MỤC LỤC

[ Nội dung 1: Các khái niệm cơ bản 4](#_Toc90680764)

[1.1. Trình bày các khái niệm CSDL, hệ QTCSDL, hệ CSDL. Nêu tên một số hệ QT CSDL có mặt trên thị trường phần mềm hiện nay 4](#_Toc90680765)

[1.4. Trình bày các vai trò người dùng trong CSDL 5](#_Toc90680766)

[1.5. Trình bày ưu, nhược điểm của các hệ quản trị CSDL 5](#_Toc90680767)

[1.6. Trình bày các khái niệm: lược đồ cơ sở dữ liệu, ánh xạ các loại lược đồ và một thể hiện của cơ sở dữ liệu. 6](#_Toc90680768)

[1.7: Tính độc lập dữ liệu là gì? Phân loại tính độc lập dữ liệu. 6](#_Toc90680769)

[1.8: Trình bày khái niệm các ngôn ngữ cơ sở dữ liệu. 6](#_Toc90680770)

[2.1: Trình bày khái niệm hệ quản trị cơ sở dữ liệu. Vẽ sơ đồ và mô tả kiến trúc của một hệ quản trị cơ sở dữ liệu. 7](#_Toc90680771)

[2.2: Vẽ sơ đồ và trình bày mô hình trừu tượng ba lớp. Cho ví dụ chuyển đổi lược đồ giữa ba mức trừu tượng. 9](#_Toc90680772)

[2.3: Trình bày cách phân loại các hệ cơ sở dữ liệu dựa theo kiến trúc 11](#_Toc90680773)

[2.4: Vẽ sơ đồ và trình bày các bước chính trong quá trình thiết kế một cơ sở dữ liệu. 13](#_Toc90680774)

[ Nội dung 2: Các mô hình dữ liệu 14](#_Toc90680775)

[1.9: Trình bày khái niệm mô hình dữ liệu và các thành phần trong mô hình dữ liệu. Phân loại mô hình dữ liệu. 14](#_Toc90680776)

[1.10: Trình bày các khái niệm thực thể, tập thực thể, thuộc tính, miền giá trị của thuộc tính. 15](#_Toc90680777)

[1.11: Cho một ví dụ về mô hình thực thể liên kết E-R và giải thích các thành phần trong đó. 15](#_Toc90680778)

[1.12: Trình bày các khái niệm: thuộc tính đơn, thuộc tính kép, thuộc tính đơn trị, thuộc tính đa trị, thuộc tính dẫn xuất, thuộc tính rỗng (null). 16](#_Toc90680779)

[1.13: Trình bày về ràng buộc ánh xạ lực lượng liên kết trong mô hình quan hệ thực thể ER 16](#_Toc90680780)

[1.14: Trình bày về ràng buộc tham gia trong mô hình quan hệ thực thể E-R 17](#_Toc90680781)

[1.15: Trình bày các khái niệm siêu khóa, khóa dự bị, khóa chính của một tập thực thể 18](#_Toc90680782)

[1.16: Khóa và siêu khóa khác nhau như thế nào? Cho ví dụ và giải thích. 18](#_Toc90680783)

[1.17: Trình bày các khái niệm tập thực thể mạnh và tập thực thể yếu. Cho một ví dụ cụ thể. 18](#_Toc90680784)

[1.18: Trình bày các khái niệm cụ thể hóa và tổng quát hóa. Cho một ví dụ cụ thể 19](#_Toc90680785)

[1.19: Trình bày cơ chế kế thừa thuộc tính trong cụ thể hóa/tổng quát hóa và lợi ích của nó. 20](#_Toc90680786)

[1.20: Trình bày khái niệm tập thực thể kết hợp. Cho một ví dụ và giải thích 20](#_Toc90680787)

[1.21: Trình bày khái niệm ràng buộc toàn vẹn tham chiếu và các phương pháp được sử dụng để đảm bảo tính ràng buộc toàn vẹn tham chiếu. 20](#_Toc90680788)

[1.22: Trình bày các khái niệm: quan hệ, thuộc tính, miền giá trị của thuộc tính, bộ (bản ghi), bậc, lực lượng trong một quan hệ. Cho ví dụ cụ thể. 21](#_Toc90680789)

[1.23: Trình bày các đặc tính của một quan hệ. 21](#_Toc90680790)

[1.24: Giải thích tại sao các bộ trong một quan hệ không cần thiết phải được sắp xếp theo một trình tự nhất định nào đó? 22](#_Toc90680791)

[1.25: Giải thích tại sao trong một quan hệ không cho phép các bộ trùng nhau? 22](#_Toc90680792)

[1.26: Trình bày những ưu điểm của mô hình quan hệ. 22](#_Toc90680793)

[2.5: Trình bày ba loại ràng buộc chính trong quá trình cụ thể hóa/tổng quát hóa 22](#_Toc90680794)

[2.6: Trình bày sự tương ứng giữa sơ đồ ER và sơ đồ lớp UML về: các tập thực thể và thuộc tính, các quan hệ, các ràng buộc về lực lượng liên kết, tổng quát hóa và cụ thể hóa. 23](#_Toc90680795)

[2.7: Trình bày các trường hợp mà thuộc tính của tập mối quan hệ hai ngôi có thể được đặt vào trong các tập thực thể tham gia liên kết. Cho ví dụ cụ thể với mỗi trường hợp. 25](#_Toc90680796)

[2.8: Khi nào khái niệm thực thể yếu có ích trong quá trình mô hình hóa dữ liệu? Trình bày các định nghĩa tập thực thể xác định (còn gọi là tập thực thể mạnh hay tập thực thể sở hữu), tập thực thể yếu, tập mối quan hệ xác định, và khóa bán phần (hay thuộc tính phân biệt). Cho ví dụ cụ thể. 27](#_Toc90680797)

[ Nội dung 3: LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ 28](#_Toc90680798)

[Câu 1: Mục đích của việc thiết kế CSDL quan hệ là gì? 28](#_Toc90680799)

[Câu 1.27: Trình bày khái niệm phụ thuộc hàm và các đặc tính có ích của phụ thuộc hàm cho việc chuẩn hoá dữ liệu 28](#_Toc90680800)

[Câu 1.28: Trình bày 6 luật suy diễn phổ biến nhất cho các phụ thuộc hàm 28](#_Toc90680801)

[Câu 1.29: Trình bày các luật suy diễn trong hệ tiên đề Amstrong. Giải thích tính đúng đắn và đầy đủ của các hệ tiên đề 29](#_Toc90680802)

[Câu 1.30: Trình bày các định nghĩa: bao đóng của một tập các phụ thuộc hàm và bao đóng của một tập các thuộc tính. Trình bày thuật toán tính bao đóng 29](#_Toc90680803)

[Câu 1.33: Trình bày các khái niệm: các thuộc tính dư thừa, phụ thuộc hàm tối giản vế trái và phụ thuộc hàm tối giản vế phải 29](#_Toc90680804)

[Câu 1: 31](#_Toc90680805)

[Câu 1.34: Trình bày định nghĩa dạng chuẩn 1(1NF), dạng chuẩn 2(2NF), dạng chuẩn 3(3NF). Cho ví dụ về các dạng chuẩn này 32](#_Toc90680806)

[Câu 1.36: Trình bày định nghĩa dạng chuẩn Boyee-Codd(BCNF). So sánh dạng chuẩn 3NF với BCNF. Cho ví dụ về BCNF 33](#_Toc90680807)

[Câu 1.37: Trình bày 2 tính chất mà người thiết kế CSDL phải tuân theo khi thực hiện phân tách các lược đồ về dạng chuẩn 3NF, BCNF hoặc các dạng chuẩn ở mức cao hơn 34](#_Toc90680808)

[Câu 1.38: Tính chất bảo toàn phụ thuộc hàm khi phân tách một lược đồ quan hệ thành một tập các lược đồ nhỏ hơn là gì? Tính chất kết nối không tổn thất thông tin là gì? 34](#_Toc90680809)

[Câu 2.19: Trình bày những bất thường có thể xảy ra khi thực hiện các thao tác cập nhật dữ liệu, bao gồm: bổ sung, loại bỏ và sửa đổi dữ liệu. Cho ví dụ cụ thể: 34](#_Toc90680810)

[Câu 2. 20: Trình bày định nghĩa 2 tập phụ thuộc hàm tương đương và phương pháp để kiểm tra sự tương đương của 2 tập phụ thuộc hàm 36](#_Toc90680811)

NỘI DUNG 4: CÁC NGÔN NGỮ CƠ SỞ DỮ LIỆU

[Câu 1: SQL là gì? 36](#_Toc90680812)

[Câu 1.38: Trình bày cú pháp các câu lệnh tạo bảng và cập nhật bản ghi bằng ngôn ngữ SQL. Cho ví dụ với mỗi câu lệnh 37](#_Toc90680813)

[Câu 1.39: Trình bày cú pháp các câu lệnh thêm mới và xoá bản ghi bằng ngôn ngữ SQL. Cho ví dụ với mỗi câu lệnh 39](#_Toc90680814)

[Câu 1.40: Trình bày cú pháp các câu lệnh tìm kiếm không có điều kiện, tìm kiếm với điều kiện đơn giản và tìm kiếm có sử dụng các mệnh đề GROUP BY, ORDER BY bằng ngôn ngữ SQL 39](#_Toc90680815)

[Câu 2.21: Trình bày khái niệm và cho ví dụ về 5 phép toán cơ bản của đại số quan hệ: phép chọn, phép chiếu, phép hợp, phép trừ và phép tích Đề-các 41](#_Toc90680816)

[Câu 2.22: Trình bày khái niệm và cho ví dụ về các phép toán mở rộng của đại số quan hệ: phép giao, phép kết nối theta, phép kết nối bằng, phép kết nối tự nhiên và phép chia. 47](#_Toc90680817)

* Nội dung 1: Các khái niệm cơ bản
  1. Trình bày các khái niệm CSDL, hệ QTCSDL, hệ CSDL. Nêu tên một số hệ QT CSDL có mặt trên thị trường phần mềm hiện nay
* CSDL:
  + Là 1 bộ sưu tập rất lớn (tài nguyên)
  + Bao gồm các loại dữ liệu: âm thanh, hình ảnh, văn bản, tiếng nói, … 🡪gắn kết với nhau 🡪 bảng
  + Được mã hóa thành các chuỗi bit và được lưu trữ dưới dạng file
  + Có 2 loại dữ liệu:
    - Dữ liệu vật lý: thô, thực được nhập vào CSDL
    - Dữ liệu dẫn xuất: dữ liệu được tính toán từ dữ liệu thuộc CSDL.
    - VD: CSDL lưu thông tin về SV, trong đó lưu các điểm thi của SV. Giá trị tb điểm thi của SV sẽ đc tính toán khi cần
* Hệ QTCSDL:
  + Hệ thống phần mềm
  + Cho phép tạo lập và điều khiển mọi truy nhập đến CSDL đó
  + Có 4 đặc tính:
    - Tạo mới CSDL = ngôn ngữ định nghĩa dữ liệu (DDLs)
    - Truy vấn = ngôn ngữ thao tác dữ liệu (DMLs)
    - Hỗ trỡ lưu trữ số lượng lớn trong time dài
    - Kiểm soát truy nhập dữ liệu từ nhiều người tại cùng 1 thời điểm
* Hệ CSDL:
  + là 1 CSDL được quản trị bởi 1 hệ QTCSDL
  + có 4 tp:
    - phần mềm hệ QTCSDL
    - phần cứng
    - CSDL hợp nhất:
      * Tối ưu hóa dư thừa
      * Được chia sẻ
    - Người dùng

- VD về hệ QTCSDL: MS Access, My SQL, SQL server, …

1.4. Trình bày các vai trò người dùng trong CSDL

- Người dùng (Người dùng cuối): là “khách hàng” của CSDL, và có thể được phân thành 2 nhóm dựa theo cách mà họ sd hthong:

+ Nhóm người dùng kb đến k/n CSDL or hệ QTCSDL: Truy nhập CSDL thông qua ct UD đc viết riêng biệt, giúp cho các thao tác của ng dùng đơn giản nhất có thể

+ Nhóm ng dùng nhận biết đc cấu trúc CSDL và các phương tiện đc cung cấp bởi hệ QTCSDL: Thường dùng các ngôn ngữ truy vấn bậc cao như SQL để thực hiện những thao tác đc yêu cầu và thậm trí có thể viết những ct UD để phục vụ cho mục đích riêng.

