**Câu 1.1 Trình bày các khái niệm cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu,hệ cơ sở dữ liệu**

**Khái niệm cơ sở dữ liệu**

Theo nhận thức chung nhất, cơ sở dữ liệu (database) đơn giản là một tập thông tin (dữ liệu) có liên quan đến nhau. định nghĩa này rất mơ hồ, vì theo đây, có thể coi một trang văn bản là một CSDL.

Khái niệm “dữ liệu” trong CSDL có thể bao gồm một phạm vi rất rộng các đối tượng: chữ số, văn bản, đồ họa, video,...

Định nghĩa cụ thể hơn của một CSDL bao gồm một tập các đặc tính không tường minh được xem xét cùng nhau để định nghĩa một CSDL.

CSDL thể hiện các khía cạnh khác nhau của thế giới thực.  
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.

**Khái niệm hệ quản trị cơ sở dữ liệu :**Hệ quản trị CSDL (DBMS – Database management system) là một hệ thống phần mềm cho phép tạo lập CSDL và điều khiển mọi truy nhập đến CSDL đó.

**Khái niệm về 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.

**Câu 1.2 : Trình bày các khái niệm về hệ quản trị cơ sở dữ liệu và hệ cơ sở dữ liệu.Nêu tên một số hệ quản trị cơ sở dữ liệu có mặt trên thị trường hiện nay.  
  
Khái niệm hệ quản trị cơ sở dữ liệu :**Hệ quản trị CSDL (DBMS – Database management system) là một hệ thống phần mềm cho phép tạo lập CSDL và điều khiển mọi truy nhập đến CSDL đó. **Khái niệm về 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.  
Một số hệ quản trị dữ liệu hiện nay : SQL Server, My SQL , No SQL , Mongo DB,…

**Câu 1.3:Trình bày khái niệm dữ liệu vật lý và dữ liệu dẫn xuất.Cho ví dụ**  
**Dữ liệu vật lý** là những dữ liệu có thực, được nhập vào trong CSDL.  
**Dữ liệu dẫn xuất**: Là những dữ liệu được tính toán từ những dữ liệu nằm trong CSDL.  
Ví dụ trong CSDL quản lý sinh viên có thuộc tính về tuổi sinh viên và thuộc tính ngày sinh của sinh việt thì thuộc tính ngày sinh của sinh viên là thuộc tính vật lý còn thuộc tính tuổi là thuộc tính dẫn xuất vì từ ngày sinh ta có thể tính toán được tuổi của sinh viên đó.  
Dữ liệu dẫn xuất thường không cần lưu trong CSDL.

**Câu 1.4 Trình bày các vai trò người dùng trong cơ sở dữ liệu  
Người quản trị dữ liệu (DA –Data Administrator)**:có trách nhiệm quản lý tài nguyên dữ liệu, bao gồm lập kế hoạch cho CSDL, phát triển và duy trì các chuẩn, chính sách và thủ tục, và thiết kế CSDL mức khái niệm/logic.

**Người quản trị CSDL (DBA –Database Administrator)**:có trách nhiệm với việc lưu trữ vật lý CSDL, bao gồm thiết kế và cài đặt CSDL vật lý, kiểm soát bảo mật và toàn vẹn dữ liệu, duy trì hệ điều hành, và đảm bảo thỏa mãn hiệu năng cho các ứng dụng người dùng. *=> Vai trò của DBA liên quan đến nhiều đặc tính kỹ thuật hơn DA.*

**Người thiết kế CSDL**:trong các dự án thiết kế CSDL lớn, cần phân biệt 2 loại thiết kế:

+Người thiết kế CSDL logic liên quan tới việc xác định CSDL (các thực thể và thuộc tính), các mối quan hệ giữa dữ liệu, và các ràng buộc đối với dữ liệu sẽ được lưu trữ trong CSDL.

+ Người thiết kế CSDL vật lý phụ thuộc nhiều vào hệ quản trị CSDL đích, và có thể có nhiều cách để cài đặt CSDL. Người thiết kế CSDL vật lý phải có hiểu biết đầy đủ về tính năng của hệ quản trị CSDL đích.

**Người phát triển ứng dụng**:Có trách nhiệm xây dựng chương trình ứng dụng cung cấp các chức năng cần thiết cho người dùng cuối, sau khi CSDL đã được cài đặt.

