn tại trong thế giới khách quan mà ta có thể liên kết thực thể này với 1 thực thể khác. Thực thể có thể là một thực thể cụ thể hoặc trừu tượng. Mỗi thực thể được mô tả bởi một số thuộc tính.

Ví dụ : - Sinh viên là 1 thực thể cụ thể. Ta có thể phân biệt sinh viên này với sinh viên khác dựa vào mã số, ta có thể liên kết sinh viên này với sinh viên kia dựa vào mối liên kết cùng lớp.

- Môn học là 1 thực thể trừu tượng, ta có thể phân biệt môn học này với môn học khác dựa vào tên môn học, ta có thể liên kết môn học này với môn học kia dựa vào liên kết môn học nào là cơ sở của môn học kia.

2.2.2 Tập thực thể (Entity set) :

Tập thực thể là một tập bao gồm những thực thể giống nhau.

Ví dụ : - Nhiều thực thể sinh viên hợp lại thành tập thể sinh viên.
- Nhiều môn học hợp lại tạo ra tập thể môn học.

2.2.3 Các thuộc tính và các khoá :

Thuộc tính: Các tập thực thể thì có những đặc tính, những đặc tính này gọi là thuộc tính.

Ví dụ : Tập thực thể Mặt hàng có các thuộc tính: tên mặt hàng, số lượng.

Mỗi thuộc tính có một kiểu dữ liệu (số nguyên, số thực, các chuỗi ký tự ..) và có giá trị trong một miền giá trị.

Ví dụ : Mã vật tư là một chuỗi các ký tự.

- Số lượng là 1 số nguyên.
- Khối lượng là 1 số thực.

Khóa là một thuộc tính hoặc là một tập các thuộc tính mà giá trị của nó được dùng để xác định duy nhất 1 thực thể trong một tập thực thể.

Ví dụ : Để phân biệt các sinh viên người ta có thể dùng khóa sau:

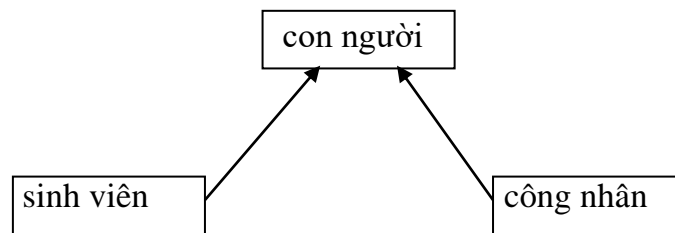
- Mã sinh viên có 1 thuộc tính mà người ta gọi là khóa đơn (single key).
- Dùng họ tên và năm sinh có nhiều thuộc tính, khóa này gọi là khóa tổ hợp.

- Dùng mã sinh viên và họ tên để làm khóa thì khoá này được gọi là siêu khoá (*super key*). Siêu khóa là 1 khóa mà có tập con các thuộc tính khóa của nó lại là khóa.

2.2.4 Hệ phân cấp isa hay sự tổng quát hoá - chuyên biệt hóa

Ta nói A isa B được đọc là “A là B” nghĩa là nếu tập thực thể B là sự tổng quát hóa của tập thực thể A hay nói cách khác đi A là 1 loại riêng biệt của B.

Ví dụ : Sinh viên Isa con người.



Ta nói:

$$A \text{ Isa } B \Leftrightarrow \forall a \in A \text{ thì } a \in B \text{ và } \exists b \in B \text{ sao cho } a = b$$

Tính chất của hệ phân cấp Isa:

- A thừa hưởng mọi thuộc tính của B.
- A có những thuộc tính mà B không có.
- B có tồn tại 1 thực thể mà không tương ứng 1 thực thể trong A.

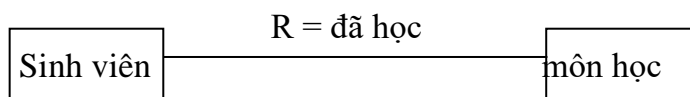
Ví dụ : Sinh viên chứa mọi thuộc tính của Con người, khi đó thuộc tính khóa của Con Người cũng sẽ là thuộc tính khóa của Sinh viên, và các thuộc tính riêng của Sinh viên trong đó có thuộc tính nhận dạng thực thể sinh viên. Và trong Con người chắc chắn tồn tại thực thể không phải là Sinh viên mà là Công nhân.

2.2.5 Mối liên kết

Một mối liên kết giữa các tập thực thể là 1 danh sách có thứ tự của các tập thực thể. Nếu có 1 mối liên kết R giữa các tập thực thể E_1, E_2, \dots, E_k thì 1 minh họa của R là 1 tập hợp các bộ k. Ta gọi 1 tập hợp như thế là 1 mối liên kết.

Một bộ k (e_1, e_2, \dots, e_k) trong 1 tập mối liên kết R gồm các thực thể (e_1, e_2, \dots, e_k) với $e_i \in E_i, i = 1 \dots k$.

Ví dụ : Xét mối liên kết giữa tập thực thể Sinh viên và môn học.

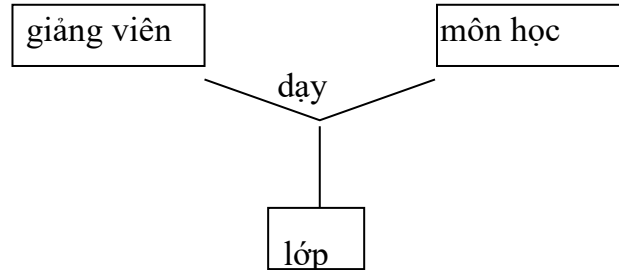


Đã học (sinh viên, môn học)

Bộ (x, y) : x đã học môn y
x thuộc sinh viên

y thuộc môn học

Ví dụ : Xét mối liên kết giữa môn học, giảng viên, lớp.



dạy (giảng viên, môn học, lớp)

Bộ (g, m, p) \Leftrightarrow gv g dạy môn học m ở lớp p

Mỗi liên kết sẽ chứa các thuộc tính khóa của các các tập thực thể tham gia liên kết (còn gọi là thuộc tính khóa vay mượn) và các thuộc tính riêng đặc tả cho mỗi liên kết đó.

Ví dụ: Các thuộc tính khóa vay mượn:

Sinh viên có khóa là mã SV.

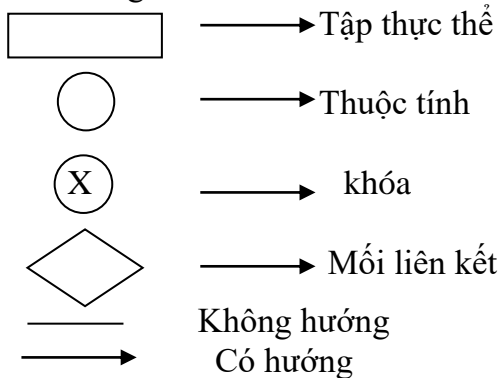
Môn học có khóa là mã MH.

Mỗi liên kết “Đã học” liên kết hai tập thực thể Sinh viên và Môn học sẽ có khóa là mã SV và mã MH. Vậy khóa của chúng là khóa vay mượn của các tập thực thể Sinh viên và Môn học.

2.2.6 Sơ đồ thực thể mối liên kết

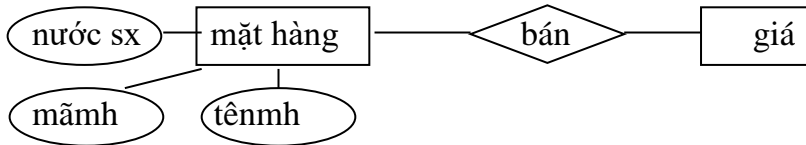
Qui ước :

- Hình chữ nhật tương ứng với tập thực thể.
- Hình tròn tương ứng với thuộc tính.
- Hình tròn mà trong đó có gạch dưới ta gọi đó là thuộc tính khóa.
- Hình thoi dùng để thể hiện mối liên kết.
- Gạch nối giữa hình chữ nhật và hình thoi có thể có hướng hoặc không hướng.

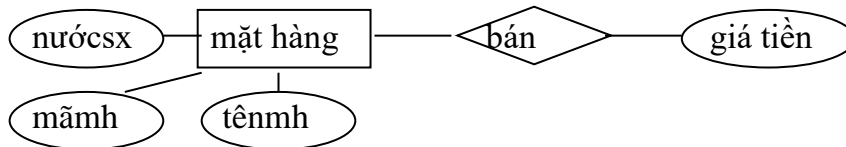


Đặc biệt : nếu 1 tập thực thể chỉ có một thuộc tính thì ta nên gọi tên tập thực thể bằng tên thuộc tính và tập thực thể đó được ghi trong hình tròn.

Ví dụ: Mặt hàng được bán với nhiều giá

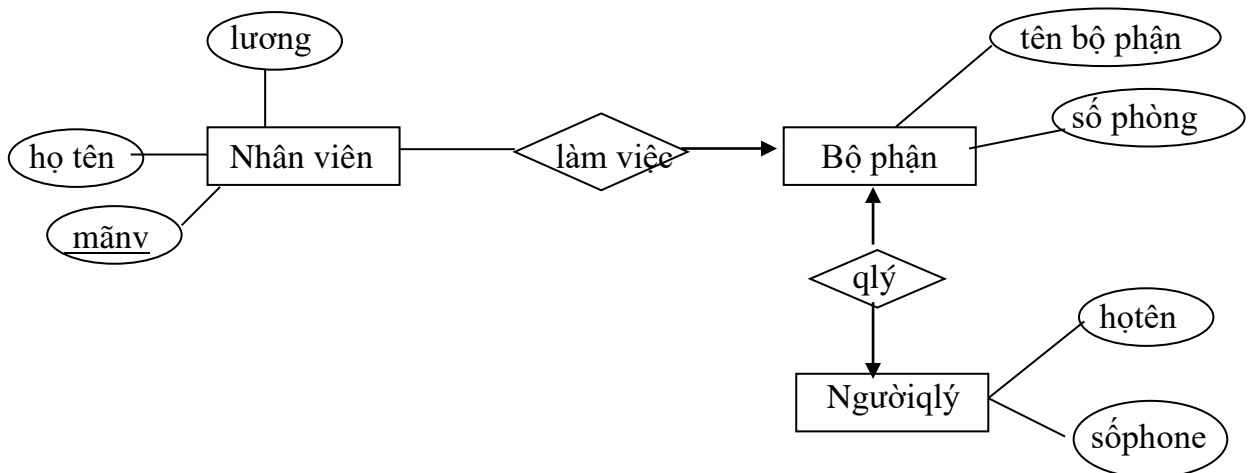


- Giá chỉ có 1 thuộc tính là đơn giá nên ta vẽ hình chữ thành hình tròn.



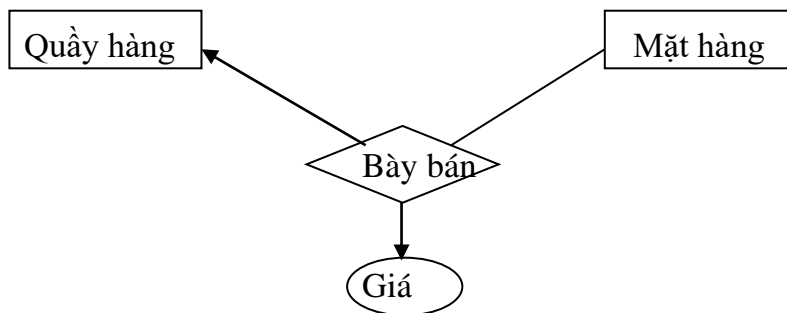
Ví dụ : Một công ty có nhiều bộ phận. Mỗi bộ phận có nhiều nhân viên, nhưng nhân viên chỉ làm việc ở 1 bộ phận. Mỗi bộ phận có 1 người quản lý và người quản lý chỉ quản lý 1 bộ phận mà thôi. Hãy trình bày sơ đồ thực thể mối liên kết công ty trên.

Tập thực thể	Thuộc tính	Mối liên kết	Tập thực thể
Bộ phận	- tên bộ phận - số phòng	có (nhiều) bị quản lý (bởi 1)	Nhân viên Người quản lý
Nhân viên	mã NV họ tên lương	làm việc (tại 1)	Bộ phận
Người quản lý	họ tên số phone	quản lý (1)	Bộ phận

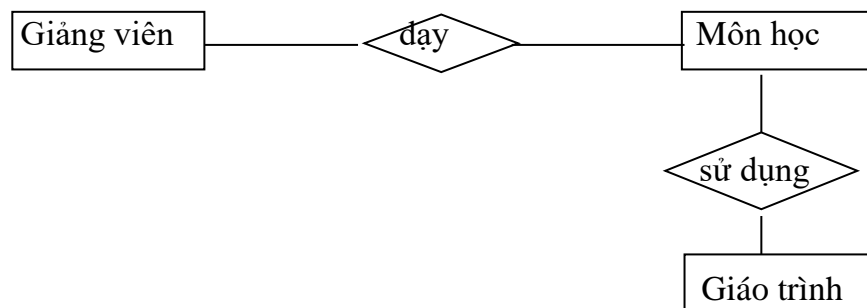


Ví dụ : Một cửa hàng bày bán mặt hàng trong nhiều quầy hàng. Một quầy hàng thì bán nhiều mặt hàng nhưng 1 mặt hàng chỉ được bày bán ở 1 quầy hàng với 1 giá cố định.

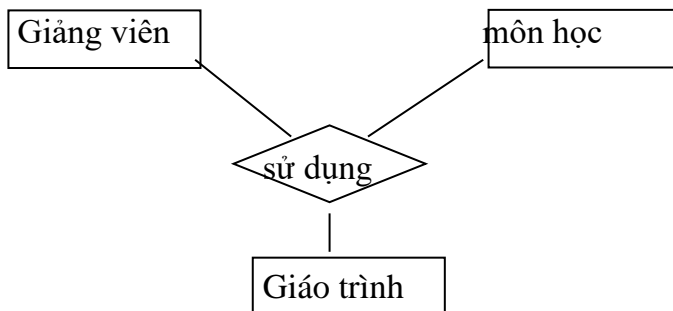
Tập thực thể	Thuộc tính	Mối liên kết	Tập thực thể
Quầy hàng	- tên - số	bày (nhiều)	mặt hàng
Mặt hàng	- tên - mã hàng	bán (ở một) với một giá (thuộc tính riêng)	quầy hàng



Ví dụ : Một trường có nhiều giảng viên, các giảng viên dạy nhiều môn học, môn học có thể được nhiều giảng viên dạy. Mỗi giảng viên dạy môn học thì sử dụng nhiều giáo trình.



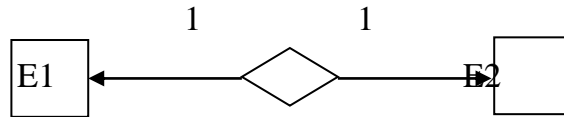
Câu hỏi : Cho biết thầy Hạnh dạy môn cơ sở dữ liệu thì sử dụng giáo trình gì thì mô hình trên sẽ không trả lời được câu hỏi trên. Phải sử dụng mô hình liên kết 3.



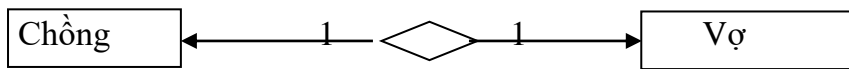
2.2.7 Tính hàm của mối liên kết

Ta cần phải phân loại các mối liên kết dựa vào tính chất là 1 thực thể của tập thực thể liên kết được với bao nhiêu thực thể của tập kia, do đó giá trị của tính chất hàm chỉ có 2 giá trị đó là 1 hoặc nhiều.

2.2.7.1 Mối liên kết 1-1 : là mối liên kết giữa 2 tập thực thể mà ứng với 1 thực thể của tập này liên kết nhiều nhất với 1 thực thể của tập kia và ngược lại.

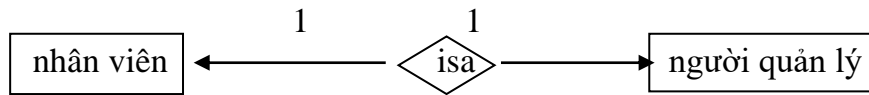


Ví dụ : Tập thực thể chồng và tập thực thể vợ là liên kết 1-1 theo quan điểm hiện đại.



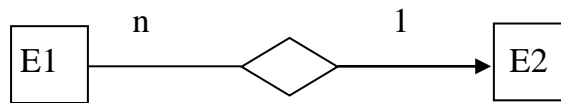
- Mỗi liên kết isa cũng là mỗi liên kết 1-1

Nhân viên nếu là 1 người quản lý thì nhân viên chính là người quản lý.



2.2.7.2 Mỗi liên kết nhiều -1 :

Xét mỗi liên kết nhiều -1 từ tập thực thể E1 và tập thực thể E2 thì 1 thực thể của tập thực thể E2 liên kết với 0 hoặc nhiều thực thể của tập E1 nhưng ngược lại, 1 thực thể của tập E1 chỉ liên kết nhiều nhất với 1 thực thể của tập E2.



Ví dụ : Xét tập thực thể cha mẹ và con cái.

Cha mẹ có thể có nhiều con cái, nhưng con cái chỉ có 1 cặp cha mẹ.

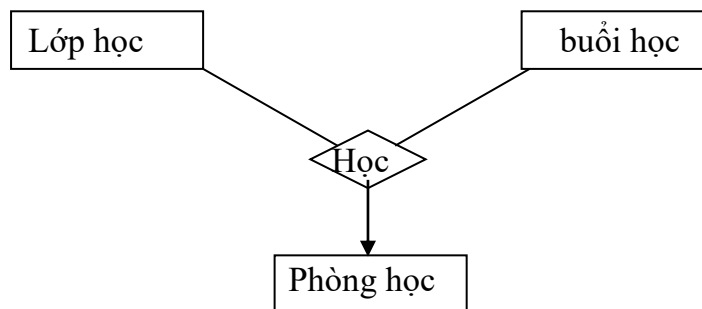
□ Khái niệm về mỗi liên kết nhiều -1 :

Từ 2 tập thực thể có thể được tổng quát hoá thành nhiều mỗi liên kết nhiều -1 với k tập thực thể ($k > 2$)

Nếu có 1 mỗi liên kết giữa các tập thực thể $E_1, E_2, E_3, \dots, E_k$ gọi là R ứng với các thực thể $e_1, e_2, e_{i-1}, e_{i+1}, \dots, e_k$ của các tập thực thể $E_1, E_2, \dots, E_{i-1}, E_{i+1}, \dots, E_k$ thì liên kết với nhiều nhất 1 thực thể e_i của tập E_i thì ta nói R là mỗi liên kết nhiều 1.

Chú ý : Khi ta xây dựng 1 mỗi liên kết giữa k tập thực thể điều đó chứng tỏ rằng ta chỉ xác định được 1 thực thể e_i nào đó nếu đã biết k-1 thực thể của các tập thực thể kia.

Ví dụ : Có 3 tập thực thể lớp học, buổi học và phòng học.

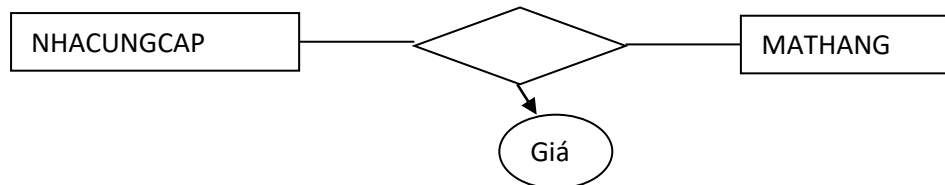


2.2.7.3 Mỗi liên hệ nhiều - nhiều

Một mỗi liên kết được gọi là nhiều - nhiều giữa các tập thực thể nếu như không có 1 giới hạn nào đó về số lượng của các tập thực thể tham gia trong mỗi liên kết nghĩa

là 1 thực thể của tập thể này có thể liên kết được với 0 hoặc nhiều thực thể của tập thực thể kia và ngược lại.

Ví dụ : Người cung cấp có thể cung cấp nhiều mặt hàng, và 1 mặt hàng có thể do nhiều người cung cấp; nhưng cùng 1 mặt hàng có thể có giá khác nhau bởi các nhà cung cấp khác nhau:



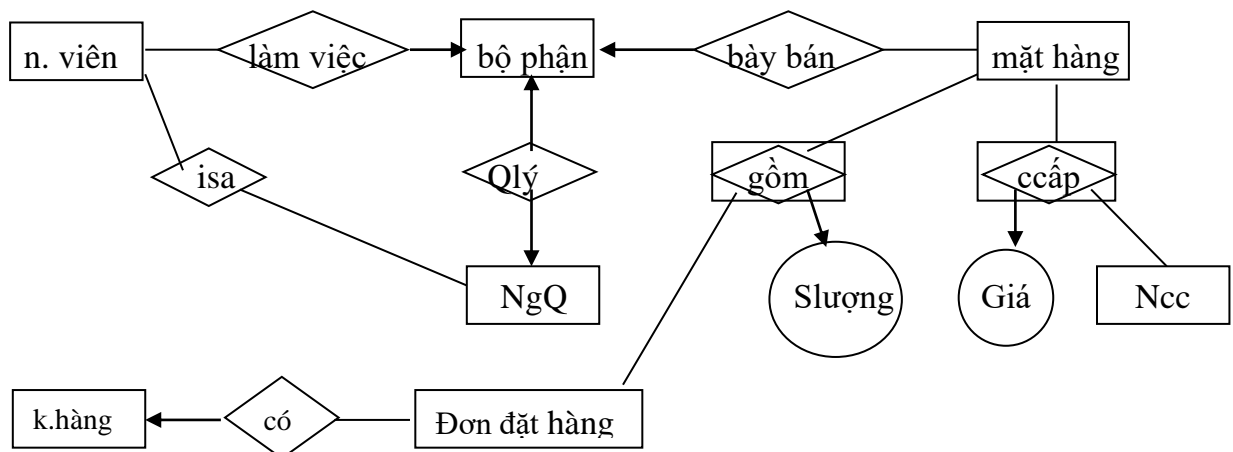
2.2.8 Ví dụ về sơ đồ thực thể mối liên kết (ERD)

Ví dụ 1: Xây dựng 1 sơ đồ ERD của siêu thị.

