

Chương 1: Khái niệm cơ bản về cơ sở dữ liệu

I. Khái niệm về CSDL

- CSDL được coi là một tập dữ liệu gắn kết logic với nhau. Các dữ liệu ngẫu nhiên không được coi là một CSDL (mặc dù chúng là những ngoại lệ).
- Một CSDL được thiết kế, xây dựng và sử dụng cho một số mục đích cụ thể. Nó được sử dụng bởi một tập người dùng và ứng dụng cụ thể ngay từ khi mới thiết kế.

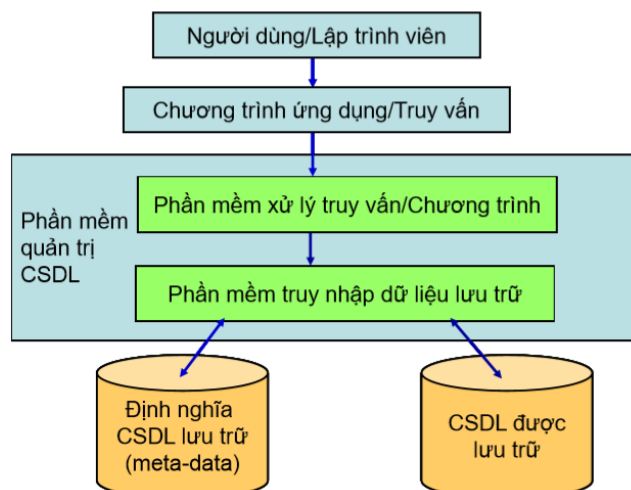
II. Hệ quản trị CSDL

- Hệ quản trị CSDL là một hệ thống phần mềm cho phép tạo lập CSDL và điều khiển mọi truy nhập đến CSDL đó.
- Các đặc điểm (chức năng) quan trọng của một hệ quản trị CSDL:
 - Cho phép người dùng tạo mới CSDL, thông qua ngôn ngữ định nghĩa dữ liệu (DDLs – Data Definition Languages).
 - Cho phép người dùng truy vấn cơ sở dữ liệu, thông qua ngôn ngữ thao tác dữ liệu (DMLs – Data Manipulation Languages).
 - Hỗ trợ lưu trữ số lượng lớn dữ liệu, thường lên tới hàng GB hoặc nhiều hơn trong thời gian dài. Duy trì tính bảo mật và tính toàn vẹn trong quá trình xử lý.
 - Kiểm soát truy nhập dữ liệu từ nhiều người dùng tại cùng một thời điểm.

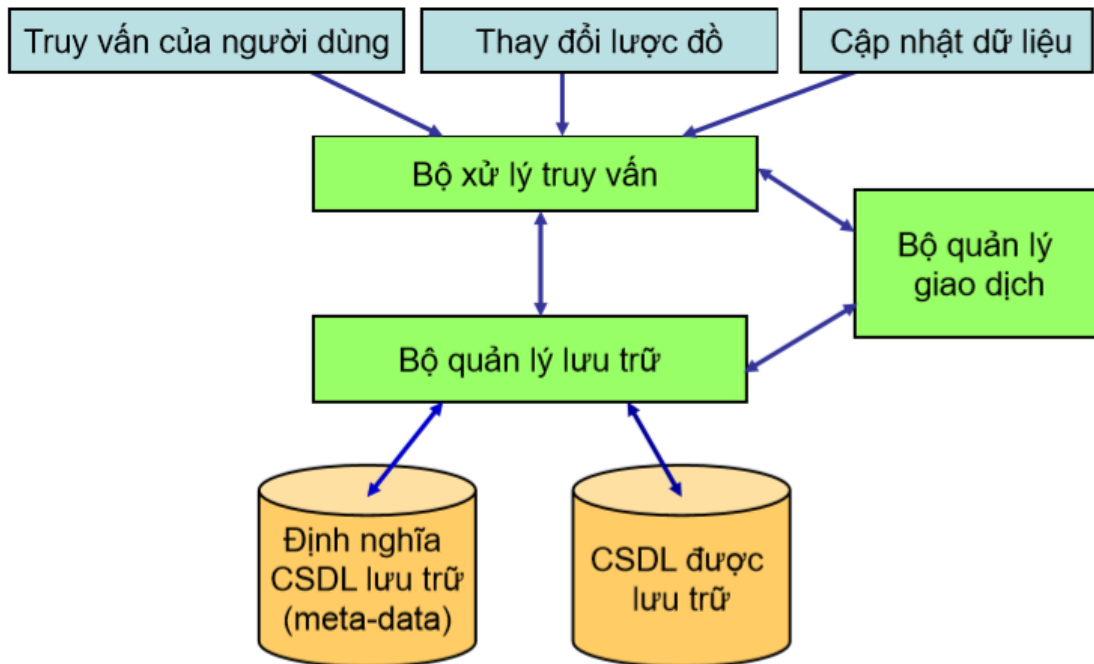
III. Hệ cơ sở dữ liệu

- Một CSDL được quản lý bởi một hệ quản trị CSDL thường được gọi là một hệ cơ sở dữ liệu.
- Khi làm việc với Hệ CSDL, gồm 4 thành phần:
 - CSDL hợp nhất: có 2 tính chất là tối thiểu hóa dư thừa và được chia sẻ.
 - Người dùng: là những người có nhu cầu truy nhập vào CSDL (người dùng cuối, người viết chương trình ứng dụng, người quản trị CSDL).
 - Phần mềm hệ quản trị CSDL.
 - Phần cứng: gồm các thiết bị nhớ thứ cấp được sử dụng để lưu trữ CSDL

IV. Các thành phần của 1 hệ quản trị CSDL



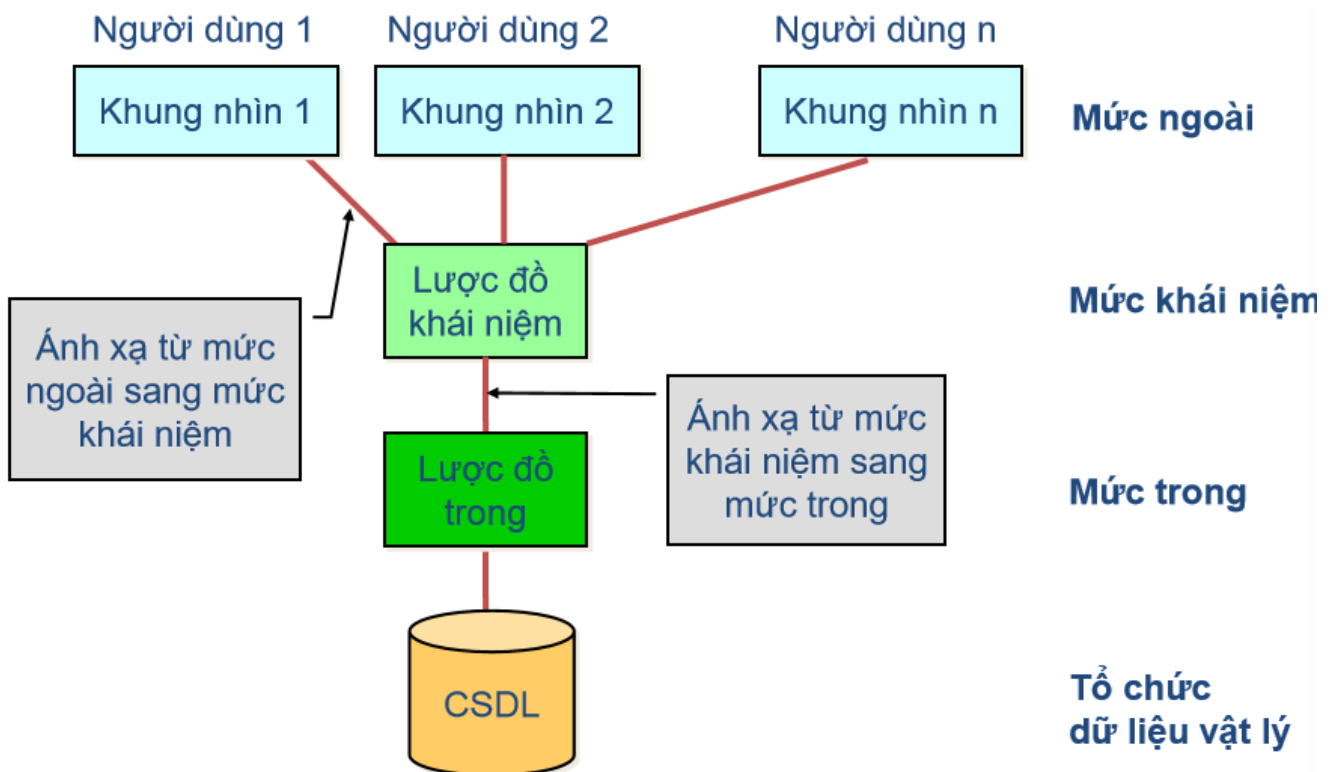
V. Kiến trúc của 1 hệ quản trị CSDL



- CSDL lưu trữ và meta-data:
 - CSDL được lưu trữ tại thiết bị nhớ thứ cấp hoặc cấp 3.
 - Meta-data (siêu dữ liệu) là dữ liệu về dữ liệu: Mô tả các thành phần dữ liệu của CSDL (vị trí tương đối của các trường trong bản ghi, thông tin về lược đồ, thông tin về chỉ mục, ...).
 - Với mỗi CSDL, hệ quản trị CSDL có thể duy trì nhiều chỉ mục khác nhau được thiết kế để cung cấp truy nhập nhanh tới dữ liệu ngẫu nhiên.
 - Trong các CSDL hiện đại, hầu hết các chỉ mục được biểu diễn dưới dạng B-tree (cây tìm kiếm nhị phân). Các B-tree có xu hướng “ngắn và béo” giúp truy nhập nhanh từ gốc đến lá.
- Bộ quản lý lưu trữ: Trong các hệ CSDL đơn giản, bộ quản lý lưu trữ chỉ như là hệ thống tệp trong hệ điều hành. Với các hệ thống lớn hơn, để hiệu quả, hệ quản trị CSDL thường quản lý việc lưu trữ trực tiếp trên ổ đĩa.
Bộ quản lý lưu trữ có 2 thành phần cơ bản:
 - Bộ quản lý tệp: Lưu vị trí các tệp trên ổ đĩa và lấy ra được khối hoặc các khối chứa tệp theo yêu cầu từ bộ quản lý vùng đệm.
 - Bộ quản lý vùng đệm: Quản lý bộ nhớ chính. Lấy các khối dữ liệu từ ổ đĩa, qua bộ quản lý tệp, và chọn một trang trong bộ nhớ chính để lưu trữ. Thuật toán tạo trang sẽ xác định trang sẽ tồn tại bao lâu trong bộ nhớ chính.
- Bộ xử lý truy vấn: Biến đổi một câu truy vấn hoặc một thao tác CSDL, đang được biểu diễn tại một mức rất cao (ví dụ, ngôn ngữ SQL), thành một chuỗi các yêu cầu đối với dữ liệu được lưu trữ trong CSDL.
Phản phức tạp nhất của bộ xử lý truy vấn là tối ưu hóa truy vấn, nghĩa là chọn ra được chiến lược tốt nhất để thực thi truy vấn.