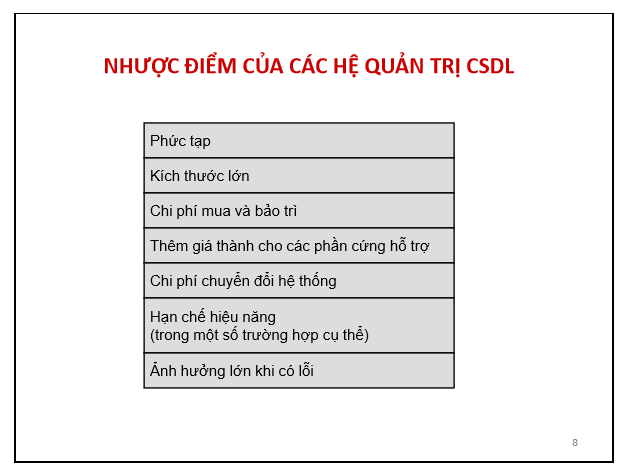
**Người dùng cuối**:là “khách hàng” của CSDL, và có thể được phân thành 2 nhóm dựa theo cách mà họ sử dụng hệ thống:

 Nhóm người dùng không biết đến khái niệm CSDL hoặc hệ quản trị CSDL:Truy nhập CSDL thông qua chương trình ứng dụng được viết riêng biệt, giúp cho các thao tác của người dùng đơn giản nhất có thể.

 Nhóm người dùng nhận biết được cấu trúc CSDL và các phương tiện được cung cấp bởi hệ quản trị CSDL:Thường dùng các ngôn ngữ truy vấn bậc cao như SQL để thực hiện những thao tác được yêu cầu và thậm chí có thể viết những chương trình ứng dụng để phục vụ cho mục đích riêng.

**Câu 1.5:Trình bày ưu, nhược điểm của các hệ quản trị cơ sở dữ liệu.**





**Câu 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 khác nhau được định nghĩa theo các mức trừu tượng của mô hình 3 lớp:

 Trên cùng có rất nhiều **lược đồ ngoài**(hay gọi là lược đồ con) tương ứng với các khung nhìn dữ liệu khác nhau.   
 Tại mức khái niệm có một **lược đồ khái niệm,** mô tả tất cả các thực thể, thuộc tính và các quan hệ cùng với những ràng buộc toàn vẹn.  
+ Mức trừu tượng thấp nhất có một **lược đồ trong**, mô tả toàn bộ mô hình trong, gồm định nghĩa các bản ghi được 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

**=> Tập hợp các thông tin được lưu trữ trong CSDL tại một thời điểm đặc biệt được gọi là một thể hiện của cơ sở dữ liệu đó.**Ví dụ: Giá trị của một bản ghi tại một thời điểm được gọi là một thể hiện của CSDL.

**Câu 1.7: Tính độc lập dữ liệu là gì ? , Phân loại tính độc lập dữ liệu.**Một trong những mục tiêu chính của kiến trúc 3 lớp là cung cấp 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.

**Câu 1.8: Trình bày khái niệm các ngôn ngữ CSDL** 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).

Hầu hết các hệ quản trị CSDL đều có một môi trường cho phép nhúng các ngôn ngữ con dữ liệu vào trong một ngôn ngữ lập trình bậc cao như COBOL, Pascal, C, C++, Java hay Visual Basic (được gọi là các ngôn ngữ chủ).

**Câu 1.9 : Trình bày các khái niệm mô hình dữ liệu và các thành phần trong mô hình dữ liệu . Phân loại mô hình dữ liệu.**Một mô hình dữ liệu là một tập tích hợp các khái niệm nhận thức để mô tả và thao tác dữ liệu, các mối quan hệ giữa dữ liệu và các ràng buộc trên dữ liệu của một tổ chức**.**Một mô hình dữ liệu là một cách biểu diễn các đối tượng trong thế giới thực và các sự kiện cũng như các liên hệ giữa chúng. Nó là một khái niệm trừu tượng tập trung vào những khía cạnh cần thiết, sống còn của một tổ chức và cần bỏ qua những thuộc tính ngẫu nhiên.  
Mô hình dữ liệu phải cung cấp các khái niệm và ký hiệu cơ bản, cho phép người thiết kế CSDL và người dùng trao đổi với nhau những hiểu biết về dữ liệu của tổ chức một cách chính xác và không đa nghĩa.

Dựa vào mô hình kiến trúc 3 lớp, có thể xác định 3 loại mô hình dữ liệu khác nhau:

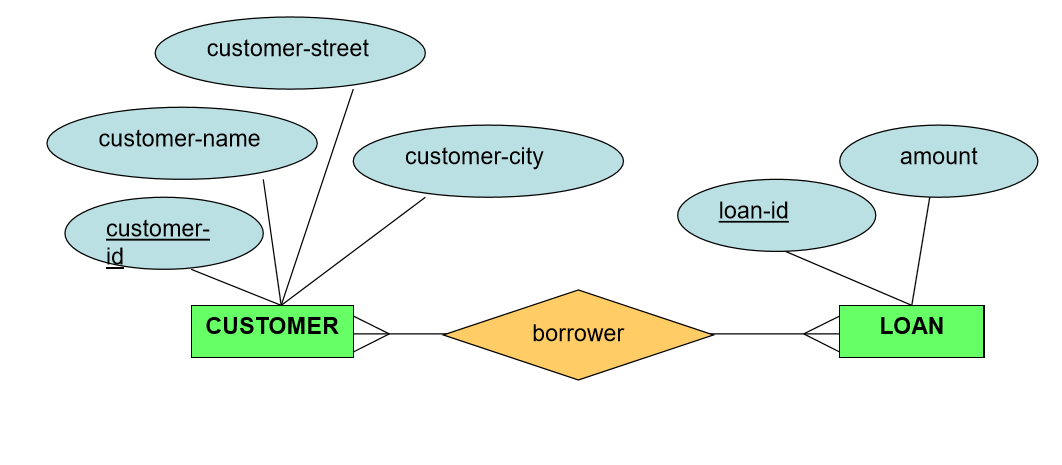
1. Mô hình dữ liệu ngoài biểu diễn từng khung nhìn của người dùng trong tổ chức.