Một siêu thị có nhiều nhân viên. Thông tin về nhân viên: MaNV, TênNV, Luong. Mỗi nhân viên làm việc ở 1 bộ phận, 1 bộ phận có thể có nhiều nhân viên. Thông tin về bộ phận: MaBP và tênBP. Trong các nhân viên có người làm quản lý, một người chỉ quản lý tối đa 1 bộ phận, và 1 bộ phận chỉ có 1 người quản lý.

Ở 1 bộ phận có nhiều mặt hàng , nhưng 1 mặt hàng chỉ được đặt ở 1 bộ phận mà thôi. Thông tin về mặt hàng: MSMH, TenMH. 1 mặt hàng do nhiều người cung cấp bán, 1 người cung cấp chỉ thể cung cấp nhiều mặt hng, mỗi mặt hng sẽ chỉ giá khc nhau. Thông tin về người cung cấp: MaNCC,TenCC, Diachi.

Siêu thị có nhiều khách hàng. Thông tin về khách hàng: MAKH, TenKH, Diachi. Mỗi khách hàng có nhiều đơn đặt hàng nhưng 1 đơn đặt hàng thì chỉ có 1 khách hàng mà thôi. Thông tin về đơn đặt hàng: SoHieuDDH, NgayDH. Mỗi đơn đặt hàng bao gồm nhiều mặt hàng với mỗi mặt hàng có 1 số lượng mua và đơn giá xác định.

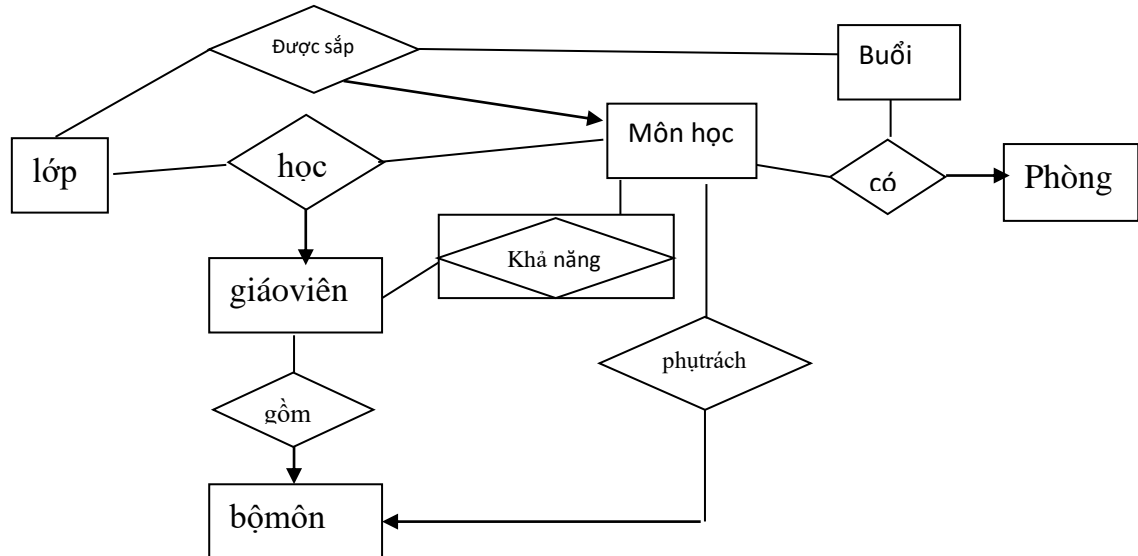


Ví dụ 2: Một trường có nhiều lớp, một lớp học thì học nhiều môn, mỗi môn trong 1 lớp chỉ có 1 người dạy. Với 1 lớp, 1 môn có thể học ở nhiều buổi; 1 lớp, 1 buổi có thể học nhiều môn; và 1 môn học ở 1 buổi chỉ học ở 1 phòng.

Một trường có nhiều bộ môn, một bộ môn thì có nhiều giáo viên nhưng 1 giáo viên chỉ thuộc 1 bộ môn.

1 bộ môn thì đảm nhận nhiều môn học và 1 môn học chỉ thuộc 1 bộ môn. Một giáo viên có khả năng dạy nhiều môn, một môn thì có thể có nhiều giáo viên dạy.

Hãy trình bày mô hình ERD ở trường trên.



Tìm lỗi sai trong ERD trên ??

2.3 Mô hình dữ liệu quan hệ.

2.3.1 Các khái niệm cơ bản.

2.3.1.1 Thuộc tính

- ✓ Mỗi đối tượng được khảo sát đều có những đặc tính. Những đặc tính này được gọi là thuộc tính.
- ✓ Mỗi thuộc tính đều thuộc một kiểu dữ liệu.
- ✓ Mỗi thuộc tính chỉ lấy giá trị trong một tập con của kiểu dữ liệu được gọi là miền (domain) giá trị của thuộc tính, ký hiệu là Dom (A) với A là thuộc tính.
- ✓ Miền giá trị có thể chứa thêm một giá trị đặc biệt gọi là giá trị rỗng (NULL).

2.3.1.2 Lược đồ quan hệ

- ✓ Một lược đồ quan hệ được đặc trưng bởi một tên phân biệt (Q) và một tập hợp hữu hạn các thuộc tính của lược đồ quan hệ đó.
- ✓ Tập hợp các thuộc tính của lược đồ quan hệ Q, ký hiệu là $Q^+ = \{A_1, A_2, \dots, A_N\}$; với các miền giá trị DOM (A_i).

- ✓ Số phần tử của Q^+ , ký hiệu $\text{Card}(Q^+)$ được gọi là số ngôi của lược đồ quan hệ.
- ✓ Thứ tự các thuộc tính trong Q không quan trọng.
- ✓ Một lược đồ quan hệ Q với tập thuộc tính $Q^+ = \{A_1, A_2, \dots, A_n\}$, cũng có thể viết gọn thành $Q(A_1, A_2, \dots, A_n)$.
- ✓ Tên từ của lược đồ quan hệ Q , ký hiệu $\|Q\|$ dùng để mô tả ý nghĩa (ngữ nghĩa) của lược đồ quan hệ Q .

Ví dụ:

- SINHVIÊN (MASV, HOSV, TENSV, NGSINH, MALOP, HOCBONG).
 - Tên từ $\|SINH VIÊN\|$: Mỗi sinh viên được nhận diện qua MASV, phải có họ, tên, ngày sinh, lớp xác định và có thể có học bổng.
- KETQUA (MASV, MAMH, LANTHI, DIEM).

Tên từ $\|KET QUA\|$: Mỗi sinh viên có thể dự thi các môn đã đăng ký. Mỗi môn được thi tối đa 2 lần.

- ✓ Lược đồ quan hệ có thể biểu diễn cho một loại thực thể cụ thể hoặc trừu tượng.
- ✓ Một lược đồ cơ sở dữ liệu C bao gồm nhiều lược đồ quan hệ con :

$$C = \{ Q_i \mid i : 1 \rightarrow n$$

2.3.1.3. Bộ (tuple)

- ✓ Một bộ là các thông tin của 1 đối tượng thuộc 1 lược đồ quan hệ. Nó còn được gọi là 1 mẫu tin hay bản ghi.
- ✓ Về mặt hình thức một bộ q là 1 vectơ gồm n thành phần thuộc tập con của tích Descartes giữa các miền giá trị của các thuộc tính và thỏa mãn tên từ $\|Q\|$.

$$q = (a_1, a_2, \dots, a_n) \in \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n).$$

2.3.1.4. Quan hệ (Relation)

Một quan hệ T_Q định nghĩa trên 1 lược đồ quan hệ Q là 1 thể hiện (hay 1 tình trạng) của lược đồ quan hệ Q ở 1 thời điểm nào đó. Khi đó T_Q chứa các bộ q thỏa tên từ của lược đồ quan hệ Q .

$$T_Q = \{ q = (a_1, a_2, \dots, a_n) \mid a_i \in \text{Dom}(A_i) \text{ và } q \text{ thỏa } \|q\| \}$$

Một quan hệ còn được gọi là bảng (table).

Ví dụ : Quan hệ Sinh viên

MASV	HOSV	TENSV	NGSINH	MALOP	HOCBONG
C981001	Nguyễn Văn	Anh	12/8/1976	C98CQCN01	
C981002	Lê Văn	Anh	1/5/1977	C98CQCN01	1000000
C981003	Trần Thị	Bé	6/7/1976	C98CQCN01	1000000

Một lược đồ cơ sở dữ liệu $C = \{Q_i\}$ là tập hợp các lược đồ quan hệ con. T_{Q_i} định nghĩa trên những lược đồ quan hệ con Q_i tại 1 thời điểm gọi là tình trạng của lược đồ cơ sở dữ liệu C và ký hiệu là $T_C = \{T_{Q_i}\}$

2.3.1.5. Siêu khoá - Khoá chính.

□ Khái niệm :

Một tập con S của Q^+ được gọi là siêu khoá của lược đồ quan hệ Q nếu S có thể dùng làm cơ sở để phân biệt 2 bộ khác nhau tùy ý trong 1 quan hệ T_Q bất kỳ được định nghĩa trên lược đồ quan hệ Q .

□ Định nghĩa

Cho quan hệ T_Q . Tập con $K \subset Q^+$ được gọi :

- 1/ Là siêu khoá của Q nếu $\forall t, t' \in T_Q$ ta có $t(K) = t'(K) \Rightarrow t = t'$. Trong đó $t(K)$ là bộ t nhưng chỉ xét các thuộc tính K .
- 2/ Là 1 khoá của Q nếu K là siêu khoá và không có tập con thật sự nào của K thoả tính chất này. K còn được gọi là khoá chính.

Ví dụ: SINHVIEN(MASV, HOSV, TENSX, NGSINH, MALOP, HOCBONG)

Lược đồ quan hệ SINHVIEN có các siêu khoá sau :

$S1 = \{MASV\}$

$S2 = \{MASV, HOSV, TENSX\}$

$S3 = \{MASV, NGSINH\}$

Trong đó $S1$ là khoá chính; còn $S2, S3$ là siêu khoá.

Trong trường hợp lược đồ quan hệ Q có nhiều khoá chính, người sử dụng sẽ chọn một trong các khoá đó làm khoá chính, các khoá còn lại gọi là khoá tương đương.

Các thuộc tính tham gia vào khoá được gọi là thuộc tính khoá và khi liệt kê khoá trong 1 lược đồ quan hệ sẽ được gạch dưới, ngược lại là thuộc tính không khoá.

- Quy ước :
- Khoá không chứa giá trị rỗng.
 - Không nên sửa đổi giá trị của thuộc tính khoá.

□ Giải thuật : kiểm tra khoá của một thể hiện T_Q

Giải thuật : Satisfy_key (T_Q, K)

Vào : T_Q và khoá $K \subset Q^+$

Ra : Đúng hoặc sai

Các bước :

1. $T'_Q := \{t(K) | t \in T_Q\}$
2. Return Card(T_Q) = Card(T'_Q).

2.3.2 Sự chuyển hóa từ sơ đồ ER sang mô hình dữ liệu quan hệ.

Để biến đổi biểu đồ ERD thành các quan hệ thì ta phải xác định các loại quan hệ. Có 2 loại quan hệ : quan hệ thực thể và quan hệ mối liên kết.

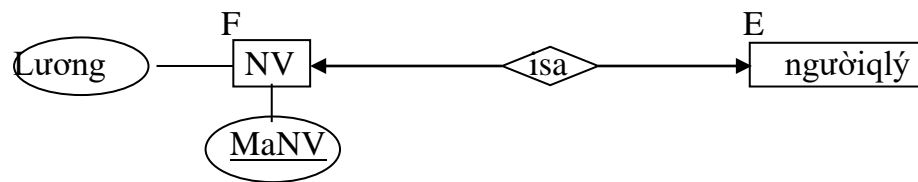
□ **Xây dựng quan hệ thực thể :**

Một tập thực thể E có thể được biểu diễn bởi 1 quan hệ mà lược đồ quan hệ bao gồm tất cả các thuộc tính của tập thực thể. Quan hệ này là quan hệ thực thể.

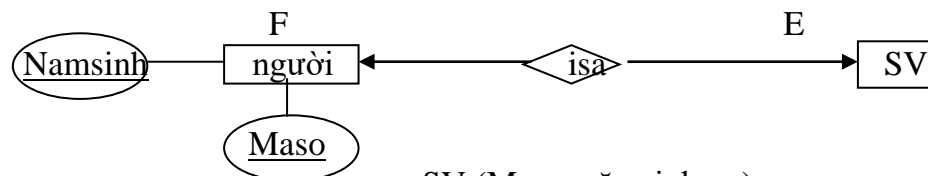
Ví dụ : Tập thực thể khách hàng (Tênkh, Diachi, Sodu) suy ra quan hệ thực thể khách hàng gồm có các thuộc tính tênkh, Diachi, sodu.

* **Chú ý :** Nếu E là tập thực thể chuyên biệt hóa bởi tập thực thể F nào đó thì lược đồ quan hệ của E có chứa các thuộc tính của F mà chúng được dùng làm khóa.

Ví dụ :



Quan hệ E (Manv, ...)



SV (Maso, nămsinh,...)

□ **Xây dựng quan hệ liên kết.**

Một mối liên kết R giữa các tập thực thể E₁, E₂,...E_n được biểu diễn bằng quan hệ mà sơ đồ quan hệ bao gồm các thuộc tính dùng trong khóa cho mỗi tập E₁, E₂,... E_n.

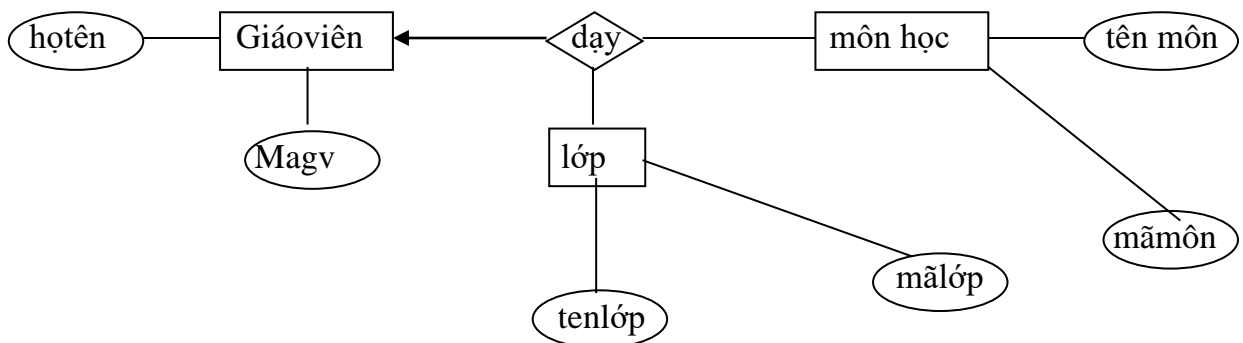
Ví dụ : Cho các thực thể:

Giaovien (magv, tengv),

MonHoc (mamon, tenmon, sotiet)

Lop (malop, tenlop)

- Giáo viên X dạy môn Y cho lớp Z, và 1 môn trên 1 lớp chỉ có 1 giảng viên dạy.



Quan hệ liên kết Dạy có lược đồ quan hệ sau: Dạy(magv, mamon,malop)

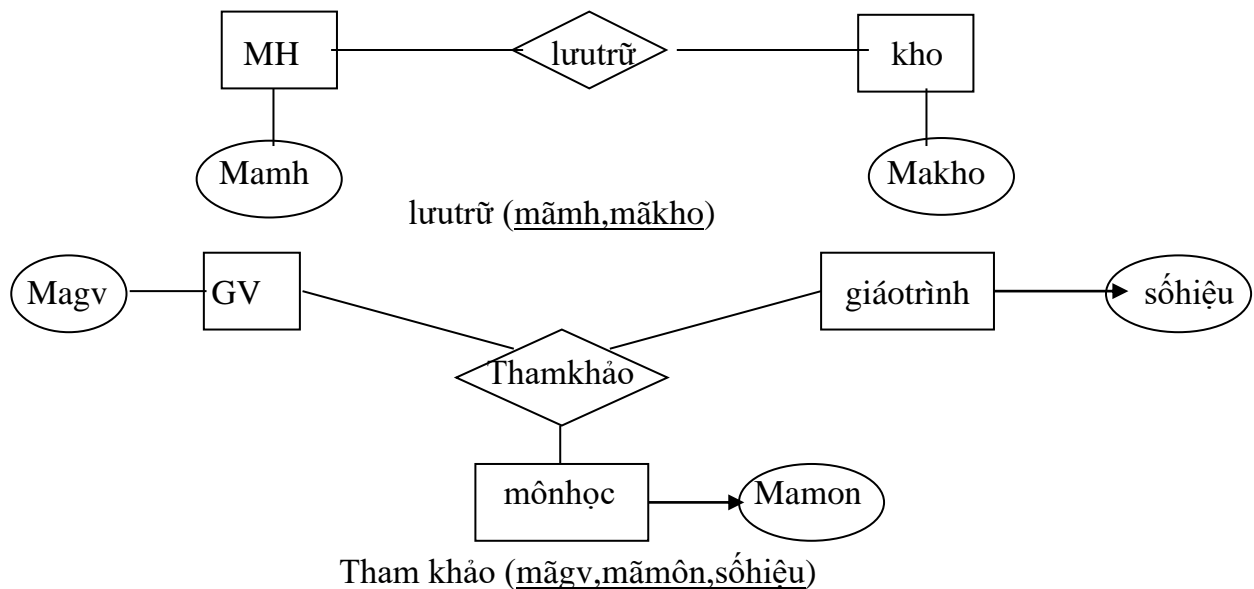
Các khoá của quan hệ.

Quy tắc xác định khóa :

Để xác định khóa của 1 quan hệ ta phải dựa vào ý nghĩa của quan hệ đó chứ không nên dựa vào minh họa của quan hệ. Từ ý nghĩa của các quan hệ thực thể, quan hệ liên kết một-một, nhiều-một, nhiều-nhiều, ta rút ra được 1 số nguyên tắc sau :

- Nếu 1 quan hệ được xác định từ 1 tập thực thể thì tập các thuộc tính khóa của tập thực thể cũng chính là tập các thuộc tính khóa của quan hệ.
- Nếu 1 quan hệ được xác định từ 1 mối liên kết nhiều-nhiều thì khóa dùng cho quan hệ liên kết là tất cả các thuộc tính khóa của các quan hệ đầu nhiều.

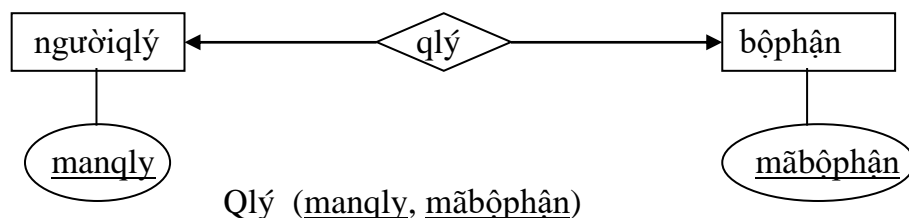
Ví dụ : Xét mỗi quan hệ:



- Khóa của mỗi liên kết từ quan hệ một-một.

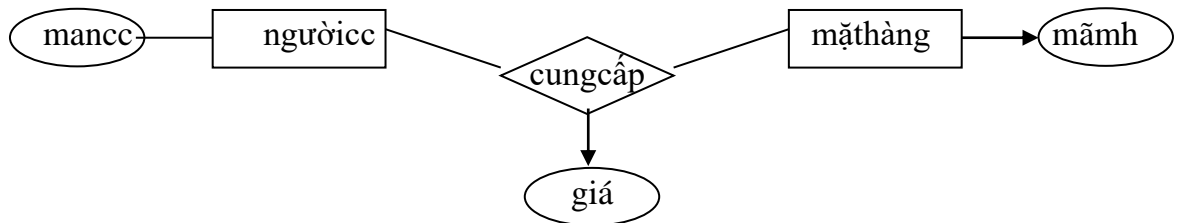
Nếu 1 quan hệ được xác định từ mỗi liên kết một-một giữa các tập thực thể E và F thì khóa dùng chung cho E và F là những khóa dùng cho quan hệ (1 trong 2 khóa đó là khóa)

Ví dụ : Giữa QuảnLý và BộPhận.



- Khóa của mỗi liên kết nhiều-một.

Một quan hệ được xác định từ mỗi liên kết nhiều- một từ các tập E_1, E_2, \dots, E_{k-1} vào tập E_k thì khóa của quan hệ là các thuộc tính khóa của các quan hệ thực thể nhiều.



Cung cấp (mancc, mãmh, giá)

Kết luận : Khóa của các quan hệ liên kết là khóa của các quan hệ thực thể theo đầu nhiều.

□ **Chuyển đổi ERD sang Mô hình dữ liệu quan hệ:** Các thực thể trong ERD sẽ trở thành quan hệ thực thể, còn mỗi liên kết thì theo qui tắc sau:

- Mỗi liên kết 1-1 : quan hệ này sẽ chứa khóa chính của quan hệ kia, và ngược lại
- Mỗi liên kết 1-n : quan hệ đầu nhiều sẽ chứa khóa chính của quan hệ đầu 1.
- Mỗi liên kết n-n : sẽ có thêm quan hệ mới với các thuộc tính là khóa chính ở thực thể đầu nhiều, và các thuộc tính riêng của nó. Khóa chính của Quan hệ mới này sẽ là khóa tổ hợp (khóa chính của các thực thể đầu nhiều), có thể có thêm thuộc tính riêng trong khóa chính.