- Bộ quản lý giao dịch: Giao dịch là một tập các thao tác được xử lý như một đơn vị không chia cắt được. Để đảm bảo được tính chất này, bộ quản lý giao dịch phải đảm bảo 4 tính chất (được gọi là thuộc tính ACID):
 - Tính nguyên tử (Atomicity): tất cả các thao tác của giao dịch được thực hiện hoặc không thao tác nào được thực hiện.
 - Tính nhất quán (Consistency): các thao tác phải đảm bảo tính nhất quán của CSDL.
 - Tính biệt lập (Isolation): các giao dịch đồng thời phải được tách riêng biệt nhau.
 - Tính duy trì (Durability): những thay đổi tới CSDL bởi một giao dịch sẽ không bị mất đi ngay cả khi hệ thống có lỗi ngay sau khi giao dịch hoàn thành.
- Ba kiểu thao tác:
 - Truy vấn của người dùng: là các thao tác hỏi đáp về dữ liệu được lưu trữ trong CSDL. Chúng được sinh ra theo 2 cách: (1) Thông qua giao diện truy vấn chung, (2) Thông qua giao diện chương trình ứng dụng.
 - Cập nhật dữ liệu: là các thao tác thay đổi dữ liệu, như thêm, sửa, xóa dữ liệu trong CSDL. Chúng cũng được sinh ra theo 2 cách (1) và (2) như trên.
 - Thay đổi lược đồ: là các lệnh được sinh ra bởi người dùng được cấp phép, thường là người quản trị CSDL.

VI. Mô hình trừu tượng 3 lớp



Chương 2: Các mô hình dữ liệu

I. Giới thiệu

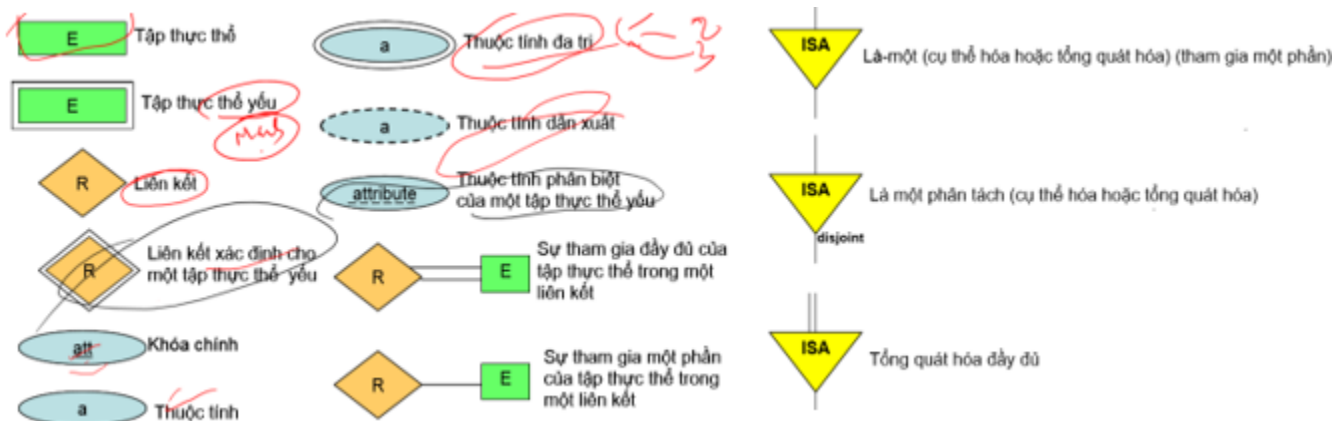
- Mô hình dữ liệu là một tập hợp các khái niệm dùng cho việc mô tả và thao tác dữ liệu, các mối quan hệ và các ràng buộc trên dữ liệu của tổ chức.
- Một mô hình dữ liệu có thể gồm 3 thành phần:
 - Phần cấu trúc gồm tập các luật mà theo đó CSDL được xây dựng.
 - Phần thao tác, định nghĩa các thao tác được phép trên dữ liệu (gồm cả các thao tác cập nhật, lấy dữ liệu từ CSDL, và thay đổi cấu trúc của CSDL).
 - Có thể có một tập các luật về tính toán vụn nhằm đảm bảo CSDL luôn chính xác.
- Dựa vào mô hình kiến trúc 3 lớp, có thể xác định 3 loại mô hình dữ liệu khác nhau:
 - Mô hình dữ liệu ngoài biểu diễn từng khung nhìn của người dùng trong tổ chức.
 - Mô hình dữ liệu khái niệm biểu diễn khung nhìn mức logic, độc lập với hệ quản trị CSDL.
 - Mô hình dữ liệu trong biểu diễn lược đồ khái niệm theo cách mà hệ quản trị CSDL hiểu được.

II. Quá trình thiết kế một CSDL

Gồm 6 bước:

- Bước 1: Phân tích yêu cầu.
- Bước 2: Thiết kế CSDL mức khái niệm: Những thông tin có được từ bước phân tích yêu cầu sẽ được dùng để phát triển một mô tả mức tổng quát dữ liệu được lưu trong CSDL, cùng với các ràng buộc cần thiết trên dữ liệu này.
- Bước 3: Thiết kế CSDL mức logic: Một hệ quản trị CSDL sẽ được chọn để cài đặt CSDL và chuyển thiết kế CSDL mức khái niệm thành một lược đồ CSDL với mô hình dữ liệu của hệ quản trị CSDL đã chọn.
- Bước 4: Cải tiến lược đồ: Các lược đồ được phát triển sẽ được phân tích các vấn đề tiềm ẩn. Tại đây, CSDL sẽ được chuẩn hóa, dựa trên lý thuyết toán học.
- Bước 5: Thiết kế CSDL mức vật lý: Khối lượng công việc tiềm ẩn và các phương pháp truy nhập được mô phỏng để xác định các điểm yếu tiềm ẩn trong CSDL mức khái niệm. Quá trình này thường là nguyên nhân tạo ra các tệp chỉ mục hoặc/và các quan hệ nhóm. Trong trường hợp đặc biệt, toàn bộ mô hình khái niệm sẽ được xây dựng lại.
- Bước 6: Thiết kế an toàn bảo mật: Xác định các nhóm người dùng và phân tích vai trò của họ để định nghĩa các phương pháp truy nhập dữ liệu.
Trong quá trình phát triển, thường có bước cuối cùng (bước thứ 7), gọi là pha điều chỉnh (tuning phase), trong đó CSDL sẽ được thực hiện (mặc dù nó có thể chỉ được chạy mô phỏng) và sẽ được cải tiến, chỉnh sửa để đáp ứng nhu cầu thực thi trong môi trường mong đợi.

III. Lược đồ thực thể liên kết E-R



1. Thuộc tính

- Thuộc tính đơn hoặc thuộc tính kép: Thuộc tính đơn không bao gồm các thành phần cấu thành, trong khi thuộc tính kép bao gồm các thành phần con cấu thành.
- Thuộc tính đơn trị hoặc thuộc tính đa trị: Thuộc tính đơn trị có nhiều nhất một giá trị tại một thời điểm cụ thể. Thuộc tính đa trị có thể có nhiều giá trị khác nhau tại một thời điểm.
- Thuộc tính dẫn xuất: là thuộc tính mà giá trị của nó được dẫn xuất (hoặc được tính toán) từ những giá trị của các thuộc tính hoặc các thực thể có liên quan.
- Thuộc tính rỗng (Null): thuộc tính nhận giá trị rỗng khi một thực thể không có giá trị cho nó.

2. Liên kết

Một cách hình thức, một tập quan hệ là một tập các liên hệ cùng loại. Nó là một quan hệ toán học của n tập thực thể (có thể giao nhau, $n \geq 2$):

Nếu E_1, E_2, \dots, E_n là các tập thực thể, thì một tập quan hệ R là một tập con của:

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, \dots, e_n \in E_n\}$$

Với (e_1, \dots, e_n) là 1 quan hệ.

3. Khoá của 1 tập thực thể

- Khóa (key) cho phép xác định một tập các thuộc tính đủ để phân biệt các thực thể với nhau. Khóa cũng giúp cho việc xác định duy nhất các mối quan hệ, và vì vậy, phân biệt các mối quan hệ với nhau.
- Siêu khóa (superkey) là một tập gồm một hoặc nhiều thuộc tính được lựa chọn cho phép xác định duy nhất một thực thể trong tập thực thể.
 - ⇒ Khái niệm siêu khóa là một định nghĩa không đầy đủ của khóa vì siêu khóa còn chứa nhiều thuộc tính dư thừa.
- Khóa dự bị (candidate key) là những siêu khóa mà không có tập con nào của nó là siêu khóa.
 - ⇒ Với mỗi một tập thực thể E cho trước, tồn tại một hoặc nhiều khóa dự bị.

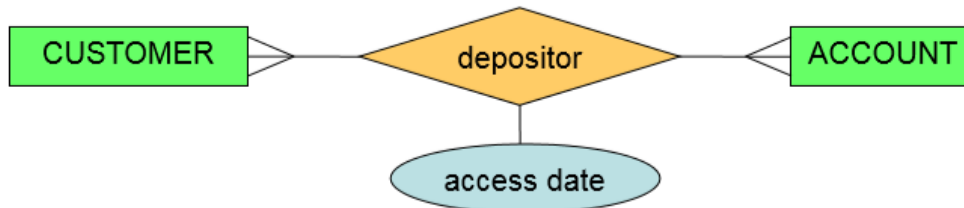
⇒ Người thiết kế CSDL chỉ chọn một khóa dự bị làm khóa chính, hay còn gọi là khóa của tập thực thể.

IV. Một số vấn đề cần quan tâm khi thiết kế lược đồ E-R

1. Ảnh hưởng của ánh xạ lược lượng liên kết lên các khóa

Khoá chính của quan hệ sẽ là hợp các khoá chính của tập thực thể bên nhiều (N:N thì là khoá chính của cả 2 thực thể; 1:N thì chỉ bên N;...)

VD:



⇒ Lược đồ này thể hiện một quan hệ M-M cho quan hệ **depositor** với thuộc tính **access-date** liên quan tới tập quan hệ giữa hai thực thể **CUSTOMER** và **ACCOUNT**.

Khoá chính của quan hệ này sẽ bao gồm hợp của các khóa chính của hai tập thực thể **CUSTOMER** và **ACCOUNT**.

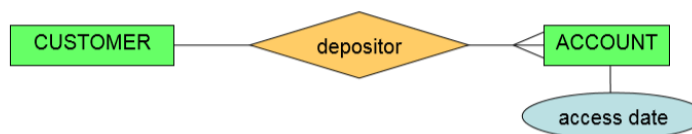
2. Đặt vị trí cho các mối quan hệ

Các thuộc tính của 1 tập quan hệ:

- Luôn có thể thuộc luôn quan hệ.
- Thuộc vào các tập thực thể tham gia liên kết (bên nhiều trong 1:N, bên nào cũng được trong 1:1).
- Bắt buộc thuộc quan hệ (N:N).

VD:

- Thuộc tính **access-date** có thể được đặt liên quan tới thực thể **ACCOUNT** mà không làm tổn thất thông tin:



3. Tập thực thể hay các thuộc tính

Đẩy 1 thuộc tính ra làm thực thể (kiểu số điện thoại đa trị).

4. Tập thực thể hay tập quan hệ

Đẩy quan hệ ra làm thực thể.

5. Tập thực thể mạnh hay yếu

- Tập thực thể không đủ các thuộc tính để hình thành một khóa chính gọi là tập thực thể yếu. Tập thực thể có khóa chính được gọi là tập thực thể mạnh.