2. Mô hình dữ liệu khái niệm biểu diễn khung nhìn mức logic, độc lập với hệ quản trị CSDL.

3. Mô hình dữ liệu trong biểu diễn lược đồ khái niệm theo cách mà hệ quản trị CSDL hiểu được.

**Câu 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 thế giới thực và có thể phân biệt được với các đối tượng khác. Thực thể có thể cụ thể (một người, một quyển sách, …) hoặc cũng có thể trừu tượng (một khoản vay ngân hàng, một khái niệm, …).  
Thực thể được biểu diễn bởi một tập các **thuộc tính** (là các thuộc tính mô tả hoặc các đặc tính của thực thể).  
**Tập thực thể** là một nhóm các thực thể có cùng thuộc tính. Ví dụ: tập tất cả khách hàng của ngân hàng có thể được định nghĩa là tập khách hàng.  
Mỗi thuộc tính có một tập giá trị cho phép, được gọi là **miền** (hay tập giá trị) của thuộc tính đó.

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

CSDL trên gồm 2 tập thực thể . Customer và Loan. Trong đó Customer bao gồm các thuộc tính về customer-id,customer-name,customer-street,customer-city . Trong đó mỗi customer khác nhau về customer-id.  
Tập thực thể LOAN bao gồm các thuộc tính loan-id và amount.Mỗi LOAN khác nhau phân biệt bởi loan-id.   
Mỗi customer là một borrower của loan và mỗi LOAN có thể borrow nhiều customer.

**Câu 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 là thuộc tính không bao gồm các thành phần cấu thành giá trị của nó chỉ phụ thuộc bởi chính nó.  
Thuộc tính kép bao gồm nhiều thành phần con cấu thành,giá trị của nó phụ thuộc vào các giá trị con cấu thành nó.

Thuộc tính đơn trị là thuộc tính **có nhiều nhất** một giá trị tại một thời điểm cụ thể.

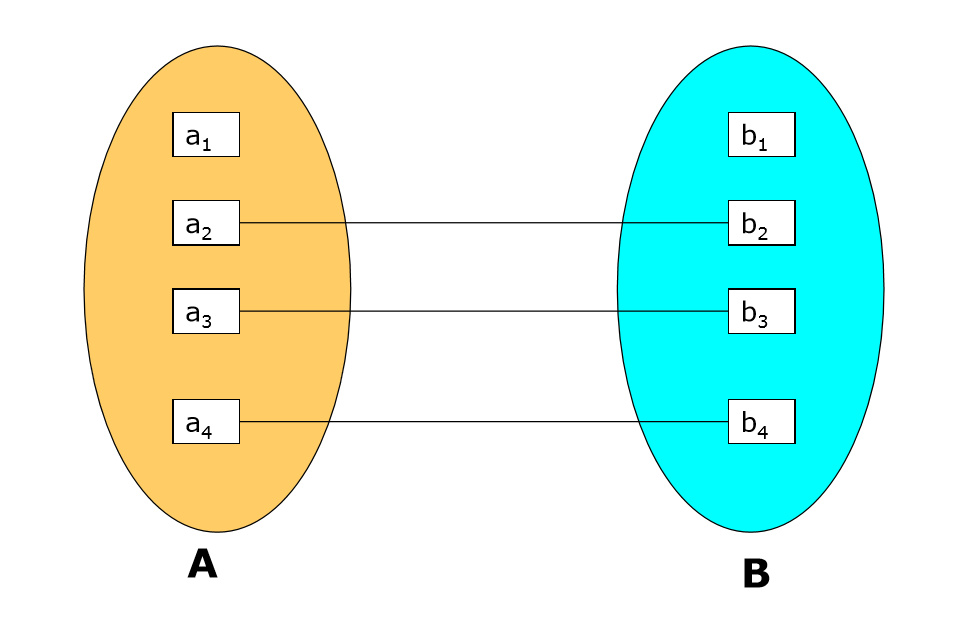
Thuộc tính đa trị là thuộc tínhcó thể có nhiều giá trị khác nhau tại một thời điểm.   
Thuộc tính dẫn xuất: là thuộc tính mà giá trị của nó được dẫn xuất (hoặc được tính toán) từ những giá trị của các thuộc tính hoặc các thực thể có liên quan.  
Thuộc tính rỗng (Null): là thuộc tính nhận giá trị rỗng khi một thực thể không có giá trị cho nó.