Ví dụ: Ta chuyển đổi ERD của Sieuthi qua MHDQLH như sau:

Nhanvien(MANV, Tennv, lương, **MaBP**)

Nguoiquy(MANVQL, **MaBP**)

Bophan(MaBP, TenBP)

Nguoicc(MANCC, TenCC, Diachi)

MatHang(MaMH, TenMH, **MaBP**)

KhachHang(MAKH, TenKH, Diachi)

DonDatHang(SoHieu, Ngay, **MAKH**)

CungCap(MaMH, MANCC,
Gia)

CTDDH (SoHieu, MaMH,
SoLuong, Dongia)

Sau đó, ta tổ hợp các quan hệ có cùng khóa chính lại với nhau:

Bophan + Nguoiquy → Bophan(MaBP, TenBP, MANVQL)

2.3.3 Các phép toán đại số trong mô hình dữ liệu quan hệ

2.3.3.1 Các phép toán đại số quan hệ cơ bản.

- Toán hạng : Toán hạng của các phép toán đại số là hằng hay là biến có kiểu quan hệ
- Các phép toán : Phép hợp, giao, hiệu, tích đề các, chiếu và chọn, chia
 - ✓ Phép hợp : Phép hợp của 2 quan hệ R và S là tập hợp các bộ thuộc R hoặc thuộc S với điều kiện R và S có cùng bậc.

Phép hợp được ký hiệu $R \cup S = \{ t \mid t \in R \vee t \in S \}$

Ví dụ

R(A,B,C)			U	S(A,B,C)			=	R U S		
a	b	c		a	b	d		a	b	c
a	e	f		a	e	f		a	e	f
c	b	d						c	b	d
								a	b	d

- ✓ Phép giao : Cho 2 quan hệ R và S có cùng bậc, phép giao R và S là các bộ vừa thuộc R và vừa thuộc S.

Ký hiệu là $R \cap S = \{ t \mid t \in R \wedge t \in S \}$

Ví dụ : $R \cap S$

a	e	f
---	---	---

- ✓ Phép hiệu : Phép hiệu của 2 quan hệ là tập hợp các bộ thuộc R mà không thuộc S với điều kiện R và S có cùng bậc.

Ví dụ:

$R - S = \{ t \mid t \in R \wedge t \notin S \}$

a	b	c
c	b	d

- ✓ Tích đề các : Cho 2 quan hệ R và S có bậc k_1 và k_2 , Tích đề các của R và S là tập hợp tất cả các bộ có $k_1 + k_2$ thuộc tính, với k_1 thành phần đầu tiên thuộc R và k_2 thành phần cuối thuộc S.

$R \times S = Q(r_1, r_2, \dots, r_{k_1}, s_1, s_2, \dots, s_{k_2}) = \{ t \mid t(R^+) \in R \wedge t(S^+) \in S \}$

Ví dụ

$R \times S =$

a	b	c	a	b	d
a	b	c	a	e	f
a	e	f	a	b	d
a	e	f	a	e	f
c	b	d	a	b	d

c	b	d	a	e	f
---	---	---	---	---	---

✓ Phép chiếu : Cho 1 quan hệ có bậc k, phép chiếu của R trên các thành phần (thuộc tính) y_1, y_2, \dots, y_m là tập hợp các bộ m (a_1, a_2, \dots, a_m) sao cho có 1 bộ b nào đó của R là $b(b_1, b_2, \dots, b_k)$ mà $a_j = b_j$ (j chạy từ 1 đến m).

Ký hiệu có phép chiếu của R trên các thành phần i_1, i_2, \dots, i_m là

$$\pi_{i_1, i_2, \dots, i_m}(R) = Q(i_1, i_2, \dots, i_m) = \{t(Q^+) \mid t \in R\}$$

Ví dụ : $\pi_{A,B}(R)$

a	b
a	e
c	b

✓ Phép chọn: Cho 1 quan hệ R, phép chọn lựa trong quan hệ R theo 1 điều kiện F là tập hợp các bộ μ trong R mà các thành phần của bộ này thoả được công thức F.

_ Điều kiện F bao gồm :

- Các toán hạng là những hằng hoặc số hiệu thành phần. Thành phần thứ i được ký hiệu là $\$i$.
- Các phép toán so sánh =, <>, >, <, <=, >=, IN (ds), between ... and, like
- Các phép toán luận lý not, and, or (\wedge , \vee , \neg)
- Phép chọn lựa được ký hiệu là $\sigma_F(R)$

Ví dụ : $\sigma_{\$1 > \$2}(R)$

c	b	d
---	---	---

✓ Các phép toán khác :

• Phép chia : Cho 2 quan hệ R và S có bậc r và s và $r > s, s \neq 0$. Phép chia R cho S ký hiệu $R \div S$ là tập hợp các bộ $r - s$ (a_1, a_2, \dots, a_{r-s}) sao cho đối với tất cả bộ s (a_{r-s+1}, \dots, a_r) trong quan hệ S thì tồn tại bộ r (a_1, a_2, \dots, a_r) ở trong R

Ví dụ : $R(A, B, C, D)$

a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

$S(C, D)$

c	d
e	f

$R \div S$

a	b
e	d

Để tính $R \div S$ ta làm như sau :

- $T = \pi_{1,2,\dots,r-s}(R)$
- Tính $T \times S - R$
- Tính $V = \pi_{1,2,\dots,r-s}(T \times S - R)$
- $R \div S = T - V$

Ví dụ: $T = \pi_{1,2}(R)$

a	b
b	c
e	d

$T \times S - R$

b	c	c	d
---	---	---	---

$V = \pi_{1,2}(T \times S - R) =$

b	c
---	---

$T \times S$

a	b	c	d
a	b	e	f
b	c	c	d
b	c	e	f
e	d	c	d
e	d	e	f

$T - V$

a	b
e	d

Ví dụ :

$R(A, B)$

1018	Cs30
1050	Cs10
1070	Cs10
1050	Cs20
1070	Cs20
1021	Cs10
1050	Cs30
1021	Cs30
1070	Cs30

$S(B)$

Cs10
Cs20
Cs30

$R \div S = T - V$

1050
1070

$T = \pi_1(R)$

1080
1050
1070
1021

$T \times S$

1080	Cs10
1080	Cs20
1080	Cs30
1050	Cs10
1050	Cs20
1050	Cs30
1070	Cs10
1070	Cs20
1070	Cs30

$T \times S - R$

1080	Cs10
1080	Cs20
1021	Cs30

$V = \pi_1(T \times S - R)$

1080
1021

1021	Cs10
1021	Cs20
1021	Cs10

Ý nghĩa của phép chia:

Phép chia được dùng để trả lời cho câu truy vấn thông tin “ đầy đủ “

Ví dụ 1 Cho hai lược đồ quan hệ :

cungcấp (têncc, mấmh, giá)

Mặthàng (tênmh, mấmh).

Hãy tìm ra tên của các nhà cung cấp có cung cấp đầy đủ các mặt hàng

$\pi_{tenc} (Cungcấp \div \pi_{mãm} (mặthàng)) \rightarrow$ tên người cung cấp tất cả các mặt hàng.

Ví dụ 2: Cho hai lược đồ quan hệ :

Mônđạt (mãsv, mấmh, điểm).

MônThiTN (mấmh,).

Cho biết sinh viên tốt nghiệp (với điều kiện thi đạt hết các môn tốt nghiệp)?

$\pi_{mãsv, mãm} (mônđạt) \div \pi_{mãm} (mônthiTN)$

• **Phép kết nối :**

Phép kết nối R và S trên các thuộc tính i và j được ký hiệu là :

$R \bowtie S$

$i \theta j$

trong đó θ là phép toán so sánh được định nghĩa là:

$R \bowtie S = \sigma_{i \theta (r+j)} (R \times S)$

$i \theta j$

với r là bậc quan hệ của R.

Ví dụ : R

S

A	B	C
1	2	3
4	5	6
7	8	9

D	E
3	1
6	2

$R \bowtie S$

$B < D$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

• **Phép kết nối tự nhiên :**

Phép kết nối tự nhiên giữa 2 quan hệ R và S được ký hiệu là $R \bowtie S$ trong đó R và S có 1 các thuộc tính cùng tên thì sẽ được tính như sau.

- Tính $R \times S$

- Đối với các thuộc tính chung A_1, A_2, \dots, A_k thì ta sẽ có:

$$R \bowtie S = \pi_{i_1, i_2, \dots, i_m} (\sigma_{R.A_1 = S.A_1 \wedge R.A_2 = S.A_2 \wedge \dots \wedge R.A_k = S.A_k} (R \times S))$$

$$i_t \neq S.A_j \quad (t : 1 \dots m ; j = 1 \dots k)$$

Ví dụ :

A	B	C
A	b	c
E	b	c
D	e	f

B	C	D
b	c	g
k	e	f
e	f	k

$R \times S$

A	R.B	R.C	S.B	S.C	D
a	b	c	b	c	g
a	b	c	k	e	f
a	b	c	e	f	k
e	b	c	b	c	g
e	b	c	k	e	f
e	b	c	e	f	k
d	e	f	b	c	g
d	e	f	k	e	f
d	e	f	e	f	k

$$R \bowtie S = \sigma_{R.B = S.B \wedge R.C = S.C} (R \times S)$$

A	B	C	D
a	b	c	g
e	b	c	g
d	e	f	k

Ví dụ : Cho hai lược đồ quan hệ:

phòngban (mãpb, tênpb)

Quản lý (tênngười, mãph)

Tên người trưởng phòng quản lý phòng nào và mã ?

Phòngban \bowtie quản lý

Tên người quản lý khoa 'CNTT'

$$\pi_{tênngười} (\sigma_{mãph = 'CNTT'} (\text{phòngban} \bowtie \text{quản lý}))$$

Ví dụ: Cho lược đồ CSDL sau:

sinhviên (mãsv, họsv, tênsv,...)

môn học (tênmh, mãmh)

đăng ký (mãsv, mãmh)

kết quả (mãmh, mãsv, điểm,)

Cho biết họ tên sinh viên đăng ký môn 'CSDL'

$$\pi_{\text{họsv, tênsv}} (\sigma_{\text{mãmh}} = \text{'CSDL'} (\text{sinhviên } l > < l \text{ đăng ký}))$$

Cho biết các sinh viên đã thi đậu môn 'NNLT' ?

Cách 1: $\sigma_{\text{tênmh}} = \text{'NNLT'}$ (môn học)

$$S = \sigma_{\text{điểm}} \geq 5 (\text{kết quả } l > < l \sigma_{\text{mãmh}} = \text{'NNLT' (môn học)})$$

$$\pi_{\text{họsv, tênsv}} (\text{sinhviên } l > < l S)$$

Cách 2: $\pi_{\text{họsv, tênsv}} (\sigma_{\text{tênmh}} = \text{'NNLT'} \wedge \text{điểm} \geq 5 ((\text{sinhviên } l > < l \text{ kết quả}) l > < l \text{ môn học}))$

- Phép nối kết nội :

Phép nối kết là phép kết nối tự nhiên nhưng chỉ chiếu trên các thuộc tính của R:

$$R \bowtie S = \pi_R (R \bowtie S)$$

□ Nhận xét :

Ngôn ngữ đại số quan hệ là 1 ngôn ngữ hỏi ta có thể sử dụng các phép toán của đại số quan hệ để trả lời 1 số câu hỏi.

Ví dụ : Ta có quan hệ cung cấp trong mô hình siêu thị :
cung cấp(têncc, mãmh, giá).

- Cho biết các mặt hàng và giá của người cung cấp có tên là 'ABC'

$$\pi_{\text{mãmh, giá}} (\sigma_{\text{têncc}} = \text{'ABC'} (\text{cung cấp}))$$

- Hãy cho biết mã các mặt hàng có người cung cấp

$$\pi_{\text{mãmh}} (\text{cung cấp})$$

2.2.4 Thiết kế cơ sở dữ liệu quan hệ

Để thiết kế cơ sở dữ liệu, ta tiến hành theo các bước sau :

- ✓ Xác định các tập thực thể và các mối liên kết giữa các tập thực thể để từ đó xây dựng sơ đồ ER
- ✓ Biến đổi sơ đồ ER thành các quan hệ dự tuyến
- ✓ Chuẩn hóa các quan hệ dự tuyến

2.2.4.1 Xác định các tập thực thể và các mối liên kết

- ✓ Xác định các tập thực thể :

Ta có 1 số gợi ý sau :

- Nếu có thông tin mô tả 1 đối tượng thì đối tượng này được xem như 1 thực thể.

Ví dụ : Để mô tả 1 thành phố người ta có các thuộc tính như dân số, số quận huyện, tên thành phố. Từ đó ta xem thành phố như 1 thực thể

- Nếu có nhiều hơn 1 giá trị 1 thuộc tính mô tả tương ứng với 1 giá trị của danh hiệu thì thuộc tính mô tả này nên được xem là thực thể

Ví dụ : kích thước gồm có chiều cao, rộng... của căn nhà nên ta coi kích thước như 1 thực thể.

Trong các thuộc tính thì có 2 loại thuộc tính:

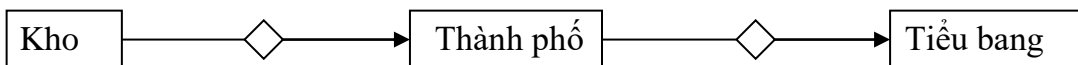
- ☐ Thuộc tính mô tả
- ☐ Thuộc tính danh hiệu (khóa)

- Nếu 1 thuộc tính mô tả của 1 thực thể có mối liên kết nhiều-một với thực thể khác thì nên xem thuộc tính mô tả này là 1 thực thể

Ví dụ : Ta có tập thực thể : kho (mãkho, tênthànhphố, tiểubang (sốtb))

Kho		tiểu bang	
Mã kho	T - phố	MãTB	
001	HCM	Nam	
002	CT	Trung	
003	HUE	Bắc	
004	ĐN		

Vì vậy ta tách thành phố thành 1 tập thực thể



- Nên gán các thuộc tính cho tập thực thể mà chúng mô tả trực tiếp nhất

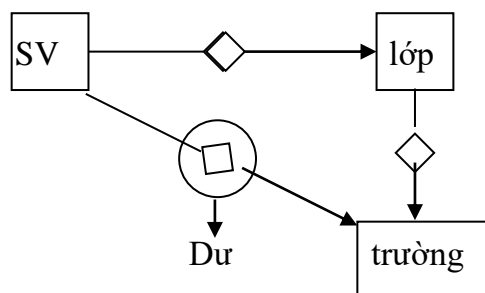
Ví dụ : Ta có 2 tập thực thể là công nhân và phòng ban, (thì người quản lý nằm trong phòng ban)

- Nên tránh các khóa tổ hợp trong các tập thực thể.

- ☐ Xây dựng các mối liên kết :

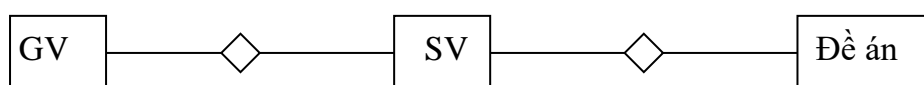
- Nên loại bỏ các mối liên kết dư thừa thường xảy ra khi ta dùng mối liên kết bắc cầu.

Ví dụ : Ta có 3 tập thực thể SV, lớp, trường

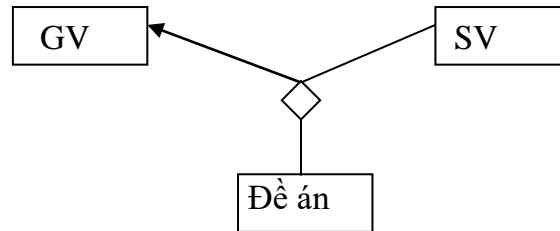


- Hạn chế sử dụng mối liên kết 3.

Ví dụ : một sinh viên có thể tham gia nhiều đề án và 1 giảng viên có thể hướng dẫn nhiều sinh viên ở bất kỳ đề án nào.



Mỗi sinh viên có thể tham gia đề án và có nhiều giáo viên hướng dẫn nhưng ở 1 đề án mà sinh viên tham gia chỉ có 1 giáo viên hướng dẫn.



- Ta chỉ nên xây dựng 3 loại mối liên kết.

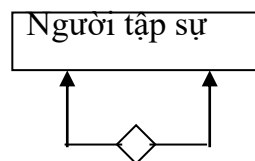
Liên kết 1 ngôi

Liên kết 2 ngôi

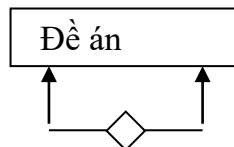
Liên kết 3 ngôi

□ Liên kết 1 ngôi :

Ví dụ : Mỗi người tập sự chỉ có 1 người tập sự khác là cộng sự viên của mình trong công việc

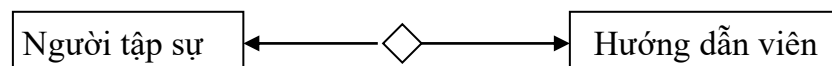


Ví dụ : mọi đề án đều có sự liên hệ với nhau



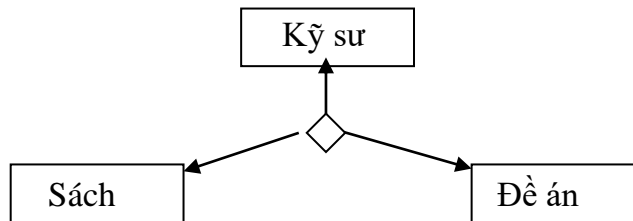
□ Mối liên kết 2 ngôi

Ví dụ : một người tập sự chỉ có 1 hướng dẫn viên và hướng dẫn viên chỉ hướng dẫn 1 người tập sự



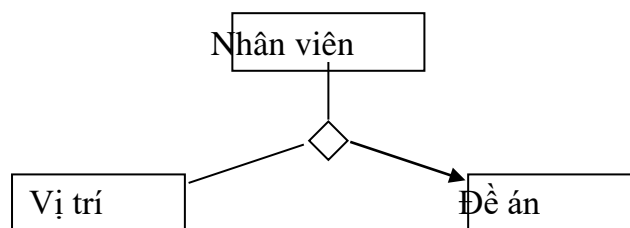
□ Mối liên kết 3 ngôi : được thiết lập khi phải cần 2 thực thể mới suy ra được thực thể còn lại.

Ví dụ : Một kỹ sư chỉ sử dụng 1 sách cho 1 đề án. Các kỹ sư khác nhau sử dụng các sách khác nhau cho cùng 1 đề án. Không kỹ sư nào sử dụng cùng 1 sách cho các đề án khác nhau.



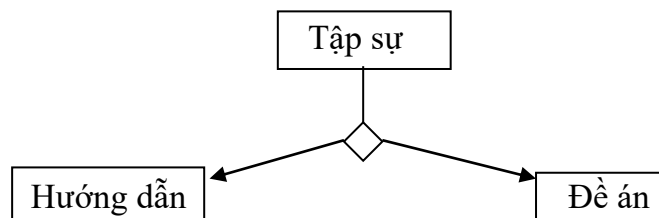
Mô hình 1

- Các nhân viên có thể tham gia nhiều đề án nhưng chỉ có thể tham gia nhiều nhất vào 1 đề án ở vị trí đã cho.



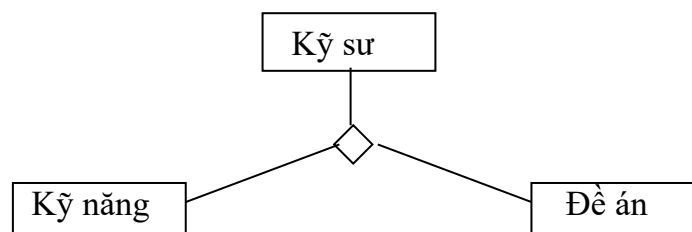
Mô hình 2

- Người tập sự làm việc trong các đề án dưới sự chỉ dẫn của người hướng dẫn. Một người hướng dẫn chỉ có thể hướng dẫn nhiều nhất 1 đề án cho 1 người tập sự nào đó. Một người tập sự tham gia vào 1 đề án nào đó dưới sự chỉ dẫn của nhiều nhất 1 người hướng dẫn.



Mô hình 3

- Các kỹ sư sử dụng các kỹ năng cho mỗi đề án mà họ tham gia



Mô hình 4

2.2.4.2 Xây dựng các quan hệ dự tuyển từ mô hình ER

- Biến đổi quan hệ thực thể thành mối liên kết thực thể với khóa của quan hệ thực thể cũng chính là khóa.

- Biến đổi mối liên kết thành quan hệ liên kết.

+ Nếu là mối liên kết 1-1 thì khóa của quan hệ liên kết này sẽ là 1 trong 2 khóa của 2 hoặc 3 tập thực thể tham gia vào mối liên kết.


+ Nếu là mối liên kết nhiều-1 thì quan hệ của liên kết này có khóa là khóa của tập thực thể phía nhiều.

+ Nếu là mối liên kết nhiều-nhiều thì khóa của quan hệ này sẽ là tất cả các khóa của các đối tượng tập thực thể.

Chú ý : Trong mỗi liên kết 3 ngôi vì phải có 2 đối tượng để xác định đối tượng còn lại do đó cần phải có 2 khóa của 2 đối tượng này để suy ra đối tượng còn lại.

Ví dụ : Theo mô hình số 1 ta có :

Kỹ sư (mãks, ...)
 Sách (mãsách, ...)
 Đề án (sốđềán, ...)
 Tham gia (mãks, mãsách, sốđềán)



- Theo mô hình 2 ta có :

Nhân viên (mãnv, ...)
 Vị trí (mãpb, ...)
 Đề án (sốđềán, ...)
 Tham gia (mãnv, mãpb, sốđềán)

- Theo mô hình số 3

Tập sự (mãts, ...)
 Hướng dẫn (mãnv, ...)
 Đề án (sốđềán, ...)
 Tham gia (mãnv, mãts, sốđềán)

- Theo mô hình số 4

⇒ Tham gia (mãks, mãkn, sốđềán)

2.2.4.3 Chuẩn hóa các quan hệ :

Chuẩn hóa quan hệ là tiến hành kiểm tra và xử lý sao cho các lược đồ quan hệ đều đạt tối thiểu ở dạng chuẩn 3. Phần này sẽ được nghiên cứu ở chương 5.