1.5. Trình bày ưu, nhược điểm của các hệ quản trị CSDL

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| * Kiểm soát dư thừa dữ liệu * Đảm bảo tính nhất quán dữ liệu * Thêm thông tin từ cùng dữ liệu * Hỗ trợ tính sẵn dùng của khối lượng lớn dữ liệu * Chia sẻ dữ liệu * Cải tiến sự toàn vẹn dữ liệu * Cải tiến tính bảo mật dữ liệu * Ép buộc chuẩn hóa dữ liệu * Kinh tế khi tăng số lượng dlieu * Cân = những yêu cầu bị xung đột * Cải tiến vc truy nhập dữ liệu * Tăng hiệu suất của hệ thống * Cải tiến vc bảo trì * Tăng xử lý đồng thời * Cải tiến sao lưu và khôi phục dữ liệu * Cải tiến đáp ứng truy vấn | * Phức tạp * Kích thước lớn * Chi phí mua và bảo trì * Thêm giá thành cho các phần cứng hỗ trợ * Chi phí chuyển đổi hệ thống * Hạn chế hiệu năng (trong một số TH cụ thể) * Ảnh hưởng lớn khi có lỗi |

1.6. Trình bày các khái niệm: lược đồ cơ sở dữ liệu, ánh xạ các loại lược đồ và một thể hiện của cơ sở dữ liệu.

- Lược đồ CSDL:

+ là thiết kế tổng thể của CSDL

+ Có 3 loại lược đồ:

. Lược đồ ngoài: tương ứng với các khung nhìn CSDL

. Lược đồ khái niệm: mô tả thực thể, thuộc tính, các mqh

. Lược đồ trong: đ/n các bản ghi lưu trữ, phương thức biểu diễn, …

*=> Hệ quản trị CSDL có trách nhiệm ánh xạ các loại lược đồ với nhau*

- **Lược đồ khái niệm liên kết với lược đồ trong** thông qua ánh xạ mức khái niệm/mức trong.  
=> Ánh xạ này giúp cho hệ quản trị CSDL tìm ra được bản ghi thực tế hoặc  
kết nối các bản ghi trong bộ lưu trữ vật lý để tạo bản ghi logic trong lược đồ  
khái niệm, cùng với một số ràng buộc được gắn với các thao tác cho bản ghi  
logic đó.

- **Lược đồ ngoài liên kết với lược đồ khái niệm** thông qua ánh xạ mức ngoài/mức khái niệm.  
=> Ánh xạ này giúp cho hệ quản trị CSDL ánh xạ các tên trong khung nhìn  
của người dùng thành các phần có liên quan trong lược đồ khái niệm.

- Một thể hiện CSDL:

+ tập tin được lưu trữ trong CSDL

+ tại 1 thời điểm xđ

1.7: Tính độc lập dữ liệu là gì? Phân loại tính độc lập dữ liệu.

- Tính độc lập dữ liệu: nghĩa là các mức cao hơn không bị ảnh hưởng bởi bất kỳ sự thay đổi nào từ các mức thấp hơn

- Có 2 loại độc lập dữ liệu:

▪ ***Độc lập dữ liệu mức logic (mức khái niệm):*** các lược đồ ngoài  
không bị ảnh hưởng bởi sự thay đổi của lược đồ khái niệm.  
▪ ***Độc lập dữ liệu mức vật lý:*** lược đồ khái niệm không bị ảnh  
hưởng bởi sự thay đổi của lược đồ trong

1.8: Trình bày khái niệm các ngôn ngữ cơ sở dữ liệu.

- Một ngôn ngữ con dữ liệu bao gồm 2 phần: Một ngôn ngữ  
định nghĩa dữ liệu (DDL - Data Definition Language) và một ngôn  
ngữ thao tác dữ liệu (DML - Data Manipulation Language).

- DDL được dùng để xác định lược đồ CSDL, và DML dùng để đọc và cập nhật CSDL.

- Các ngôn ngữ này được gọi là *ngôn ngữ con dữ liệu* vì chúng không bao gồm các cấu trúc lập trình cần thiết cho việc tính toán như là các cấu trúc điều khiển hoặc câu lệnh lặp (được cung cấp bởi các ngôn ngữ lập trình bậc cao)

\* **NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU (DDL):** là ngôn ngữ cho phép người quản trị CSDL hoặc người dùng mô tả và đặt tên các thực thể, thuộc tính và các quan hệ cần thiết cho ứng dụng, cùng với những ràng buộc về bảo mật và toàn vẹn liên quan.

=> Kết quả của việc thực thi/biên dịch câu lệnh DDL là một  
tập các bảng được lưu trong các tệp đặc biệt, được gọi là  
danh mục hệ thống (system catalog) (hay từ điển dữ liệu/ thư  
mục dữ liệu).

\* **NGÔN NGỮ THAO TÁC DỮ LIỆU (DML):** là ngôn ngữ cung cấp một tập các thao tác hỗ trợ cho các phép toán thao tác dữ liệu cơ bản trên dữ liệu được lưu trong CSDL.

- Các thao tác của DML bao gồm:  
▪ Chèn dữ liệu mới vào CSDL  
▪ Sửa đổi dữ liệu đã được lưu trữ trong CSDL  
▪ Lấy dữ liệu từ CSDL  
▪ Xóa dữ liệu trong CSDL

- Phần thao tác lấy dữ liệu ra được gọi là ngôn ngữ truy vấn

- Có 2 loại DML chính:

▪ ***DMLs có thủ tục*** là các ngôn ngữ trong đó người dùng có thông báo với hệ thống những dữ liệu nào cần và cách thức chính xác để lấy nó ra.  
▪ ***DMLs không có thủ tục*** là các ngôn ngữ trong đó người dùng chỉ thông  
báo cho hệ thống dữ liệu nào cần và hệ thống sẽ tự xác định cách thức lấy dữ liệu đó ra

- DMLs có thủ tục thường được nhúng vào các ngôn ngữ lập trình bậc cao.

- DMLs có thủ tục có xu hướng tập trung vào từng bản ghi đơn còn DMLs không thủ tục có xu hướng thực hiện trên một tập các bản ghi.

\* **NGÔN NGỮ THẾ HỆ THỨ TƯ:** là một ngôn ngữ lập trình rất nhanh. Những yêu cầu được thực hiện với hàng trăm dòng lệnh trong ngôn ngữ thế hệ thứ ba sẽ được thể hiện chỉ trong một vài dòng lệnh của ngôn ngữ thế hệ thứ tư.

- Ngôn ngữ thế hệ thứ ba là loại có thủ tục, còn ngôn ngữ thế hệ thứ tư là loại không có thủ tục.

- Ngôn ngữ thế hệ thứ tư gồm các ngôn ngữ làm việc trên bảng tính và trên cơ sở dữ liệu.  
Ví dụ: Ngôn ngữ SQL

2.1: Trình bày khái niệm hệ quản trị cơ sở dữ liệu. Vẽ sơ đồ và mô tả kiến trúc của một hệ quản trị cơ sở dữ liệu.

* Hệ QTCSDL:
  + Hệ thống phần mềm
  + Cho phép tạo lập và điều khiển mọi truy nhập đến CSDL đó
  + Có 4 đặc tính:
    - Tạo mới CSDL = ngôn ngữ định nghĩa dữ liệu (DDLs)
    - Truy vấn = ngôn ngữ thao tác dữ liệu (DMLs)
    - Hỗ trỡ lưu trữ số lượng lớn trong time dài
    - Kiểm soát truy nhập dữ liệu từ nhiều người tại cùng 1 thời điểm
* Sơ đồ mô tả kiến trúc của 1 hệ QT CSDL

Diagram

Description automatically generated

* + 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 của tin, thông tin về lược đồ, thông tin về chỉ mục, ...)
  + ***Bộ quản lý lưu trữ:*** lưu trữ chỉ như là hệ thống tệp trong hệ điều hành và 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:***
    - Xử lý câu truy vấn = ngôn ngữ bậc cao
    - Phần phức tạp nhất 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 tố (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.

2.2: Vẽ sơ đồ và trình bày mô hình trừu tượng ba lớp. Cho ví dụ chuyển đổi lược đồ giữa ba mức trừu tượng.

- Sơ đồ mô hình trừu tượng ba lớp:

Diagram

Description automatically generated

* ***Mức ngoài bao gồm một số khung nhìn khác nhau về CSDL.***
  + Mức ngoài chỉ bao gồm các thực thể, thuộc tính, quan hệ trong thế giới thực mà người dùng quan tâm. (Những thực thể, thuộc tính, quan hệ khác có thể tồn tại, nhưng người dùng sẽ không cần biết đến sự tồn tại của chúng.)
  + Thường xảy ra trường hợp các khung nhìn ngoài khác nhau sẽ  
    có những biểu diễn khác nhau đối với cùng dữ liệu.  
    ***Ví dụ:*** Một khung nhìn có thể thể hiện ngày tháng dưới dạng (ngày, tháng, năm), trong khi một khung nhìn khác lại thể hiện dưới dạng (tháng, ngày, năm).
  + Một số khung nhìn có thể bao gồm các dữ liệu dẫn xuất. Các  
    dữ liệu này sẽ không được lưu thực trong CSDL mà chúng sẽ  
    được tạo ra khi cần.

***Ví dụ:*** Một khung nhìn cần biết tuổi của một người. Tuy nhiên, dữ liệu về tuổi không cần thiết phải được lưu trong CSDL vì nó được cập nhật hàng ngày: Tuổi sẽ được tính dựa theo dữ liệu ngày sinh của người đó và ngày hiện tại trong hệ thống.

* ***Mức khái niệm*** là khung nhìn của người thiết kế CSDL, mô tả dữ liệu nào được lưu trong CSDL và mối quan hệ giữa chúng.
  + Người quản trị CSDL nhìn thấy toàn bộ cấu trúc logic của  
    CSDL. Cấu trúc này thể hiện khung nhìn hoàn chỉnh về những yêu cầu dữ liệu của tổ chức mà không liên quan tới bất kỳ phương thức lưu trữ nào.
  + bất kỳ dữ liệu nào chuyển tới người dùng đều phải được lưu lại hoặc được sinh ra từ mức khái niệm.
  + Mức khái niệm không liên quan tới bất kỳ thông tin nào về việc lưu trữ dữ liệu.
* ***Mức trong*** thể hiện biểu diễn về mặt vật lý của CSDL  
  trong máy tính, mô tả cách thức lưu trữ dữ liệu trong  
  CSDL.
  + Mô tả cài đặt vật lý cần thiết để đạt được tối ưu về thời  
    gian thực thi và việc sử dụng không gian lưu trữ.
  + Bao gồm các cấu trúc dữ liệu và tổ chức tệp lưu trữ dữ  
    liệu trong các thiết bị nhớ.
  + Có giao diện với các phương thức truy nhập của hệ điều  
    hành (các kỹ thuật quản lý tệp để lưu trữ và lấy các bản  
    ghi dữ liệu) để đưa dữ liệu vào các thiết bị nhớ, xây dựng  
    chỉ mục, lấy dữ liệu, ...
* ***Mức vật lý*** nằm dưới mức trong, được quản lý bởi hệ điều hành  
  dưới chỉ dẫn của hệ quản trị CSDL.  
  ❖ Chức năng của hệ quản trị CSDL và hệ điều hành tại mức vật lý là  
  không có ranh giới rõ ràng và thay đổi từ hệ thống này sang hệ  
  thống khác.  
  ❖ Một số hệ quản trị CSDL tận dụng ưu điểm của nhiều phương thức  
  truy nhập của hệ điều hành, trong khi một số hệ quản trị CSDL khác  
  lại chỉ sử dụng những phương thức cơ bản và tự tạo ra tổ chức tệp  
  của riêng chúng.  
  ❖ Mức vật lý dưới hệ quản trị CSDL gồm các mục chỉ được biết đến  
  bởi hệ điều hành. Ví dụ: làm thế nào để tạo một chuỗi các thực thi? Liệu các trường của các bản ghi trong CSDL có được lưu trữ bởi các byte liền nhau trong ổ đĩa hay không?
* ***Ví dụ chuyển đổi lược đồ giữa ba mức trừu tượng***

Timeline

Description automatically generated

2.3: Trình bày cách phân loại các hệ cơ sở dữ liệu dựa theo kiến trúc

***Có 2 loại kiến trúc CSDL:***❖ Hệ cơ sở dữ liệu tập trung:  
▪ Hệ CSDL cá nhân  
▪ Hệ CSDL trung tâm  
▪ Hệ CSDL khách/chủ (client/server)  
❖ Hệ cơ sở dữ liệu phân tán  
▪ Hệ CSDL phân tán thuần nhất  
▪ Hệ CSDL phân tán không thuần nhất

* **Hệ CSDL tập trung**:
  + Là các hệ CSDL mà trong đó phần CSDL được lưu trữ tại một vị trí nhất định.
  + Người dùng tại các trạm từ xa nói chung có thể truy nhập CSDL thông qua các công cụ truyền thông dữ liệu.
  + Cung cấp một sự kiểm soát lớn đối với việc truy nhập và cập nhật dữ liệu.
  + Dễ bị lỗi do phụ thuộc vào tính sẵn sàng của các tài nguyên.
* **Hệ CSDL cá nhân**
  + Là hệ CSDL nhỏ, trong đó người quản trị CSDL chính là người  
    viết chương trình ứng dụng, đồng thời cũng là người dùng  
    cuối.
  + Ứng dụng: trong các tổ chức nhỏ và vừa

ví dụ: quản lý nhân sự ở một đơn vị hành chính.