**Câu 1.13 : Trình bày về ràng buộc ánh xạ lực lượng liên kết trong mô hinh quan hệ thực thể E – R**

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

Ví dụ về ánh xạ 1 : 1 từ A đến B (các th còn lại tương tự)



**Câu 1.13 : Trình bày về ràng buộc tham gia trong mô hinh quan hệ thực thể E – R**

Sự tham gia của một tập thực thể E trong một tập các quan hệ R được gọi là đầy đủ (total) nếu mọi thực thể của E tham gia vào ít nhất một 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.

**Câu 1.14 : Trình bày các khái niệm về 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.  
Nếu tập K là một siêu khóa của tập thực thể E thì mọi tập cha của K cũng là siêu khóa.  
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.  
 Hai thực thể khác nhau sẽ không có cùng giá trị trên tất cả các thuộc tính khóa (khóa chính, khóa dự bị hay siêu khóa) tại cùng một thời điểm. => ràng buộc khóa   
 Người thiết kế CSDL phải cẩn thận khi lựa chọn tập các thuộc tính cấu thành khóa của một tập thực thể để đảm bảo:

1. Chắn chắn rằng tập các thuộc tính xác định duy nhất thực thể.

2. Tập thuộc tính khóa sẽ không bao giờ hoặc rất hiếm khi bị thay đổi.

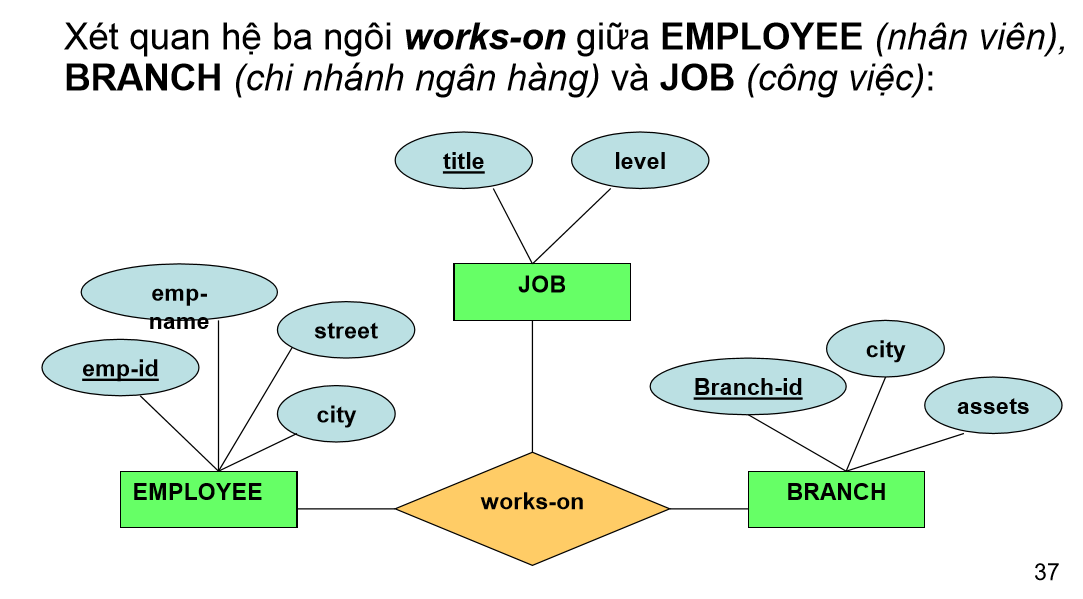
**Câu 1.16 :** **Khóa và siêu khóa khác nhau thế nào ? Cho ví dụ giải thích.** Khái niệm siêu khóa là một định nghĩa không đầy đủ của khóa vì siêu khóa còn chứa nhiều thuộc tính dư thừa. Giả sử có một tập thực thể sinh viên trong một lớp học với lược đồ sau: STUDENTS (SS#, name, address, age, major, minor, gpa, spring - sch)  
Siêu khóa cho tập thực thể STUDENTS có thể là: (SS#, name, major, minor) hoặc (SS#, name)