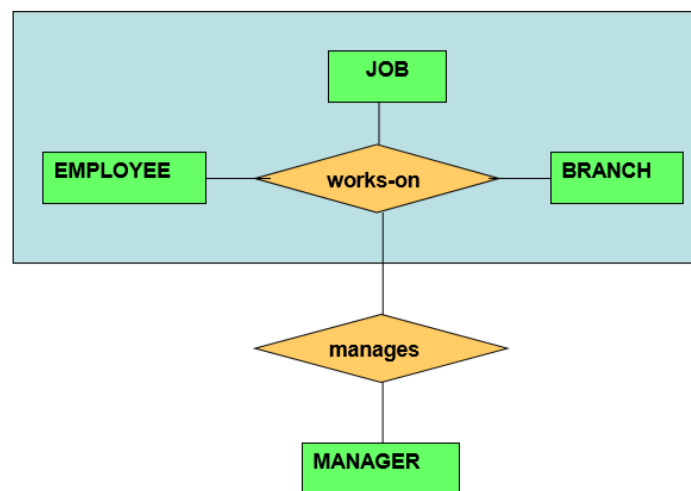
- Để một tập thực thể yếu có ý nghĩa, nó phải liên hệ với một tập thực thể khác, được gọi là tập thực thể xác định hay tập thực thể sở hữu.
- Mỗi quan hệ giữa tập thực thể yếu với tập thực thể xác định được gọi là mối quan hệ xác định.
- Mỗi quan hệ xác định là quan hệ M:1 từ tập thực thể yếu tới tập xác định và sự tham gia của tập thực thể yếu trong quan hệ là đầy đủ.
- Mặc dù tập thực thể yếu không có khóa chính, nhưng có một phương thức để phân biệt tất cả các thực thể trong tập thực thể yếu phụ thuộc vào một thực thể mạnh cụ thể.
=> Tập các thuộc tính của một thực thể yếu cho phép phân biệt các thực thể được gọi là thuộc tính phân biệt (hay khóa bán phần).
- Khóa chính của một tập thực thể yếu được cấu thành bởi khóa chính của tập thực thể xác định và thuộc tính phân biệt của tập thực thể yếu.

6. Cụ thể hoá và tổng quát hoá

- Cụ thể hóa xuất phát từ tập thực thể riêng biệt; nhấn mạnh sự khác nhau giữa các thực thể trong cùng tập bằng cách tạo ra các tập thực thể phân biệt nhau ở mức thấp hơn.
- Tổng quát hóa xuất phát từ việc nhận ra một số tập thực thể có chung một số các đặc tính. Trên cơ sở đó, tổng quát hóa tổng hợp các tập thực thể này thành một tập thực thể ở mức cao hơn. Quá trình tổng quát hóa nhấn mạnh tính tương đồng giữa các tập thực thể ở mức thấp hơn và giấu đi những khác biệt.
- Kế thừa thuộc tính: Các tập thực thể ở mức thấp hơn kế thừa các thuộc tính từ tập thực thể ở mức cao hơn.

7. Kết hợp

- Kết hợp là một sự trừu tượng thông qua việc coi các mối quan hệ như là các thực thể ở mức cao.



Sự kết hợp trong mô hình E-R

V. Mô hình dữ liệu quan hệ

1. Các khái niệm cơ bản

- Quan hệ (Bảng/ Tập): là một bảng (ma trận) với các hàng và các cột, lưu giữ thông tin về các đối tượng được mô hình hóa trong CSDL.
- Thuộc tính (Cột/ Trường): là các cột được đặt tên trong một quan hệ. Mỗi thuộc tính là một đặc tính của một thực thể (hay một quan hệ) được mô hình hóa trong CSDL. Các thuộc tính có thể xuất hiện theo bất kỳ thứ tự nào trong quan hệ.
- Miền giá trị: là một tập các giá trị có thể có của một hoặc nhiều thuộc tính. Mỗi thuộc tính được xác định trên một miền giá trị.
- Bộ (Hàng/ Bản ghi): là một hàng của một quan hệ. Các bộ có thể xuất hiện theo bất kỳ thứ tự nào trong quan hệ.
- Bậc (cấp): của một quan hệ là số lượng các thuộc tính mà nó có.
- Lực lượng: là số lượng các bộ mà một quan hệ có.
- Cơ sở dữ liệu quan hệ: là một tập hợp các quan hệ được chuẩn hóa với các tên phân biệt nhau.

2. Lược đồ quan hệ và thể hiện quan hệ

- Lược đồ quan hệ bao gồm tên và các thuộc tính cho quan hệ và thường không thay đổi. Một lược đồ cơ sở dữ liệu quan hệ là một tập các lược đồ quan hệ, mỗi lược đồ có một tên riêng $R = \{R1, R2, ..., Rn\}$.
- Một thể hiện của quan hệ là một tập các bộ của quan hệ và có thể thay đổi thường xuyên.
Hầu hết các quá trình cập nhật, chèn thêm hay xóa các bộ sẽ làm thay đổi thể hiện của quan hệ.
Một CSDL hiện thời (snapshot database) thể hiện trạng thái hiện tại của thể giới thực tại một thời điểm. Nếu thể giới thực thay đổi, CSDL cũng sẽ thay đổi theo để duy trì biểu diễn đó.

VI. Ánh xạ mô hình E-R sang mô hình quan hệ

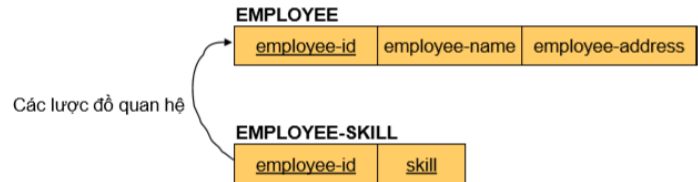
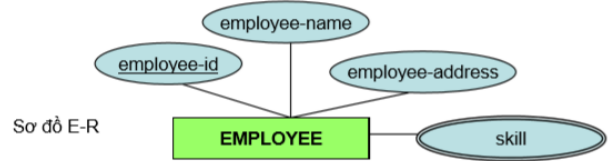
1. Thiết kế logic

- Thiết kế logic là quá trình chuyển đổi thiết kế mức khái niệm thành các lược đồ CSDL quan hệ. Đầu vào là các sơ đồ E-R và đầu ra là các lược đồ quan hệ.
- Cần nhớ 3 loại thực thể đã định nghĩa sau:
 - Thực thể thường (khỏe/ mạnh).
 - Thực thể yếu.
 - Thực thể kết hợp: hình thành từ những mối quan hệ nhiều-nhiều giữa các loại thực thể khác nhau.

2. Ánh xạ thực thể mạnh

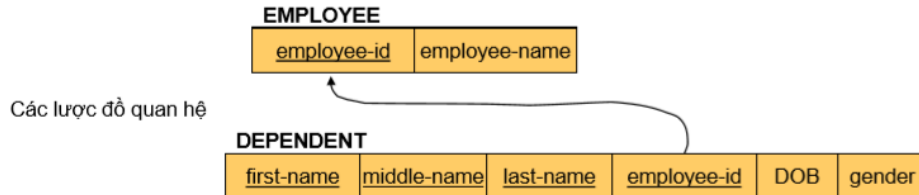
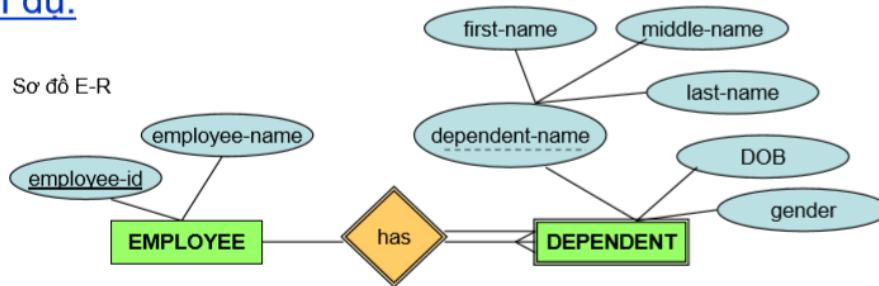
- Thuộc tính kép: chỉ những thuộc tính đơn của thuộc tính kép này được đưa vào lược đồ quan hệ.

- Thuộc tính đa trị: Nếu một thực thể thường có một thuộc tính đa trị thì hai lược đồ quan hệ mới sẽ được tạo ra.
 - Lược đồ quan hệ thứ nhất chứa tất cả các thuộc tính của thực thể trừ thuộc tính đa trị.
 - Lược đồ quan hệ thứ hai sẽ có hai thuộc tính cấu thành khóa chính: khóa chính của lược đồ quan hệ thứ nhất (khóa ngoại trong lược đồ thứ hai) và thuộc tính đa trị.



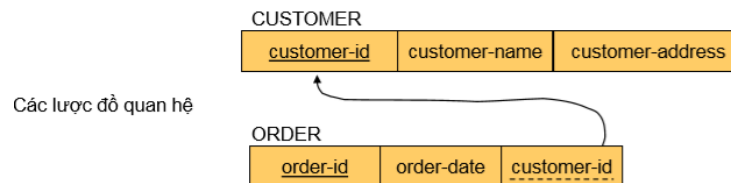
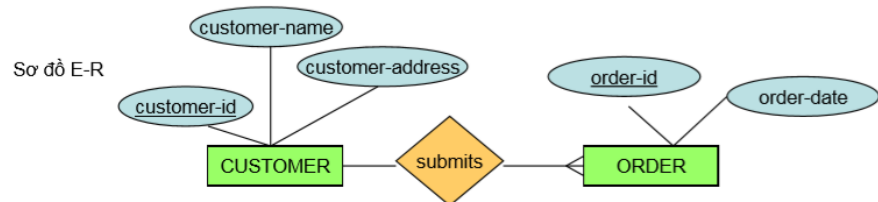
3. Ảnh xạ thực thể yếu

Ví dụ:

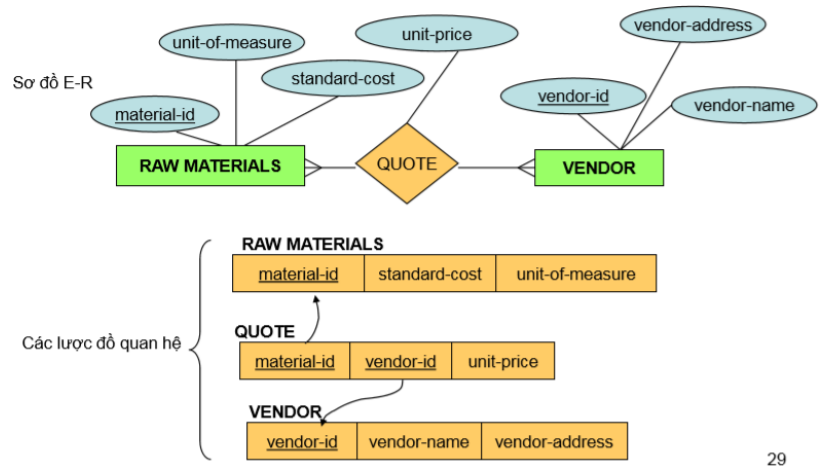


4. Ảnh xạ các quan hệ 2 ngôi

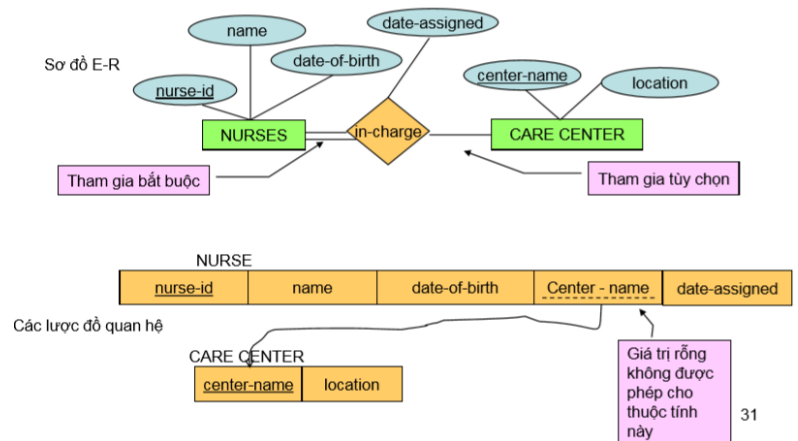
- Quan hệ 1-N: các thuộc tính khóa chính của thực thể bên phía 1 của mỗi quan hệ thành khóa ngoại cho quan hệ nằm ở bên phía N của mỗi quan hệ.