* + Việc phát triển và sử dụng các hệ CSDL cá nhân là khá đơn giản và dễ dàng.
  + Có nguy cơ phải chịu rủi ro, vì CSDL chỉ được lưu trữ tại một trạm đơn lẻ.
  + Dữ liệu khó được chia sẻ cho nhiều ứng dụng khác nhau.
* **Hệ CSDL trung tâm**
  + Trong các tổ chức lớn, dữ liệu mà hầu hết các ứng dụng có thể  
    truy nhập được lưu trữ trên một máy tính trung tâm.
  + Người dùng từ xa có thể truy nhập CSDL này thông qua các  
    thiết bị đầu cuối và các kết nối truyền thông dữ liệu.
  + CSDL trung tâm thường lưu trữ các CSDL tích hợp rất lớn và  
    được nhiều người dùng truy nhập.
  + Việc sử dụng thường có cường độ lớn với hàng trăm đến hàng  
    nghìn giao dịch trong một giây.

Ví dụ: hệ thống đặt vé máy bay, hoặc các hệ thống ngân hàng.

* **Hệ CSDL Khách/ Chủ**
  + Mục đích chính của kiến trúc khách/chủ là cho phép các  
    ứng dụng máy khách truy nhập dữ liệu được quản lý bởi  
    máy chủ. Giao diện người dùng và logic của chương trình  
    ứng dụng được xử lý bên máy khách, trong khi xử lý CSDL  
    được thực hiện bên máy chủ.
  + Máy chủ không cần có cấu hình quá mạnh như trong các  
    hệ CSDL trung tâm.
* **Hệ CSDL phân tán**
  + Hiện nay, nhiều tổ chức phân bố trên nhiều vị trí địa lý  
    khác nhau (các thành phố hay các quốc gia khác nhau).  
    => Việc xây dựng các hệ CSDL tập trung là không thực tế  
    và không kinh tế.
  + CSDL phân tán là một CSDL logic đơn lẻ được trải ra về  
    mặt vật lý trên nhiều máy tính ở nhiều vị trí địa lý khác  
    nhau.

Ví dụ: CSDL của một ngân hàng sẽ được phân bổ theo các  
chi nhánh tại từng địa phương.

* **Hệ CSDL phân tán thuần nhất**
  + Các hệ điều hành máy tính tại mỗi vị trí địa lý là như nhau  
    hoặc có khả năng tương thích cao.
  + Các mô hình dữ liệu được sử dụng tại mỗi vị trí địa lý là  
    như nhau. Mô hình quan hệ được sử dụng chung nhất đối  
    với các hệ CSDL phân tán ngày nay.
  + Các hệ quản trị CSDL được sử dụng tại mỗi vị trí địa lý là  
    như nhau hoặc có khả năng tương thích cao.
  + Dữ liệu tại các vị trí khác nhau có định nghĩa và khuôn  
    dạng chung.
* **Hệ CSDL phân tán không thuần nhất**
  + Các máy tính khác nhau và các hệ điều hành khác nhau có  
    thể được sử dụng tại các vị trí địa lý khác nhau.
  + Các mô hình dữ liệu khác nhau và các hệ quản trị CSDL  
    khác nhau cũng có thể được sử dụng.
  + Dữ liệu trên các vị trí thường không tương thích. Có khác  
    biệt về cú pháp và ngữ nghĩa.

*=> Khi có nhu cầu chia sẻ dữ liệu, giải pháp là phát triển  
một hệ CSDL mới hoàn toàn, hợp nhất tất cả các hệ CSDL  
đang tồn tại. Tuy nhiên, giải pháp này rất khó khăn về mặt  
kỹ thuật và kinh tế.*

2.4: Vẽ sơ đồ và trình bày các bước chính trong quá trình thiết kế một cơ sở dữ liệu.

**-** Sơ đồ:

Diagram

Description automatically generated

* Các bước chính trong quá trình thiết kế một cơ sở dữ liệu.

1. Phân tích yêu cầu
   * + Dữ liệu nào đc lưu trữ?
     + UD nào đc sd?
     + Thao tác nào sd thường xuyên
2. Thiết kế CSDL mức k/n: ttin 🡪 phát triển thành 1 mô tả tổng quát dữ liệu có ràng buộc cần thiết với nhau
3. Thiết kế CSDL mức logic: 1 hệ QT CSDL 🡪 chọn 🡪 cài đặt (mức k/n 🡪 lược đồ CSDL)
4. Cải tiến lược đồ: Các lược đồ đc phát triển => phân tích vấn đề tiềm ẩn => giải quyết => chuẩn hóa CSDL
5. Thiết kế CSDL mức vật lý:
   * + Khối lượng cvc tiêm ẩn
     + Phương pháp truy nhập đc mô phỏng

* Giải quyết => tiềm ẩn dlieu

1. Thiết kế an toàn bảo mật: ? giải quyết hacker

* Nội dung 2: Các mô hình dữ liệu

1.9: Trình bày khái niệm mô hình dữ liệu và các thành phần trong mô hình dữ liệu. Phân loại mô hình dữ liệu.

* ***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** bao gồm 3 thành phần:  
  1. Phần cấu trúc: tập các luật xd CSDL  
  2. Phần thao tác: cập nhật, lấy dữ liệu và thay đổi cấu trúc của CSDL  
  3. Có thể có một tập các luật về tính toàn vẹn, nhằm đảm bảo CSDL luôn chính xác.
* Có 3 loại mô hình:
  + MH dlieu ngoài: bdien khung nhìn của ng dùng
  + MH dlieu k/n: bdien k/n mức logic
  + MH dlieu trong: bdien lược đồ k/n mà hệ CSDL hiểu đc

1.10: Trình bày các khái niệm thực thể, tập thực thể, thuộc tính, miền giá trị của thuộc tính.

* Thực thể:
  + Là một đối tượng trong thể giới thực, có thể phân biệt đc với các đối tượng khác.
  + Thực thể có thể cụ thể (một ng, một quyển sách, …) or cũng có thể trừu tượng (một khoản vay ngân hàng, một k/n, …)
  + Thực thể đc biểu diễn bởi một tập các thuộc tính
* Tập thực thể:
  + Là một nhóm các thực thể có cùng thuộc tính
  + Các tập thực thể k nhất thiết phải tách rời nhau
* Miền giá trị: là một tập giá trị cho phép của mỗi thuộc tính

▪ Một thuộc tính của một tập thực thể là một hàm ánh xạ từ một tập thực thể vào một miền giá trị.  
▪ Một tập thực thể có thể có nhiều thuộc tính.  
=> Mỗi thực thể trong tập có thể được mô tả bởi một tập các cặp *<thuộc tính, giá trị>*, ứng với từng thuộc tính trong tập thực thể.

1.11: Cho một ví dụ về mô hình thực thể liên kết E-R và giải thích các thành phần trong đó.

Diagram

Description automatically generated

* Tập thực thể: CUSTOMER, LOAN
* Thuộc tính: borrower
* Thuộc tính của CUSTOMER: khóa chính: customer-id; customer-name; customer-street; customer-city
* Thuộc tính của LOAN: khóa chính: loan-id; amount.
* CUSTOMER và LOAN là liên kết nhiều nhiều (M:M)

1.12: Trình bày các khái niệm: thuộc tính đơn, thuộc tính kép, thuộc tính đơn trị, thuộc tính đa trị, thuộc tính dẫn xuất, thuộc tính rỗng (null).

* Thuộc tính đơn không bao gồm các thành phần cấu thành
* Thuộc tính kép bao gồm các thành phần con cấu thành.

*Ví dụ: thuộc tính* ***tên****: Nếu tên biểu diễn một thuộc tính đơn thì có thể  
coi bộ ba cấu thành* ***tên*** *là* ***họ****,* ***tên đệm*** *và* ***tên gọi*** *là một thuộc tính  
nguyên tố, không phân chia được nữa. Còn nếu coi* ***tên*** *là một thuộc  
tính kép thì có thể lựa chọn thao tác với thuộc tính này là một tên đầy  
đủ hoặc có thể thao tác với từng thành phần cấu thành tên*

* 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.

*Ví dụ: Tại một trường học sinh viên được đăng ký học theo tín chỉ.  
Tại một kỳ học nào đó, số tín chỉ một sinh viên đăng ký là đơn trị, ví  
dụ là 7 (tín chỉ) => số tín chỉ không thể nhận giá trị đa trị.  
Thuộc tính số điện thoại của sinh viên có thể chứa nhiều giá trị  
cùng lúc do tại một thời điểm, một sinh viên có thể có một vài số  
điện thoại khác nhau. => thuộc tính số điện thoại là đa trị.*

* 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.

*Ví dụ: Giả sử thực thể* ***KHÁCH HÀNG*** *của một ngân hàng có một  
thuộc tính tên là* ***loans-held****, chứa số lượng các khoản vay của một  
khách hàng tại ngân hàng. Giá trị của thuộc tính này có thể được  
tính bằng cách đếm số lượng thực thể các khoản vay liên quan tới  
khách hàng.*

* Thuộc tính rỗng (Null): thuộc tính nhận giá trị rỗng khi một thực thể không có giá trị cho nó.

1.13: Trình bày về ràng buộc ánh xạ lực lượng liên kết trong mô hình quan hệ thực thể ER

❖ Ánh xạ lực lượng liên kết thể hiện số lượng các thực thể mà một thực thể khác có thể liên hệ thông qua một tập quan hệ.  
❖ Ràng buộc ánh xạ lực lượng liên kết có ích nhất khi mô tả các quan hệ hai ngôi.  
❖ Với một tập quan hệ hai ngôi *R* giữa tập thực thể *A* và *B*, ánh xạ lực lượng liên kết gồm các loại sau:  
▪ *(1:1) một tới một từ A đến B*▪ *(1:M) một tới nhiều từ A đến B*▪ *(M:1) nhiều tới một từ A đến B*▪ *(M:M) nhiều tới nhiều từ A đến B*

Diagram

Description automatically generated Diagram

Description automatically generated

Diagram

Description automatically generated Diagram

Description automatically generated

1.14: Trình bày về ràng buộc tham gia trong mô hình quan hệ thực thể E-R

* 1 Tập thực thể E 🡪 Tham gia 🡪 1 tập các quan hệ R được gọi là đầy đủ (total): nếu mọi thực thể của E tham gia vào ít nhất 1 quan hệ trong R
* Nếu chỉ có một vài thực thể của *E* tham gia vào một quan hệ trong *R,* thì sự tham gia của tập thực thể *E* trong tập các quan hệ *R* được gọi là một phần (partial).
* Ví dụ, xem xét hệ thống ngân hàng.  
  ▪ Mỗi thực thể **LOAN** (khoản vay) có liên hệ tới ít nhất một **CUSTOMER** (khách hàng) thông qua mối quan hệ ***borrower***(vay mượn) => sự tham gia của thực thể LOAN trong tập quan  
  hệ ***borrower*** là **đầy đủ**.  
  ▪ Ngược lại, có những khách hàng không liên quan tới các khoản  
  vay mượn ngân hàng. Như vậy, có thể nói rằng chỉ một số các  
  thực thể **CUSTOMER** có liên hệ tới một thực thể **LOAN** thông  
  qua quan hệ ***borrower*** => sự tham gia của thực thể  
  **CUSTOMER** trong tập quan hệ ***borrower*** là **một phần**

1.15: Trình bày các khái niệm siêu khóa, khóa dự bị, khóa chính của một tập thực thể

* 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.  
  VD: Có một tập thực thể sinh viên trong một lớp học với lược đồ sau:  
  STUDENTS(SS#, name, address, age, major, minor, gpa, spring-sch)

Siêu khóa cho tập thực thể STUDENTS có thể là:  
(SS#, name, major, minor) hoặc (SS#, name)  
*=> 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.

1.16: Khóa và siêu khóa khác nhau như thế nào? Cho ví dụ và giải thích.