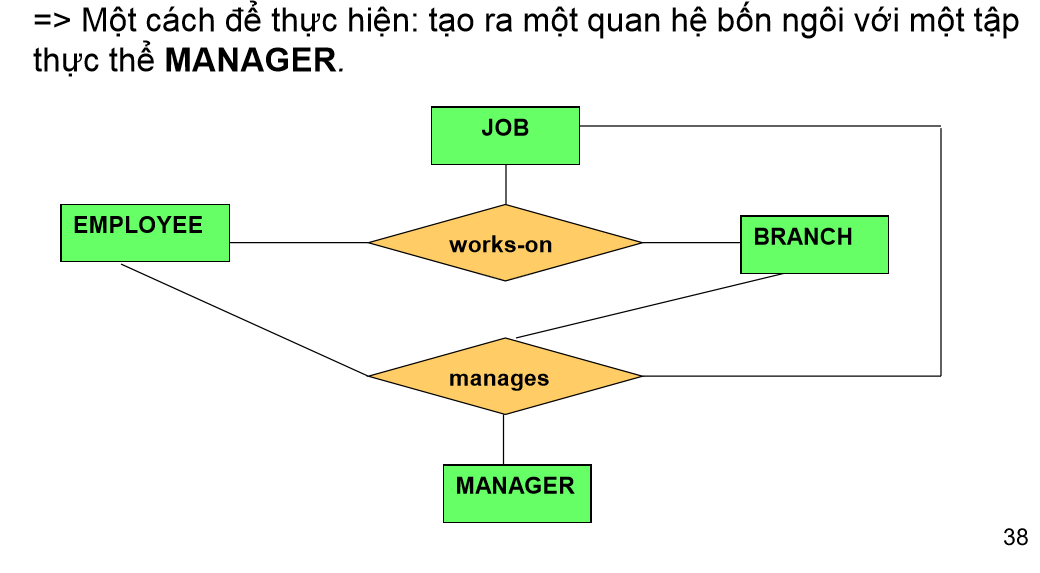
**Câu hỏi 1.17: Trình bày các khái niệm tập thực thể mạnh và tập thực thể yếu. Cho một ví dụ cụ thể.**

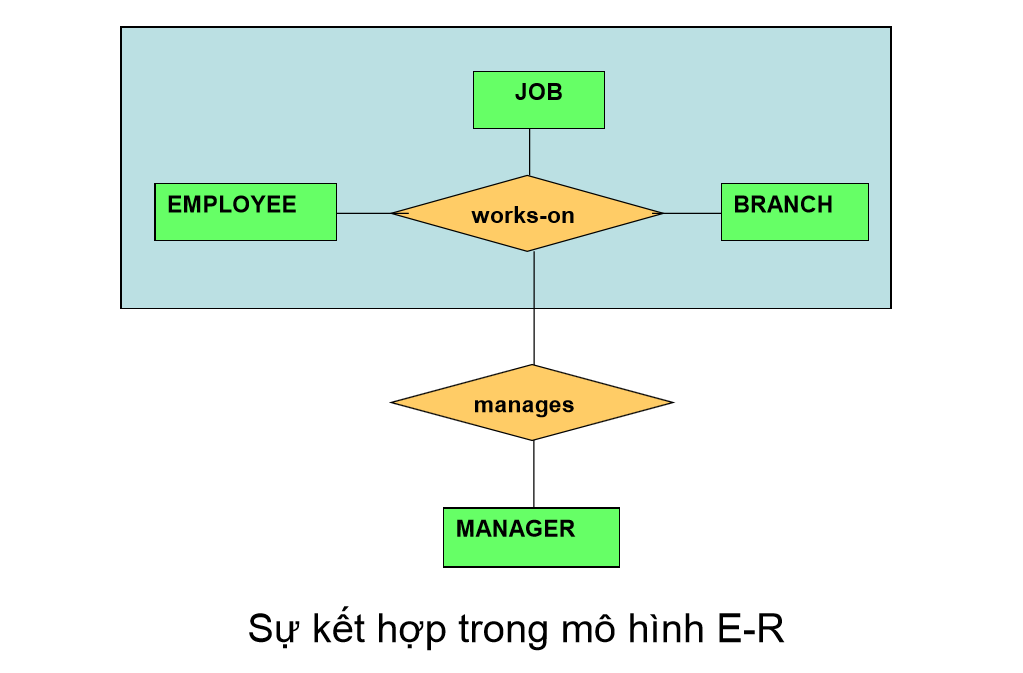
Tập thực thể không đủ các thuộc tính để hình thành một khóa chính gọi là tập thực thể yếu.   
Tập thực thể có khóa chính được gọi là tập thực thể mạnh.  
Ví dụ: Xét tập thực thể trả tiền:   
PAYMENT( payment - number, payment date, payment amount ).   
=> Mã số trả tiền (payment-number) thường là các số liên tiếp, bắt đầu từ 1 và được sinh ra riêng rẽ 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 không có khóa chính và chỉ là một tập thực thể yếu.

**Câu hỏi 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.  
 **xét Ví dụ trước** : 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, years- employed).   
 Sau đó, tìm điểm chung giữa các thuộc tính để xây dựng tập thực thể: PERSON (name, address, city)

**Câu hỏi 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.  
Tập thực thể mức thấp hơn cũng kế thừa các mối quan hệ thuộc về tập thực thể mức cao hơn định nghĩa nó.  
Lợi ích ở đây để thể hiện những đặc tính riêng của kịch bản trong thế giới thực , giúp cơ sở dữ liệu thiết kế đúng với kịch bản đã đặt ra.  
Ví dụ tập thực thể Động Vật là tập thực thể cha của 2 tập thực thể con là Chó và Mèo thì Chó và Mèo đều kế thừa tất cả các thuộc tính của Động Vật, mặt khác chó và mèo vẫn có những đặc tính riêng .