- Quan hệ N-N: Khóa của lược đồ mới C là sự kết hợp khóa chính của các tập thực thể tham gia vào quan hệ và các khóa chính này cũng là khóa ngoại của C. Các thuộc tính không phải là khóa mà liên quan tới quan hệ N-N giữa A và B cũng được đưa vào C.



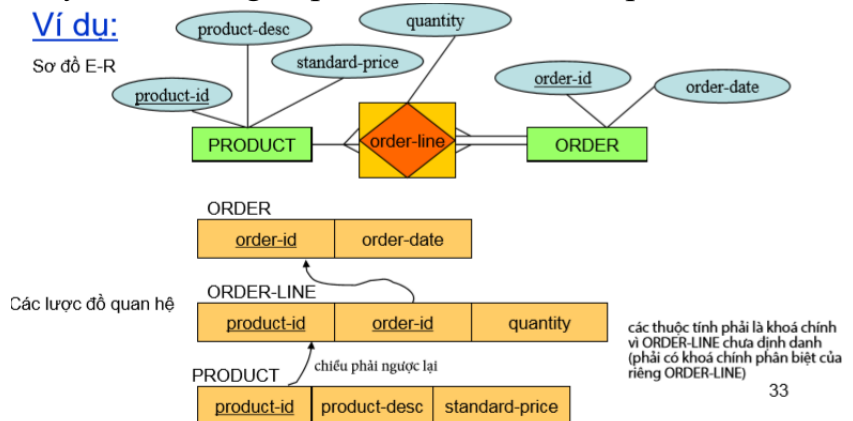
- Quan hệ 1-1: Khóa chính của một quan hệ sẽ thành khóa ngoại trong quan hệ còn lại. Trong liên kết 1-1, việc tham gia vào liên kết trong một bên thường là tùy chọn và bên kia là bắt buộc => Nên thêm vào quan hệ của bên tham gia bắt buộc khóa ngoại của tập thực thể tham gia tùy chọn nhằm tránh việc lưu trữ các giá trị rỗng cho thuộc tính khóa ngoại.



5. Ảnh xạ các thực thể kết hợp

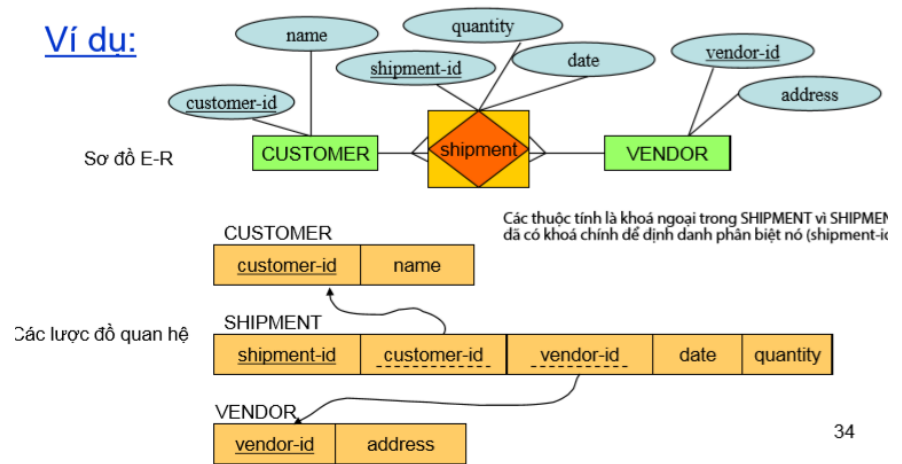
Tương tự trường hợp N-N, chú ý có 2 trường hợp cho thực thể kết hợp:

- Không gán định danh: khóa chính ngầm định cho quan hệ kết hợp gồm các thuộc tính khóa chính từ hai quan hệ còn lại. Các thuộc tính này sẽ là khóa ngoại tham chiếu tới hai quan hệ đó.



- Gán định danh: tạo một quan hệ kết hợp mới thể hiện thực thể kết hợp, với khóa chính là định danh được gán trên sơ đồ E-R. Khóa chính của 2 thực thể tham gia sẽ được thêm vào làm khóa ngoại trong quan hệ kết hợp.

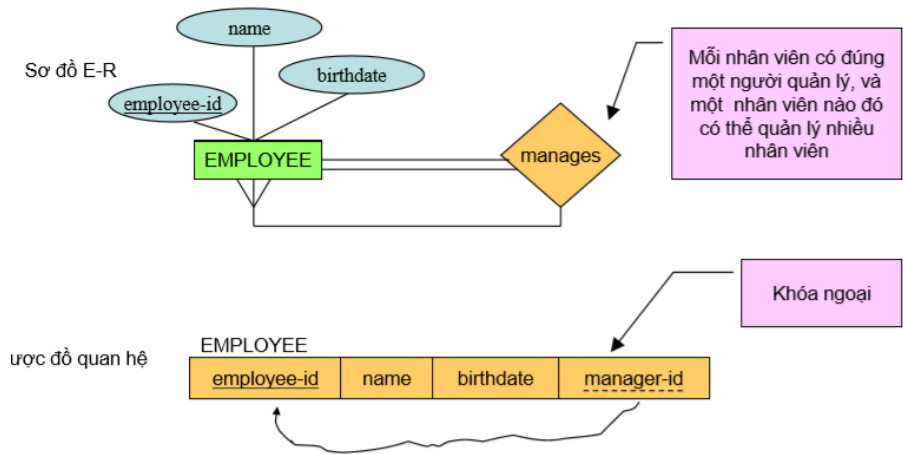
Ví dụ:



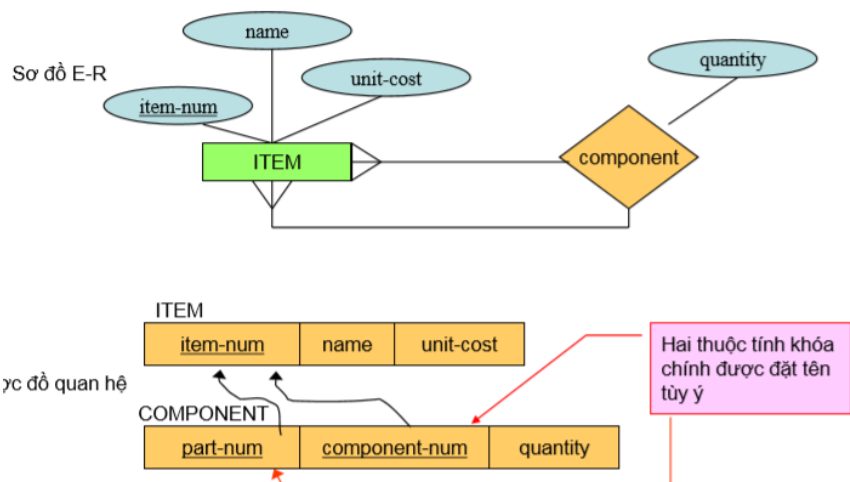
34

6. Ảnh xạ các quan hệ đệ quy (một ngôi)

- Loại 1-N: thêm thuộc tính khóa ngoại vào quan hệ có tham chiếu tới các giá trị của khóa chính (khóa ngoại này phải có cùng miền giá trị với khóa chính, khóa ngoại này được gọi là khóa ngoại đệ quy).

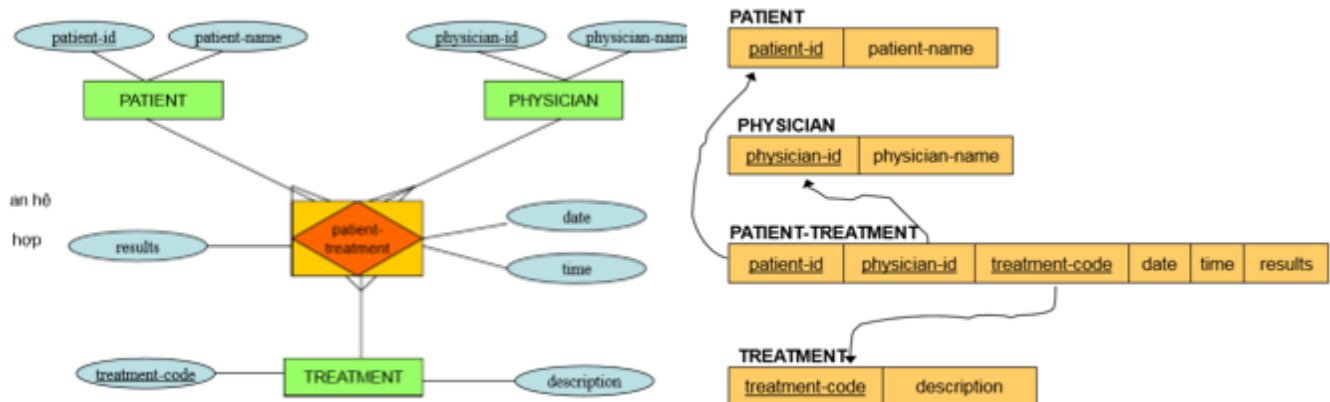


- Loại N-N: Khóa chính của quan hệ liên kết gồm hai thuộc tính. Các thuộc tính này (không nhất thiết phải có cùng tên) đều lấy giá trị từ các khóa chính của quan hệ còn lại.



7. Ánh xạ các quan hệ nhiều ngôi

- Quan hệ nhiều ngôi nên được chuyển đổi thành các thực thể kết hợp trước khi được tiếp tục xử lý.
- Để ánh xạ một thực thể kết hợp kết nối ba tập thực thể loại thường, cần tạo ra một quan hệ kết hợp mới.
- Khóa chính ngầm định cho quan hệ kết hợp gồm các thuộc tính khóa chính của các thực thể tham gia liên kết (trong một số trường hợp cần thêm các thuộc tính khác để hình thành một khóa chính duy nhất). Các thuộc tính này được coi là các khóa ngoại tham chiếu tới từng khóa chính của các tập thực thể tham gia liên kết.
- Mỗi thuộc tính của thực thể kết hợp trở thành thuộc tính trong quan hệ kết hợp mới.



8. Ánh xạ các mối liên kết lớp cha/ con

Chương 3: Phụ thuộc hàm

I. Giới thiệu

1. Định nghĩa phụ thuộc hàm

- Một phụ thuộc hàm thể hiện ngữ nghĩa của các thuộc tính trong một quan hệ: một thuộc tính có quan hệ với thuộc tính khác như thế nào và xác định các phụ thuộc hàm giữa các thuộc tính đó. Phụ thuộc hàm là ràng buộc giữa các thuộc tính.
- Xét một quan hệ với các thuộc tính A và B, với thuộc tính B là phụ thuộc hàm vào thuộc tính A: Nếu biết giá trị của A sẽ tìm thấy một giá trị duy nhất của B. Nhưng với mỗi giá trị của B cho trước có thể có nhiều giá trị khác nhau tương ứng của A.
- Đối tượng xác định của một phụ thuộc hàm là thuộc tính hoặc một nhóm các thuộc tính nằm bên trái của mũi tên trong một phụ thuộc hàm.
- Đối tượng (thuộc tính) hệ quả của phụ thuộc hàm là một thuộc tính hoặc nhóm thuộc tính nằm phía bên phải mũi tên trong phụ thuộc hàm.