* Siêu khóa: là một tập thuộc tính cho phép xác định duy nhất một thực thể trong tập thực thể
* Khóa: là siêu khóa mà khi loại bỏ bất kì thuộc tính nào từ khóa này thì nó không còn là một siêu khóa nữa. Khóa có số thuộc tính là nhỏ nhất
* Ví dụ: Có một tập thực thể sinh viên trong một lớp học với lược đồ sau:  
  STUDENTS(SS#, name, address, age, major, minor, gpa, spring-sch)

Siêu khóa cho tập thực thể STUDENTS có thể là: (SS#, name, major, minor) hoặc (SS#, name)

Khóa của tập thực thể là (SS#)

1.17: Trình bày các khái niệm tập thực thể mạnh và tập thực thể yếu. Cho một ví dụ cụ thể.

* Tập thực thể mạnh: là tập thực thể có khóa chính
* Tập thực thể yếu: là tập thực thể k đủ các thuộc tính để hình thành một khóa chính
* Ví dụ: Xét tập thực thể trả tiền:

PAYMENT (payment-number, payment-date, payment-amount)

* Mã số trả tiền (**payment-number**) thường là các số liên tiếp, bắt đầu từ 1 và đc sinh ra riêng rẻ cho mỗi khoản nợ. Do đó, mặc dù mỗi thực thể **PAYMENT** là khác nhau, việc trả tiền cho các khoản nợ khác nhau có thể có cùng mã số **payment-number**
* Tập **PAYMENT** k có khóa chính và chỉ là một tập thực thể yếu

Khóa chính của tập thực thể PAYMENT sẽ là (loan-number, payment-number):

* + Loan-number: là khóa chính của tập thực thể LOAN
  + Payment-number: là thuộc tính phân biệt của tập thực thể yếu PAYMENT
* Tập PAYMENT là một tập thực thể mạnh

Diagram

Description automatically generated

1.18: Trình bày các khái niệm cụ thể hóa và tổng quát hóa. Cho một ví dụ cụ thể

* Cụ thể hóa là quá trình thiết kế các phân nhóm trong một tập thực thể. Nghĩa là, phân biệt các tập thực thể con trong một tập thực thể khi chúng có các thuộc tính không giống nhau

Ví dụ: Xét tập thực thể người: PERSON(name, street, city).  
Một **PERSON** có thể được phân chia nhỏ hơn thành **STUDENT**(sinh viên) hoặc **INSTRUCTOR** (giảng viên). Mỗi loại này được mô  
tả bởi một tập các thuộc tính bao gồm tất cả các thuộc tính của tập  
thực thể **PERSON**, và một số thuộc tính bổ sung riêng.  
▪ Thực thể **STUDENT** có thể bổ sung thêm các thuộc tính ***gpa*** và  
***credit-hours-earned****;*▪ Thực thể **INSTRUCTOR** bổ sung thêm các thuộc tính ***salary*** và  
***years-employed.***

* Tổng quát hóa là nhiều tập thực thể được đồng bộ vào một tập thực thể ở mức cao hơn trên cơ sở các thuộc tính chung.
* Ví dụ: PERSON là tập thực thể ở mức cao (gọi là *cha*), và INSTRUCTOR và STUDENT là các tập thực thể ở mức thấp  
  (gọi là *con*).
  + Đầu tiên có thể xác định 2 tập thực thể:  
    STUDENT*(name, address, city, gpa, credit-hours-earned),*

INSTRUCTOR*(name, address, city, salary, yearsemployed)*.

* + Sau đó, tìm điểm chung giữa các thuộc tính để xây dựng  
    tập thực thể: PERSON *(name, address, city).*

1.19: Trình bày cơ chế kế thừa thuộc tính trong cụ thể hóa/tổng quát hóa và lợi ích của nó.

❖ Các tập thực thể ở mức thấp hơn kế thừa các thuộc tính từ tập thực thể ở mức cao hơn.  
Ví dụ: **INSTRUCTOR** và **STUDENT** đều kế thừa tất cả các thuộc tính của **PERSON**.  
❖ Tập thực thể mức thấp hơn cũng kế thừa các mối quan hệ thuộc về tập thực thể mức cao hơn định nghĩa nó.  
❖ Cây phân cấp các tập thực thể:  
• Thực thể ở mức cao nhất nằm trên đỉnh của cây phân cấp.  
• Kế thừa đơn: Tập thực thể ở mức thấp hơn có một mối quan hệ ISA.  
• Đa kế thừa: Tập thực thể mức thấp có nhiều hơn một mối quan hệ ISA.

1.20: Trình bày khái niệm tập thực thể kết hợp. Cho một ví dụ và giải thích

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

- Ví dụ:

**Diagram

Description automatically generated**

Có thể coi tập quan hệ **works-on** (liên quan tới các tập thực thể **EMPLOYEE**, **BRANCH** và **JOB**) như một tập thực thể ở mức cao, được đặt tên là **WORKS-ON**.  
Tạo một quan hệ hai ngôi **manages** giữa **WORKS-ON** và  
**MANAGER** để thể hiện ai quản lý các công việc này.

1.21: Trình bày khái niệm ràng buộc toàn vẹn tham chiếu và các phương pháp được sử dụng để đảm bảo tính ràng buộc toàn vẹn tham chiếu.

❖ Ràng buộc toàn vẹn tham chiếu có thể được hiểu đơn giản là việc đảm bảo một thuộc tính nào đó có một giá trị khác rỗng. Tuy nhiên, các ràng buộc toàn  
vẹn tham chiếu thường liên quan tới các quan hệ giữa các tập thực thể.  
❖ Ràng buộc toàn vẹn tham chiếu yêu cầu mỗi thực thể “được tham chiếu tới” bởi một quan hệ phải tồn tại trong cơ sở dữ liệu.

❖ Các phương pháp được sử dụng để đảm bảo tính ràng buộc toàn vẹn tham chiếu:  
• Không được phép xóa bỏ một thực thể được tham chiếu đến.  
• Nếu một thực thể được tham chiếu bị xóa bỏ thì tất cả các bản ghi tham chiếu tới thực thể đó cũng bị xóa.

1.22: Trình bày các khái niệm: quan hệ, thuộc tính, miền giá trị của thuộc tính, bộ (bản ghi), bậc, lực lượng trong một quan hệ. Cho ví dụ cụ thể.

❖ Quan hệ: 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: 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: 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.

- Ví dụ:

**Diagram, table

Description automatically generated**

1.23: Trình bày các đặc tính của một quan hệ.

1. Quan hệ có một tên gọi phân biệt với tên của các quan  
hệ khác trong lược đồ quan hệ.  
2. Mỗi thuộc tính có một tên gọi riêng.  
3. Mỗi thuộc tính có một miền giá trị.  
4. Mỗi thuộc tính chứa một giá trị nguyên tố.  
5. Các bộ là phân biệt nhau (không có hai bộ nào giống hệt nhau).  
6. Thứ tự của các thuộc tính không quan trọng.  
7. Thứ tự của các bộ cũng không quan trọng (về mặt lý thuyết).

=> Tuy nhiên, trong thực tế, thứ tự này có thể ảnh hưởng đến hiệu  
quả truy nhập vào các bộ

**Table

Description automatically generated**

1.24: Giải thích tại sao các bộ trong một quan hệ không cần thiết phải được sắp xếp theo một trình tự nhất định nào đó?

Bộ là một hàng của một quan hệ. Trong một quan hệ các bộ với các thuộc tính không thể trùng nhau

1.25: Giải thích tại sao trong một quan hệ không cho phép các bộ trùng nhau?

Giữa các thực thể dữ liệu tồn tại các mqh, ràng buộc lẫn nhau. Mục đích vc xây dựng các ràng buộc dữ liệu là nhằm đảm bảo tính độc lập và toàn vẹn dữ liệu. Dữ liệu lưu trữ trong CSDL hiện thực khách quan, không dư thừa và mâu thuẫn thông tin.

1.26: Trình bày những ưu điểm của mô hình quan hệ.

* Tính đơn giản: Các ttin về các thực thể và các ràng buộc của các thực thể đc bdien duy nhất là các bảng
* Tính độc lập dữ liệu: thể hiện đc tính độc lập dữ liệu ở mức vật lý và tính độc lập dữ liệu ở mức logic
* Tính đối xứng: Do cấu trúc bdien dlieu trong các hệ CSDL quan hệ, kết quả cảu các câu hỏi cũng là quan hệ đối xứng nhau
* Có cơ sở lí thuyết vững chắc: Khi nghiêm cứu của thiết kế và cài đặt các hệ CSDL quan hệ có nên tảng lí thuyết vững chắc, chặt chẽ và logic

2.5: Trình bày ba loại ràng buộc chính trong quá trình cụ thể hóa/tổng quát hóa

* ***Ràng buộc thứ nhất:*** Xác định thực thể nào có thể là thành viên của tập thực thể mức thấp hơn.  
  Thành viên được định nghĩa theo một trong 2 cách:
  + Mệnh đề xác định: Thành viên được đánh giá trên cơ sở xác định xem thực thể có thỏa mãn một mệnh đề (điều kiện) tường minh nào đó không.
  + Người dùng xác định: Tập các thực thể mức thấp do người dùng xác định không bị ràng buộc bởi các điều kiện thành viên mà người dùng cơ sở dữ liệu gán các thực thể tới tập thực thể nào đó.
* ***Ràng buộc thứ hai:*** Trong quá trình tổng quát hóa, liệu các thực thể có thuộc vào nhiều hơn một tập thực thể ở mức thấp hơn không.  
  Các tập thực thể ở mức thấp hơn có thể:
  + Không giao nhau: Một thực thể không thể thuộc vào nhiều tập thực thể ở mức thấp hơn. (Trong mô hình E-R, ràng buộc không giao nhau được thể hiện bằng từ “disjoint” đặt cạnh biểu tượng hình tam giác).
  + Giao nhau: Một thực thể có thể thuộc vào nhiều tập thực thể ở mức thấp hơn trong cùng một quá trình tổng quát hóa.
* ***Ràng buộc thứ ba:*** Là ràng buộc dựa trên tính toàn bộ, xác định xem một thực thể trong tập thực thể ở mức cao có thuộc vào một trong các tập thực thể ở mức thấp hơn hay không.  
  Có 2 loại ràng buộc này:  
  • Tổng quát hóa/cụ thể hóa toàn bộ: mỗi thực thể ở mức cao phải thuộc vào một thực thể ở mức thấp hơn.  
  • Tổng quát hóa/cụ thể hóa một phần: Tồn tại thực thể ở mức cao không thuộc vào tập thực thể nào ở mức thấp hơn. (Đây là trường hợp mặc định).

2.6: Trình bày sự tương ứng giữa sơ đồ ER và sơ đồ lớp UML về: các tập thực thể và thuộc tính, các quan hệ, các ràng buộc về lực lượng liên kết, tổng quát hóa và cụ thể hóa.

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**A picture containing diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

2.7: Trình bày các trường hợp mà thuộc tính của tập mối quan hệ hai ngôi có thể được đặt vào trong các tập thực thể tham gia liên kết. Cho ví dụ cụ thể với mỗi trường hợp.

* Các thuộc tính của một tập quan hệ dạng 1:1 hoặc 1:M thường được đặt vào trong các tập thực thể tham gia liên kết, hơn là được đặt vào bản thân tập các mối quan hệ đó.

**Diagram

Description automatically generated**

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

**Diagram

Description automatically generated**

=> Vì một tài khoản cụ thể thuộc sở hữu bởi nhiều nhất là một khách hàng, và tài khoản đó có thể có nhiều nhất một ***accessdate***.

- Xét trường hợp: một tài khoản được làm chủ bởi nhiều nhất một khách hàng và một khách hàng chỉ có thể sở hữu duy nhất một tài khoản.

**Diagram

Description automatically generated**

=> Thuộc tính ***access-date*** có thể gắn vào hoặc thực thể **CUSTOMER** hoặc tập thực thể **ACCOUNT** mà không làm tổn thất thông tin.

- Nếu thuộc tính ***access-date*** được lưu trữ với tập **CUSTOMER** thì nó phải tham chiếu tới lần truy nhập cuối cùng của khách hàng tới tài khoản duy nhất mà họ có.

**Diagram

Description automatically generated**

**-** Nếu thuộc tính ***access-date*** được lưu trữ trong thực thể **ACCOUNT** thì nó sẽ tham chiếu tới lần truy nhập cuối cùng tới tài khoản bởi người khách hàng duy nhất sở hữu nó.

**Diagram

Description automatically generated**

**-** Xét trường hợp: quan hệ ***depositor*** có ràng buộc N:N

**Diagram

Description automatically generated**

=> Việc gắn thuộc tính ***access-date*** với bất kỳ tập thực thể tham gia nào cũng không mô hình hóa được tình huống này mà không làm tổn thất thông tin.