CHƯƠNG 3 NGÔN NGỮ SQL

3.1 Giới thiệu

Ngôn ngữ SQL (Structured Query Language) được phát triển từ ngôn ngữ SEQUEL (Structured English Query Language) là sản phẩm của nhóm nghiên cứu tại trung tâm nghiên cứu của IBM tại San Jose, California, USA. Ngay sau khi ra đời vào những năm đầu của thập niên 70, ngôn ngữ này được ứng dụng rộng rãi và nhanh chóng trở thành ngôn ngữ chuẩn trong lĩnh vực khai thác cơ sở dữ liệu quan hệ trên máy lớn cũng như trên máy vi tính sau này. Hiện nay, hầu hết các hệ quản trị cơ sở dữ liệu đều có ngôn ngữ SQL.

Ngôn ngữ SQL là một ngôn ngữ truy vấn dựa trên đại số quan hệ và là ngôn ngữ phi thủ tục. Ngôn ngữ SQL được chia thành các phần sau:

- Ngôn ngữ định nghĩa dữ liệu (Data Definition Language DDL): Cung cấp các lệnh định nghĩa lược đồ quan hệ, xóa quan hệ, tạo chỉ mục thay đổi lược đồ quan hệ.
- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language DML): gồm ngôn ngữ truy vấn dựa trên đại số quan hệ và các lệnh thêm, xóa, sửa các bộ trong cơ sở dữ liệu.
- Ngôn ngữ thao tác dữ liệu được nhúng (Embedded Data Manipulation Language): Dạng nhúng của ngôn ngữ SQL trong các ngôn ngữ lập trình như Pascal, C, Cobol, Fortran ...
- Ngôn ngữ điều khiển dữ liệu (Data Control Language DCL) chứa các lệnh bắt đầu, kết thúc, phục hồi giao tác và khoá dữ liệu cho việc truy xuất đồng thời.

3.2 Ngôn ngữ định nghĩa dữ liệu (Data Definition Language DDL)

*** Tạo CƠ SỞ DỮ LIỆU: CREATE DATABASE <ten CƠ SỞ DỮ LIỆU >**

3.2.1 Tạo cấu trúc bảng

Cú pháp: **CREATE TABLE < tên_bảng >**

(<Thuộc tính > <Kiểu> [<Kích thước>][NOT NULL], ...

PRIMARY KEY (<khoá chính>)

[UNIQUE (<khoá>), ...]

[FOREIGN KEY (<khoá ngoại>) REFERENCE <tên_bảng> ,...]

[CHECK <điều kiện ràng buộc>, ...]

)

Các kiểu dữ liệu:

nchar(10), nvarchar(n) : Kiểu chuỗi ký tự gồm n ký tự ($1 \leq n \leq 8000$)

smallint : Kiểu số nguyên 1 byte

int: Kiểu số nguyên 2 bytes

decimal(n,t): Kiểu số thực có n chữ số với t chữ số thập phân.

float: Kiểu thực dấu chấm động

date: kiểu ngày dương lịch

bit: Kiểu luận lý (Đúng, Sai)

Các từ khoá:

NOT NULL: thuộc tính không lấy giá trị rỗng

PRIMARY KEY (khoá chính): Khai báo khoá chính của bảng

UNIQUE KEY (khoá) : Khai báo khoá duy nhất khác.

FOREIGN KEY (khoá ngoại) : Khai báo các khoá ngoại của bảng

CHECK <điều kiện ràng buộc> : Mô tả các ràng buộc về miền giá trị.

Ví dụ : Tạo lập quan hệ cung cấp(MANCC,Mamh,gia)

```
Create      Table cungcấp
(MANCC      char (10)    FOREIGN KEY REFERENCE NguoiCC (MANCC) ,
Mamh       char (5)      FOREIGN KEY REFERENCE mathang (MAMH),
Gia        Num (7)       NOT NULL,
PRIMARY KEY (tencc,mamh),
CHECK gia >0
)
```

3.2.2 Thay đổi cấu trúc bảng

Cú pháp: **ALTER TABLE <tên_bảng>**
 ADD <thuộc tính> <kiểu> [(<kích thước>)] [NOT NULL],...
 DROP <thuộc tính>, ...

Ví dụ : Thêm thuộc tính NGAYGH , SOCMND vào bảng cungcấp

```
ALTER TABLE cungcấp
ADD ngaygh date not null, socmnd char(9) not null
```

Ví dụ: Loại bỏ thuộc tính SOCMND

```
ALTER TABLE cungcấp
DROP socmnd
```

Chú ý: Thao tác hủy là không hợp lệ nếu thuộc tính bị hủy là thuộc tính khoá.

3.2.3 Hủy bỏ một bảng

Cú pháp: **DROP TABLE <bảng>**

Ví dụ DROP TABLE cungcấp

3.2.4. Tạo khung nhìn (view)

Cú pháp: **CREATE VIEW <tên_khung_nhìn> [<thuộc tính>,...]**
 AS [<tên_bảng>] | [<câu truy vấn Select ...>]

Ví dụ Danh sách tên nhà cung cấp các mặt hàng

```
CREATE VIEW danhsachcc
AS SELECT tenc, tênmh
FROM cungcap, mathang
WHERE cungcap.mamh = mathang.mamh
```

3.2.5 Xoá khung nhìn

Cú pháp: **DROP VIEW** <tên_khung_nhìn>

Ví dụ: Xoá khung nhìn danhsachcc

```
DROP VIEW danhsachcc
```

3.3 Ngôn ngữ thao tác dữ liệu (Data Manipulation Language DML)

3.3.1 Thêm bộ mới vào quan hệ

Cú pháp: **INSERT INTO** <tên_bảng>
VALUES (V1, V2, ..., Vn)

V1, V2, ..., Vn : là các giá trị của các thuộc tính tương ứng được đưa vào bảng.

Ví dụ : Thêm 1 bộ mới vào quan hệ cung cấp có tên người cung cấp = 'Nguyễn Mai Chi'

```
INSERT INTO cungcap
VALUES ('Nguyễn mai chi', '007', 10)
```

Lệnh Insert còn cho phép lấy dữ liệu từ các table khác chuyển vào qua cú pháp :

```
INSERT INTO <table> (ds field)
SELECT <ds cột> ...
```

Ví dụ 3: Lệnh Insert sau sẽ copy tất cả các record từ 1 version cũ của table Nhanvien (OldEmp) vào version mới của Nhanvien (có thêm field NoiSinh với giá trị là ' ')

Insert Into Nhanvien (MANV, HO, TEN, NOISINH)

```
Select MANV, HO, TEN, ' '
From OldEmp
```

Lưu ý: Lệnh Select Into để tạo ra 1 table mới có các mẫu tin lấy từ 1 hoặc nhiều tables.

3.3.2 Cập nhật:

Cú pháp: **UPDATE** <tên_bảng>
SET A1 = V1, A2 = V2, ..., An = Vn
[WHERE ĐK]

Thuộc tính A_i có giá trị V_i

Ví dụ : Tăng giá của các mặt hàng lên 10%

```
UPDATE cungcap
```


Ví dụ 3 : Hãy cho biết tên người cung cấp bán mặt hàng coca-cola

```

Select tênNCC
From cungcấp, bàymặthàng
Where      tênmh = 'coca-cola' and
           Bàymặthàng.mãmh = cungcấp.mãmh
⇔  $\pi_{\text{tênNCC}} (\sigma_{\text{tênmh} = \text{'coca-cola' .and (cungcấp x bàymặthàng)})$ 
           bàymặthàng, mãmh = cungcấp, mãmh

```

Ví dụ 4 : Hãy cho biết tên và địa chỉ của người cung cấp mặt hàng coca-cola.

```

Select      nguoiCC.tênNCC, địachỉ
From        cungcấp, bàymặthàng, nguoiCC
Where bàymặthàng.mãmh = cungcấp.mãmh and
        Cungcấp.tênNCC = nguoiCC.tênCC and
        Tênmh= 'Coca-cola'
 $\pi_{\text{tênNCC, địachỉ}} (\sigma_{\text{bàymặthàng, mãmh} = \text{cungcấp, mãmh, and cungcấp, tênNCC} = \text{nguoiCC, tênNCC. (cungcấp x bàymặthàng x nguoiCC)})$ 

```

Ví dụ 5 : Hãy cho biết tên của người cung cấp mặt hàng được đặt mua bởi khách hàng Nguyễn Minh Tuấn.

```

Select      DISTINCT TênNCC
From        cungcấp, đơndặthàng, chứa
Where tênKH = 'Nguyễn Minh Tuấn' .and.
        Đơndặthàng.sốhiệu = chứa. sốhiệu and
        cungcấp.mãmh = chứa.mãmh
⇔  $\pi_{\text{tênNCC, địachỉ}} (\sigma_{\text{tênKH} = \text{'Nguyễn Minh Tuấn' .and đơndặthàng, sốhiệu} = \text{chứa, sốhiệu ,and chứa, mãmh} = \text{cungcấp, mãmh (cungcấp x đơndặthàng x chứa)})$ 

```

Chú ý : Để chỉ xuất 1 bộ trong các bộ giống nhau ta dùng từ khóa '**DISTINCT**'

□ **Biến kiểu bộ :**

Biến kiểu bộ là 1 biến mà giá trị của nó là 1 bộ của quan hệ, vậy biến của bộ thực chất là 1 biến Record.

Cách đặt biến kiểu bộ cho 1 quan hệ.

From Qhệ 1 Q1 , Qhệ 2 Q2

Cách sử dụng giống như 1 biến của Record

Ví dụ : Qhệ (họ, tên, địachỉ, ngàysinh) Q1

Truy xuất Q1.Họ ; Q1.Tên; Q1.Ngàysinh

Ứng dụng:

- Để thay thế tên quan hệ
- So sánh các bộ trong cùng 1 quan hệ.

Ví dụ : Hãy cho biết tên và địa chỉ của khách hàng có số dư nhỏ hơn số dư của khách hàng Nguyễn Minh Tuấn.

```
Select      tênKH, địachỉ, sốdư
From        kháchhàng   KH
Where KH.sodu < (Select sodu from kháchhàng
                  where tênKH = 'Nguyễn Minh Tuấn' )
```

*** Các toán tử được dùng trong điều kiện:**

> >= (!<) < <= (!>) = <> (!=)

Is Null

Is Not Null

Between ... And ...

In ('N', 'X')

Like : _ ñaïi dieän 1 kýù töï
 % ñaïi dieän 1 string

Not And Or

☐ **So trùng mẫu :**

Một số ký tự đặc biệt dùng để so trùng mẫu.

- ☐ % : bất kỳ chuỗi ký tự nào cũng được
- ☐ _ : bất kỳ ký tự nào cũng được.

- Cách sử dụng :

Where biến ký tự LIKE 'CHUỖI SO TRÙNG'

Ví dụ : Hãy cho biết tên các mặt hàng được bán có tên bắt đầu bằng chữ E

```
Select      tênmh
From        mặthàng
Where tênmh like 'E%'
```

Ví dụ : Hãy cho biết toàn bộ các thông tin về đơn đặt hàng có số hiệu từ 1000 - >1999

```
Select      *
From        đơndặthàng
Where sốhiệu like '1_ _ _'
```

☐ **Các phép toán trong mệnh đề WHERE:**

- ☐ Trong mệnh đề WHERE thì có các phép so sánh
- ☐ Trong mệnh đề WHERE có thể chứa cả phát biểu SELECT, FROM, WHERE.
- ☐ Kết quả của lệnh SELECT là 1 quan hệ do đó mệnh đề WHERE chứa lệnh Select, From, Where phải có các phép toán trên quan hệ

IN	\subseteq	()
NOT IN	$\not\subseteq$	
ANY	\exists	
ALL	\forall	
EXISTS	\exists	phần tử trong tập
NOT EXISTS		

Ví dụ : Hãy cho biết tên người cung cấp các mặt hàng được đặt mua bởi ông Nguyễn Minh Tuấn.

Cung cấp (têncc, mãmh, giá)
 Đơn đặt hàng (sốhiệu, tênkh, ngày)
 Chứa (mãmh, sốhiệu, sốlượng)

```

Select têncc
From cung cấp
Where mãmh in (
    select mãmh
    From chứa
    Where sốhiệu in (Select sốhiệu
                    From đơn đặt hàng
                    Where tênkh = 'Nguyễn Minh Tuấn'))
  
```

Ví dụ : Hãy cho biết tên và địa chỉ của khách hàng có số dư lớn hơn số dư của khách hàng 'Nguyễn Minh Tuấn'

```

Select tênkh, địachỉ
From khách hàng
Where sốdư > ALL (Select sốdư
                  From khách hàng
                  Where tênkh = 'Nguyễn Minh Tuấn')
  
```

Ví dụ : Hãy cho biết tên mặt hàng được bán với giá cao nhất

```

Select tênmh
From mặt hàng MH, cung cấp CC
Where giá >= ALL (
    Select giá
    From cung cấp)
And CC.mãmh = MH.mãmh
  
```

Ví dụ: Hãy cho biết tên các mặt hàng do ông Nguyễn Minh Tuấn cung cấp.

```

Select mamh, tenmh
From mathang
Where Exist(
    Select mamh
    From cungcap
    Where tenc= 'Nguyễn Minh Tuấn')
  
```

□ Các hàm dùng trong SELECT

AVG () Giá trị trung bình

COUNT ()	Đếm số bộ
MAX ()	Giá trị lớn nhất
MIN ()	Giá trị nhỏ nhất
SUM ()	Tổng giá trị

Ví dụ : Hãy cho biết số dư trung bình của khách hàng

```
Select      AVG (sốdư)
From        kháchhàng
```

Ví dụ : Cho biết có bao nhiêu người cung cấp mặt hàng.

```
Select      COUNT (DISTINCT TENCC)
From        cungcấp
```

Ví dụ : Hãy cho biết có bao nhiêu người cung cấp mặt hàng coca-cola.

```
Select      COUNT (têncc)
From        cungcấp, mặthàng
Where tênmh = 'coca-cola'
And         cungcấp.mãmh = mặthàng.mãmh
```

□ Phân nhóm của *SELECT*

```
Group by    A1, A2, ..., Ak
Having      ĐK
```

Ghi sau lệnh WHERE dùng để gom nhóm theo các thuộc tính A₁, A₂,..., A_k

Ví dụ : Hãy cho biết tên các mặt hàng và đơn giá bán trung bình

```
Select      tênmh, AVG (giá)
From        mặthàng MH, cungcấp CC
Where CC.mãmh = MH. mãmh
Group by    tênmh
```

Ví dụ : Hãy cho biết tên các mặt hàng, đơn giá bán trung bình và điều kiện những mặt hàng này có nhiều hơn 1 người cung cấp

```
Select      tênmh, AVG (giá)
From        mặthàng MH, cungcấp CC
Where CC.mãmh = MH.mãmh
Group by    tênmh      Having      Count (*) > 1
```

Ví dụ : Hãy cho biết tên các mặt hàng, đơn giá bán trung bình và điều kiện mặt hàng có 2 đơn giá khác nhau trở lên:

```
Select      tênmh, AVG (giá)
From        mặthàng MH, cungcấp CC
Where CC.mãmh = MH.mãmh
Group by    tênmh      Having      Count (DISTINCT giá) > 1
```

□ **Chuyển kết quả vào một bảng :**

INTO TABLE <tên_bảng>

Nếu tên_bảng bắt đầu bởi # hoặc ## thì bảng đó là bảng ảo, và nó sẽ tự động mất đi khi kết nối tạo ra nó đã bị ngắt.

Ví dụ: Lưu lại danh sách các mặt hàng với đơn giá trung bình vào bảng ảo Gia_TB_MatHang

```
Select      tênmh, AVG (giá)
Into Table # Gia_TB_MatHang
From        mặthàng MH, cungcấp CC
Where CC.mãmh = MH. mãmh
Group by    tênmh
```

□ **Sắp xếp dữ liệu xuất**

ORDER BY <thuộc tính [DESC] > [, ...]

Mặc định thì dữ liệu sẽ được sắp xếp theo thứ tự tăng dần, nếu ta muốn sắp xếp dữ liệu theo thứ tự giảm dần thì sử dụng từ khóa DESC

Ví dụ: Hãy cho biết tên các mặt hàng và đơn giá bán trung bình theo thứ tự tên mặt hàng.

```
Select      tênmh, AVG (giá)
From        mặthàng MH, cungcấp CC
Where CC.mãmh = MH.mãmh
Group by    tênmh      Order by    tênmh
```

□ **Các phép toán trên tập hợp**

Các phép toán trên tập hợp gồm có phép hội UNION [ALL], phép giao INTERSECT, phép hiệu EXCEPT

Cú pháp :

Phát biểu SELECT 1 <phép toán trên tập hợp> Phát biểu SELECT 2

với điều kiện các phát biểu select phải cùng dạng kết xuất.

Ví dụ : Hãy liệt kê tên các mặt hàng được cung cấp với giá 10000 và 20000, và cho biết cụ thể đối tác cung cấp.

```
Select Tenmh, TenCC, gia
From Mathang mh, cungcap cc
Where mh.mamh=cc.mamh and gia=10000
Intersect      Select Tenmh, TenCC, gia
From Mathang mh, cungcap cc
Where mh.mamh=cc.mamh and gia=20000
```

3.4 Ngôn ngữ điều khiển dữ liệu (Data Control Language)

Ngôn ngữ điều khiển dữ liệu chứa các lệnh để điều khiển giao tác và khoá dữ liệu.

- **Giao taùc (Transaction):** Khi thay ñoái döõ lieäu trên nhiều table hay nhiều records trên 1 table, ta phải ñaùm baùu tính nhaát quaùn veà döõ lieäu trên cô sôû döõ lieäu. Ví dụ ta ñang thöïc hieän vieäc taêng möùc giaùm giaù cho caùc customer thì tieán trình ñang thì haønh bò ngaét quaõng vì 1 nguyeân nhaân naøo ñoù (maát nguoaùn). Nhö vaäy, roõ raøng laø chæ coù 1 soá khaùch haøng ñöôïc taêng Discount, coøn 1 soá khaùch thì khoâng ; ñieàu naøy seõ daãn ñeán khoâng nhaát quaùn veà döõ lieäu.

Ñeå traùnh tình traëng naøy xaùy ra, SQL Server cung caáp 1 khaù naêng cho pheùp ta phuïchoài laïi döõ lieäu cuõ neáu coâng vieäc ñang thì haønh bò loaïi : giao taùc .

Moät coâng vieäc cuûa ta coù khaù naêng laø 1 leänh hay nhiều leänh SQL taùc ñoäng lên nhiều Table; moät coâng vieäc nhö vaäy ta goïi laø 1 giao taùc. Ñeå baét ñaàu 1 giao taùc, ta duøng : Begin Transaction; xaùc nhaän 1 giao taùc ñaõ hoàøn thaønh : Commit; huý boû giao taùc vaø traû laïi döõ lieäu cuõ : RollBack.

Ví dụ: Trong ngaân haøng, 1 giao taùc laø vieäc chuyeän soá tieàn töø taøi khoaùn tieát kieäm cuûa khaùch haøng coù taøi khoaùn @TKCHUYEN qua taøi khoaùn @TKNHAN với số tiền @SOTIEN :

```
CREATE PROC [dbo].[SP_CHUYENTHEN]
@TKCHUYEN NVARCHAR (10) , @TKNHAN NVARCHAR (10), @SOTIEN BIGINT
AS
SET XACT_ABORT ON

BEGIN TRANSACTION
BEGIN TRY
    UPDATE TAIKHOAN
    SET SODU = SODU+ @SOTIEN
    WHERE SOTK= @TKNHAN

    UPDATE TAIKHOAN
    SET SODU = SODU - @SOTIEN
```



```
WHERE SOTK= @TKCHUYEN
```

```
COMMIT
```

```
END TRY
```

```
BEGIN CATCH
```

```
ROLLBACK
```

```
DECLARE @ErrorMessage VARCHAR(2000)
```

```
SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
```

```
RAISERROR(@ErrorMessage, 16, 1)
```

```
END CATCH
```

* Ghi chú **Về tùy chọn XACT_ABORT**: Đây là tùy chọn ở mức kết nối, chỉ có tác dụng trong phạm vi kết nối của ta. XACT_ABORT nhận hai giá trị ON và OFF (OFF là giá trị mặc định). Khi tùy chọn này được đặt là OFF, SQL Server sẽ chỉ hủy bỏ lệnh gây ra lỗi trong transaction và vẫn cho các lệnh khác thực hiện tiếp, nếu lỗi xảy ra được đánh giá là không nghiêm trọng. Còn khi XACT_ABORT được đặt thành ON, SQL Server mới cư xử đúng như mong đợi – khi gặp bất kỳ lỗi nào nó hủy bỏ toàn bộ transaction và quay lại trở lại như lúc ban đầu.

□ Khoá dữ liệu:

Khi 2 user cùng truy xuất 1 table, có khả năng tiến trình update của user này sẽ xung đột với tiến trình Select hay update của user thứ 2. Chúng ta hãy thử khi user 1 đang thực hiện lệnh: Select Avg (Discount) From Customer thì user thứ 2 lại đang thực hiện lệnh Update:

```
Update Customer
```

```
Set Discount = 0.01
```

```
Where ShipCity = 'Khanh Hoa'
```

Trong trường hợp này, user 1 có thể lấy giá trị trung bình của Discount để tính trung bình. Ta sẽ thực hiện Transact-SQL cung cấp 1 tùy chọn khóa table, hạn chế xung đột xảy ra trong quá trình truy xuất dữ liệu.