Ví dụ: $A \rightarrow B$: A là đối tượng xác định và B là đối tượng hệ quả của A \Rightarrow A xác định B hay B phụ thuộc hàm vào A.

2. Phụ thuộc hàm hiển nhiên

- Trong quá trình xác định các phụ thuộc hàm, cần bỏ qua các phụ thuộc hàm hiển nhiên đúng.
- Phụ thuộc hàm hiển nhiên đúng: khi và chỉ khi vế phải của phụ thuộc hàm là một tập con của vế trái. Ví dụ: $\{\text{staff\#}, \text{sname}\} \rightarrow \text{sname}$

3. Các luật suy diễn cho các phụ thuộc hàm

Gồm 6 luật suy diễn, 3 cái đầu là hệ tiên đề Amstrong suy ra 3 cái còn lại.

- Luật phản xạ: nếu $Y \subseteq X$ thì $X \rightarrow Y$
- Luật tăng trưởng (gia tăng): nếu $X \rightarrow Y$, thì $XZ \rightarrow YZ$ hoặc $XZ \rightarrow Y$
- Luật bắc cầu: nếu $X \rightarrow Y$ và $Y \rightarrow Z$, thì $X \rightarrow Z$
- Luật chiếu: Nếu $X \rightarrow YZ$, thì $X \rightarrow Y$ và $X \rightarrow Z$
- Luật cộng thêm: Nếu $X \rightarrow Y$ và $X \rightarrow Z$, thì $X \rightarrow YZ$
- Luật giả bắc cầu: Nếu $X \rightarrow Y$ và $YZ \rightarrow W$, thì $XZ \rightarrow W$

II. Bao đóng và cách xác định bao đóng

1. Định nghĩa

- Gọi F là tập các phụ thuộc hàm xác định trên lược đồ quan hệ R.
- Ngoài các phụ thuộc hàm hiển nhiên còn nhiều các phụ thuộc hàm khác cũng thỏa mãn với các thể hiện của quan hệ đã thỏa mãn các phụ thuộc hàm trong F.
- Tập tất cả các phụ thuộc hàm được suy diễn từ một tập phụ thuộc hàm F được gọi là bao đóng của F và được ký hiệu là F^+ .
- Ký hiệu: $F \models X \rightarrow Y$ cho biết phụ thuộc hàm $X \rightarrow Y$ được suy diễn từ tập phụ thuộc hàm F.
- $F^+ \equiv \{X \rightarrow Y \mid F \models X \rightarrow Y\}$

2. Xác định bao đóng

- F^+ là bao đóng của tập phụ thuộc hàm F. F^+ là tập (nhỏ nhất) tất cả các phụ thuộc hàm được sinh ra nhờ hệ tiên đề Amstrong.
- F^+ là hữu hạn nhưng có kích thước tăng theo cấp số nhân so với số thuộc tính của R.
- Để xác định liệu một phụ thuộc hàm $X \rightarrow Y$ có thỏa mãn lược đồ quan hệ R với tập phụ thuộc hàm F hay không thì cần xem $F \models X \rightarrow Y$ không, hoặc chính xác hơn là xem $X \rightarrow Y$ có nằm trong F^+ hay không.
 - \Rightarrow Mong muốn kiểm tra được $X \rightarrow Y$ có nằm trong F^+ hay không mà không cần sinh ra tất cả các thành phần của bao đóng.
 - \Rightarrow Thực hiện bằng cách: chỉ cần sinh ra bao đóng của tập thuộc tính X, ký hiệu là X^+ , và kiểm tra xem liệu Y thuộc X^+ hay không.

3. Thuật toán tính bao đóng

X^+ được gọi là bao đóng của tập thuộc tính X trên tập phụ thuộc hàm F nếu mọi thuộc tính trong X^+ đều được sinh ra từ X nhờ F .

Thuật toán Closure {trả về X^+ trên F }:

```
Closure ( $X, F$ ) {  
     $X \rightarrow X^+$ ;  
    while ( $\text{old}X^+ \neq X^+$ ) {  
         $X^+ \rightarrow \text{old}X^+$ ;  
        for mỗi phụ thuộc hàm  $W \rightarrow Z$  trong  $F$  do  
            if  $W \subseteq X^+$  then  $X^+ \cup Z \rightarrow X^+$ ; } }
```

VD: Cho $F = \{A \rightarrow D, AB \rightarrow E, BI \rightarrow E, CD \rightarrow I, E \rightarrow C\}$. Tính $(AE)^+$.

Vòng 1:

$X^+ = \{A, E\}$

Có $A \rightarrow D, A \subseteq X^+ \Rightarrow$ thêm D vào $X^+, X^+ = \{A, E, D\}$

Có $E \rightarrow C, E \subseteq X^+ \Rightarrow$ thêm C vào $X^+, X^+ = \{A, E, D, C\}$

Vòng 2:

$X^+ = \{A, E, D, C\}$

Có $CD \rightarrow I, CD \subseteq X^+ \Rightarrow$ thêm I vào $X^+, X^+ = \{A, E, D, C, I\}$

Vòng 3: Thuật toán kết thúc

Vậy $(AE)^+ = \{A, E, D, C, I\}$

\Rightarrow Điều này có nghĩa là các phụ thuộc hàm sau sẽ thuộc F^+ : $AE \rightarrow AECDI$

\Rightarrow Phụ thuộc hàm $AE \rightarrow CDI$ có suy diễn từ F hay không? \Rightarrow Có vì CDI là tập con của $(AE)^+$.

III. Phủ và sự tương đương của phụ thuộc hàm

- Một tập phụ thuộc hàm F được phủ bởi một tập phụ thuộc hàm G (hay nói cách khác G phủ F) nếu mọi phụ thuộc hàm trong F đều nằm trong G^+ .
 $\Rightarrow F$ được phủ nếu mọi phụ thuộc hàm trong F có thể được suy diễn từ G .
- Hai tập phụ thuộc hàm F và G là tương đương ($F \equiv G$) nếu $F^+ = G^+$.
 \Rightarrow Mọi phụ thuộc hàm trong G có thể được suy diễn từ F và mọi phụ thuộc hàm trong F có thể được suy diễn từ G .
- Để xác định xem G có phủ F hay không: tính X^+ trên G cho mỗi phụ thuộc hàm $X \rightarrow Y$ trong F , nếu $Y \subseteq X^+$ cho mọi $X \rightarrow Y$ thì G phủ F .

IV. Phủ không dư thừa

1. Định nghĩa

- Một tập các phụ thuộc hàm F được cho là không dư thừa nếu không có tập con thực sự G nào của F mà G tương đương với F . Nói cách khác, trong F không có phụ thuộc hàm dư thừa.
- F là một phủ không dư thừa của G nếu F là một phủ của G và F không dư thừa.

- Với một tập các phụ thuộc hàm cho trước có thể có nhiều hơn một phủ không dư thừa.
- Phủ không dư thừa của một tập các phụ thuộc hàm G có thể chứa những phụ thuộc hàm không nằm trong G .

2. Thuật toán sinh ra 1 phủ không dư thừa

Thuật toán Nonredundant {sinh ra một phủ không dư thừa}

Từ 1 tập phụ thuộc hàm $G \Rightarrow$ 1 phủ không dư thừa của G

Nonredundant (G) {

$G \rightarrow F$;

for mỗi phụ thuộc hàm $X \rightarrow Y \in G$ do

if (Member($F - \{X \rightarrow Y\}$, $X \rightarrow Y$)) then $F - \{X \rightarrow Y\} \rightarrow F$;

return (F); }

Đơn giản: Xét tất cả phụ thuộc hàm trong F , nếu 1 phụ thuộc hàm có thể suy diễn từ {tập các phụ thuộc hàm đã bỏ nó đi} thì tức là nó thừa \Rightarrow bỏ phụ thuộc hàm này.

VD: Tìm một phủ không dư thừa của $G = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C\}$.

Ta có:

$G \rightarrow F$

Xét $A \rightarrow B$:

Member($\{B \rightarrow A, B \rightarrow C, A \rightarrow C\}, A \rightarrow B$)

Closure($A, \{B \rightarrow A, B \rightarrow C, A \rightarrow C\}$)

$A^+ = \{A, C\} \Rightarrow A \rightarrow B$ là không dư thừa

Xét $B \rightarrow A$:

Member($\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}, B \rightarrow A$)

Closure($B, \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$)

$B^+ = \{B, C\} \Rightarrow$ nên $B \rightarrow A$ là không dư thừa

Xét $B \rightarrow C$:

Member($\{A \rightarrow B, B \rightarrow A, A \rightarrow C\}, B \rightarrow C$)

Closure($B, \{A \rightarrow B, B \rightarrow A, A \rightarrow C\}$)

$B^+ = \{B, A, C\} \Rightarrow B \rightarrow C$ dư thừa \Rightarrow bỏ đi trong F

Xét $A \rightarrow C$:

Member($\{A \rightarrow B, B \rightarrow A\}, A \rightarrow C$)

Closure($A, \{A \rightarrow B, B \rightarrow A\}$)

$A^+ = \{A, B\} \Rightarrow A \rightarrow C$ là không dư thừa

Vậy $F = \{A \rightarrow B, B \rightarrow A, A \rightarrow C\}$ là 1 phủ không dư thừa của G .

V. Tập phụ thuộc hàm tối giản

1. Định nghĩa

Cho F là một tập các phụ thuộc hàm trên lược đồ R và cho $X \rightarrow Y \in F$.

- $X \rightarrow Y$ được gọi là tối giản trái nếu X không chứa các thuộc tính dư thừa A .
 \Rightarrow Một phụ thuộc hàm tối giản về trái cũng được gọi là 1 phụ thuộc hàm đầy đủ.

- $X \rightarrow Y$ được gọi là tối giản phải nếu Y không chứa các thuộc tính dư thừa A .
- $X \rightarrow Y$ được gọi là tối giản nếu nó tối giản trái, tối giản phải và Y khác rỗng.

2. Thuật toán tối giản trái

Thuật toán tối giản trái {trả lại tập các phụ thuộc hàm tối giản về trái F }

Từ tập các phụ thuộc hàm $G \Rightarrow$ một tập các phụ thuộc hàm tối giản về trái F

Left-Reduce (G) {

$G \rightarrow F$;

for mỗi phụ thuộc hàm $X \rightarrow Y$ trong G do

for mỗi thuộc tính A trong X do

if Member($F, (X - A) \rightarrow Y$) then loại bỏ A khỏi X trong $X \rightarrow Y$ của F

return(F); }

Đơn giản: Xét tất cả phụ thuộc hàm trong F có vế trái lớn hơn 1 thuộc tính, xét tất cả thuộc tính thuộc phụ thuộc hàm đó, nếu phụ thuộc hàm đã bỏ đi thuộc tính đang xét vẫn có thể suy diễn ra từ $F \Rightarrow$ thuộc tính đang xét là thừa \Rightarrow bỏ đi

VD: Tìm tập các phụ thuộc hàm tối giản về trái của $F = \{A \rightarrow BC, BD \rightarrow A, AB \rightarrow CD\}$.