⇒ Nếu cần lưu trữ ngày truy nhập cuối cùng của một khách hàng cụ  
thể tới một tài khoản cụ thể thì thuộc tính ***access-date*** nhất thiết  
phải là một thuộc tính của tập quan hệ ***depositor***, chứ không thể  
là thuộc tính của bất kỳ tập thực thể tham gia nào.  
 + Nếu ***access-date*** là một thuộc tính của **ACCOUNT** thì không  
thể xác định được khách hàng nào đã thực hiện việc chuyển  
tiền vào tài khoản đó.  
 + Còn nếu ***access-date*** là một thuộc tính của **CUSTOMER**, thì  
cũng không thể xác định được tài khoản nào khách hàng đã  
truy nhập vào lần cuối

2.8: Khi nào khái niệm thực thể yếu có ích trong quá trình mô hình hóa dữ liệu? Trình bày các định nghĩa tập thực thể xác định (còn gọi là tập thực thể mạnh hay tập thực thể sở hữu), tập thực thể yếu, tập mối quan hệ xác định, và khóa bán phần (hay thuộc tính phân biệt). Cho ví dụ cụ thể.

* Tập thực thể yếu: là tập thực thể k đủ các thuộc tính để hình thành một khóa chính
* Ví dụ: Xét tập thực thể trả tiền:

PAYMENT (payment-number, payment-date, payment-amount)

* Mã số trả tiền (**payment-number**) thường là các số liên tiếp, bắt đầu từ 1 và đc sinh ra riêng rẻ cho mỗi khoản nợ. Do đó, mặc dù mỗi thực thể **PAYMENT** là khác nhau, việc trả tiền cho các khoản nợ khác nhau có thể có cùng mã số **payment-number**
* Tập **PAYMENT** k có khóa chính và chỉ là một tập thực thể yếu
* Tập thực thể xác định hay tập thực thể sở hữu: là một tập thực thể có ý nghĩa, nó phải liên hệ với một tập thực thể khác
* Mqh xác định: là mqh giữa tập thực thể yếu với tập thực thể xác định
* Thuộc tính phân biệt hay khóa bán phần: là tập các thuộc tính của một thực thể yếu cho phép phân biệt các thực thể
* Ví dụ: Thuộc tính phân biệt của tập thực thể yếu PAYMENT là payment-id, vì với mỗi khoản nợ, một mã số trả tiền sẽ xác định duy nhất một lần trả tiền riêng biệt cho khoản nợ này

* Nội dung 3: LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ

Câu 1: Mục đích của việc thiết kế CSDL quan hệ là gì?

Mục đích của việc thiết kế CSDL quan hệ: nhóm các thuộc tính vào thành các quan hệ sao cho tối thiểu hóa sự dư thừa dữ liệu => giảm không gian lưu trữ và tránh dị thường thông tin khi cập nhật dữ liệu.

Câu 1.27: Trình bày khái niệm phụ thuộc hàm và các đặc tính có ích của phụ thuộc hàm cho việc chuẩn hoá dữ liệu

***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.**

***Các đặc tính có ích của các phụ thuộc hàm cho việc chuẩn hóa:***

* Tồn tại mối quan hệ 1:1 giữa các thuộc tính trong đối tượng xác định và đối tượng hệ quả.
* Phụ thuộc hàm là bất biến theo thời gian, nghĩa là nó thỏa mãn tất cả các thể hiện có thể của quan hệ.
* Các phụ thuộc hàm là không hiển nhiên. Tất cả các phụ thuộc hàm hiển nhiên đúng đều được bỏ qua.

Câu 1.28: Trình bày 6 luật suy diễn phổ biến nhất cho các phụ thuộc hàm

IR1: Luật phản xạ: nếu X ⊇ Y, thì X → Y  
IR2: Luật tăng trưởng: nếu X → Y, thì XZ → YZ  
IR3: Luật bắc cầu: nếu X → Y và Y → Z, thì X → Z

* *3 luật suy diễn này là Hệ tiên đề Amstrong: đóng vai trò là một tập luật cần thiết và đầy đủ cho việc tạo ra bao đóng của một tập các phụ thuộc hàm.*

IR4: Luật chiếu:  
Nếu X → YZ, thì X → Y và X → Z  
IR5: Luật cộng thêm:  
Nếu X → Y và X → Z, thì X → YZ  
IR6: Luật giả bắc cầu:  
Nếu X → Y và YZ → W, thì XZ → W

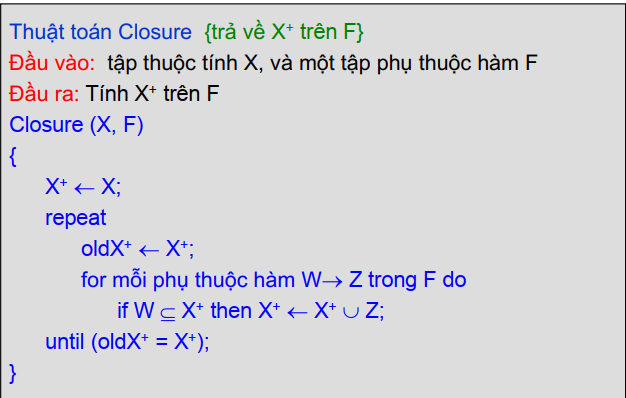
Câu 1.29: Trình bày các luật suy diễn trong hệ tiên đề Amstrong. Giải thích tính đúng đắn và đầy đủ của các hệ tiên đề

IR1: Luật phản xạ: nếu X ⊇ Y, thì X → Y  
IR2: Luật tăng trưởng: nếu X → Y, thì XZ → YZ  
IR3: Luật bắc cầu: nếu X → Y và Y → Z, thì X → Z

* *3 luật suy diễn này là Hệ tiên đề Amstrong: đóng vai trò là một tập luật cần thiết và đầy đủ cho việc tạo ra bao đóng của một tập các phụ thuộc hàm.*

Câu 1.30: Trình bày các định nghĩa: bao đóng của một tập các phụ thuộc hàm và bao đóng của một tập các thuộc tính. Trình bày thuật toán tính bao đóng

* 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+
* X+ 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



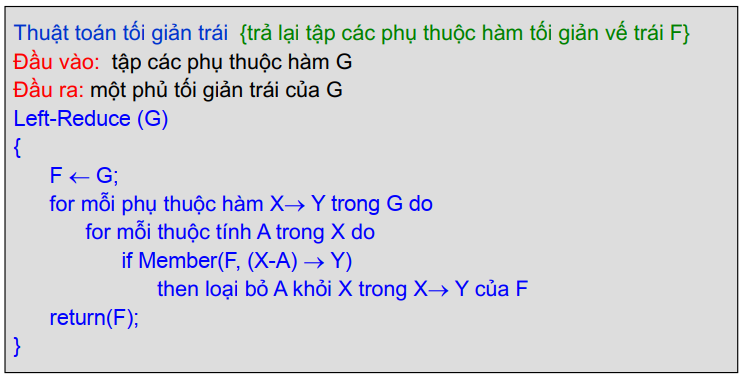
Câu 1.33: Trình bày các khái niệm: các thuộc tính dư thừa, phụ thuộc hàm tối giản vế trái và phụ thuộc hàm tối giản vế phải

\* Các thuộc tính dư thừa:

Nếu F là tập các phụ thuộc hàm trên lược đồ quan hệ R và X→Y ∈ F thì thuộc tính A được gọi là dư thừa trong X→Y ∈ F nếu:  
1. X = AZ, X ≠ Z and {F – {X → Y}} ∪ {Z → Y} ≡ F, hoặc  
2. Y = AW, Y ≠ W and {F – {X → Y}} ∪ {X → W} ≡ F  
⇒ ***Một thuộc tính A dư thừa trong X***→***Y nếu A có thể được loại bỏ khỏi vế trái hoặc vế phải của phụ thuộc hàm mà không làm thay đổi F+.***

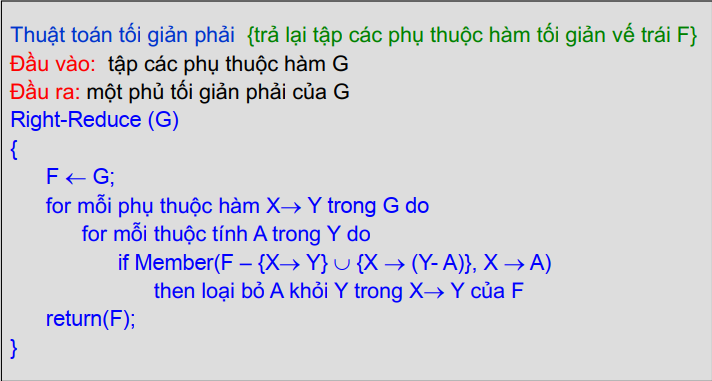
\* Phụ thuộc hàm tối giản vế trái:

X → Y được gọi là tối giản trái nếu X không chứa các thuộc tính dư thừa A



\* Phụ thuộc hàm tối giản vế phải:

X → Y được gọi là tối giản phải nếu Y không chứa các thuộc tính dư thừa A



***X → Y được gọi là tối giản nếu nó tối giản trái, tối giản phải và Y khác rỗng.***

**CHƯƠNG 7: CHUẨN HOÁ**

Câu 1:

***Chuẩn hóa*** là một kỹ thuật chính thức dùng cho việc phân tích các quan hệ dựa trên khóa chính (hoặc khóa dự bị), các thuộc tính và các phụ thuộc hàm.

* Chuẩn hóa về dạng 1NF: loại bỏ dữ liệu dư thừa
* Chuẩn hóa về 2NF: loại bỏ các phụ thuộc hàm bộ phận
* Chuẩn hóa về 3NF: loại bỏ các phụ thuộc hàm bắc cầu

***Các yêu cầu chuẩn hoá:***

Đối với mô hình quan hệ, một vấn đề rất quan trọng và thiết yếu là phải nhận ra được một quan hệ vừa tạo ra đã ở dạng chuẩn 1 (1NF) hay chưa. Tất cả các dạng chuẩn ở mức cao hơn sau đó là tùy theo từng trường hợp, có thể có hoặc không

Người thiết kế CSDL thường được khuyến nghị là phải đưa toàn bộ các quan hệ trong CSDL về ít nhất là dạng chuẩn 3 (3NF)

Việc chuẩn hóa sẽ được thực hiện từ dạng chưa chuẩn hóa, đưa về dạng chuẩn 1, sau đó đưa về dạng chuẩn 2, dạng chuẩn 3, … (đến các dạng chuẩn ở các mức cao hơn)

***Dạng chưa chuẩn hoá:***

Các quan hệ ở dạng chưa chuẩn hóa đồng nghĩa với việc chúng chưa ở dạng chuẩn 1.  
- ***Các quan hệ chưa ở dạng chuẩn 1*** chứa một hoặc một số thuộc tính không nguyên tố, các thuộc tính lặp, và các thuộc tính dẫn xuất.  
- ***Thuộc tính chứa giá trị nguyên tố:*** là những thuộc tính chứa giá trị đơn và không thể phân rã được nữa.

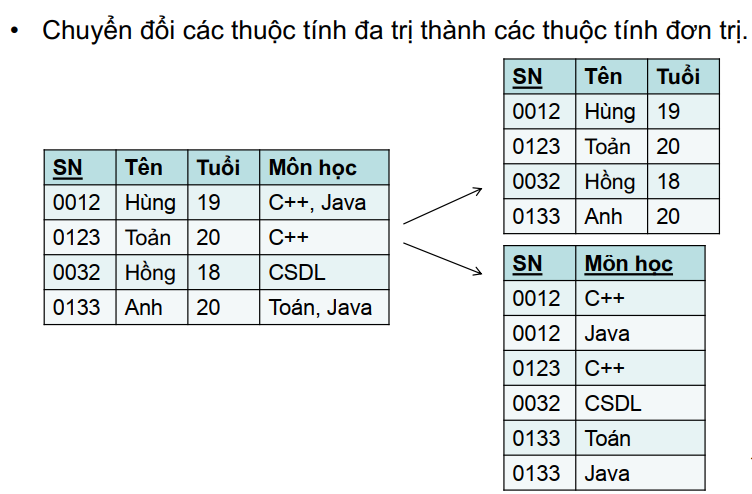
Một số khái niệm về khoá:

- ***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:*** 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.