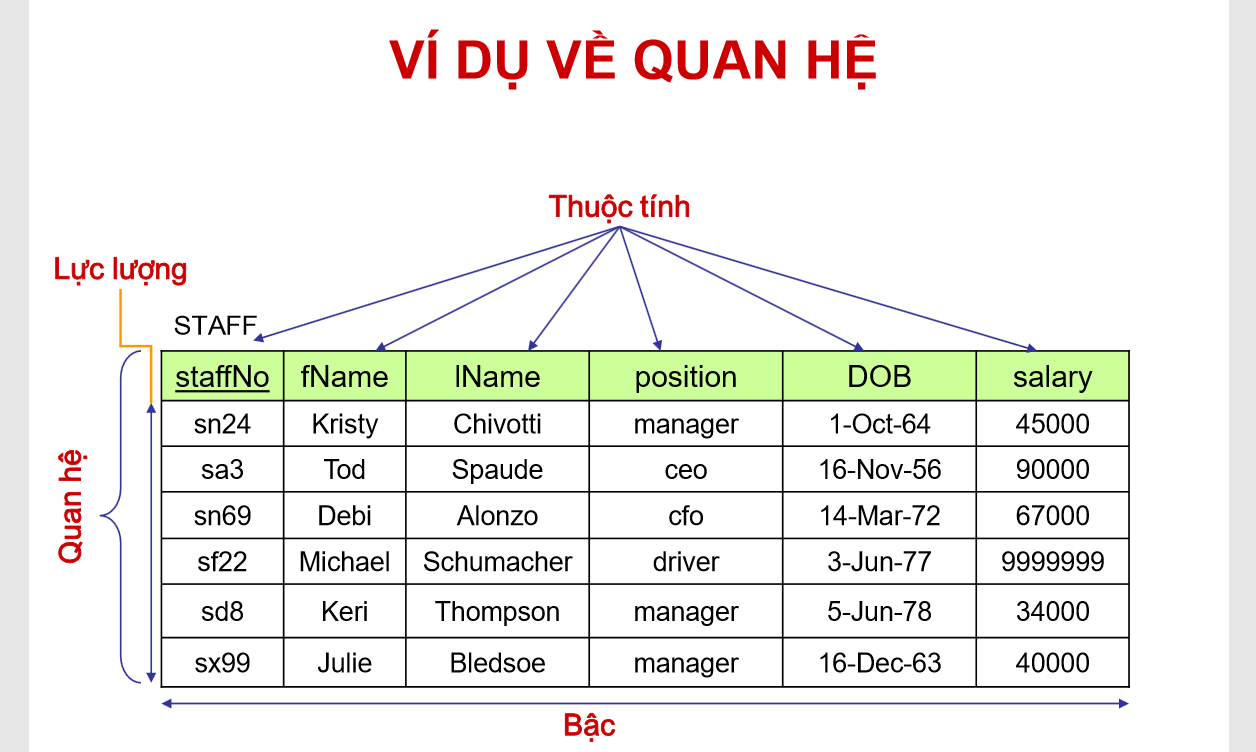
**Câu hỏi 1.20: Trình bày khái niệm tập thực thể kết hợp. Cho một ví dụ và giải thích.**Kết hợp là một sự trừu tượng thông qua việc coi các mối quan hệ như là các thực thể ở mức cao.   
Tập thực thể kết hợp là tập hợp những thực thể ở mức cao đó.  
Ví dụ :  


Ở đây ta thấy Giả sử muốn ghi nhận việc quản lý từng công việc được thực hiện bởi một nhân viên tại một chi nhánh; nghĩa là muốn lưu trữ người quản lý cho quan hệ ba ngôi (EMPLOYEE, BRANCH, JOB).  


không thể thể hiện mối quan hệ hai ngôi giữa MANAGER và EMPLOYEE vì Một quan hệ hai ngôi sẽ không cho phép thể hiện liên kết (BRANCH, JOB) của một EMPLOYEE (nhân viên) được quản lý bởi người quản lý.  
Trong lược đồ E-R mô hình hóa tình huống trên, tập các mối quan hệ works-on và manages có thể kết hợp thành một tập quan hệ duy nhất.   
=> Tuy nhiên, không thể thực hiện việc kết hợp vì một bộ ba (EMPLOYEE, BRANCH, JOB) có thể không có người quản lý.   
Có thông tin dư thừa trong lược đồ E-R trên vì từng bộ ba (EMPLOYEE, BRANCH, JOB) của quan hệ manages cũng giống của works-on.   
Nếu người quản lý là một giá trị chứ không phải là một thực thể, thì có thể biến MANAGER thành một thuộc tính đa trị của quan hệ works-on. => Biến đổi này sẽ làm việc tìm kiếm khó khăn hơn (cả về logic lẫn về chi phí thực hiện). (Ví dụ, bộ ba employee-branch-job sẽ do người quản lý nào chịu trách nhiệm?).  
**Cách tốt nhất là dùng kết hợp**   
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   