$F \rightarrow F'$

Xét $BD \rightarrow A$:

Xét B :

Member($F', D \rightarrow A$)

Closure(D, F')

$D^+ = \{D\} \Rightarrow D \rightarrow A$ là không dư thừa $\Rightarrow B$ không dư thừa

Xét D :

Member($F', B \rightarrow A$)

Closure(B, F')

$B^+ = \{B\} \Rightarrow B \rightarrow A$ là không dư thừa $\Rightarrow D$ không dư thừa

$\Rightarrow BD \rightarrow A$ không dư thừa về trái

Xét $AB \rightarrow CD$:

Xét A :

Member($F', B \rightarrow CD$)

Closure(B, F')

$B^+ = \{B\} \Rightarrow B \rightarrow CD$ là không dư thừa $\Rightarrow A$ không dư thừa

Xét B :

Member($F', A \rightarrow CD$)

Closure(A, F')

$A^+ = \{A, B, C, D\} \Rightarrow A \rightarrow CD$ là dư thừa $\Rightarrow B$ dư thừa, bỏ B

$\Rightarrow AB \rightarrow CD$ dư thừa thuộc tính $B \Rightarrow$ Thay thế thành $A \rightarrow CD$

Vậy $F' = \{A \rightarrow BC, BD \rightarrow A, A \rightarrow CD\}$ là 1 tập các phụ thuộc hàm tối giản về trái của F .

3. Thuật toán tối giản phải

Thuật toán tối giản phải {trả lại tập các phụ thuộc hàm tối giản về phải F}

Từ tập các phụ thuộc hàm $G \Rightarrow$ một tập các phụ thuộc hàm tối giản về phải F

Right-Reduce (G) {

$G \rightarrow F$;

for mỗi phụ thuộc hàm $X \rightarrow Y$ trong G do

for mỗi thuộc tính A trong Y do

if $\text{Member}(F - \{X \rightarrow Y\} \cup \{X \rightarrow (Y - A)\}, X \rightarrow A)$

then loại bỏ A khỏi Y trong $X \rightarrow Y$ của F

return(F); }

Đơn giản: Xét tất cả phụ thuộc hàm trong F có vế phải lớn hơn 1 thuộc tính, tách nó ra thành các phụ thuộc hàm theo luật chiếu rồi xét các phụ thuộc hàm đã bị tách ra đó, nếu phụ thuộc hàm nào có thể được suy diễn từ các phụ thuộc hàm còn lại \Rightarrow phụ thuộc hàm đó thừa \Rightarrow thuộc tính phải tương ứng phụ thuộc hàm đang xét là thừa \Rightarrow bỏ đi

VD: Tìm tập các phụ thuộc hàm tối giản về phải của $F = \{A \rightarrow BC, BD \rightarrow A, A \rightarrow CD\}$.

Cách 1: $F \rightarrow F'$

Xét $A \rightarrow BC$:

Xét B:

$\text{Member}(\{A \rightarrow C, BD \rightarrow A, A \rightarrow CD\}, A \rightarrow B)$

$\text{Closure}(A, \{A \rightarrow C, BD \rightarrow A, A \rightarrow CD\})$

$A^+ = \{A, C, D\} \Rightarrow A \rightarrow B$ là không dư thừa $\Rightarrow B$ không dư thừa

Xét C:

$\text{Member}(\{A \rightarrow B, BD \rightarrow A, A \rightarrow CD\}, A \rightarrow C)$

$\text{Closure}(A, \{A \rightarrow B, BD \rightarrow A, A \rightarrow CD\})$

$A^+ = \{A, B, C, D\} \Rightarrow A \rightarrow C$ là dư thừa $\Rightarrow C$ dư thừa, bỏ C

$\Rightarrow A \rightarrow BC$ dư thừa thuộc tính C về phải \Rightarrow Thay thế thành $A \rightarrow B$

$\Rightarrow F' = \{A \rightarrow B, BD \rightarrow A, A \rightarrow CD\}$

Xét $A \rightarrow CD$:

Xét C:

$\text{Member}(\{A \rightarrow B, BD \rightarrow A, A \rightarrow D\}, A \rightarrow C)$

$\text{Closure}(A, \{A \rightarrow B, BD \rightarrow A, A \rightarrow D\})$

$A^+ = \{A, B, D\} \Rightarrow A \rightarrow C$ là không dư thừa $\Rightarrow C$ không dư thừa

Xét D:

$\text{Member}(\{A \rightarrow B, BD \rightarrow A, A \rightarrow C\}, A \rightarrow D)$

$\text{Closure}(A, \{A \rightarrow B, BD \rightarrow A, A \rightarrow C\})$

$A^+ = \{A, B, C\} \Rightarrow A \rightarrow D$ là không dư thừa $\Rightarrow D$ không dư thừa

Vậy $F' = \{A \rightarrow B, BD \rightarrow A, A \rightarrow CD\}$ là 1 tập các phụ thuộc hàm tối giản về phải của F .

Cách 2:

$F \rightarrow F'$

Xét $A \rightarrow BC$, tách thành 2 phụ thuộc hàm $A \rightarrow B$ và $A \rightarrow C$.

Có $F' = \{A \rightarrow B, A \rightarrow C, BD \rightarrow A, A \rightarrow CD\}$

Để thấy $A \rightarrow C$ dư thừa vì có thể suy ra từ $A \rightarrow CD \Rightarrow$ bỏ $A \rightarrow C$

$\Rightarrow F' = \{A \rightarrow B, BD \rightarrow A, A \rightarrow CD\}$

Xét $A \rightarrow CD$, tách thành 2 phụ thuộc hàm $A \rightarrow D$ và $A \rightarrow C$.

Có $F' = \{A \rightarrow B, BD \rightarrow A, A \rightarrow C, A \rightarrow D\}$

Không thấy phụ thuộc hàm dư thừa \Rightarrow Kết luận F' .

4. Thuật toán tối giản phụ thuộc hàm

Thuật toán tối giản phụ thuộc hàm {trả lại tập các phụ thuộc hàm tối giản F }

Từ tập các phụ thuộc hàm $G \Rightarrow$ một tập các phụ thuộc hàm tối giản G

Reduce (G) {

Right-Reduce(Left-Reduce(G)) $\rightarrow F$;

loại bỏ tất cả các phụ thuộc hàm dạng $X \rightarrow \text{null}$ từ F

return(F); }

VI. Phủ tối thiểu

1. Định nghĩa

Một tập phụ thuộc hàm F là tối thiểu nếu:

- Mọi phụ thuộc hàm đều có về phải là một thuộc tính.
- F là không dư thừa.
- F là một tối giản trái.

2. Thuật toán tìm phủ tối thiểu

Thuật toán tìm phủ tối thiểu {trả lại phủ tối thiểu của F }

Từ tập các phụ thuộc hàm $F \Rightarrow$ một phủ tối thiểu của F .

MinCover (F) {

$F \rightarrow G$;

Tách tất cả phụ thuộc hàm có về phải hơn 1 thuộc tính thành các phụ thuộc hàm theo luật chiếu;

Left-Reduce(G);

Nonredundant(G);

return(G); }

Chương 4: Chuẩn hoá lược đồ quan hệ

I. Chuẩn hoá dựa trên khoá chính

1. Nhắc lại khái niệm

- Siêu khóa: Là một tập các thuộc tính xác định duy nhất thực thể trong quan hệ.
- Khóa: là một siêu khóa mà khi loại bỏ bất kỳ thuộc tính nào từ khóa này thì nó không còn là một siêu khóa nữa. Nghĩa là, khóa có số thuộc tính là nhỏ nhất.
- Khóa dự bị (khái niệm tương đương khoá): là một tập các thuộc tính khóa nhỏ nhất của lược đồ quan hệ.
- Khóa chính: là một khóa dự bị được chọn ra. Tất cả các khóa dự bị còn lại trở thành khóa phụ hay khóa thứ cấp.
- Thuộc tính khóa: là thuộc tính của quan hệ R và là thành viên của một khóa dự bị nào đó.
- Thuộc tính không khóa: là thuộc tính của quan hệ R mà không phải là thành viên của một khóa dự bị nào.

2. Xác định khoá cho 1 lược đồ quan hệ

ĐN: Nếu R là một lược đồ quan hệ với các thuộc tính A_1, A_2, \dots, A_n và một tập các phụ thuộc hàm F trong đó $X \subseteq \{A_1, A_2, \dots, A_n\}$ thì X là một khóa của R khi:

- $X \rightarrow A_1 A_2 \dots A_n \in F^+$, và
- Không có tập con thực sự nào $Y \subseteq X$ mà $Y \rightarrow A_1 A_2 \dots A_n \in F^+$.

VD: Cho $r = (C, T, H, R, S, G)$ với tập phụ thuộc hàm $F = \{C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R\}$. Xác định 1 khoá của r.

Bước 1: Tính $(A_i)^+$ với $1 \leq i \leq n$

$C^+ = \{CT\}$, $T^+ = \{T\}$, $H^+ = \{H\}$ $R^+ = \{R\}$, $S^+ = \{S\}$, $G^+ = \{G\}$

\Rightarrow Không có thuộc tính đơn nào là khoá của r.

Bước 2: Tính $X^+ = (A_i A_j)^+$ với $1 \leq i \leq n, 1 \leq j \leq n$

$(CT)^+ = \{C, T\}$, $(CH)^+ = \{CHTR\}$, $(CR)^+ = \{CRT\}$, $(CS)^+ = \{CSGT\}$,

$(CG)^+ = \{CGT\}$, $(TH)^+ = \{THRC\}$, $(TR)^+ = \{TR\}$, $(TS)^+ = \{TS\}$,

$(TG)^+ = \{TG\}$, $(HR)^+ = \{HRCT\}$, $(HS)^+ = \{HSRCTG\}$, $(HG)^+ = \{HG\}$,

$(RS)^+ = \{RS\}$, $(RG)^+ = \{RG\}$, $(SG)^+ = \{SG\}$

\Rightarrow Tập thuộc tính (HS) là 1 khoá của r

\Rightarrow Các tập thuộc tính khác có số thuộc tính lớn hơn bao gồm cả (HS) (VD: CHS, THSG, CTHRS,...) đều là siêu khoá.

II. Dạng chuẩn 1 (1NF)

ĐN: Một quan hệ ở dạng chuẩn 1:

- Mọi giá trị thuộc tính của quan hệ đều ở dạng nguyên tố.
- Không có thuộc tính đa trị.
- Không có thuộc tính dẫn xuất.

III. Dạng chuẩn 2 (2NF)

1. Định nghĩa

- Dạng chuẩn 2 dựa trên khái niệm phụ thuộc hàm đầy đủ.
- Một phụ thuộc hàm $X \rightarrow Y$ được gọi là phụ thuộc hàm đầy đủ nếu loại bỏ bất kỳ thuộc tính A nào của X thì sẽ làm phụ thuộc hàm này không còn đúng nữa. Nói cách khác $X \rightarrow Y$ tối giản trái.
 \Rightarrow Không có bất kỳ thuộc tính $A \in X$ nào mà $X - \{A\} \rightarrow Y$
- ĐN: Một lược đồ quan hệ R với tập phụ thuộc hàm F ở dạng chuẩn 2NF khi thỏa mãn:
 - Là dạng chuẩn 1NF.
 - Mọi thuộc tính không khóa phụ thuộc hàm đầy đủ vào mọi khóa dự bị của R.
- Nói cách khác, một lược đồ quan hệ R với tập phụ thuộc hàm F ở dạng chuẩn 2NF khi thỏa mãn một trong các điều kiện sau:
 - Tất cả các khóa dự bị đều có một thuộc tính.
 - Không có thuộc tính không khóa nào.
 - Không tồn tại thuộc tính không khóa mà phụ thuộc hàm một phần vào một khóa dự bị của R.

2. Chuẩn hoá từ 1NF sang 2NF

- Loại bỏ các thuộc tính không khóa mà phụ thuộc một phần vào khóa chính để tách thành bảng riêng. Khóa chính của bảng riêng là bộ phận khóa mà chúng phụ thuộc vào.
- Các thuộc tính còn lại lập thành một bảng với khóa chính là khóa ban đầu.