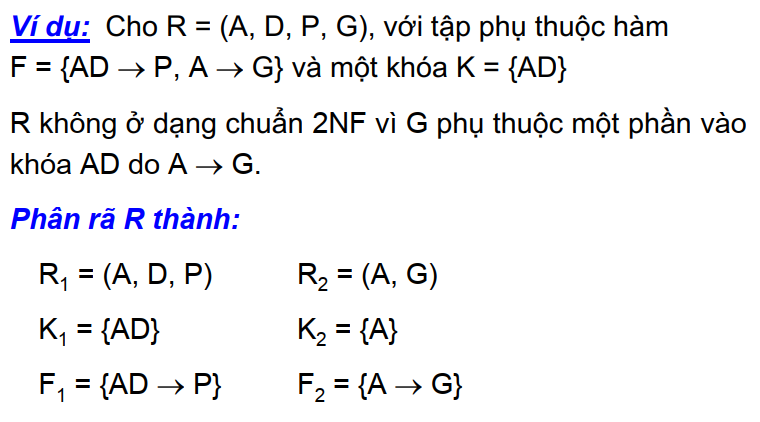
Câu 1.34: Trình bày định nghĩa dạng chuẩn 1(1NF), dạng chuẩn 2(2NF), dạng chuẩn 3(3NF). Cho ví dụ về các dạng chuẩn này

***\*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

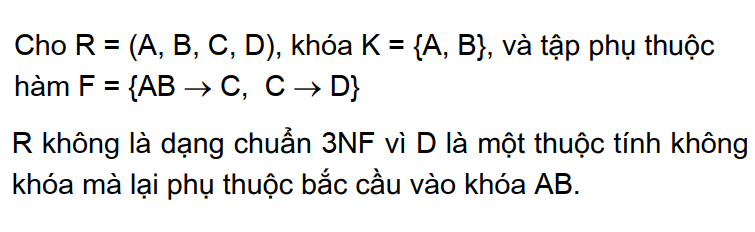


***\* Định nghĩa dạng chuẩn 2:*** Một lược đồ quan hệ R với tập phụ thuộc hàm F ở dạng chuẩn 2NF khi thỏa mãn:  
- Là dạng chuẩn 1NF  
- Mọi thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào mọi khóa dự bị của R.

\* 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.

******

***\*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 → 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 có thuộc tính không khóa nào phụ thuộc hàm bắc cầu vào khóa của R



Câu 1.36: Trình bày định nghĩa dạng chuẩn Boyee-Codd(BCNF). So sánh dạng chuẩn 3NF với BCNF. Cho ví dụ về BCNF

***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 → A nào và A ⊈ X, thì X là một siêu khóa của R.  
ϖ ***Ví dụ:*** Cho lược đồ quan hệ R = (A, B, C), tập phụ thuộc hàm F = {AB → C, C → A} và khóa K = {AB} R không ở dạng chuẩn BCNF vì có C → 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 → 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* → *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.*

Câu 1.37: Trình bày 2 tính chất mà người thiết kế CSDL phải tuân theo khi thực hiện phân tách các lược đồ về dạng chuẩn 3NF, BCNF hoặc các dạng chuẩn ở mức cao hơn

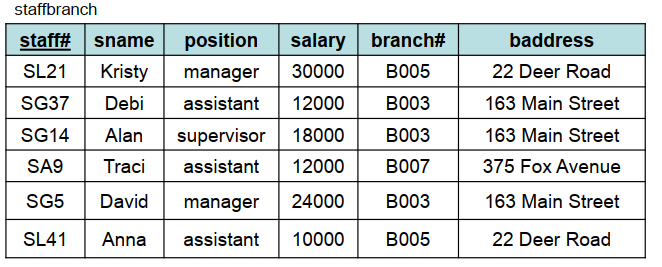
1. Kết nối không mất mát thông tin  
2. Các phụ thuộc hàm được bảo toàn sau khi phân tách

Câu 1.38: Tính chất bảo toàn phụ thuộc hàm khi phân tách một lược đồ quan hệ thành một tập các lược đồ nhỏ hơn là gì? Tính chất kết nối không tổn thất thông tin là gì?

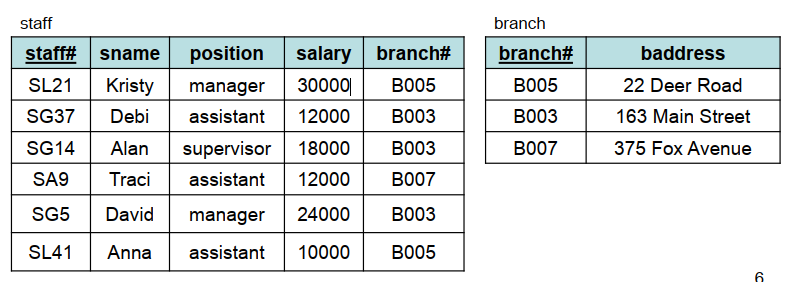
***2 tính chất quan trọng khi tách một lược đồ quan hệ thành một tập các lược đồ nhỏ hơn:  
1. Kết nối không tổn thất thông tin:*** đảm bảo mọi thể hiện của quan hệ ban đầu có thể xác định được từ các thể hiện liên quan trong các quan hệ nhỏ hơn.  
***2. Bảo toàn sự phụ thuộc hàm:*** đảm bảo một ràng buộc trên quan hệ ban đầu có thể được duy trì bằng cách sử dụng đơn giản một số ràng buộc trên mỗi quan hệ nhỏ hơn.  
=> Các quan hệ nhỏ hơn không cần kết nối với nhau để kiểm tra xem một ràng buộc trên các quan hệ ban đầu có bị vi phạm hay không.

Câu 2.19: Trình bày những bất thường có thể xảy ra khi thực hiện các thao tác cập nhật dữ liệu, bao gồm: bổ sung, loại bỏ và sửa đổi dữ liệu. Cho ví dụ cụ thể:

* Xét ví dụ lược đồ quan hệ *staffbranch*



* Nếu tách riêng quan hệ *staff* và *branch* thì thông tin về từng chi nhánh ngân hàng chỉ xuất hiện duy nhất một lần.



***\* Khi thực hiện chèn thêm dữ liệu:***

- Để chèn thêm thông tin chi tiết cho các nhân viên mới vào staffbranch, cần phải thêm thông tin chi tiết về chi nhánh tương ứng cho mỗi bản ghi nhân viên.

Ví dụ: Nếu thêm nhân viên mới vào branch B007, thì phải thêm cả địa chỉ của B007. Còn đối với lược đồ (staff, branch) thì chỉ cần thêm thông tin vào quan hệ staff là đủ.

- Để chèn thêm thông tin cho một branch mới mà hiện tại chưa có nhân viên nào, cần chèn thêm các giá trị Null cho các thuộc tính về nhân viên, như staff#, sname, …

+ Tuy nhiên, thuộc tính staff# là khóa chính nên điều này không thể được (vi phạm tính toàn vẹn của khóa).

* Vậy, không thể nhập thông tin cho một chi nhánh mới mà không có nhân viên nào.

\* ***Khi thực hiện xóa dữ liệu:***- Giả sử trong *staffbranch* có một chi nhánh chỉ còn một nhân viên cuối cùng. Nếu xóa thông tin về nhân viên này thì các thông tin về chi nhánh đó cũng sẽ bị xóa khỏi CSDL.  
***Ví dụ:*** Nếu xóa thông tin về nhân viên Traci khỏi quan hệ *staffbranch* thì thông tin về chi nhánh B007 cũng sẽ bị xóa.  
⇒ Điều này không xảy ra đối với lược đồ (*staff, branch*) vì thông tin về nhân viên được lưu trữ tách riêng khỏi thông tin về chi nhánh.

\* ***Khi thực hiện cập nhật dữ liệu:***- Giả sử muốn thay đổi giá trị của một trong các thuộc tính của một chi nhánh nào đó trong quan hệ staffbranch, ví dụ: địa chỉ chi nhánh, thì cần phải cập nhật tất cả các bộ của các nhân viên làm việc tại chi nhánh đó.  
- Nếu việc cập nhật thay đổi địa chỉ chi nhánh không được thực hiện trên tất cả các bộ liên quan trên quan hệ staffbranch thì CSDL sẽ không nhất quán (một chi nhánh sẽ có 2 địa chỉ khác nhau).  
⇒ *Như vậy: trên quan hệ staffbranch, khi thực hiện các thao tác cập nhật dữ liệu thì xuất hiện dị thường thông tin. Việc này sẽ tránh được bằng cách tách lược đồ staffbranch thành 2 quan hệ staff và branch.*

Câu 2. 20: Trình bày định nghĩa 2 tập phụ thuộc hàm tương đương và phương pháp để kiểm tra sự tương đương của 2 tập phụ thuộc hàm

Hai tập phụ thuộc hàm F và G là tương đương (ký hiệu: F ≡ G) nếu F+=G+.  
⇒ 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.

PHẦN 4: CÁC NGÔN NGỮ CƠ SỞ DỮ LIỆU

**Câu 1: SQL là gì?**

SQL là một ngôn ngữ tuân thủ chuẩn cho việc tạo ra và truy vấn các CSDL quan hệ.

SQL là một ngôn ngữ CSDL quan hệ đầy đủ. Nó bao gồm cả ngôn ngữ định nghĩa dữ liệu (DDL) và ngôn ngữ thao tác dữ liệu (DML).

Trước khi sử dụng một hệ CSDL quan hệ, bạn phải thực hiện 2 việc:  
(1) tạo một cấu trúc CSDL, (2) tạo các bảng lưu dữ liệu người sử dụng.

**\*Khai báo khoá:**

Có thể dùng PRIMARY KEY hoặc UNIQUE  
• SQL chỉ cho phép đánh chỉ số (index) bằng PRIMARY KEY.  
• SQL không cho phép thuộc tính PRIMARY KEY chứa giá trị rỗng.  
Tuy nhiên, nó lại cho phép các thuộc tính UNIQUE mang giá trị rỗng (có thể có nhiều hơn một bản ghi mang giá trị rỗng, tuy nhiên các giá trị khác rỗng của các bản ghi khác nhau phải khác nhau).

CREATE TABLE Bia (

|  |  |
| --- | --- |
| tên  Nhà\_SX | CHAR(20) PRIMARY KEY, CHAR(20) |

);  
CREATE TABLE Bán (  
quán CHAR(20),  
bia CHAR(20) REFERENCES Bia(tên),  
giá REAL  
);  
Hoặc:  
CREATE TABLE Bán (  
quán CHAR(20),  
bia CHAR(20),  
giá REAL,  
FOREIGN KEY bia REFERENCES Bia(tên)  
);

**\* Khoá ngoại:**

CREATE TABLE Bia (

|  |  |
| --- | --- |
| tên  Nhà\_SX | CHAR(20) PRIMARY KEY, CHAR(20) |

);  
CREATE TABLE Bán (  
quán CHAR(20),  
bia CHAR(20) REFERENCES Bia(tên),  
giá REAL  
);  
Hoặc:  
CREATE TABLE Bán (  
quán CHAR(20),  
bia CHAR(20),  
giá REAL,  
FOREIGN KEY bia REFERENCES Bia(tên)  
);

**\*Kiểm tra trên từng thuộc tính:**

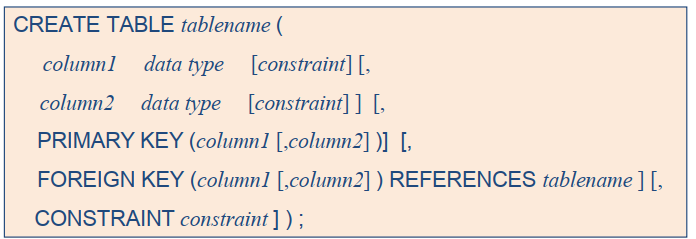
• CHECK (*condition*) : kiểm tra giá trị của từng bản ghi tại thuộc tính nào đó theo điều kiện có trong *condition.*• Các điều kiện trong *condition* chỉ được kiểm tra khi giá trị của các thuộc tính liên quan đến nó bị thay đổi (chèn, cập nhật).