Lệnh sau đây sẽ bảo vệ khỏi sự xung đột với update:

```
Select Avg (Discount)
```

```
From Customer With (TabLock)
```

TabLock là khóa chung (share lock) trên table – nó ngăn cản user khác update dữ liệu trên table đang được đọc bởi 1 user nào đó, nhưng vẫn cho phép đọc dữ liệu.

Tổng cộng, phát biểu update có thể dùng From với **TabLockX** để thiết lập khóa riêng (exclusive lock) – không cho bất cứ loại truy xuất nào trên table đang
dùng.

CHƯƠNG 4 PHỤ THUỘC HÀM

4.1 Phụ Thuộc hàm (Functional Dependency)

4.1.1 Định nghĩa phụ thuộc hàm (FD)

Cho $R(U)$ là một lược đồ quan hệ với $U = \{A_1, A_2, \dots, A_n\}$ là tập thuộc tính. X và Y là tập hợp con của U .

Ta nói $X \rightarrow Y$ (đọc là X xác định hàm Y hoặc Y phụ thuộc hàm X), nếu r là một thể hiện của quan hệ xác định trên $R(U)$ sao cho bất kỳ 2 bộ $t_1, t_2 \in r$ mà $t_1[X] = t_2[X]$ thì $t_1[Y] = t_2[Y]$.

4.1.2 Định nghĩa phụ thuộc hàm đầy đủ (Full Functional Dependency "FFD")

Cho thể hiện r thỏa phụ thuộc hàm $X \rightarrow Y$ và nếu không tồn tại $X' \subset X$ sao cho $X' \rightarrow Y$ ta nói X xác định đầy đủ Y trong r hay Y phụ thuộc hàm đầy đủ X trong r .

4.1.3 Mệnh đề

Cho thể hiện quan hệ r của lược đồ R .

a) $X \rightarrow Y$ là phụ thuộc hàm trên r nếu và chỉ nếu X là siêu khoá của $r[XY]$.

b) $X \rightarrow Y$ là phụ thuộc hàm đầy đủ trên r nếu và chỉ nếu X là khoá của $r[XY]$.

4.1.4 Giải thuật kiểm tra phụ thuộc hàm

a) Giải thuật kiểm tra phụ thuộc hàm:

SATISFY (r, f).

Vào: Thể hiện quan hệ r và $f: X \rightarrow Y$.

Ra: Đ, S.

Phương pháp:

1. $r' = r[XY]$.
2. Return ($\text{card}(r'[x]) = \text{card}(r')$)

b) Giải thuật kiểm tra phụ thuộc hàm đầy đủ:

SATISFYFULL (r, f)

Vào: Thể hiện quan hệ r và $f: X \rightarrow Y$.

Ra: Đ, S.

Phương pháp:

1. For $\forall A \in X$ DO.
If SATISFY ($r, X - A \rightarrow Y$) Then return (FALSE)
2. Return (true).

Ví dụ: Cho quan hệ $r(ABCD)$.

r	(A	B	C	D)	với $f: AC \rightarrow B$
a1	b1	c1	d1				xem xét r có thỏa f ?
a1	b1	c1	d2				
a2	b1	c3	d2				
a3	b2	c1	d1				

Tính $r' = r \mid ABC$

$r' (A \ B \ C)$

a1 b1 c1

a2 b1 c3

a3 b2 c1

- r' thoả mãn khoá AC nên r thoả $f = AC \rightarrow B$.

- Tuy nhiên AC là siêu khoá vì A là khoá nên f không đầy đủ trong r .

4.1.5 Tập phụ thuộc hàm

Cho thể hiện quan hệ $r (R)$, F là tập các phụ thuộc hàm. Quan hệ r được gọi là thoả F nếu r thoả $f \ \forall f \in F$.

Ví dụ cho quan hệ $r (ABCDE)$.

$r (A \ B \ C \ D \ E)$

a1 b1 c1 d1 e1

a1 b2 c2 d2 e1

a2 b1 c3 d3 e1

a2 b1 c4 d3 e1

a3 b2 c5 d1 e1

$F = \{A \rightarrow D, AB \rightarrow D, C \rightarrow BDE, E \rightarrow A, A \rightarrow E\}$.

r không thoả $A \rightarrow D$ nên r không thoả F .

4.2 Hệ luật dẫn ARMSTRONG

$R(U)$ là 1 lược đồ quan hệ với $U = \{A_1, A_2, \dots, A_n\}$ là tập thuộc tính.

Xét $X, Y, Z, W \subset U$, hệ luật dẫn Armstrong:

1/ Luật phản xạ.

$Y \subseteq X \Rightarrow X \rightarrow Y$.

2/ Luật thêm vào.

$X \rightarrow Y$ và $Z \subseteq W \Rightarrow XW \rightarrow YZ$.

$X \rightarrow Y$ thì $XZ \rightarrow YZ$ (tăng trưởng).

3/ Luật bắc cầu :

$X \rightarrow Y$ và $Y \rightarrow Z \Rightarrow X \rightarrow Z$.

4/ Luật tựa bắc cầu.

$X \rightarrow Y$ và $WY \rightarrow Z \Rightarrow XW \rightarrow Z$.

5/ Luật phân rã.

$X \rightarrow Y$ và $Z \subseteq Y \Rightarrow X \rightarrow Z$.

6/ Luật hợp.

$X \rightarrow Y$ và $X \rightarrow Z \Rightarrow X \rightarrow YZ$.

Ví dụ: $AB \rightarrow C, C \rightarrow A$

Chứng minh: $BC \rightarrow ABC$

a) $C \rightarrow A$ gt.

b) $B \in U, C \rightarrow A \Rightarrow BC \rightarrow AB$ (tăng trưởng)

c) $C \in U, BC \rightarrow AB \Rightarrow BC \rightarrow ABC$ (tăng trưởng).

4.3 Bao Đóng.

4.3.1 Định nghĩa phụ thuộc hàm suy dẫn

Cho tập các phụ thuộc hàm, F, X, Y là 2 tập thuộc tính của R . Ta nói:

$f : X \rightarrow Y$ là phụ thuộc hàm suy dẫn từ F nếu nó được suy từ F bằng các luật dẫn của hệ tiên đề Armstrong. Ký hiệu $F \models f$.

Tập $F^+ = \{f / F \models f\}$ được gọi là bao đóng của F .

4.3.2 Mệnh đề.

Cho tập F các phụ thuộc hàm ta có :

a) $F \subseteq F^+$

b) Với f là phụ thuộc hàm tùy ý, f là hệ quả của $F \Leftrightarrow F \models f \Leftrightarrow f \in F^+$.

Ví dụ: Cho lược đồ quan hệ $Q(A B C D E G)$ với :

$F = \{f_1 : AE \rightarrow C, f_2 : CG \rightarrow A, f_3 : BD \rightarrow G, f_4 : GA \rightarrow E\}$

Chứng minh $BDC \rightarrow Q^+ \in F^+$

1) $BD \rightarrow G$

2) $BDC \rightarrow GC$ (tăng trưởng) (a)

3) $GC \rightarrow A \Rightarrow GC \rightarrow AG$ (tăng trưởng)

4) Từ 3 và 2 $\Rightarrow BDC \rightarrow AG$ (bắc cầu)

5) $AG \rightarrow E$ và 4 $\Rightarrow BDC \rightarrow E$ (bắc cầu)

6) Từ 2,4,5, $BDC \rightarrow BDC \Rightarrow BDC \rightarrow Q^+$

4.3.3 Bao đóng của tập thuộc tính.

□ Định nghĩa

Cho lược đồ quan hệ Q với tập $FD F = \{f_1, f_2, \dots, f_n\}$. Giả sử $X \subseteq Q^+$

Bao đóng của X dựa trên F ký hiệu $X_F^+ = \{Y \in Q^+ / X \rightarrow Y \in F^+\}$

Nhận xét :

□ $X \subseteq X^+$

□ $f: X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X_F^+$

□ Giải thuật tính X_F^+

Vào : F và $X \subseteq Q^+$

Ra : X_F^+

Phương pháp :

1. Temp = 0

2. While temp \neq X DO

```

{ Temp = X
for   $\forall f = W \rightarrow Z \in F$  Do
    if  $W \subset X$  then  $X = X \cup Z$ 
}
3. Return (X)

```

Ví dụ: $F = \{A \rightarrow D, AB \rightarrow E, BI \rightarrow E, CD \rightarrow I, E \rightarrow C\}$

$X = AE$ thì $X^+ = A C D E I$.

Ví dụ: $F = \{B \rightarrow A, DA \rightarrow CF, D \rightarrow H, GH \rightarrow C, AC \rightarrow D\}$

$BD_F^+ = B D A C E H$; $AC_F^+ = A C D E H$

4.3.4 Bài toán thành viên :

- Cho tập FD F , $f : X \rightarrow Y$ tùy ý, $f \in F^+ \Leftrightarrow Y \subset X^+$.
- Giải thuật Thành viên (f, F)

Vào: $F, f : X \rightarrow Y \in F^+$

Ra : Đ, S

Phương pháp : Return ($Y \subset X_F^+$)

Ví dụ: $F = \{A \rightarrow D, AB \rightarrow E, BI \rightarrow E, CD \rightarrow I, E \rightarrow C\}$

Xét $F : AE \rightarrow DI \in F^+$

$AE_F^+ = A E D C I$, $DI \subseteq AE_F^+$ nên $f \in F^+$

4.4 Khoá của quan hệ

4.4.1 Xác định khoá

- Tìm các nút gốc X
- Tính bao đóng của các nút gốc X
Nếu $X_F^+ = U$ thì X là khoá
Ngược lại $X = X \cup Y$ với $Y \in U$
Quay lại bước 2.

Ví dụ: Cho R (A B C D E H)

$F = \{AB \rightarrow C, CD \rightarrow E, EC \rightarrow A, CD \rightarrow H, H \rightarrow B\}$

Tìm 1 khoá của R

- Nút gốc $X = \{\text{vế trái}\} - \{\text{vế phải}\} + \{\text{các thuộc tính không có trong các PTH}\}$
 $= \{A B C D E H\} - \{C E A H B\}$
 $X = \{D\}$
- $X_F^+ = \{D\}$
 $X = DC$; $DC_F^+ = \{C D E A H B\}$
 Nên CD là khoá.

4.4.2 Xác định tất cả khoá chính.

Nhận xét : Nếu K là khoá của Q thì K chứa tất cả các thuộc tính nguồn và không chứa bất kỳ thuộc tính đích nào.

B1. Tập gốc $N = U - \{\text{vế phải}\}$

Tập trung gian $L = U - N$

Xây dựng 2^K tập con của L : L_1, L_2, \dots, L_{2^K} bằng phương pháp đường chạy nhị phân.

B2. Xây dựng tập K chứa siêu khoá.

For $i = 1$ to 2^K do.

Begin

$X_i := N \cup L_i$.

Tính bao đóng $(X_i)^+_F$

if $(X_i)^+_F = Q^+$ theo $K = K \cup X_i$

End

B3. Loại bỏ dần các siêu khoá lớn trong tập hợp

Các siêu khoá $K = \{K_1, K_2, \dots, K_3\}$

Ví dụ. $Q(A B C D E G)$

$F = \{f_1 : AE \rightarrow C, f_2 : CG \rightarrow A, f_3 : BD \rightarrow G, f_4 : GA \rightarrow E\}$

Xác định tất cả khoá của Q

B1. $N = \{A E C G B D\} - \{A C G E\}$

$= \{BD\}$

$L = Q^+ \setminus N = \{A C E G\}$

A C E G	L_i	$X_i = N \cup L_i$	$(X_i)^+_F$
0 0 0 0		BD	BDG
1 0 0 0	A	<u>BDA</u>	Q^+
0 1 0 0	C	<u>BDC</u>	Q^+
1 1 0 0	AC	BDAC	/
0 0 1 0	E	BDE	BDEG
1 0 1 0	AE	BDAE	/
0 1 1 0	CE	BDCE	/
1 1 1 0	ACE	BDACE	/
0 0 0 1	G	BDG	BDG
1 0 0 1	AG	BDAG	/
0 1 0 1	CG	BDCG	/
1 1 0 1	ACG	BDACG	/
0 0 1 1	EG	BDEG	BDEG
0 1 1 1	CEG	BDCEG	/
1 1 1 1	ACEG	BDACEG	/
1 0 1 1	AEG	BDAEG	/

Kết luận : Khóa chính của Q : BDA, BDC

4.5 Phủ tối thiểu của tập phụ thuộc hàm.

4.5.1 Định nghĩa: Hai tập phụ thuộc hàm F và G.

a) Ta nói F suy ra G ký hiệu là $F \models G$ nếu $G^+ \subseteq F^+$

b) Ta nói F tương đương G ký hiệu là $F \equiv G$ nếu $F \models G$ và $G \models F$

Cho tập phụ thuộc hàm F. Xét tập phụ thuộc hàm G nếu $G \models F$ ta nói G là một phủ của F (và ngược lại).

Ví dụ: $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C, AB \rightarrow C, B \rightarrow BC\}$
 $G = \{A \rightarrow B, B \rightarrow C\}$

Ta thấy $F \equiv G$ tuy nhiên ta cần nói G là 1 phủ của F theo nghĩa là G tốt hơn F vì ít phụ thuộc hàm hơn.

Chú ý : Ta có các mệnh đề sau

1. $F \subset F^+$
2. $F \subset G \rightarrow F^+ \subset G^+$
3. $(F^+)^+ = F^+$
4. Cách CM $F^+ \subset G^+$

Ta chứng minh $F \subset G^+$ nghĩa là với $\forall FD X \rightarrow Y \in F$ ta chứng minh $Y \subseteq X_{G^+}$

4.5.2 Định Nghĩa: Phủ tối thiểu.

Một tập phụ thuộc hàm F được gọi là phủ tối thiểu nếu thỏa :

- a) $\forall f, f \in F, f$ có dạng $X \rightarrow A, A$ chỉ có 1 thuộc tính.
- b) Không tồn tại phụ thuộc hàm $f: X \rightarrow A$ và $Z \subset X$ sao cho
 $F \equiv F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$
- c) Không tồn tại phụ thuộc hàm $f: X \rightarrow A$ để $F - \{X \rightarrow A\}$ còn tương đương với F.

Tính chất (c) bảo đảm không dư thừa.

Tính chất (b) bảo đảm không có phụ thuộc hàm không đầy đủ.

Tính chất (a) bảo đảm không có thuộc tính nào của vế phải là tổ hợp.

4.5.3 Giải thuật tìm phủ tối thiểu.

Vào : Tập phụ thuộc hàm F.

Ra: Phủ tối thiểu của F.

Phương pháp :

$G := \emptyset$

a) For $\forall f \in F, f: X \rightarrow Y$ Do $G = G \cup \{X \rightarrow A, A \in Y\}$.

b) For $\forall f: X \rightarrow A \in G$ Do

if $\exists Z \subset X, Z \neq X, G \equiv G \setminus \{f\} \cup \{Z \rightarrow A\}$ Then

$G = G - \{f\} \cup \{Z \rightarrow A\}$

c) For $\forall f: X \rightarrow A \in G$ Do

If $G \setminus \{f\} \equiv G$ then $G = G - \{f\}$

Ví dụ : Cho tập phụ thuộc hàm F:

$$F = \{ AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, \\ BE \rightarrow C, CG \rightarrow BD, CE \rightarrow AG \}$$

Tìm phủ tối thiểu của F ?

Thực hiện giải thuật trên ta có:

$$\text{Sau bước 1,2: } G = \{ AB \rightarrow C, C \rightarrow A, BC \rightarrow D, \text{ACD} \rightarrow B, \\ D \rightarrow E, D \rightarrow G, BE \rightarrow C, CG \rightarrow B, \\ CG \rightarrow D, \text{CE} \rightarrow A, CE \rightarrow G \}$$

$$\text{Sau bước 2: } G = \{ AB \rightarrow C, C \rightarrow A, BC \rightarrow D, CD \rightarrow B, \\ D \rightarrow E, D \rightarrow G, BE \rightarrow C, CG \rightarrow B, \\ CG \rightarrow D, CE \rightarrow G \}$$

// loại CE \rightarrow A vì có C \rightarrow A ; ta có CD \rightarrow B nên loại ACD \rightarrow B

$$\text{Sau bước 3: } G = \{ AB \rightarrow C, C \rightarrow A, BC \rightarrow D, \\ D \rightarrow E, D \rightarrow G, BE \rightarrow C, CG \rightarrow B, CE \rightarrow G \}$$

// loại CD \rightarrow B, CG \rightarrow D vì có thể được suy từ các PTH còn lại

4.6. Phép tách các lược đồ quan hệ

4.6.1 Định nghĩa

Phân rã (decomposition) 1 lược đồ quan hệ $R = \{A_1, A_2, \dots, A_n\}$ là thay đổi R bằng 1 tập hợp $\rho = \{R_1, R_2, \dots, R_K\}$ với R_i là các lược đồ con của R sao cho $R_1^+ \cup R_2^+ \cup \dots \cup R_K^+ = R^+$ trong đó R không nhất thiết phải rời nhau. Mục tiêu của phép tách chủ yếu là để nâng cao chất lượng quan hệ sao cho đạt dạng chuẩn cao hơn.

Ví dụ: Cho lược đồ quan hệ người cung cấp:

$$S = (SNAME, ADD, PRO, PRICE) \\ SNAME \rightarrow ADD \\ SNAME, PRO \rightarrow PRICE$$

S được thay thế bằng 2 lược đồ con khác.

$$S1 (SNAME, ADD) \\ S2 (SNAME, PRO, PRICE)$$

Các tính chất của một phân rã mà ta cần quan tâm đến là:

- ☐ Phân rã bảo toàn thông tin
- ☐ Phân rã bảo toàn phụ thuộc
- ☐ Tính các phụ thuộc hàm được bao trong một lược đồ con của ρ .

4.6.2 Phân rã bảo toàn thông tin

☐ Nếu R là 1 lược đồ quan hệ được tách thành các lược đồ con R_1, R_2, \dots, R_K và D là tập các phụ thuộc hàm. Ta nói rằng phép tách là tách kết nối không mất mát thông tin đối với D, nếu với mỗi quan hệ r trên R thỏa D:

$$r = \prod (r) \bowtie \prod (r) \bowtie \dots \bowtie \prod (r)$$

- Kiểm tra phép kết nối không mất mát thông tin.

Vào: Lược đồ quan hệ R , tập FD F và phép tách $P = (R_1, \dots, R_K)$.

Ra: Kết luận phép tách P có mất mát thông tin ?

Phương pháp:

- a) Thiết lập 1 bảng với n cột và K hàng.

Cột thứ j ứng với thuộc tính A_j .

Hàng thứ i ứng với lược đồ R_i .

Tại hàng i và cột j điền ký hiệu a_j nếu $A_j \in R_i^+$, nếu không thì điền b_{ij} .

- b) Xét các phụ thuộc hàm từ F áp dụng cho bảng trên.

Xét $X \rightarrow Y \in F$, xét các hàng nếu có giá trị bằng nhau trên thuộc tính X thì làm bằng nhau trên thuộc tính Y .

Chú ý: Khi làm bằng giá trị trên Y , nếu 1 trong 2 giá trị là a_j thì ưu tiên làm bằng ký hiệu a_j . Ngoài ra làm chúng bằng 1 trong các ký hiệu b_{ij} .

Tiếp tục áp dụng phụ thuộc hàm cho bảng (kể cả việc lập lại các phụ thuộc hàm đã được áp dụng) cho tới khi không còn áp dụng được nữa.

Nếu xuất hiện 1 hàng gồm ký hiệu (a_1, a_2, \dots, a_n) thì phép kết nối không mất mát thông tin.

Ví dụ: $S (SNAME, ADD, PRO, PRICE)$.

$S_1 (SNAME, ADD)$

$S_2 (SNAME, PRO, PRICE)$

$SNAME \rightarrow ADD$

$SNAME, PRO \rightarrow PRICE$

	SNAME	ADD	PRO	PRICE
S_1	a_1	a_2	b_{13}	b_{14}
S_2	a_1	b_{22}	a_3	a_4

Áp dụng $SNAME \rightarrow ADD$ ở hàng 2 (S_2)

Ta có hàng 2: $a_1 \quad a_2 \quad a_3 \quad a_4$

Do đó phép kết nối không mất mát thông tin.

- **Định lý Delobel**

Nếu $\rho (R_1, R_2)$ là 1 phép tách của R và F , thì ρ là tách không mất mát thông tin đối với F khi và chỉ khi.

$$R_1 \cap R_2 \rightarrow R_1 - R_2 \quad \text{hoặc} \quad R_1 \cap R_2 \rightarrow R_2 - R_1.$$

Chú ý: Các phụ thuộc hàm nêu trên không nhất thiết phải thuộc tập F ban đầu nhưng phải thuộc F^+ .

4.6.3 Phân rã bảo toàn phụ thuộc

4.6.3.1 Định nghĩa phụ thuộc hàm được bao trong một lược đồ quan hệ

- Một phụ thuộc hàm $f: X \rightarrow Y$ được gọi là phụ thuộc hàm được bao trong lược đồ quan hệ R nếu $XY \subset R^+$

□ Một phụ thuộc hàm $f: X \rightarrow Y$ được gọi là phụ thuộc hàm được bao trong lược đồ cơ sở dữ liệu $D = \{ R_1, R_2, \dots, R_p \}$ nếu f được bao trong $R_i, i \in [1..p]$.

4.6.3.2 Định nghĩa phụ thuộc hàm bị ép thoả trong lược đồ cơ sở dữ liệu

□ Gọi G là tập tất cả các phụ thuộc hàm được bao trong D thì

□ $\forall f \in G^+, f$ được gọi là bị ép thoả trong lược đồ cơ sở dữ liệu D .

□ $\forall f \in (F^+ - G^+)$, f được gọi là không bị ép thoả trong lược đồ cơ sở dữ liệu D .

□ Gọi G là tập tất cả các phụ thuộc hàm được bao trong D ta nói tập phụ thuộc hàm F được gọi là bị ép thoả trong D nếu $F \equiv G$.