**Câu hỏi 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.

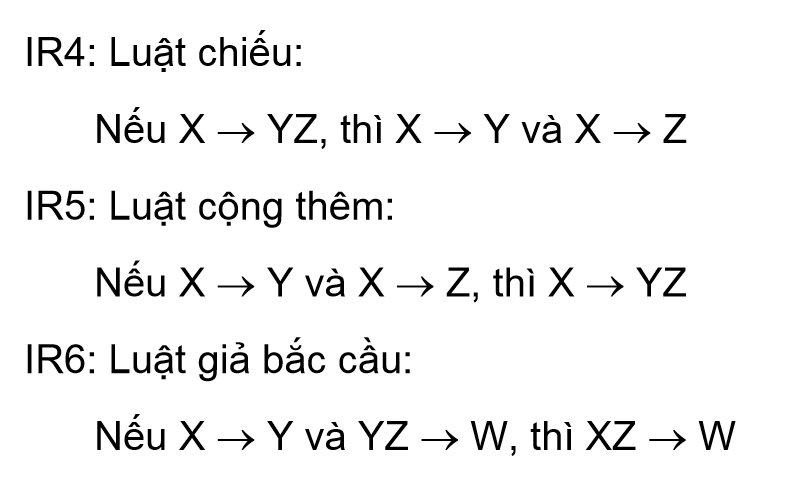
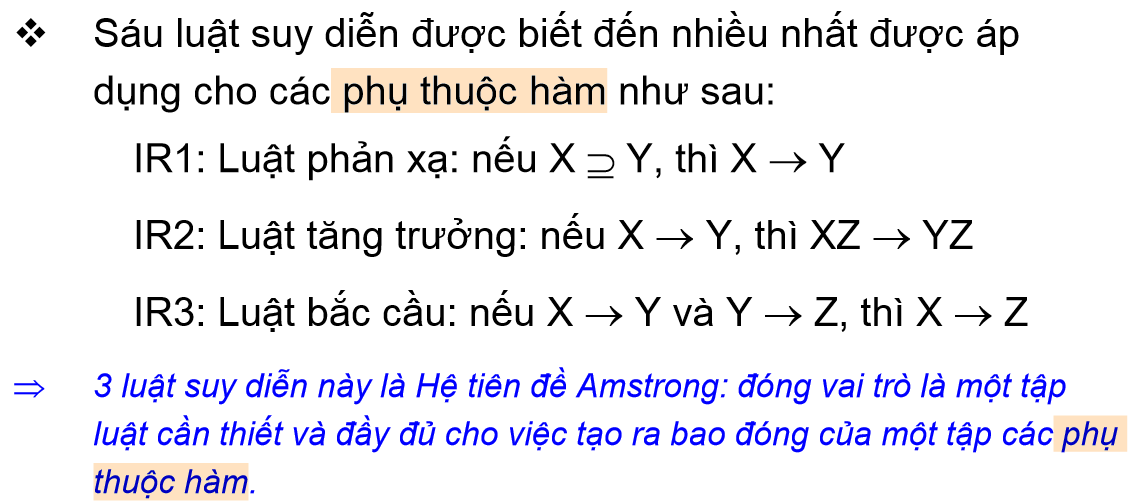
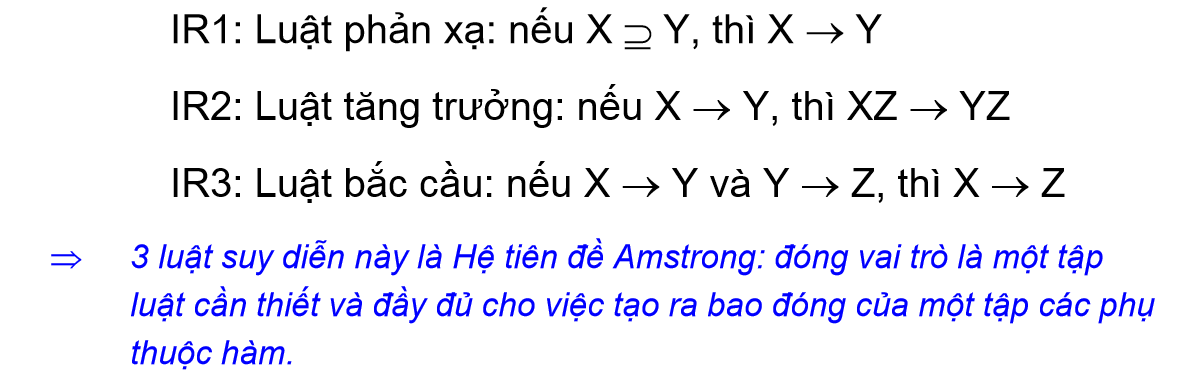
**Câu hỏi 1.22: Trình bày các khái niệm: quan hệ, thuộc tính, miền giá trị của thuộc tính, bộ (bản ghi), bậc, lực lượng trong một quan hệ. Cho ví dụ cụ thể.  
Quan hệ:** là một bảng (ma trận) với các hàng và các cột, lưu giữ thông tin về các đối tượng được mô hình hóa trong CSDL.  
**Thuộc tính**: là 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.