CREATE TABLE Bán (  
quán CHAR(20),  
bia CHAR(20) CHECK(  
bia IN (SELECT tên  
FROM Bia) ),  
giá REAL CHECK(  
giá <= $5.00 )  
);

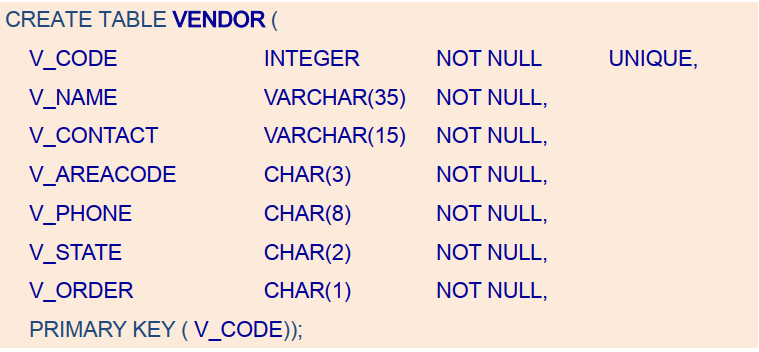
**\* Kiểm tra trên từng bản ghi:**  
• Được khai báo riêng sau khi đã khai báo các thuộc tính,  
• Điều kiện *condition* có thể liên quan đến bất kỳ thuộc tính nào trong bảng  
• Kiểm tra trong quá trình thêm, sửa bản ghi.  
Ví dụ:  
CREATE TABLE Bán (  
quán CHAR(20),  
bia CHAR(20),  
giá REAL,  
CHECK(quán = ‘Hải Xồm' OR giá <= $5.00)  
);  
=> Chỉ quán Hải Xồm được phép bán bia đắt hơn $5.00

**Câu 1.38: Trình bày cú pháp các câu lệnh tạo bảng và cập nhật bản ghi bằng ngôn ngữ SQL. Cho ví dụ với mỗi câu lệnh**

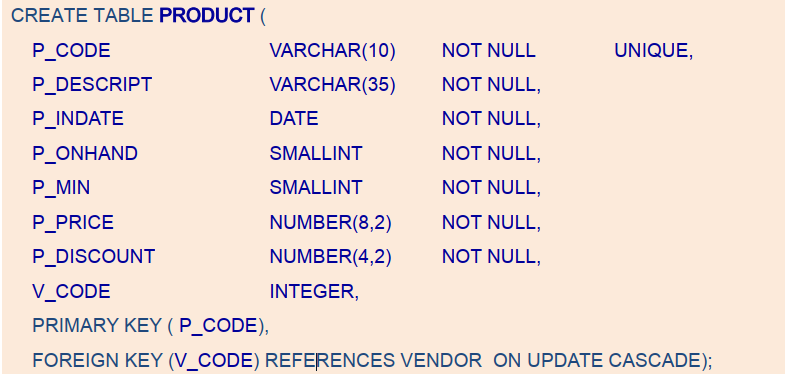
Câu lệnh CREATE TABLE có cú pháp như sau:



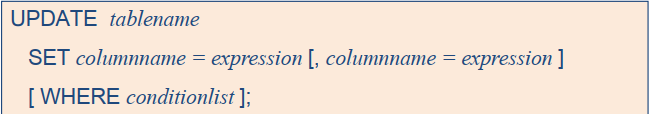
Ví dụ: tạo một bảng VENDOR



bảng PRODUCT



Cập nhật bản ghi bằng câu lệnh UPDATE

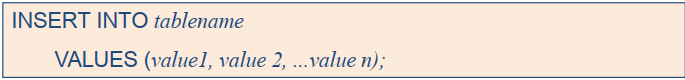


Lưu ý rằng điều kiện WHERE là không bắt buộc trong câu lệnh UPDATE. Nếu không có điều kiện WHERE, thì câu lệnh UPDATE sẽ được thực hiện trên tất cả các bản ghi của bảng đó.

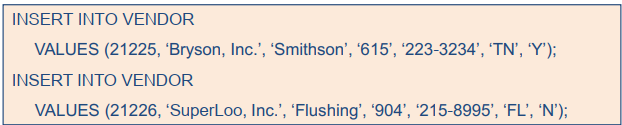
**Câu 1.39: Trình bày cú pháp các câu lệnh thêm mới và xoá bản ghi bằng ngôn ngữ SQL. Cho ví dụ với mỗi câu lệnh**

**\*Thêm dữ liệu vào bảng**

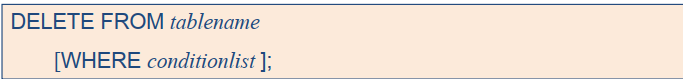
• SQL dùng câu lệnh INSERT để thêm dữ liệu mới vào một bảng.  
• Cú pháp của câu lệnh này như sau:



Ví dụ: Thêm hai bản ghi mới vào bảng VENDOR chúng ta cần thực hiện hai câu lệnh  
SQL dưới đây:



**\*Xoá dữ liệu:**

****

Ví dụ: Để xoá một bản ghi từ một bảng dựa trên giá trị của khoá chính, bạn có thể sử dụng câu lệnh như sau:  
DELETE FROM PRODUCT  
WHERE P\_CODE = ‘23114-AA’;

**Câu 1.40: Trình bày cú pháp các câu lệnh tìm kiếm không có điều kiện, tìm kiếm với điều kiện đơn giản và tìm kiếm có sử dụng các mệnh đề GROUP BY, ORDER BY bằng ngôn ngữ SQL**

**\*Cú pháp các câu lệnh tìm kiếm không có điều kiện:**

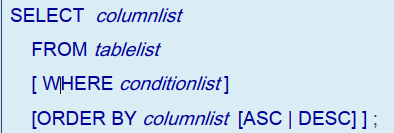
SELECT [ ALL | DISTINCT] *columnlist*FROM *tablelist*

**\* Tìm kiếm với điều kiện đơn giản:**

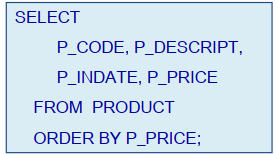
SELECT *columnlist*FROM *tablelist*WHERE *conditionlist* ;

**\*Tìm kiếm có sử dụng các mệnh đề GROUP BY, ORDER BY:**

Vế câu **ORDER BY** rất hữu dụng trong việc sắp xếp danh sách theo yêu cầu  
nào đó  
• Cú pháp:

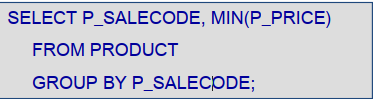
  
• Nếu cột được sắp xếp bao gồm giá trị rỗng thì các giá trị này được cho lên  
đầu hoặc xuống cuối tùy thuộc vào các hệ CSDL quan hệ cụ thể.  
• Vế câu ORDER BY luôn nằm cuối cùng trong tập lệnh SELECT.  
• Ta có thể chọn cụ thể kiểu sắp xếp (tăng dần hoặc giảm dần). Nếu không  
chọn, mặc định của hệ thống sẽ là tăng dần.

Ví dụ: Câu truy vấn sau liệt kê các bản ghi của bảng PRODUCT được sắp xếp theo thứ tự tăng dần của P\_PRICE (giảm dần thêm DESC ở cuối):



Tần suất phân bổ các giá trị trả về sẽ được tự động tạo ra bởi vế câu lệnh  
**GROUP BY** bên trong câu lệnh SELECT.  
• Cú pháp là:

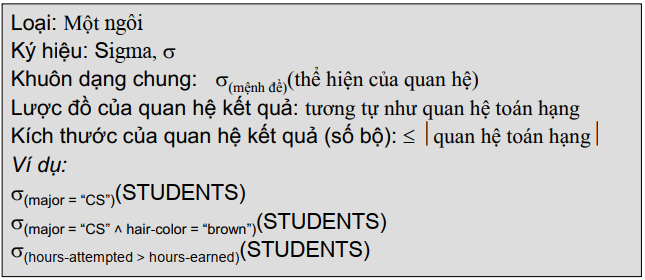
  
• Vế câu lệnh GROUP BY thường được sử dụng trong các cột thuộc tính có các  
hàm thống kê (MAX, MIN, AVG, SUM, COUNT) trong câu lệnh SELECT.  
• Ví dụ, tìm giá bán nhỏ nhất cho mỗi mã hàng. Câu lệnh được thực hiện:

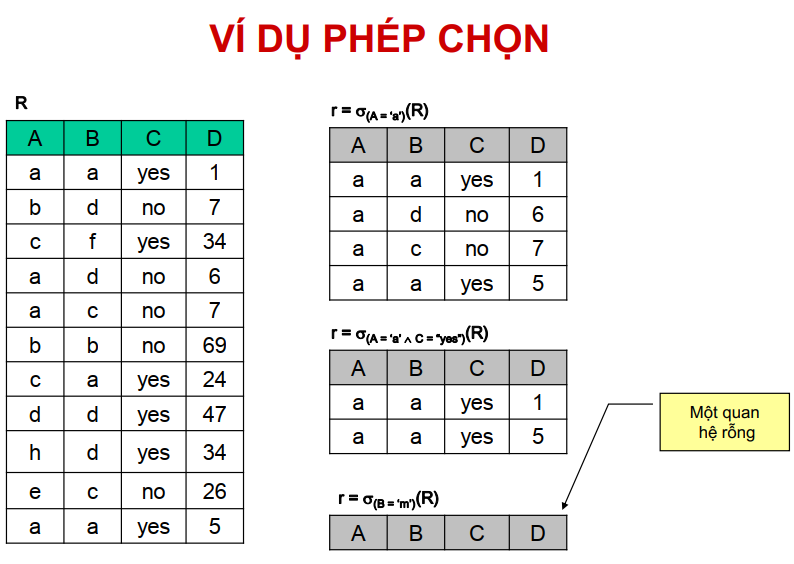


**Câu 2.21: Trình bày khái niệm và cho ví dụ về 5 phép toán cơ bản của đại số quan hệ: phép chọn, phép chiếu, phép hợp, phép trừ và phép tích Đề-các**

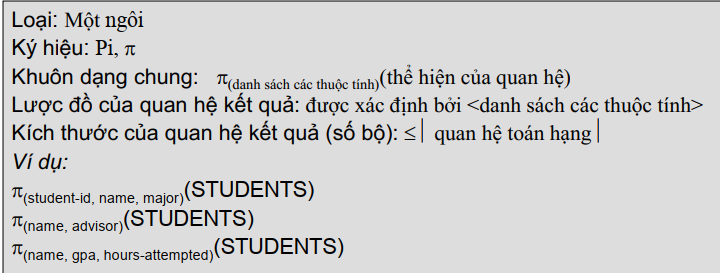
**\*Phép chọn:**

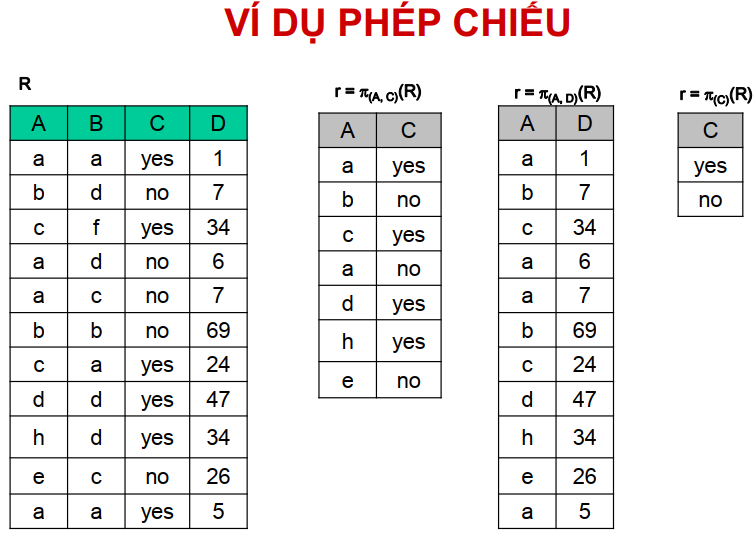
- Phép chọn lựa chọn ra các bộ từ thể hiện của quan hệ sao cho thỏa mãn mệnh đề điều kiện cụ thể nào đó.  
- Mệnh đề có thể chứa các toán tử so sánh, như =, ≠, <, ≤, >, ≥. Hoặc kết hợp với các phép toán liên kết *và* (∧), *hoặc* (∨), và *phủ định* (¬).  
- Phép chọn có thể được coi như một lát cắt ngang của quan hệ toán hạng.



**\*Phép chiếu:**

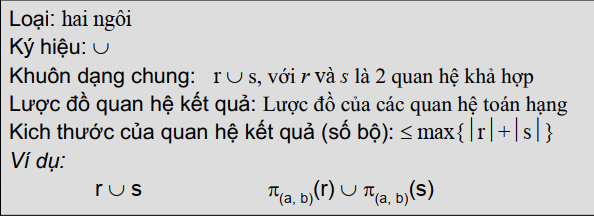
- Phép chiếu có thể được coi như một lát cắt dọc của quan hệ toán hạng.  
- Nếu phép toán sinh ra các bộ giống hệt nhau, thì sẽ chỉ giữ lại một bộ và loại bỏ đi các bộ bị trùng.