VD: Cho $R = (A, D, P, G)$, với tập phụ thuộc hàm $F = \{AD \rightarrow P, A \rightarrow G\}$ và một khóa $K = \{AD\}$.

R không ở dạng chuẩn 2NF vì G phụ thuộc một phần vào khóa AD do $A \rightarrow G$.

Phân rã R thành 2 bảng:

- $R1 = (A, D, P); K1 = \{AD\}; F1 = \{AD \rightarrow P\}$
- $R2 = (A, G); K2 = \{A\}; F2 = \{A \rightarrow G\}$

IV. Dạng chuẩn 3 (3NF)

1. Định nghĩa

- Dạng chuẩn 3 dựa trên khái niệm phụ thuộc hàm bắc cầu.
- Định nghĩa phụ thuộc hàm bắc cầu: Cho một lược đồ quan hệ R và một tập các phụ thuộc hàm F của R, một tập con $X \subseteq R$ và một thuộc tính $A \in R$. A được gọi là phụ thuộc hàm bắc cầu vào X nếu tồn tại $Y \subseteq R$ mà $X \rightarrow Y$, $Y \rightarrow A$ và $A \notin X \cup Y$.
- Một cách định nghĩa khác cho phụ thuộc hàm bắc cầu: Một phụ thuộc hàm $X \rightarrow Y$ trong lược đồ quan hệ R là phụ thuộc bắc cầu nếu có một tập thuộc tính $Z \subseteq R$ mà Z không phải là tập con của bất kỳ khóa nào của R nhưng $X \rightarrow Z$ và $Z \rightarrow Y$.

- Định nghĩa dạng chuẩn 3NF: Một lược đồ quan hệ R ở dạng 3NF với một tập phụ thuộc hàm F nếu thỏa mãn: Với bất kỳ phụ thuộc hàm $X \rightarrow A$ trong F thì hoặc X là một siêu khóa của R, hoặc A là một thuộc tính khóa.
- Một cách định nghĩa khác về dạng chuẩn 3NF: Một lược đồ quan hệ R ở dạng 3NF với một tập phụ thuộc hàm F nếu thỏa mãn:
 - R ở dạng 2NF
 - Không thuộc tính không khóa nào phụ thuộc bắc cầu vào một khóa của R.

2. Chuẩn hoá về 3NF

- Cách 1:
 - Tìm phủ tối thiểu của tập phụ thuộc hàm của lược đồ ban đầu.
 - Loại bỏ các thuộc tính phụ thuộc bắc cầu ra khỏi quan hệ, lập thành bảng mới với khóa chính là thuộc tính bắc cầu.
 - Các thuộc tính còn lại lập thành một bảng với khóa chính là khóa ban đầu.
- Cách 2:
 - Tìm phủ tối thiểu (G) của tập phụ thuộc hàm (F) của lược đồ (R) ban đầu.
 - Ứng với mỗi phụ thuộc hàm trong G, lập một bảng quan hệ mới.
 - Nếu trong tất cả các bảng mới tạo ra không có bảng nào chứa một khóa dự bị nào của R thì tạo ra thêm một bảng mới chứa một khóa dự bị của R.

VD1: Cho $R = (A, B, C, D, E)$, và tập phụ thuộc hàm $F = \{AB \rightarrow C, C \rightarrow B, A \rightarrow D\}$. Chuẩn hoá R về dạng chuẩn 3NF.

LG

- Dễ thấy F đã là phủ tối thiểu (vế phải đơn, không quan hệ dư thừa, tối giản trái).
- Xác định khoá dự bị:
 - $(AE)^+ = \{A, B, C, D\}$;
 - $(ABE)^+ = \{A, B, C, D, E\}$; $(ACE)^+ = \{A, C, E, B, D\}$; $(ADE)^+ = \{A, D, E\}$
 - \Rightarrow Khoá dự bị của R là (ABE) và (ACE)
- Cách 1:
 - Dễ thấy chưa chuẩn 2NF vì có $A \rightarrow D$: D là thuộc tính không khoá phụ thuộc không đầy đủ vào khoá.
 - \Rightarrow Đưa về chuẩn 2 sẽ là chuẩn 3.
 - Phân rã R thành 2 bảng:
 - $R_1 = (A, D)$; $K_1 = \{A\}$; $F_1 = \{A \rightarrow D\}$
 - $R_2 = (A, B, C, E)$; $K_2 = \{ABE\}$; $F_2 = \{C \rightarrow B, AB \rightarrow C\}$
- Cách 2:
 - Phân rã R thành 3 bảng, vì 3 bảng đều không chứa 1 khoá dự bị nào của R nên ta thêm 1 bảng chứa 1 khoá dự bị của R là 4 bảng, chọn khoá dự bị nào cũng được:
 - $R_1 = (A, B, C)$; $K_1 = \{AB\}$; $F_1 = \{AB \rightarrow C\}$
 - $R_2 = (C, B)$; $K_2 = \{C\}$; $F_2 = \{C \rightarrow B\}$
 - $R_3 = (A, D)$; $K_3 = \{A\}$; $F_3 = \{A \rightarrow D\}$

$$- R4 = (A, B, E); K2 = \{ABE\}$$

VD2: Cho lược đồ quan hệ $U = (BCDEG)$ với tập phụ thuộc hàm: $F = \{BC \rightarrow G, E \rightarrow DE, E \rightarrow B, E \rightarrow CD, E \rightarrow G\}$. Kiểm tra 3NF, nếu chưa thì chuẩn hoá.

LG

- Khoá dự bị: $\{E\}$
- $E \rightarrow DE, E \rightarrow B, E \rightarrow CD, E \rightarrow G$ là thoả mãn vì có E là siêu khoá
- $BC \rightarrow G$ không thoả mãn vì BC không phải siêu khoá ($(BC)^+ = (BC)$ là tập con của R)
 \Rightarrow Chưa đạt chuẩn 3NF
- Có G là thuộc tính không khoá phụ thuộc bắc cầu vào E thông qua BC . Tách R :
 - $R2 = (B, C, G); F2 = (BC \rightarrow G); K2 = (BC)$
 - $R1 = (E, D, B, C); F1 = \{E \rightarrow DE, E \rightarrow B, E \rightarrow CD, E \rightarrow G\}; K1 = (E)$

V. Dạng chuẩn Boyce-codd (BCNF)

- Dạng chuẩn Boyce-Codd (BCNF) là một chuẩn nghiêm ngặt hơn chuẩn 3NF.
- Định nghĩa: Một lược đồ quan hệ R được coi là ở dạng chuẩn Boyce-Codd với tập phụ thuộc hàm F nếu với bất kỳ phụ thuộc hàm $X \rightarrow A$ nào và $A \notin X$, thì X là một siêu khóa của R .
- Ví dụ: Cho lược đồ quan hệ $R = (A, B, C)$, tập phụ thuộc hàm $F = \{AB \rightarrow C, C \rightarrow A\}$ và khóa $K = \{AB\}$
 R không chuẩn BCNF vì có $C \rightarrow A$ và C không phải là một siêu khóa của R .
- Sự khác nhau giữa dạng chuẩn 3NF và BCNF là BCNF cho phép bỏ đi luật với phụ thuộc hàm $X \rightarrow A$ thì X phải là thuộc tính khóa.
- Trong thực tế, hầu hết các lược đồ quan hệ ở dạng chuẩn 3NF thì cũng ở dạng chuẩn BCNF. Chỉ khi trong lược đồ có $X \rightarrow A$ mà X không phải là một siêu khóa hoặc A là một thuộc tính khóa thì lược đồ này ở dạng chuẩn 3NF mà không ở dạng chuẩn BCNF.

Chương 5: Ngôn ngữ truy vấn đại số quan hệ

I. Ngôn ngữ đại số quan hệ

- Là một ngôn ngữ truy vấn thủ tục, bao gồm các phép toán tập hợp một ngôi hoặc hai ngôi, nghĩa là các toán hạng của chúng là một quan hệ hoặc hai quan hệ. Kết quả đầu ra là một quan hệ.
- Năm phép toán cơ bản của đại số quan hệ là phép chọn, phép chiếu, phép hợp, phép trừ và phép tích Đề-các.
- Một số các phép toán mở rộng khác cũng được định nghĩa trong đại số quan hệ bao gồm: phép giao, phép kết nối tự nhiên, phép chia, phép bán kết nối, và phép kết nối ngoài.
- Mệnh đề điều kiện có thể chứa các toán tử so sánh, như $=, \neq, <, \geq, \leq, >$. Hoặc kết hợp với các phép toán liên kết và (\wedge) , hoặc (\vee) , và phủ định (\neg) .

II. Năm phép toán cơ bản

1. Phép chọn (một ngôi)

- Ký hiệu: Sigma, σ
- Khuôn dạng chung: $\sigma_{(mệnh\ đề)}$ (thể hiện của quan hệ) (SELECT * + WHERE)
- VD: $\sigma_{(khoa="CNTT" \wedge gpa \geq 3.0)}(STUDENTS)$

2. Phép chiếu (một ngôi)

- Ký hiệu: Pi, π
- Khuôn dạng chung: $\pi_{(danh\ sách\ thuộc\ tính)}$ (thể hiện của quan hệ) (SELECT)
- Chú ý: Nếu phép toán sinh ra các bộ giống hệt nhau, thì sẽ chỉ giữ lại một bộ và loại bỏ đi các bộ bị trùng.
- VD: $\pi_{(name,gpa,major)}(STUDENTS)$

3. Phép hợp (hai ngôi)

- Ký hiệu: \cup
- Khuôn dạng chung: $r \cup s$, với r và s là 2 quan hệ khả hợp.
- VD: $\pi_{(a,b)}(r) \cup \pi_{(a,b)}(s)$
- 2 quan hệ r và s được gọi là khả hợp khi thoả mãn 2 điều kiện:
 - Chúng phải có cùng số bậc hay cùng số lượng thuộc tính.
 - Miền giá trị của thuộc tính thứ (i) của r và thuộc tính thứ (j) của s phải giống nhau, cho mọi giá trị của i, j.

A	B	D
a	a	1
b	d	7
c	f	34
a	d	6
a	c	7

A	B
a	a
b	d
c	f
a	d
a	c

$r = R \cup T$

Không hợp lệ – R và T không phải là 2 quan hệ khả hợp

E	F	G
a	a	1
b	d	7
c	f	34
a	d	6
a	c	7
a	m	4
b	c	22
a	d	16

$r = T \cup X$

A	B
a	a
b	d
c	f
a	d
a	c

X	Y	Z
a	m	4
b	c	22
a	d	16
a	c	7

A	B
a	a
b	d
a	c

4. Phép trừ (hai ngôi)

- Ký hiệu: $-$
- Khuôn dạng chung: $r - s$, với r và s là 2 quan hệ khả hợp.

A	B	D
a	a	1
b	d	7
c	f	34
a	d	6
a	c	7

A	B
a	a
b	d
c	f
a	d
a	c

$r = R - T$

Không hợp lệ – R và T không phải là 2 quan hệ khả hợp

E	F	G
a	a	1
b	d	7
c	f	34
a	d	6

$r = T - X$

A	B
c	f
a	d

X	Y	Z
a	m	4
b	c	22
a	d	16
a	c	7

A	B
a	a
b	d
a	c

$r = X - T$

Quan hệ rỗng

E	F	G
a	m	4
b	c	22
a	d	16

5. Phép tích Đề-các (hai ngôi)

- Ký hiệu: \times
- Khuôn dạng chung: $r \times s$.
- Tích Đề-các cho phép kết nối 2 quan hệ bất kỳ thành một quan hệ đơn.
- Một quan hệ là một tập con của tích Đề-các tập các miền giá trị.