Ví dụ: Cho $D = \{ R_1, R_2, R_3 \}$ với $R_1(A, B, C), R_2(B, C, D), R_3(D, E)$

Xét $F = \{ A \rightarrow BC, C \rightarrow A, A \rightarrow D, C \rightarrow D, D \rightarrow E, A \rightarrow E \}$

Ta nhận thấy phụ thuộc hàm $A \rightarrow D$ và $A \rightarrow E$ không được bao trong D nhưng F bị ép thoả trong D vì ta tìm được tập $G = \{ A \rightarrow BC, C \rightarrow A, D \rightarrow E, C \rightarrow D \}$ gồm các phụ thuộc hàm được bao trong D và $G \equiv F$.

4.6.3.3 Điều kiện phân rã bảo toàn phụ thuộc

Một phân rã ρ là bảo toàn phụ thuộc F nếu F bị ép thoả trong ρ với tư cách là một lược đồ cơ sở dữ liệu.

4.6.3.4 Giải thuật kiểm tra phân rã ρ có bảo toàn phụ thuộc

Cho lược đồ quan hệ R , tập phụ thuộc hàm F định nghĩa trên R và phân rã $\rho = \{ R_j \mid j = 1..n \}$.

Giải thuật $\text{Eclosure}(X, F, \rho)$

Vào: tập thuộc tính X , Tập phụ thuộc hàm F , phân rã ρ

Ra : X_F^+

Phương pháp:

1. $Y := \emptyset$
2. While $Y \neq X$ do

Begin $Y := X$;
 for $\forall R_j \in \rho$ do
 $X := X \cup ((X \cap R_j)_F^+ \cap R_j)$;
 End;
3. Return X ;

Giải thuật $\text{Enforce}(F, \rho)$

Vào : Tập phụ thuộc hàm F , phân rã ρ

Ra : Đúng, Sai

Phương pháp:

1. For $\forall f \in F$, f không được bao trong ρ , $f: X \rightarrow Y$ do

if Not $Y \subset \text{Eclosure}(X, F, \rho)$ then return False;

2. Return true;

Ví dụ Cho $F = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A \}$ và phân rã $\rho = \{(AB), (BC), (CD)\}$. Kiểm tra phân rã ρ có bảo toàn phụ thuộc F không?

Ta nhận thấy chỉ có phụ thuộc hàm $D \rightarrow A$ không được bao trong ρ . Áp dụng giải thuật Enforce(F, ρ) ta kiểm tra A có chứa trong $\text{Eclosure}(D)$?

Tính toán ta được $\text{Eclosure}(D) = ABCD$ tức $A \subset \text{Eclosure}(D)$. Suy ra phân rã ρ bảo toàn phụ thuộc trong F .

4.6.4 Tính các phụ thuộc hàm được bao trong một lược đồ con của ρ

□ Định nghĩa chiều của tập phụ thuộc hàm trên lược đồ con

Cho lược đồ quan hệ R , tập phụ thuộc hàm F và phân rã ρ . Xét một lược đồ con $R_i \in \rho$, ta định nghĩa chiều của tập phụ thuộc hàm trên lược đồ con $\pi_{R_i}(F) = \{ f \in F^+ \mid f \text{ được bao trong } R_i \}$

□ Nhận xét:

Xét $X \subset R$, nếu:

1. Không tồn tại $f \in F$, về trái(f) $\subset X$ thì $X^+_F = X$
2. Tồn tại $f \in F$, về trái(f) $\subset F$ thì $X \rightarrow$ về phải(f) là một phụ thuộc hàm không đầy đủ.

□ Giải thuật tính $\pi_{R_i}(F)$

Vào: Tập phụ thuộc hàm F , phân rã ρ , $R_i \subset \rho$

Ra: $F_i = \pi_{R_i}(F)$

Phương pháp:

1. $F_i := \emptyset$;
2. For $\forall f: X \rightarrow Y \in F, X \subset R_i$ do
 $F_i := F_i \cup \{ X \rightarrow (X^+_F - X) \cap R_i \}$;
3. Return (F_i);

Ví dụ Cho tập phụ thuộc hàm $F = \{ A \rightarrow C, B \rightarrow C \}$ và phân rã $\rho = \{ AC, AB \}$.

Tính $F_1 = \pi_{AC}(F)$

$F_1 := \emptyset$;

$A^+ = AC$ suy ra $F_1 = \{ A \rightarrow C \}$

Tính $F_2 = \pi_{AB}(F)$

$F_2 := \emptyset$;

$A^+ = AC$ mà $(AC - A) \cap AB = \emptyset$ nên không bổ sung $A \rightarrow C$ vào F_2

$B^+ = BC$ mà $(BC - B) \cap AB = \emptyset$ nên không bổ sung $B \rightarrow C$ vào F_2

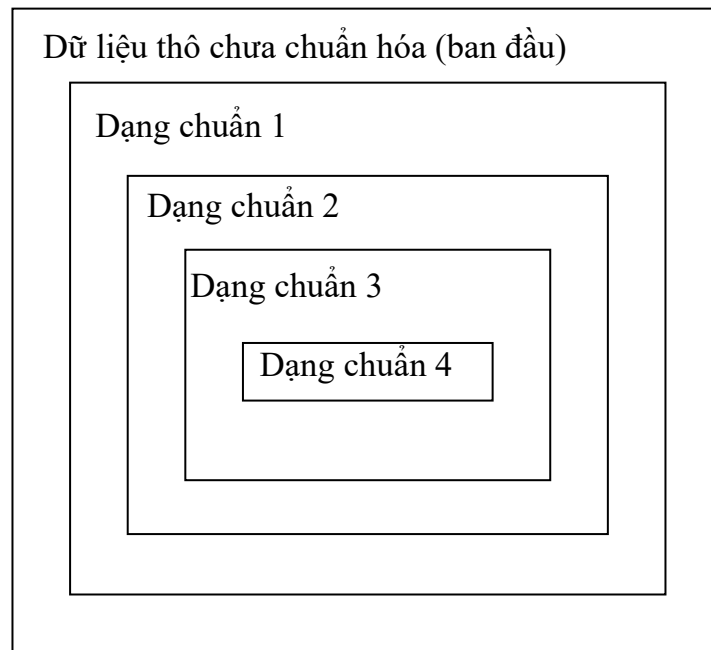
Suy ra $F_2 = \emptyset$

CHƯƠNG 5 CHUẨN HÓA CƠ SỞ DỮ LIỆU

5.1 Các dạng chuẩn

Mục đích của dạng chuẩn là loại bỏ dư thừa dữ liệu mà các phụ thuộc hàm có thể gây ra trong lược đồ cơ sở dữ liệu và những bất thường xảy ra khi thao tác dữ liệu (thêm, xóa dữ liệu).

Các dạng chuẩn được đưa ra có tính thừa kế như sau:



5.2 Dạng chuẩn 1 : 1NF (Normal Form)

□ **Định nghĩa:** Một quan hệ R được xem ở dạng chuẩn 1 nếu và chỉ nếu tất cả thuộc tính trong R đều chứa các giá trị nguyên tố (giá trị không tách ra được nữa)

Ví dụ : Ta xét quan hệ người cung cấp bán các mặt hàng.

S#	PQ	
	P#	QTY
S1	P1	300
	P2	200
	P3	400
	P4	200
	P5	100
	P6	100
S2	P1	300
	P2	400
S3	P2	200
S4	P2	200
	P4	300
	P5	400
S5	⊥ (Null)	⊥

- R (S#, PQ) không phải là 1NF
- Nếu là 1NF phải tách thuộc tính PQ → P#, QTY

□ **Nhược điểm của dạng 1NF**

- Sửa 1 mã phải sửa tất cả các mã nên tốn thời gian. Còn nếu không sửa hết thì mâu thuẫn dữ liệu. Tương tự khi xóa thì ta phải xóa tất cả.

Ví dụ : Người cung cấp mặt hàng với 1 số lượng, người cung cấp ở 1 thành phố nào đó và mỗi thành phố chỉ có 1 chỉ số thành phố.

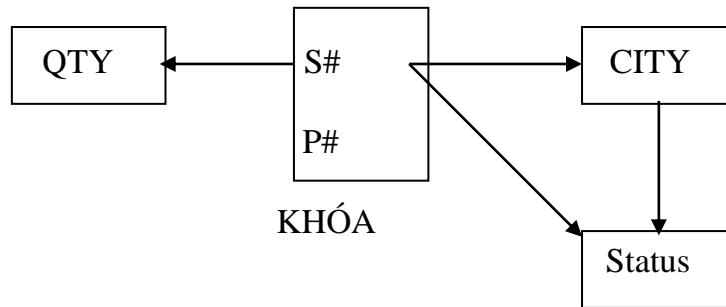
S#	Status	City	P#	Qty
S1	20	London	P1	300
S1	20	London	P2	200
S1	20	London	P3	400
S1	20	London	P4	200
S1	20	London	P5	100
S2	10	Paris	P1	300
S2	10	Paris	P2	400
S3	10	Paris	P3	200
S4	20	London	P2	200

□ **Nguyên nhân** : Có các thuộc tính không khóa không phụ thuộc hàm đầy đủ vào khóa chính.

5.3 Dạng chuẩn 2NF

Nhắc lại khái niệm phụ thuộc hàm đầy đủ :

Nếu 1 thuộc tính Y phụ thuộc hàm đầy đủ vào thuộc tính X thì Y phụ thuộc hàm vào X và không phụ thuộc hàm vào tập con bất kỳ nào của X.



□ **Định nghĩa:** Một quan hệ R ở 2NF nếu R ở 1NF và các thuộc tính không khóa phải phụ thuộc hàm đầy đủ vào khóa chính.

R1

S#	P#	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S2	P1	300
S2	P2	400
S3	P3	200
S4	P2	200

R2

S#	Status	City
S1	20	London
S2	10	Paris
S3	10	Paris
S4	20	London

□ **Nhược điểm :**

- Không thể thêm 1 thành phố mới nếu chưa có người cung cấp.
- Không thể thêm NewYork, mã số 15 nếu chưa có S#

Ví dụ : Khi sửa Status của London thành 25 sẽ tốn thời gian

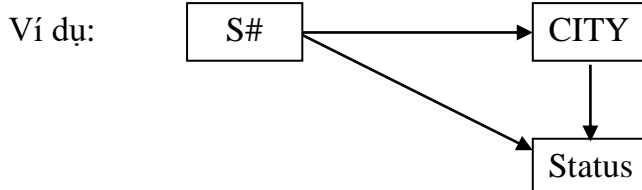
□ **Nguyên nhân :**

Có sự phụ thuộc hàm bắc cầu vào khóa chính.

5.4. Dạng chuẩn 3NF

5.4.1 Định nghĩa

Một quan hệ R ở dạng 3NF nếu R đã ở dạng chuẩn 2NF và mỗi thuộc tính không khóa không phụ thuộc hàm bắc cầu vào khóa chính.



R21

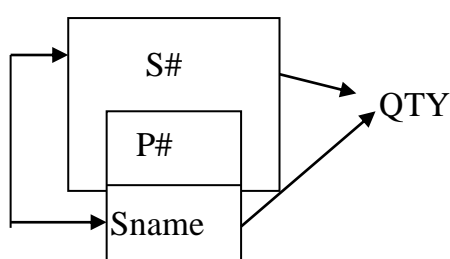
S#	City
S1	London
S2	Paris
S3	Paris
S4	London

R22

City	Status
London	20
Paris	10

Trong 3NF có thể có nhiều khóa chính chứ không phải 1 khóa chính.

Ví dụ : R (S#, P#, Sname, Qty)



- ☐ **Nhược điểm:** Quan hệ này bị dư thừa do S# và Sname lặp lại nhiều lần.
- ☐ **Nguyên nhân :** có sự phụ thuộc hàm lẫn nhau.

5.4.2 Giải thuật phân rã bảo toàn phụ thuộc hàm thành dạng chuẩn 3

Vào : Lược đồ quan hệ R, tập phụ thuộc hàm F, chúng ta giả sử F là phủ tối thiểu.

Ra : Một phân rã ρ bảo toàn phụ thuộc của R sao cho mỗi lược đồ quan hệ con đều có dạng chuẩn 3 ứng với chiếu của F trên lược đồ đó.

Phương pháp:

1. Nếu có các thuộc tính của R không liên quan tới một phụ thuộc hàm nào trong F, cả ở vế trái lẫn vế phải, thì về nguyên tắc các thuộc tính này có thể nhóm lại thành một quan hệ và ta có thể tách ra khỏi R.
2. Nếu có một phụ thuộc hàm nào liên quan đến toàn bộ các thuộc tính trong R thì R không phân chia nữa và kết quả là chính R.
3. Ngược lại, chúng ta phân rã R thành các lược đồ quan hệ XA ứng với mỗi phụ thuộc hàm $X \rightarrow A$ trong F. Tuy nhiên nếu $X \rightarrow A_1$, $X \rightarrow A_2, \dots$, $X \rightarrow A_n$ là các phụ thuộc hàm trong F thì ta sử dụng lược đồ $XA_1A_2 \dots A_n$ thay cho n lược đồ XA_1, XA_2, \dots, XA_n

Ví dụ: Xét lược đồ R = CTHRSRG với tập phụ thuộc hàm F như sau:

$$F = \{C \rightarrow T, CS \rightarrow G, HR \rightarrow C, HS \rightarrow R, HT \rightarrow R\}$$

Giải thuật trên sinh ra tập các lược đồ quan hệ CT, CGS, HRC, HSR, HTR bảo toàn phụ thuộc hàm.

5.4.3 Phân rã thành dạng chuẩn 3 bảo toàn phụ thuộc hàm và bảo toàn thông tin

Vấn đề đặt ra là ta có thể phân rã một lược đồ quan hệ về dạng chuẩn 3 trong đó vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin không? Câu trả lời là luôn luôn làm được điều này bằng cách ghép ρ với một lược đồ quan hệ X là khoá của R theo định lý sau:

Định lý : Gọi ρ là một phân rã dạng chuẩn 3 của R bằng giải thuật *phân rã bảo toàn phụ thuộc hàm thành dạng chuẩn 3* và X là khoá của R. Thế thì $\tau = \rho \cup \{X\}$ là một phân rã của R mà tất cả các lược đồ quan hệ đều có dạng chuẩn 3; phân rã này có đặc tính bảo toàn phụ thuộc hàm và bảo toàn thông tin.

Ví dụ: Sử dụng lại ví dụ ở phần 5.4.2 với SH là khoá của R thì phân rã thành các lược đồ quan hệ (CT, CGS, HRC, HSR, HTR) vừa bảo toàn phụ thuộc hàm và bảo toàn thông tin vì tính cờ HS nằm trong lược đồ quan hệ HSR.

5.5. Dạng chuẩn 4 : BCNF (Boyce Codd Normal Form)

5.5.1 Định nghĩa

Một quan hệ R ở dạng chuẩn BCNF nếu mọi định thuộc đều là khóa.

Thuộc tính X được gọi là định thuộc nếu X định trị hàm 1 thuộc tính khác nghĩa là X có phụ thuộc hàm với thuộc tính khác.

Ví dụ : R (S#, P#, SNAME, QTY)

$S\#, P\# \Rightarrow QTY$

$SNAME, P\# \Rightarrow QTY$

$S\# \Rightarrow SNAME$

$SNAME \Rightarrow S\#$

\Rightarrow R không phải dạng chuẩn BCNF

Nếu muốn đạt dạng chuẩn BCNF ta phải tách R thành 2 quan hệ R1 và R2:

R1 (S#, P#, Qty)

R2 (S#, Sname)

5.5.2 Giải thuật phân rã bảo toàn thông tin thành dạng chuẩn BCNF

□ **Bổ đề 1**

Giả sử R là một lược đồ quan hệ với tập phụ thuộc hàm F. Gọi $\rho = (R_1, R_2, \dots, R_n)$ là một phân rã của R bảo toàn thông tin tương ứng với F và giả sử $\sigma = (S_1, S_2)$ là một phân rã của R1 bảo toàn thông tin tương ứng với $\pi_{R1}(F)$. Khi đó phân rã của R thành tập $(S_1, S_2, R_2, \dots, R_n)$ là bảo toàn thông tin tương ứng với F.

□ **Bổ đề 2**

- Mọi quan hệ cấp 2 (Chỉ có hai thuộc tính) đều ở dạng chuẩn BCNF
- Nếu R không ở dạng chuẩn BCNF, ta sẽ tìm được 2 thuộc tính A,B trong R sao cho $(R-AB) \rightarrow A$ hay $(R-AB) \rightarrow B$

□ **Giải thuật phân rã bảo toàn thông tin thành dạng chuẩn BCNF**

Vào: Lược đồ quan hệ R và tập phụ thuộc hàm F

Ra: Một phân rã bảo toàn thông tin sao cho các lược đồ quan hệ con nhận được ở dạng chuẩn BCNF ứng với hình chiếu của F trên lược đồ này.

Phương pháp: Trọng tâm của giải thuật là phân rã lược đồ quan hệ R thành hai lược đồ quan hệ. Một lược đồ quan hệ có tập thuộc tính XA, nó có dạng BCNF và phụ thuộc hàm $X \rightarrow A$ là đúng. Lược đồ quan hệ thứ hai R-A, do đó phân rã R-A, XA là bảo toàn thông tin. Thực hiện đệ quy thủ tục phân rã với R-A cho đến khi đạt được lược đồ quan hệ đáp ứng với điều kiện bổ đề 2b để lược đồ quan hệ này ở dạng BCNF. Thế thì bổ đề 1 bảo đảm rằng lược đồ quan hệ này cộng với các lược đồ quan hệ BCNF được tạo ra từ mỗi đệ quy là bảo toàn thông tin.

Chương trình chính

Z := R; // T là phần còn lại của R sau mỗi đợt phân rã

Repeat

decompose((T,R₁,R₂); //T được phân rã thành R₁=XA và R₂=T-A ($X \rightarrow A$)

if $R_1 \neq \emptyset$ then Bổ sung R₁ vào phân rã ;

T := R₂;

until $R_1 = \emptyset$;

Bổ sung T vào phân rã .

Chương trình con Decompose(T,R₁,R₂)

if T không chứa cặp thuộc tính {A,B} sao cho $A \in (T-AB)^+_F$ Then

Trả lại R₁ := \emptyset ; R₂ := T; // T Đã ở dạng BCNF

else Begin

Chọn ra cặp {A,B} như vậy; // $A \in (T-AB)^+_F$

R₁ := T - B; // R₁ = XA và thoả $X \rightarrow A$ nhưng chưa chắc ở BCNF

While R₁ còn chứa cặp thuộc tính {A,B} sao cho $A \in (T-AB)^+_F$ Do

R₁ := R₁ - B;

Return R₁, R₂ := T-A;

end;

Ví dụ: Xét lược đồ quan hệ R(CTHRSG) và tập phụ thuộc hàm F
 $F = \{C \rightarrow T, CS \rightarrow G, HR \rightarrow C, HS \rightarrow R, HT \rightarrow R\}$

5.6 Dạng chuẩn 4NF :

5.6.1 Phụ thuộc đa trị

Một thuộc tính X xác định đa trị thuộc tính Y ký hiệu là $X \twoheadrightarrow Y$ ở trong R nếu với mọi r của R (r là 1 minh họa) và 2 bộ μ và ν ở trong r với $\mu[x] = \nu[x]$ thì trong R cũng có 2 bộ là ϕ, ψ sao cho thoả mãn đồng thời 3 điều kiện sau :

- 1 $\phi[x] = \psi[x] = \mu[x]$
- 2 $\phi[x] = \mu[x]$ và $\phi[R-X-Y] = \gamma[R-X-Y]$
- 3 $\psi[Y] = \nu[Y]$ và $\psi[R-X-Y] = \mu[R-X-Y]$

R (MH , GV , GT)

μ	Physics	Green	Basic mechanics
	Physics	Brown	Principle of optics
ν	Physics	Brown	Basic mechanics
	Physics	Green	Principle of optics
ϕ			

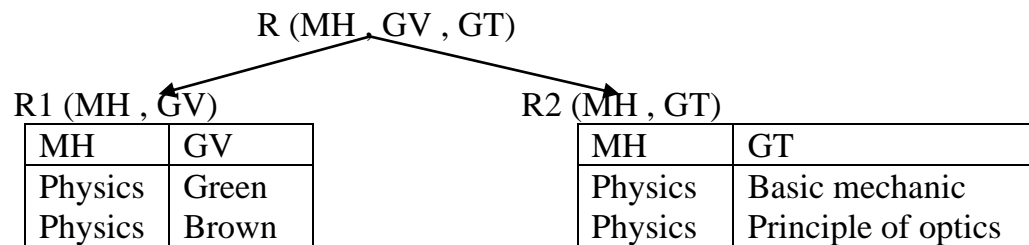
Một định nghĩa khác

Một thuộc tính Y phụ thuộc đa trị vào các thuộc tính X nếu với mọi quan hệ r của R có 2 bộ có cùng thành phần khi ta hoán đổi 2 bộ với nhau và các thành phần còn lại vẫn giữ nguyên thì ta sẽ được 2 bộ mới cùng ở trong r.

$$\left. \begin{array}{l} MH \twoheadrightarrow GT \\ GV \twoheadrightarrow GT \end{array} \right\} \begin{array}{l} MH \text{ phụ thuộc đa trị vào } GT \\ GV \text{ phụ thuộc đa trị vào } GT \end{array}$$

5.6.2 Dạng chuẩn 4NF

Qui tắc: Một quan hệ R ở dạng chuẩn 4NF nếu và chỉ nếu thuộc tính B phụ thuộc đa trị vào thuộc tính A thì các thuộc tính còn lại của R phụ thuộc hàm vào A.