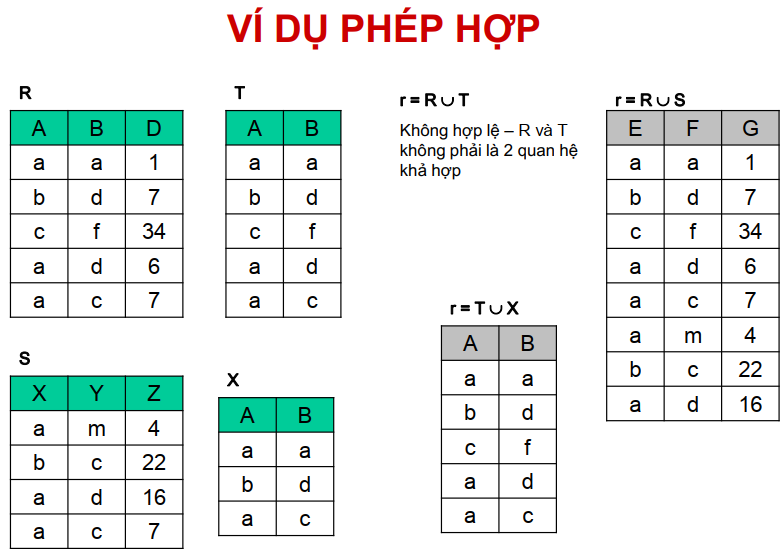




**\*Phép hợp:**

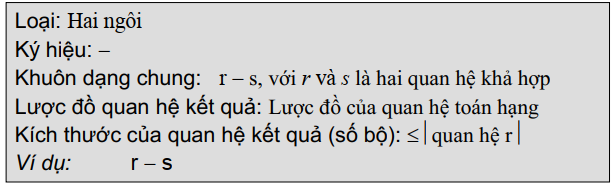
- Phép hợp cung cấp một phương tiện để trích lọc thông tin nằm trên hai quan hệ toán hạng khả hợp với nhau.  
♣ 2 quan hệ *r(R)* và *s(S)* được gọi là khả hợp khi thỏa mãn 2 điều kiện sau:  
- Chúng phải có cùng số bậc hay cùng số lượng thuộc tính.  
- Miền giá trị của thuộc tính thứ (i) của r và thuộc tính thứ (j) của s phải giống nhau, cho mọi giá trị của I, j.

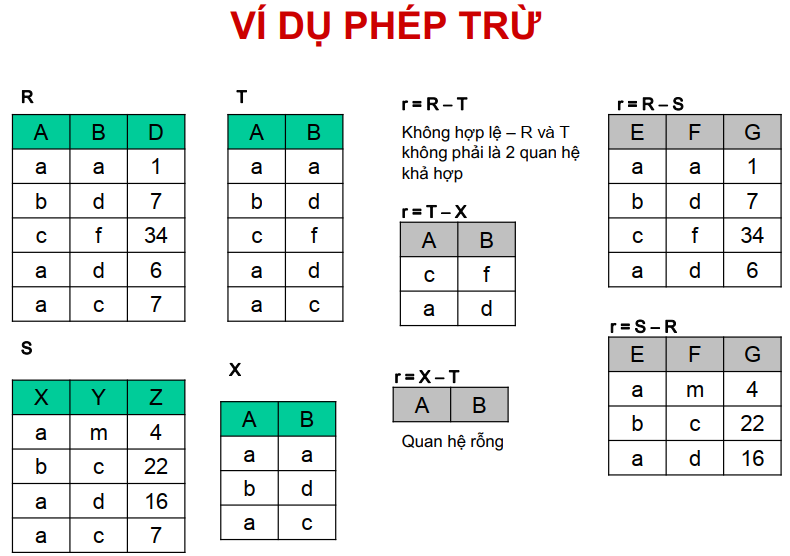




**\*Phép trừ:**

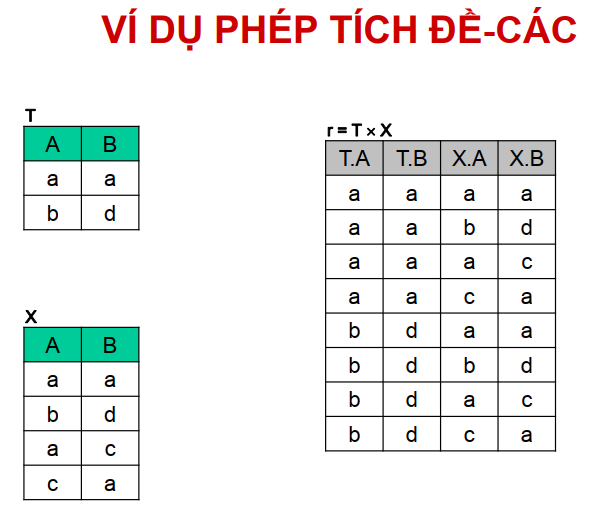
- Phép trừ cho phép trích lọc thông tin được chứa trong một quan hệ mà nó không được chứa trong quan hệ thứ hai.  
- Tương tự như phép hợp, phép trừ yêu cầu 2 quan hệ toán hạng phải là khả hợp.

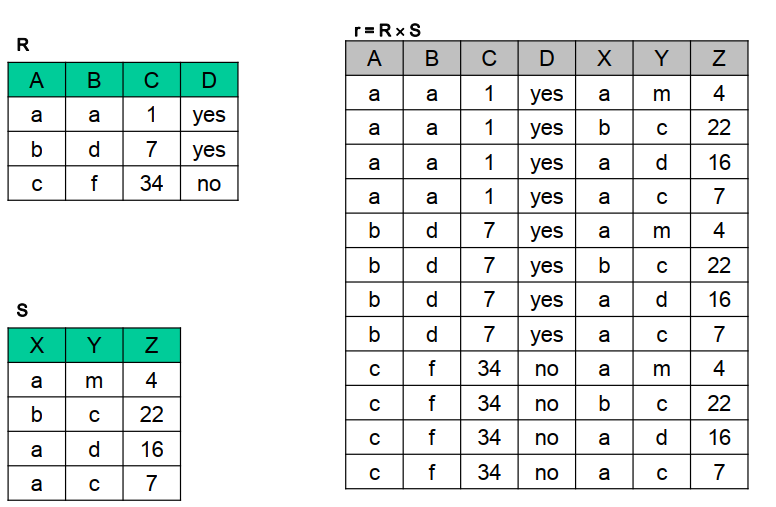




**\*Phép tích Đề-các:**

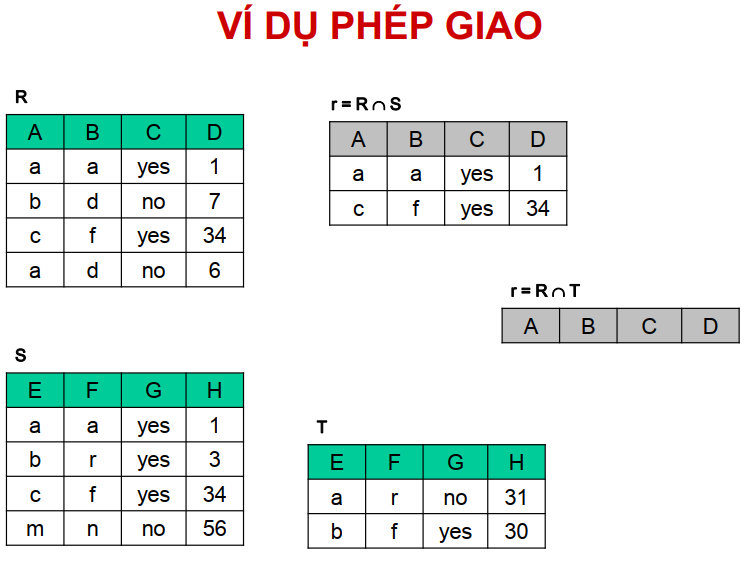
- Tích Đề-các cho phép kết nối 2 quan hệ bất kỳ thành một quan hệ đơn.  
- Một quan hệ là một tập con của tích Đề-các tập các miền giá trị.





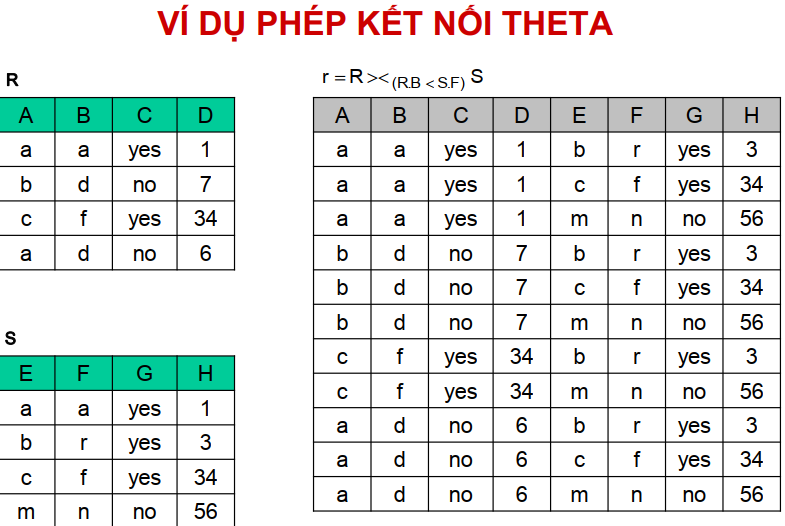
# Câu 2.22: Trình bày khái niệm và cho ví dụ về các phép toán mở rộng của đại số quan hệ: phép giao, phép kết nối theta, phép kết nối bằng, phép kết nối tự nhiên và phép chia.

**\* Phép giao:** Phép giao tạo ra tập các bộ xuất hiện ở cả hai quan hệ  
toán hạng.

****

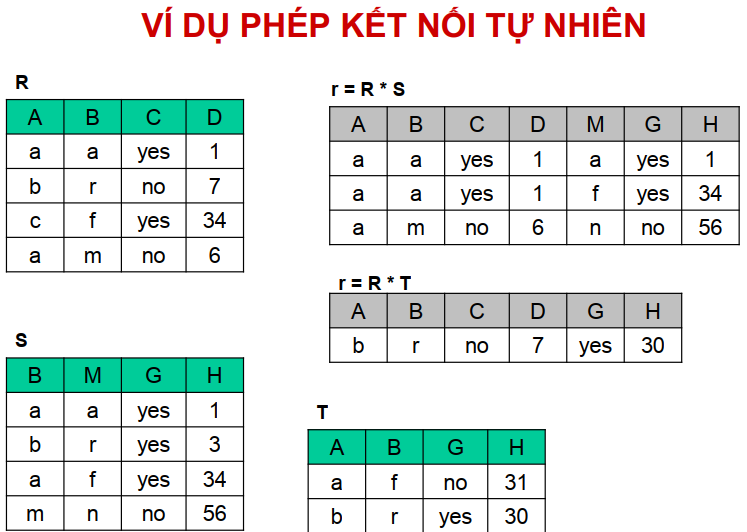
**\*Phép kết nối Theta:**

- 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, vì vậy việc loại bỏ các bộ dư thừa cần phải được thực hiện một cách tường minh thông qua mệnh đề điều kiện.

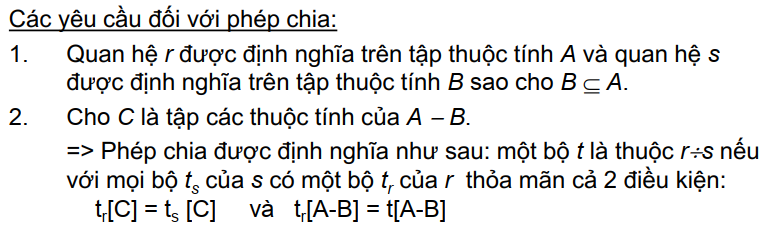


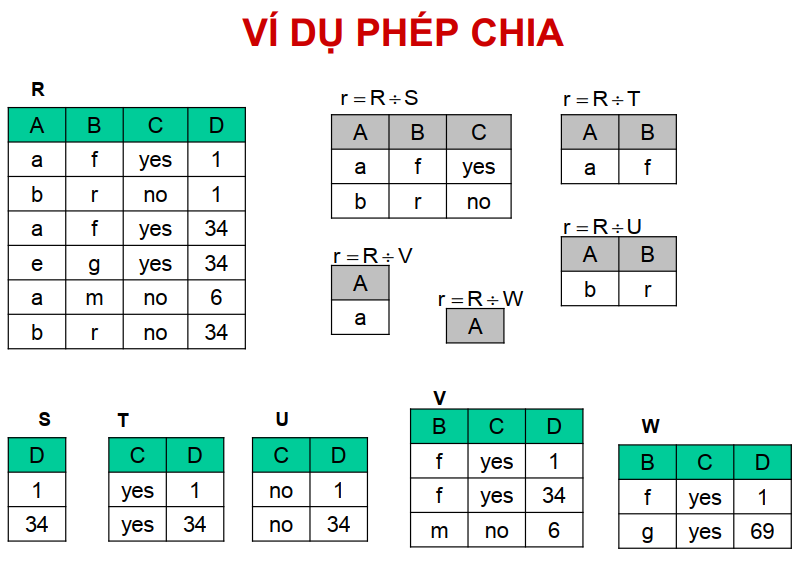
\*Phép kết nối tự nhiên:

- 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.  
- Phép kết nối tự nhiên là 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***.



**\*Phép chia:**

****

****