T	A	B
a	a	
b	d	

X	A	B
a	a	
b	d	
a	c	
c	a	

r = T x X			
T.A	T.B	X.A	X.B
a	a	a	a
a	a	b	d
a	a	a	c
a	a	c	a
b	d	a	a
b	d	b	d
b	d	a	c
b	d	c	a

VD: Ta có 1 lược đồ quan hệ như sau:

S = STUDENTS(s#, name, age, major, gpa, hours_completed)

C = COURSES(c#, term, name, dept, enrollment)

P = PROFESSORS(p#, name, dept, yrs_teaching, area)

TA = TAKES(s#, c#, term, grade)

TE = TEACH(p#, c#, term)

Hãy viết các biểu thức đại số quan hệ biểu diễn:

a) Tìm tên tất cả các giáo sư đã dạy ít nhất 1 lớp trong kỳ mùa thu 2002 (Fall 2002)

$$result = \pi_{(P.name)}(\sigma_{(TE.term="Fall\ 2002" \wedge P.p\# = TE.p\#)}(P \times TE))$$

b) Tìm tên tất cả sinh viên đã học 1 lớp trong kỳ mùa thu 2003 mà được dạy bởi 1 giáo sư có hơn 20 năm kinh nghiệm giảng dạy

$$\pi_{(S.name)}(\sigma_{(S.s\# = TA.s\# \wedge TA.c\# = TE.c\# \wedge TA.term = TE.term \wedge TE.p\# = P.p\# \wedge TA.term = "Fall\ 2003" \wedge P.yrs_{teaching} \geq 20)}(S \times P \times TA \times TE))$$

c) Tìm tên tất cả giáo sư hoặc dạy ở khoa CNTT (CS) hoặc có hơn 20 năm kinh nghiệm giảng dạy.

$$result = \pi_{(name)}(\sigma_{(dept="CS")}(P) \cup \sigma_{(yrs_{teaching} \geq 20)}(P))$$

d) Tìm số hiệu tất cả các sinh viên mà **chỉ** đăng kí học vào kì mùa xuân 2003.

$$result = \pi_{(s\#)}(\sigma_{(term="Spring\ 2003")}(TA)) - \pi_{(s\#)}(\sigma_{(term \neq "Spring\ 2003")}(TA))$$

III. Các phép toán khác

1. Toán tử đặt lại tên

- Ký hiệu: ρ , ρ
- Dạng 1: Đặt lại tên cho quan hệ: $\rho_{(tên\ mới\ của\ quan\ hệ)}(quan\ hệ)$ (AS)
- Dạng 2: Đặt lại tên cho quan hệ và các thuộc tính của quan hệ:

$$\rho_{tên\ mới\ của\ quan\ hệ\ (A1, A2, \dots, An)}(quan\ hệ)$$

VD: $\rho_X(r)$; $\rho_{X(one, two, \dots, last)}(r)$

2. Phép giao (hai ngôi)

- Ký hiệu: \cap
- Khuôn dạng chung: $r \cap s$ với r và s là 2 quan hệ khả hợp.
- Định nghĩa: $r \cap s \equiv r - (r - s)$

3. Phép kết nối theta và kết nối bằng (hai ngôi)

- Khuôn dạng chung: $r \triangleright \triangleleft_{(mệnh\ đề)} s$

- Định nghĩa: $r \bowtie_{(mệnh\ đề)} s \equiv \sigma_{(mệnh\ đề)}(r \times s)$
- Phép kết nối theta là dạng rút gọn của tích Đề-các và sau đó là thực hiện phép chọn.
- Phép kết nối bằng là một trường hợp đặc biệt của kết nối theta mà trong đó tất cả các điều kiện trong mệnh đề đều là điều kiện bằng.
- Cả phép kết nối theta và kết nối bằng đều không loại bỏ các bộ dư thừa.

R

A	B	C	D
a	a	yes	1
b	d	no	7
c	f	yes	34
a	d	no	6

s

E	F	G	H
a	a	yes	1
b	r	yes	3
c	f	yes	34
m	n	no	56

$r = R \bowtie_{(R.B < S.F)} S$

A	B	C	D	E	F	G	H
a	a	yes	1	b	r	yes	3
a	a	yes	1	c	f	yes	34
a	a	yes	1	m	n	no	56
b	d	no	7	b	r	yes	3
b	d	no	7	c	f	yes	34
b	d	no	7	m	n	no	56
c	f	yes	34	b	r	yes	3
c	f	yes	34	m	n	no	56
a	d	no	6	b	r	yes	3
a	d	no	6	c	f	yes	34
a	d	no	6	m	n	no	56

4. Phép kết nối tự nhiên (hai ngôi)

- Khuôn dạng chung: $r * s$
- Định nghĩa: $r * s \equiv r \bowtie_{(r.thuộc\ tính\ chung = s.thuộc\ tính\ chung)} s$
- Phép kết nối tự nhiên thực hiện kết nối bằng trên tất cả các thuộc tính có cùng tên của 2 quan hệ toán hạng.
- Bậc của quan hệ kết quả là tổng số bậc của 2 quan hệ toán hạng trừ đi số các thuộc tính chung của chúng.
- Phổ biến nhất trong tất cả các phép kết nối. Nó rất có ích trong việc loại bỏ đi các bộ dư thừa. Các thuộc tính chung của 2 quan hệ toán hạng thường được gọi là các thuộc tính kết nối.

R

A	B	C	D
a	a	yes	1
b	r	no	7
c	f	yes	34
a	m	no	6

s

B	M	G	H
a	a	yes	1
b	r	yes	3
a	f	yes	34
m	n	no	56

$r = R * S$

A	B	C	D	M	G	H
a	a	yes	1	a	yes	1
a	a	yes	1	f	yes	34
a	m	no	6	n	no	56

$r = R * T$

A	B	C	D	G	H
b	r	no	7	yes	30

T

A	B	G	H
a	f	no	31
b	r	yes	30

- Nếu 2 quan hệ không có thuộc tính chung sẽ trả về tích Đề-các.

5. Phép kết nối ngoài (hai ngôi)

- Khuôn dạng chung:
 - $r \bowtie s$: Kết nối ngoài trái.
 - $r \Join s$: Kết nối ngoài phải.
 - $r \bowtie \Join s$: Kết nối ngoài đầy đủ.
- Định nghĩa:
 - $r \bowtie s$: kết nối tự nhiên r và s với các bộ của r không tương ứng trong s vẫn được giữ lại trong kết quả. Tất cả giá trị còn khuyết ở thuộc tính của s được gán giá trị rỗng (null)
 - $r \Join s$: kết nối tự nhiên r và s với các bộ của s không tương ứng trong r vẫn được giữ lại trong kết quả. Tất cả các giá trị còn khuyết ở thuộc tính của r được gán giá trị rỗng (null).
 - $r \bowtie \Join s$: kết nối tự nhiên r và s với các bộ của r và s không tương ứng vẫn được giữ lại trong kết quả. Tất cả các giá trị còn khuyết được gán giá trị rỗng.

R

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

$r = R \bowtie \Join S$

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	null
7	8	9	null
null	6	7	12

S

B	C	D
2	3	10
2	3	11
6	7	12

$r = R \bowtie S$

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	null
7	8	9	null

$r = R \Join S$

B	C	D	A
2	3	10	1
2	3	11	1
6	7	12	null

6. Phép bán kết nối (hai ngôi)

- Khuôn dạng chung: $r \Join_{(mệnh\ đề)} s$
- Định nghĩa: $r \Join_{(mệnh\ đề)} s \equiv \pi_{(thuộc\ tính\ của\ r)}(r \bowtie_{(mệnh\ đề)} s)$
- Phép bán kết nối thực hiện một phép kết nối 2 quan hệ toán hạng, sau đó chiếu trên các thuộc tính của quan hệ toán hạng bên trái.
- Ưu điểm chính của phép bán kết nối là làm giảm số các bộ cần xử lý khi thực hiện phép kết nối => rất có ích trong môi trường phân tán.
- Theo một khuôn dạng chung như đã trình bày, phép bán kết nối chính là phép bán kết nối theta. Phép bán kết nối bằng và phép bán kết nối tự nhiên được định nghĩa theo cách tương tự.

VD: Ta có 1 lược đồ quan hệ như sau:

S = SUPPLIERS(s#, name, status, city)
P = PARTS(p#, name, color, weight, city)
J = JOBS(j#, name, numworkers, city)
SPJ = SHIPMENTS(s#, p#, j#, qty, date)

Hãy viết các biểu thức đại số quan hệ biểu diễn:

a) Tìm số hiệu các nhà cung cấp vận chuyển mọi linh kiện.

$$result = \pi_{(s\#)}(SPJ) - \pi_{(s\#)}(\pi_{(s\#)}(SPJ) \times \pi_{(p\#)}(P) - \pi_{(s\#,p\#)}(SPJ))$$

b) Tìm tên tất cả nhà cung cấp mà **chỉ** vận chuyển các linh kiện màu đỏ.

$$result = \pi_{(S.name)}(\pi_{(S.s\#,S.name)}(\sigma_{(P.color="red")}(S * P * SPJ)) - \pi_{(S.s\#,S.name)}(\sigma_{(P.color \neq "red")}(S * P * SPJ)))$$

c) Tìm tên các nhà cung cấp mà ở cùng thành phố với công việc mà họ vận chuyển linh kiện đến cho.

$$result = \pi_{(S.name)}(\sigma_{(S.city=J.city)}(S * J * SPJ))$$

d) Tìm số hiệu tất cả các linh kiện được vận chuyển bởi cả 2 nhà cung cấp có mã “S1” và “S2”.

$$result = \pi_{(p\#)}(\sigma_{(s\#="S1")}(SPJ)) \cap \pi_{(p\#)}(\sigma_{(s\#="S2")}(SPJ))$$

e) Tìm tất cả cặp (s#, j#) của các nhà cung cấp và công việc ở cùng thành phố nhưng các nhà cung cấp đó không vận chuyển bất kì linh kiện nào cho công việc này.

$$result = \pi_{(s\#,j\#)}(\sigma_{(S.city=J.city)}(S \times J)) - \pi_{(s\#,j\#)}(\sigma_{(S.city=J.city)}(S * J * SPJ))$$

f) Liệt kê tất cả các cặp số hiệu nhà cung cấp ở cùng 1 thành phố.

$$result = \pi_{(S.s\#,X.s\#)}(S \triangleright \triangleleft_{(S.city=X.city)}(\rho_X(S)))$$

g) Liệt kê tất cả các vận chuyển có liên quan đến linh kiện màu xanh (“green”)

$$result = SPJ \triangleright \triangleleft_{(SPJ.p\#=P.p\#)}(\sigma_{(color="green")}(P))$$

h) Liệt kê tên tất cả các nhà cung cấp mà vận chuyển tất cả các linh kiện màu xanh.

$$result = \pi_{(S.name)}(S * (\pi_{(s\#)}(SPJ) - \pi_{(s\#)}(\pi_{(s\#)}(SPJ) \times \pi_{(p\#)}(\sigma_{(color="green")}(P)) - \pi_{(s\#,p\#)}(SPJ))))$$

i) Liệt kê số hiệu của các nhà cung cấp mà chỉ vận chuyển các linh kiện màu xanh.

$$result = \pi_{(s\#)}(SPJ * (\sigma_{(color="green")}(P)) - \pi_{(s\#)}(SPJ * \sigma_{(color \neq "green")}(P)))$$