CHƯƠNG 6. TÍNH TOÀN VỆ DỮ LIỆU

Để bảo đảm cho dữ liệu trong cơ sở dữ liệu được toàn vẹn ta phải thực hiện:

- Ràng buộc toàn vẹn dữ liệu
- Phân quyền truy xuất

6.1 Ràng buộc toàn vẹn (Integrity Constraints)

6.1.1 Định nghĩa :

- Ràng buộc toàn vẹn là các quy tắc bất biến đối với các quan hệ trong cơ sở dữ liệu nhằm làm cho các dữ liệu được nhất quán, hợp lý.
- Quy tắc ở đây là 1 vị từ $\forall x, p(x)$ đúng trong đó x có thể là 1 quan hệ, 1 bộ.

6.1.2 Các yếu tố của ràng buộc toàn vẹn :

- Điều kiện.

Người ta biểu diễn điều kiện của ràng buộc toàn vẹn

R1 : "Mỗi sv có 1 mã số riêng , không trùng với 1 sv nào khác".

$\forall sv_1, sv_2 \in \text{SINHVIEN} :$

$sv_1 = sv_2 \rightarrow sv_1.MASV = sv_2.MASV$

Cuối \forall

R2 : "Mỗi sv phải đăng ký vào 1 lop của trường".

$\pi_{\text{MALOP}}(\text{SINHVIEN}) \subseteq \pi_{\text{MALOP}}(\text{LOP})$ hoặc :

$\text{SINHVIEN.MALOP} \subseteq \text{LOP.MALOP}$

R3 : "Mỗi sv chỉ có thể thi tối đa 2 lần cho mỗi môn học".

$\forall sv \in \text{SINHVIEN}$

$\text{Card}(\sigma_{\text{MASV} = sv.MASV \wedge \text{MAMH} = 'X' }(\text{KETQUA})) \leq 2.$

Cuối \forall

$\forall kq \in \text{Tketqua}$

$kq.lanthi = 1 \text{ or } kq.lanthi = 2$

Cuối \forall

- Bối cảnh

Bối cảnh của 1 ràng buộc toàn vẹn R là những quan hệ mà ràng buộc đó có hiệu lực.

Ví dụ: Bối cảnh ràng buộc R1 chỉ là 1 quan hệ Tsinhvien.

Bối cảnh ràng buộc R2 chỉ là 2 quan hệ Tsinhvien và Tkhoa.

□ Bảng tầm ảnh hưởng:

Nhằm xác định thời điểm cần kiểm tra các ràng buộc toàn vẹn đó đối với ba thao tác thêm, xoá, sửa.

Quy ước : Dấu + : cần kiểm tra ràng buộc toàn vẹn R.

Dấu - : không cần kiểm tra ràng buộc toàn vẹn R.

Dấu * : không được sửa

Ví dụ: Các bảng tầm ảnh hưởng tương ứng với các ràng buộc toàn vẹn trên.

R ₁	Thêm	Xoá	Sửa (MASV)
Tsinhvien	+	-	*

R ₂	Thêm	Xoá	Sửa
T _{LOP}	-	+	+
Tsinhvien	+	-	+ (MALOP)

R ₃	Thêm	Xoá	Sửa (LAN)
Tketqua	+	-	+

(*) Quy ước không được sửa MASV, MAMH.

Bảng tầm ảnh hưởng tổng hợp của 3 ràng buộc toàn vẹn trên như sau:

Ràng buộc toàn vẹn	Tsinhvien			Tkhoa			Tketqua		
	T	S	X	T	S	X	T	S	X
R ₁	+	*	-						
R ₂	+	+	-	-	*	+			
R ₃							+	*	-

□ Các hành động cần chọn lựa khi ràng buộc toàn vẹn bị vi phạm:

- Ngăn cản và thông báo lý do không cho phép hành động được thực thi.
- Cảnh báo hậu quả gây ra nếu cố tình thực hiện hành động đó.

Ví dụ: Khi xoá bỏ một bộ trong quan hệ LOP thì các hậu quả có thể xảy ra trong quan hệ SINHVIEN đối với ràng buộc toàn vẹn R_2 là:

- Xoá toàn bộ các sinh viên có mã lớp đó.
- Thuộc tính khoá trong các bộ sinh viên có mã lớp đó trở thành NULL tức sinh viên chưa thuộc lớp nào cả.

6.1.3 Phân loại ràng buộc toàn vẹn :

6.1.3.1 Ràng buộc toàn vẹn có bối cảnh là 1 quan hệ.

□ Ràng buộc về miền giá trị :

Ví dụ: Trong lược đồ quan hệ Ketqua, ta có MGT (DIEM) = [0..10]

Điểm có độ chính xác = 0.25

$\forall kq \in T_{ketqua}$

$Kq.diem \text{ between } 0 \text{ AND } 10 \text{ AND } (kq.DIEM * 4) \bmod 2 \text{ IN } (0, 1)$

Cuối \forall

R	Thêm	Xoá	Sửa (diem)
Tketqua	+	-	+

□ Ràng buộc toàn vẹn liên thuộc tính

Ràng buộc toàn vẹn liên thuộc tính là mối liên hệ giữa các thuộc tính trong 1 quan hệ.

Ví dụ : Hoa_don (SOHD, NGAYHD, SODH, TRIGIA_HD, NGAYXUAT)

Ràng buộc "Hàng hoá chỉ được xuất kho khi đã lập hoá đơn"

$\forall hd \in T_{hoadon}$

$Hd.NGAYHD \leq hd.NGAYXUAT.$

Cuối \forall

Ví dụ: Ctiet_hd (SOHD, MAHH, GIABAN, SL_BAN, THANHTIEN)

Ràng buộc $THANHTIEN = SLBAN * GIABAN$

$\forall cthd \in T_{ctiet_hd}$

$cthd.THANHTIEN = cthd.SLBAN * cthd.GIABAN$

cuối \forall

□ Ràng buộc toàn vẹn liên bộ

Là sự ràng buộc giữa các bộ trong quan hệ nó thường được biểu diễn bằng phụ thuộc hàm, đặc biệt là ràng buộc toàn vẹn về khoá chính

Ví dụ: Ràng buộc toàn vẹn về khoá chính trong quan hệ Sinhvien

$\forall sv_1, sv_2 \in T_{sinhvien} :$

$sv_1 = sv_2 \rightarrow sv_1.MASV = sv_2.MASV$

Cuối \forall

Ví dụ: Ràng buộc toàn vẹn trên quan hệ MUONSACH của một độc giả là không được mượn quá 5 quyển sách

MUONSACH(Madg, Masach, ngmuon, ngtra)

$\forall m \in T_{Muonsach}$

$\text{card}(\sigma_{\text{madgX=m.madg and m.ngaytra=NULL}}(\text{Muonsach})) \leq 5$

Cuối \forall

6.1.3.2 Ràng buộc có bối cảnh gồm nhiều quan hệ

□ **Ràng buộc toàn vẹn về phụ thuộc tồn tại.**

Ràng buộc toàn vẹn về phụ thuộc tồn tại còn được gọi là ràng buộc về khoá ngoại

Ví dụ: Tketqua, sự tồn tại của 1 bộ.

$Kq = (\text{masv}, \text{mamh}, \text{lanthi}, \text{diem}) \in T_{\text{Tketqua}}$

Hoàn toàn phụ thuộc vào sự tồn tại của 1 bộ $sv \in T_{\text{sinhvien}}$ sao cho $sv.MASV = kq.Masv = \text{masv1}$, nếu không tồn tại bộ sv như thế thì bộ kq không được phép tồn tại trong Tketqua.

$DIEM.MASV \subseteq SINHVIEN.MASV$

Hoặc :

$\forall d \in T_{DIEM},$

$\exists sv \in T_{SINHVIEN} : d.MASV = sv.MASV$

Hết \forall

R_2	Thêm	Xoá	Sửa (MASV)
T_{DIEM}	+	-	+
T_{sinhvien}	-	+	+

Ví dụ: Trong quan hệ Tsinhvien, sự tồn tại của bộ $sv = ('91023', \dots, 'D12CQCN01', \dots) \in T_{\text{sinhvien}}$ hoàn toàn phụ thuộc vào sự tồn tại bộ $k \in T_{\text{LOP}}$ sao cho $k.MALOP = 'D12CQCN01'$

Tổng quát :

Giả sử có 2 lược đồ quan hệ Q và R. Hai tập hợp K_Q, K_R lần lượt là các khoá của Q và R. Ta có 2 dấu hiệu của phụ thuộc tồn tại như sau :

□ Nếu $K_Q \subseteq K_R$ thì có 1 phụ thuộc tồn tại của R vào Q ký hiệu : $R[K_Q] \subseteq Q[K_Q]$

□ Nếu $K_Q \subseteq R^+$ thì có 1 phụ thuộc tồn tại của R vào Q ký hiệu : $R[K_Q] \subseteq Q[K_Q]$.

Tập hợp K_Q gọi là khoá ngoại của R .

□ **Ràng buộc toàn vẹn liên bộ :**

Ràng buộc toàn vẹn liên bộ liên quan hệ có tác dụng đối với từng nhóm các bộ của nhiều quan hệ khác nhau mà thông thường là 2 quan hệ.

Ví dụ: Xét 2 lược đồ quan hệ sau:

DOCGIA(madg, tendg, diachi, sosachdangmuon, ngaydk)

MUONSACH(madg, mash, ngmuon, ngtra)

Ta có ràng buộc số sách đang mượn của một độc giả phải bằng số bộ trong MUONSACH có cùng mã độc giả đó mà ngày trả là NULL

$\forall d \in T_{docgia}$

$d.sosachdangmuon = card(\sigma_{madg=d.madg \text{ and } NGAYTRA \text{ IS NULL}}(MUONSACH))$

cuối \forall

R	Thêm	Xoá	Sửa
T_{DOCGIA}	-	+	*
$T_{MUONSACH}$	+	*	+(ngtra)

□ **Ràng buộc toàn vẹn liên thuộc tính liên quan hệ**

Ràng buộc toàn vẹn liên thuộc tính liên quan hệ là ràng buộc toàn vẹn giữa các thuộc tính trong nhiều quan hệ.

Ví dụ: Ta xét ràng buộc toàn vẹn ngày mượn sách phải lớn hơn hoặc bằng ngày đăng ký.

$\forall d \in T_{docgia}$

$\forall m \in T_{muonsach} : d.madg = m.madg \text{ thì } d.ngaydk \leq m.ngmuon$

cuối $\forall m$

Cuối $\forall d$

□ **Ràng buộc toàn vẹn về thuộc tính tổng hợp (thống kê)**

Ràng buộc toàn vẹn về thuộc tính tổng hợp được xác định trong trường hợp một thuộc tính A trong lược đồ quan hệ Q được tính toán giá trị từ các thuộc tính của các lược đồ quan hệ khác.

6.1.4 Lệnh SQL đối với ràng buộc toàn vẹn

Trong lệnh CREATE TABLE ta có thể tạo ra các ràng buộc toàn vẹn (xem chương 3)

6.2 Phân quyền truy xuất

Việc phân quyền truy xuất là do người quản trị cơ sở dữ liệu quyết định. Nó bao gồm việc tạo ra quyền account cho một người sử dụng (user name và password) và các quyền truy xuất cho người đó:

- Quyền truy vấn trên một bảng.
- Quyền tạo một bảng mới , xoá bảng, thay đổi bảng.
- Quyền chèn, cập nhật và xoá.

Trong SQL phát biểu GRANT và REVOKE được dùng để cấp quyền và tước quyền truy xuất của một user.

Cú pháp:

GRANT <quyền> ON <đối tượng> TO <users> [WITH GRANT OPTION]

Đối tượng thường là bảng.

Mệnh đề WITH GRANT OPTION để cho phép users có quyền cấp quyền truy xuất cho user khác.

REVOKE [GRANT OPTION FOR] <quyền> ON <đối tượng> FROM <users> { RESTRICT | CASCADE }

Mệnh đề GRANT OPTION FOR chỉ tước bỏ quyền cấp quyền truy xuất cho user khác.

Từ khoá CASCADE dùng để xoá quyền của user đó và các users khác mà user đó đã cấp quyền.

Từ khoá RESTRICT dùng để chỉ xoá quyền của user đó mà thôi.

Ví dụ: User U1:

GRANT SELECT ON TABLE S TO U2 WITH GRANT OPTION

User U2:

GRANT SELECT ON TABLE S TO U3 WITH GRANT OPTION

User U3:

GRANT SELECT ON TABLE S TO U4 WITH GRANT OPTION

User U1

REVOKE SELECT ON TABLE TO U2 CASCADE

Lúc đó U2, U3, U4 sẽ bị mất quyền SELECT trên bảng S

User U1

REVOKE SELECT ON TABLE TO U2 RESTRICT

Lúc đó chỉ có U2 mất quyền SELECT

BÀI TẬP TỔNG HỢP - ĐỀ THI MẪU

Bài 1 Thiết kế mô hình dữ liệu quan niệm quản lý cơ sở dữ liệu cho một công ty hàng không dân dụng quốc tế

a/ Thiết kế sơ đồ thực thể mối liên kết cho cơ sở dữ liệu cho một công ty hàng không dân dụng quốc tế :

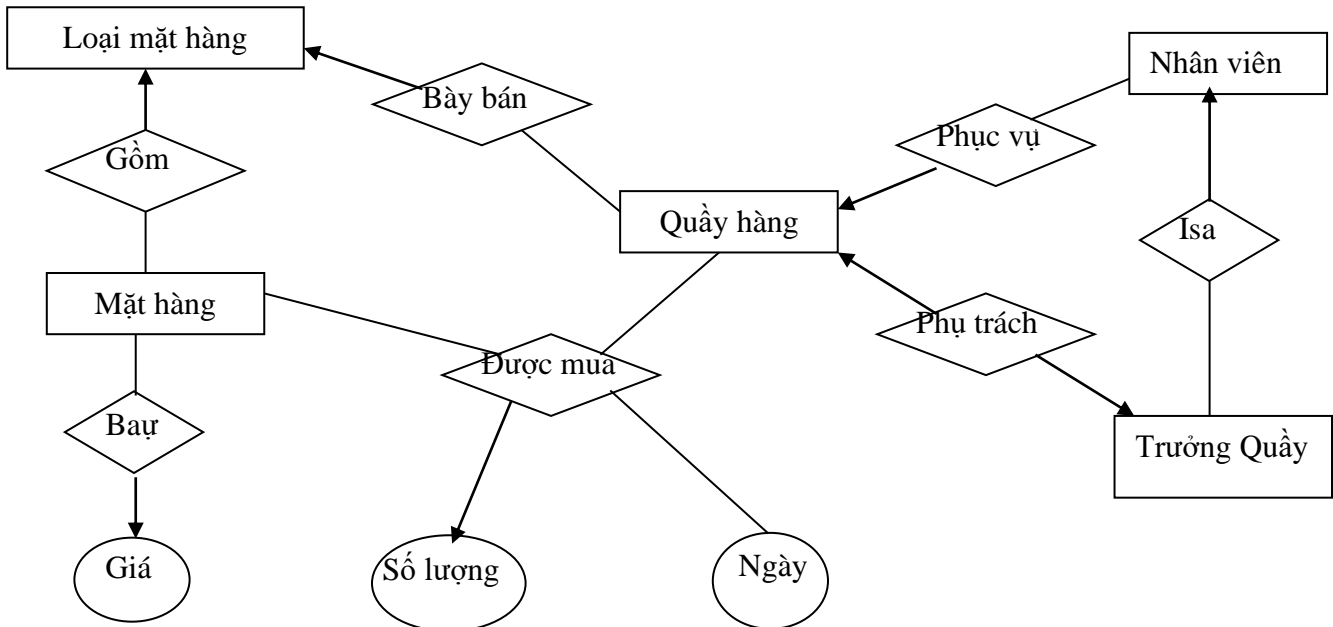
- Công ty có nhiều máy bay. Thông tin về máy bay gồm có mã số máy bay được dùng để phân biệt các máy bay trong công ty, tên loại máy bay, công dụng, sức chứa.
- Hành khách đi các chuyến bay của công ty phải đăng ký mua vé. Mỗi vé chỉ dùng cho một khách hàng.
- Thông tin về hành khách gồm có số giấy thông hành, họ tên hành khách.
- Thông tin của vé gồm có số vé dùng để phân biệt với các vé khác, vị trí chỗ trong máy bay, nước đi, nước đến. Mỗi vé chỉ sử dụng cho một chuyến bay.
- Thông tin về chuyến bay có mã chuyến bay dùng để phân biệt các chuyến bay khác nhau, ngày giờ cất cánh. Một chuyến bay phải khởi hành tại một sân bay xác định và kết thúc cũng ở một sân bay xác định.
- Thông tin về sân bay gồm có tên sân bay, tên nước, vị trí.
- Mỗi hành khách có thể có hàng hóa. Thông tin về hàng hóa gồm có mã số hàng hóa, trọng lượng, kích thước, đặc tính của hàng hóa. Hàng hóa có thể đi theo chuyến bay của khách hàng hoặc trên một chuyến bay khác.
- Lộ trình của một chuyến bay có thể quá cảnh nhiều sân bay để tiếp nhiên liệu nhưng phải đáp xuống sân bay đó trong khoảng thời gian xác định.

b/ Từ sơ đồ thực thể mối liên kết đã thiết kế ở câu a/ hãy xây dựng mô hình logic sao cho đạt dạng chuẩn 3.

c/ Dựa vào cơ sở dữ liệu ở câu b/ hãy trả lời các câu hỏi sau bằng ngôn ngữ SQL.

1. Hãy cho biết danh sách (số giấy thông hành, họ tên) các hành khách đi chuyến bay có mã 'A01' xuất phát ở 'Việt Nam' xuống ở sân bay quá cảnh tại 'Thái Lan'.
2. Hãy cho biết sân bay nào được quá cảnh nhiều nhất.
3. Hãy cho biết chuyến bay nào có lộ trình dài nhất.

Bài 2 Cho mô hình thực thể mối liên kết để quản lý việc bán hàng ở siêu thị sau: Xây dựng các quan hệ dữ liệu (mô hình logic) từ mô hình ER trên, biết:



- Thông tin về mặt hàng gồm có mã mặt hàng, tên hàng, tên nhà sản xuất
- Thông tin về loại mặt hàng gồm có mã loại mặt hàng, tên mặt hàng
- Thông tin về quầy hàng thì có số quầy
- Thông tin về nhân viên có họ tên, địa chỉ, mã nhân viên
- Thông tin về ngày gồm có ngày, tháng, năm.

Cho biết các câu hỏi sau đây đúng sai, giải thích.

- Mỗi quầy hàng bán một mặt hàng với giá khác nhau.
- Một mặt hàng chỉ được bán ở một quầy hàng duy nhất.

Trả lời các câu hỏi sau bằng ngôn ngữ SQL và cất kết quả vào một bảng tạm.

- Cho biết quầy hàng và tên trưởng quầy nào có doanh thu cao nhất trong tháng 8/1999.
- Cho biết loại mặt hàng nào được tiêu thụ ít nhất trong tháng 8/1999.

Bài 3 Thiết kế mô hình dữ liệu quan niệm để quản lý bệnh nhân của một bệnh viện sau

1/ Thiết kế sơ đồ thực thể mối liên kết cho cơ sở dữ liệu để quản lý bệnh nhân của một bệnh viện sau :

- Bệnh viện có nhiều khoa. Thông tin về khoa có mã khoa, tên khoa, số phone, số bệnh nhân tối đa có thể phục vụ.

- Trong bệnh viện có nhiều phòng bệnh. Thông tin về phòng bệnh có số phòng, vị trí, loại phòng, giá phòng (thay đổi theo thời gian).

- Mỗi phòng có nhiều giường bệnh. Các giường bệnh được đánh số riêng biệt và theo thứ tự tăng dần.

- Mỗi khoa phụ trách nhiều phòng bệnh nhưng một phòng bệnh chỉ do một khoa đảm nhiệm.

- Mỗi khoa có nhiều nhân viên nhưng một nhân viên chỉ thuộc một khoa. Thông tin về nhân viên có mã nhân viên, họ tên nhân viên, chức danh (bác sĩ, y tá, hộ lý). Mỗi khoa có một trưởng khoa là bác sĩ của khoa đó.

- Mỗi bệnh nhân nằm ở một giường cố định. Thông tin về bệnh nhân gồm có họ tên bệnh nhân (giả sử có thêm thông tin phụ để phân biệt các bệnh nhân cùng họ tên), địa chỉ, bảo hiểm y tế (có hoặc không), ngày nhập viện, ngày xuất viện. Mỗi bệnh nhân có một hồ sơ bệnh án riêng do một bác sĩ theo dõi. Thông tin về hồ sơ bệnh án gồm có mã hồ sơ, loại bệnh, trọng lượng, kết quả xét nghiệm, cách chữa trị, thuốc sử dụng, chi phí.

- Việc khám/ chữa bệnh được tổ chức theo buổi, trong một buổi khám/ chữa bệnh một bệnh nhân được một bác sĩ – y tá khám/chữa. Ngoài ra trong một buổi có một bác sĩ trực để giải quyết các sự cố có thể xảy ra.

2/ Từ sơ đồ thực thể mối liên kết ở trên hãy xây dựng mô hình logic sao cho đạt dạng chuẩn 3.

3/ Trả lời một số câu hỏi sau bằng SQL

a/ Cho biết bệnh nhân có tên X đang nằm ở khoa, phòng, giường nào? Đang bị bệnh gì? Và do bác sĩ nào trực tiếp theo dõi bệnh án?

b/ Cho biết loại bệnh nào đang được chữa chạy nhiều nhất ?

c/ Cho biết bác sĩ của khoa nào khám bệnh cho nhiều bệnh nhân nhất ?

Bài 4 Thiết kế mô hình dữ liệu quan niệm quản lý sinh viên của một trường đại học

Cho các thông tin quản lý sinh viên của một trường đại học sau:

Thông tin về sinh viên gồm có mã sinh viên dùng để phân biệt các sinh viên, họ tên sinh viên, địa chỉ, ngày sinh, năm bắt đầu vào học.

Thông tin về lớp gồm có mã lớp dùng để phân biệt các lớp và tên lớp. Một sinh viên thì ở một lớp, một lớp thì có tối đa là 50 sinh viên.

Một lớp có một giáo viên chủ nhiệm.

Thông tin về giảng viên gồm mã giảng viên dùng để phân biệt các giảng viên, tên giảng viên, học vị, chuyên môn.

Thông tin về môn học gồm mã môn học dùng để phân biệt các môn học, tên môn, số tiết.

Lớp được học môn học do một giảng viên dạy.

Lớp được học môn học bắt đầu ở một thời điểm (ngày, tháng, năm) nhất định.

Sinh viên khi theo học môn nào cũng sẽ bị điểm danh theo từng buổi học. Nếu sinh viên vắng quá 1/3 buổi học sẽ bị cấm thi. Một buổi học được tính 4 tiết.

Lớp được tổ chức cho thi môn học vào một thời điểm xác định.

Sinh viên thi môn học nào thì có một kết quả. Kết quả cho biết lần thi đó sinh viên được bao nhiêu điểm. Số lần thi cho một môn tối đa là 2 lần.

Câu hỏi

1. Thiết kế sơ đồ ER
2. Xây dựng các quan hệ dữ liệu (mô hình logic) từ sơ đồ ER trên.
3. Cho biết cách tạo ra các quan hệ tạm (bảng tạm) sau bằng ngôn ngữ SQL:
 - Danh sách sinh viên đủ điều kiện dự thi môn học X lần 1.
 - Danh sách sinh viên thi lần 2 môn học X.

Bài 5 Thiết kế mô hình dữ liệu quan niệm quản lý thư viện

Cho thông tin để quản lý thư viện sau:

- Thông tin về sách gồm mã sách dùng để phân biệt các sách, tên sách, khổ sách, số trang, tình trạng sách, giá, số bản sách (là số sách giống như nhau).

- Thông tin về tác giả gồm tên tác giả, địa chỉ.

- Thông tin về nhà xuất bản gồm tên nhà xuất bản, địa chỉ.
- Thông tin về thể loại sách gồm mã thể loại dùng để phân biệt các thể loại, tên thể loại.
- Thông tin về độc giả gồm mã độc giả dùng để phân biệt các độc giả, tên độc giả, địa chỉ.
- Thông tin về ngăn tủ gồm vị trí, số lượng sách có thể chứa vào ngăn đó.
- Các quyển sách cùng bản được cất vào một ngăn tủ.
- Một quyển sách thì có thể có nhiều tác giả và một tác giả có thể viết nhiều sách.
- Một sách thì do một nhà xuất bản in ấn.
- Một sách thuộc một loại thể loại.
- Một độc giả chỉ có thể mượn về nhà hai sách ở một thời điểm. Một sách chỉ cho mượn tối đa 30 ngày và không được mượn quá 5 quyển. Một sách chỉ có thể cho mượn nếu số bản của nó lớn hơn 1.

Câu hỏi

- Thiết kế sơ đồ ER
- Xây dựng các quan hệ dữ liệu (mô hình logic) từ sơ đồ ER trên.
- Cho biết cách tạo ra các quan hệ tạm (bảng tạm) sau bằng ngôn ngữ SQL:
- Danh sách độc giả không đủ điều kiện mượn sách.
- Danh sách tên nhà xuất bản có số lượng sách được mượn theo thứ tự giảm dần.

ĐỀ THI MÔN CƠ SỞ DỮ LIỆU

Lớp C98 A1 + C98 A4

Thời gian 120 phút

Câu 1: Cho một thể hiện của lược đồ quan hệ sau:

R(A	B	C	D	E)
a1	b1	c1	d1	e1
a1	b2	c1	d2	e1
a2	b1	c2	d3	e1
a2	b1	c2	d1	e1
a3	b2	c2	d2	e2

1. Cho biết các phụ thuộc hàm nào sau đây thỏa r? Giải thích bằng thuật toán. A -> B, AB -> C, BC -> D2. Trình bày giải thuật phát hiện khóa dựa trên thể hiện của r. Cho biết các khóa có thể có của r.

Câu 2: Cho một cơ sở dữ liệu quản lý đơn đặt hàng của một đơn vị kinh doanh sau:

1. KHACH_HANG (MKH, DC, DT, TEN_KH)

Một khách hàng có mã số (MKH) dùng để phân biệt với những khách hàng khác, có địa chỉ (DC), số điện thoại (DT), tên khách hàng (TEN_KH).

2. MAT_HANG(MHG, TEN_HG, DONGIA, NGÀY)

Mỗi mặt hàng có mã số (MHG) dùng để phân biệt với những mặt hàng khác và có một tên gọi (TEN_HG) có đơn giá (DONGIA) được thiết lập từ ngày (NGÀY).

3. DDH (MDDH, MKH, SO_HG, NGÀYDH)

Mỗi đơn đặt hàng của một khách hàng có một mã đơn dùng để phân biệt với những đơn đặt hàng khác nhau. Ứng với mỗi đơn đặt hàng ta biết số loại các mặt hàng (SO_HG) được đặt vào ngày đặt hàng(NGÀYDH).

4. CHI_TIET_DD(MDDH, MHG, LG_DAT, MKH)

Mỗi một chi tiết của đơn đặt hàng của một khách hàng liên quan đến một mặt hàng. Tương ứng với mỗi chi tiết ta biết được lượng đặt (LG_DAT).

5. GIAO_GH(MDDH, MKH, MGH, NGÀY)

Mỗi đợt giao hàng liên quan đến một đơn đặt hàng của một khách hàng. Đơn giao hàng có một mã số (MGH) dùng để phân biệt với những đợt giao khác nhau. Ta biết ngày giao (NGÀY) của đợt đó.

6. CHI_TIET_GH(MDDH, MKH, MGH, MHG, LG_GIAO)

Mỗi chi tiết của một đợt giao hàng (MGH) liên quan đến các mặt hàng (MHG) và số lượng giao mặt hàng được cho đơn đặt hàng (MDDH) của khách hàng (MKH).

Câu hỏi

- 1/ Xác định khoá và các phụ thuộc hàm của các quan hệ.
- 2/ Chuẩn hoá lược đồ cơ sở dữ liệu trên đến tối thiểu dạng chuẩn 3.
- 3/ Cho biết các ràng buộc toàn vẹn của cơ sở dữ liệu trên (không xét ràng buộc về khóa và miền giá trị).
- 4/ Trả lời câu hỏi dưới đây bằng SQL hoặc đại số quan hệ
 - Danh sách những mã hàng của đơn đặt hàng số 26 của khách hàng số 12 đã được giao làm ba đợt.

ĐỀ THI MÔN CƠ SỞ DỮ LIỆU

Lớp C98 A2 + C98 A3

Thời gian 120 phút

Câu 1: Cho $F = \{ AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A, D \rightarrow C \}$

1. Chứng minh $BG \rightarrow H$ bằng hệ tiên đề Armstrong

2. Cho biết giải thuật REPUGNANT(G) với G là tập phụ thuộc hàm sau có cho ra được phủ tối thiểu không? Giải thích?

REPUGNANT(G)

begin

$F := \emptyset;$

For mỗi $X \rightarrow Y$ trong G do

if MEMBER(G- $\{X \rightarrow Y\}$, $X \rightarrow Y$) then $F := F \cup \{X \rightarrow Y\}$

return (G- F)

end.

Cho ví dụ minh họa với F theo giải thuật REPUGNANT và so sánh với phủ tối thiểu thật sự.

Câu 2: Cho một cơ sở dữ liệu quản lý đơn đặt hàng của một đơn vị kinh doanh sau:

1. KHACH_HANG(MKH,DC,DT,TEN_KH)

Một khách hàng có mã số (MKH) dùng để phân biệt với những khách hàng khác, có địa chỉ (DC), số điện thoại (DT), tên khách hàng (TEN_KH).

2. MAT_HANG(MHG,TEN_HG, DONGIA, NGAY)

Mỗi mặt hàng có mã số (MHG) dùng để phân biệt với những mặt hàng khác và có một tên gọi (TEN_HG) có đơn giá (DONGIA) được thiết lập từ ngày (NGAY).

3. DDH(MDDH,MKH,SO_HG,NGAYDH)

Mỗi đơn đặt hàng của một khách hàng có một mã đơn dùng để phân biệt với những đơn đặt hàng khác của cùng một khách hàng. Ứng với mỗi đơn đặt hàng ta biết số loại các mặt hàng (SO_HG) được đặt vào ngày đặt hàng(NGAYDH).

4. CHI_TIET_DDH(MDDH, MHG, MKH, LG_DAT)

Mỗi một chi tiết của đơn đặt hàng của một khách hàng liên quan đến một mặt hàng. Tương ứng với mỗi chi tiết ta biết được lượng đặt (LG_DAT).

5. GIAO_GH(MDDH,MKH,MGH,NGAY)

Mỗi đợt giao hàng liên quan đến một đơn đặt hàng của một khách hàng. Đơn giao hàng có một mã số (MGH) dùng để phân biệt với những đợt giao khác nhau. Ta biết ngày giao (NGAY) của đợt đó.

6. CHI_TIET_GH(MDDH,MKH,MGH,MHG,LG_GIAO)

Mỗi chi tiết của một đợt giao hàng (MGH) liên quan đến các mặt hàng (MHG) và số lượng giao mặt hàng được cho đơn đặt hàng (MDDH) của khách hàng (MKH).

Câu hỏi

- 1/ Xác định khoá và các phụ thuộc hàm của các quan hệ.
- 2/ Chuẩn hoá lược đồ cơ sở dữ liệu trên đến tối thiểu dạng chuẩn 3.
- 3/ Cho biết các ràng buộc toàn vẹn của cơ sở dữ liệu trên (không xét ràng buộc về khóa và miền giá trị).
- 4/ Trả lời câu hỏi dưới đây bằng SQL hoặc đại số quan hệ
 - Danh sách những mã hàng của đơn đặt hàng số 26 của khách hàng số 12 đã được giao làm ba đợt.

Đề số 1

ĐỀ THI TỐT NGHIỆP MÔN CƠ SỞ DỮ LIỆU**Lớp Trung Cấp Tin Học Khoá 1**

Câu 1 Chuẩn hóa và ràng buộc toàn vẹn cơ sở dữ liệu

Cho cơ sở dữ liệu quản lý sản xuất của các công nhân trong nhà máy :

1. CONGNHAN(MACN,HOTENCN,DIACHI,SOTO,MAMAY)

MACN được dùng để phân biệt các công nhân. Mỗi công nhân có họ tên

(HOTEN), địa chỉ xác định (DIACHI) và thuộc một tổ (SOTO). Mỗi công nhân được giao phụ trách luôn một máy nhưng một máy được nhiều công nhân phụ trách tùy theo ca làm việc của họ.

2. LUONG(MACN,HOTENCN,LUONG,THANG)

Lương công nhân (MACN) có họ tên (HOTENCN) được tính theo khoán sản phẩm và được trả vào cuối tháng (THANG).

3. TO(SOTO,TOTRUONG,SOTOVIEN).

SOTO cho biết số của tổ được dùng để phân biệt các tổ. Mỗi tổ có một tổ trưởng (TOTRUONG) cũng là công nhân trong tổ. SOTOVIEN cho biết số tổ viên trong tổ.

4. SANXUAT(MACN,MAMAY,MASP,NGAY,CA,SOLUONGSP)

Quan hệ SANXUAT cho biết công nhân (MACN) đứng ở máy (MAMAY) trong ca (CA) làm việc của ngày (NGAY) đã thực hiện sản phẩm (MASP) với số lượng sản phẩm là (SOLUONGSP).

5. DINHMUC(MAMAY,MASP,TENSANPHAM,CHITIEU)

Quan hệ DINHMUC cho biết chỉ tiêu (CHITIEU) về số sản phẩm của sản phẩm có mã (MASP), tên sản phẩm (TENSANPHAM) và của một máy (MAMAY) xác định trong một ngày.

6. SANPHAM(MASP, TENSANPHAM, GIACONGDONVI)

MASP cho biết mã của sản phẩm và có một tên (TENSANPHAM) với giá công của một sản phẩm (GIACONGDONVI)

Câu hỏi

1/Xác định khóa của các quan hệ trên.

2/ Cho biết các quan hệ của cơ sở dữ liệu trên ở dạng chuẩn mấy ? Nếu chưa đạt dạng chuẩn 3 thì chuẩn hóa tiếp để đạt được.

3/ Cho biết các câu hỏi sau đây đúng hay sai , tại sao?- Một công nhân chỉ chuyên làm ra một loại sản phẩm.- Mỗi máy có chỉ tiêu riêng cho từng loại sản phẩm.

4/ Cho biết các ràng buộc toàn vẹn của cơ sở dữ liệu trên.

5/ Trả lời câu hỏi bằng ngôn ngữ SQL hoặc bằng đại số quan hệ

- Cho biết tiền lương của công nhân có tên “Nguyễn Văn Hùng” vào tháng 10/1998.

- Cho biết tiền lương trung bình của tổ có SOTO =2 vào tháng 10/1998.

- Cho biết tổ nào (SOTO) và tên tổ trưởng của tổ đó mà làm ra nhiều sản phẩm nhất trong năm vừa rồi?

- Cho biết máy nào có thể làm ra đủ các sản phẩm?

TÀI LIỆU THAM KHẢO

- | | | |
|--|------------------------|------|
| 1. Database System Concepts
Sử dụng trong các chương 1, 2, 3, 4, 7 | Henry F.Korth | 1991 |
| 2. Giáo trình nhập môn cơ sở dữ liệu
Sử dụng trong các chương 2, 4, 5,6 | Nguyễn An Tế (ĐH KHTN) | 1996 |
| 3. Guide to The SQL Standard
Sử dụng trong các chương 3, 6 | C. J. Date | 1993 |
| 4. Nguyên lý các hệ cơ sở dữ liệu Tập 2
Sử dụng trong chương 4 | J. D. Ullman | 1999 |
| 5. The Theory of relational databases
Sử dụng trong chương 4 | David Maier | 1983 |

MỤC LỤC

CHƯƠNG 1 : ĐẠI CƯƠNG VỀ CÁC HỆ CƠ SỞ DỮ LIỆU	1
1.1 Khái niệm về cơ sở dữ liệu (Database).....	1
1.1.1 Khái niệm về cơ sở dữ liệu	1
1.1.2 Mục đích của các hệ cơ sở dữ liệu	1
1.2 Kiến trúc của một hệ thống cơ sở dữ liệu	3
1.3 Hệ quản trị cơ sở dữ liệu (Database Management System : DBMS)	5
1.3.1 Khái niệm	5
1.3.2 Các chức năng của hệ quản trị cơ sở dữ liệu.....	5
1.3.3 Khái niệm về sự độc lập dữ liệu và chương trình	7
1.3.4 Bộ quản lý cơ sở dữ liệu	8
1.3.5 Người quản trị cơ sở dữ liệu	8
1.3.6 Những người sử dụng cơ sở dữ liệu.....	8
1.3.7 Cấu trúc tổng quát của hệ thống quản trị cơ sở dữ liệu	9
CHƯƠNG 2 CÁC MÔ HÌNH DỮ LIỆU	11
2.1 Mô hình dữ liệu (data model)	11
2.1.1 Các mô hình dữ liệu (Data models)	11
2.1.2 Các mô hình logic dựa trên đối tượng (Object-Based logical models)	11
2.1.3 Các mô hình dựa trên mẫu tin (Record-Based Logical Models)	13
2.2 Mô hình thực thể mối liên kết (Entity Relationship Model)	15
2.2.1 Thực thể (Entity)	15
2.2.2 Tập thực thể (Entity set) :.....	15
2.2.3 Các thuộc tính và các khoá :	15
2.2.4 Hệ phân cấp isa hay sự tổng quát hoá - chuyên biệt hóa	16
2.2.5 Mối liên kết	16
2.2.6 Sơ đồ thực thể mối liên kết	17
2.2.7 Tính hàm của mối liên kết.....	20
2.2.8 Ví dụ về sơ đồ thực thể mối liên kết (ERD)	22
2.3 Mô hình dữ liệu quan hệ	23
2.3.1 Các khái niệm cơ bản.....	23
2.3.2 Sự chuyển hóa từ sơ đồ ER sang mô hình dữ liệu quan hệ.	25

2.3.3 Các phép toán đại số trong mô hình dữ liệu quan hệ.....	29
2.2.4 Thiết kế cơ sở dữ liệu quan hệ	34
CHƯƠNG 3 NGÔN NGỮ SQL	39
3.1 Giới thiệu :	39
3.2 Ngôn ngữ định nghĩa dữ liệu (Data Definition Language DLL).....	39
3.2.1 Tạo cấu trúc bảng.....	39
3.2.2 Thay đổi cấu trúc bảng.....	40
3.2.3 Hủy bỏ một bảng.....	40
3.2.4. Tạo khung nhìn (view).....	40
3.2.5 Xóa khung nhìn	41
3.3 Ngôn ngữ thao tác dữ liệu (Data Manipulation Language DML)	41
3.3.1 Thêm bộ mới vào quan hệ.....	41
3.3.2 Cập nhật:	41
3.3.3 Hủy bỏ các bộ:	42
3.4 Nhúng ngôn ngữ SQL trong ngôn ngữ chủ	Error! Bookmark not defined.
3.5 Ngôn ngữ điều khiển dữ liệu (Data Control Language)	48
CHƯƠNG 4 PHỤ THUỘC HÀM	51
4.1 Phụ Thuộc hàm (Functional Dependency)	51
4.1.1 Định nghĩa phụ thuộc hàm (FD)	51
4.1.2 Định nghĩa phụ thuộc hàm đầy đủ (Full Functional Dependency "FFD")	51
4.1.3 Mệnh đề.....	51
4.1.4 Giải thuật kiểm tra phụ thuộc hàm.....	51
4.1.5 Tập phụ thuộc hàm.....	52
4.1.6 Biểu diễn tập phụ thuộc hàm bằng đồ thị có hướng.	Error! Bookmark not defined.
4.2 Hệ luật dẫn ARMSTRONG.....	52
4.3 Bao Đóng.	53
4.3.1 Định nghĩa phụ thuộc hàm suy dẫn.....	53
4.3. 2 Mệnh đề.....	53
4.3.3 Bao đóng của tập thuộc tính.....	53
4.3.4 Bài toán thành viên :	54
4.4 Khoá của quan hệ.....	54

4.4.1 Xác định khoá	54
4.4.2 Xác định tất cả khoá.....	55
4.5 Phủ tối thiểu của tập phụ thuộc hàm.....	55
4.5.1 Định nghĩa: Hai tập phụ thuộc hàm F và G.	55
4.5.2 Định Nghĩa: Phủ tối thiểu.	56
4.5.3 Giải thuật tìm phủ tối thiểu.	56
4.6. Phép tách các lược đồ quan hệ.....	57
4.6.1 Định nghĩa.....	57
4.6.2 Phân rã bảo toàn thông tin.....	57
4.6.3 Phân rã bảo toàn phụ thuộc	58
4.6.4 Tính các phụ thuộc hàm được bao trong một lược đồ con của ρ	60
CHƯƠNG 5 CHUẨN HÓA CƠ SỞ DỮ LIỆU	61
5.1 Các dạng chuẩn.....	61
5.2 Dạng chuẩn 1 : 1NF (Normal Form)	61
5.3 Dạng chuẩn 2NF	62
5.4. Dạng chuẩn 3NF	63
5.4.1 Định nghĩa.....	63
5.4.2 Giải thuật phân rã bảo toàn phụ thuộc hàm thành dạng chuẩn 3	64
5.4.3 Phân rã thành dạng chuẩn 3 bảo toàn phụ thuộc hàm và bảo toàn thông tin	65
5.5. Dạng chuẩn 3 : BCNF (Boyce Codd Normal Form)	65
5.5.1 Định nghĩa.....	65
5.5.2 Giải thuật phân rã bảo toàn thông tin thành dạng chuẩn BCNF	65
5.6 Dạng chuẩn 4NF :.....	67
5.6.1 Phụ thuộc đa trị	67
5.6.2 Dạng chuẩn 4NF	68
CHƯƠNG 6 TÍNH TOÀN VỆN DỮ LIỆU	69
6.1 Ràng buộc toàn vẹn (Integrity Constraints).....	69
6.1.1 Định nghĩa :.....	69
6.1.2 Các yếu tố của ràng buộc toàn vẹn :	69
6.1.3 Phân loại ràng buộc toàn vẹn :	71
6.1.4 Lệnh SQL đối với ràng buộc toàn vẹn.....	73
6.2 Phân quyền truy xuất	74

CHƯƠNG 7 PHỤC HỒI DỮ LIỆU VÀ TÍNH NGUYÊN TỐ	Error! Bookmark not defined.
7.1 Phân loại hỏng hóc.....	Error! Bookmark not defined.
7.1.1 Các loại lưu trữ	Error! Bookmark not defined.
7.1.2 Các kiểu hỏng hóc	Error! Bookmark not defined.
7.2 Sự phân cấp lưu trữ.....	Error! Bookmark not defined.
7.3 Mô hình giao tác.	Error! Bookmark not defined.
7.3.1 Ví dụ.....	Error! Bookmark not defined.
7.3.2 Tính đúng đắn và tính nguyên tố.	Error! Bookmark not defined.
7.3.3 Các trạng thái của giao tác.	Error! Bookmark not defined.
7.4 Phục hồi dựa vào nhật ký (Log - based recovery)	Error! Bookmark not defined.
7.4.1 Nhật ký cơ sở dữ liệu	Error! Bookmark not defined.
7.4.2 Kỹ thuật thay đổi CSDL được hoãn lại (Deferred database modification)	Error! Bookmark not defined.
7.4.3 Kỹ thuật thay đổi cơ sở dữ liệu tức khắc	Error! Bookmark not defined.
7.4.4 Điểm kiểm tra (check point)	Error! Bookmark not defined.
BÀI TẬP TỔNG HỢP - ĐỀ THI MẪU	75
TÀI LIỆU THAM KHẢO	86
MỤC LỤC	87