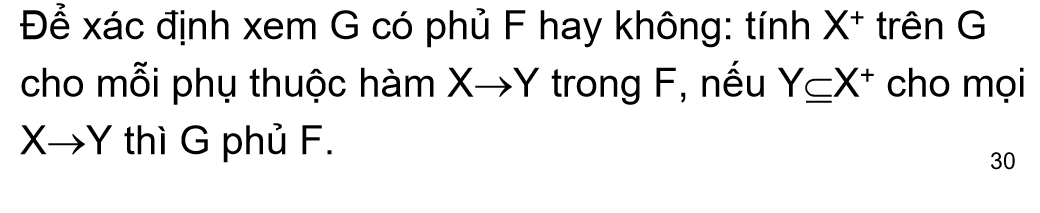
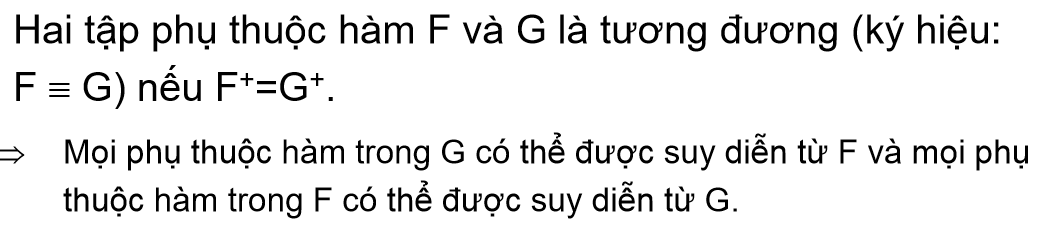

**Câu hỏi 1.23: Trình bày các đặc tính của một quan hệ.**1. Quan hệ có một tên gọi phân biệt với tên của các quan hệ khác trong lược đồ quan hệ.  
2. Mỗi thuộc tính có một tên gọi riêng.   
3. Mỗi thuộc tính có một miền giá trị.   
4. Mỗi thuộc tính chứa một giá trị nguyên tố.   
5. Các bộ là phân biệt 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.

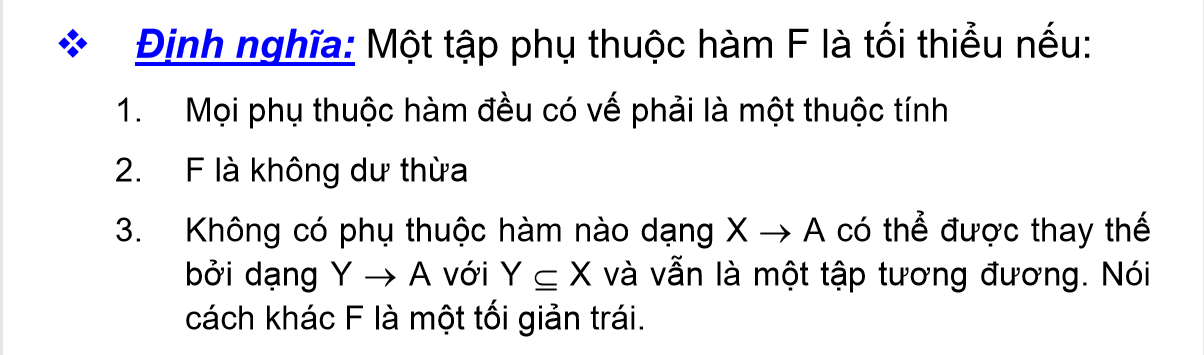
**Câu hỏi 1.24: Giải thích tại sao các bộ trong một quan hệ không cần thiết phải được sắp xếp theo một trình tự nhất định nào đó?**Sở dĩ không cần thiết phải sắp xếp các bộ theo một trật tự nhất định. Vì khi truy vấn thì nó sẽ so sánh từng bản ghi một lần lượt đến hết. Vậy nên ta k cần phải sắp xếp bản ghi làm gì cả **Câu hỏi 1.25: Giải thích tại sao trong một quan hệ không cho phép các bộ trùng nhau?**Sở dĩ không được trùng nhau là vì một thực thể không thể trùng nhau về tất cả các tập thuộc tính được. Nếu trùng nhau sẽ vi phạm về khóa chính , nó gây ra sự dư thừa dự liệu. **Câu hỏi 1.26: Trình bày những ưu điểm của mô hình quan hệ.**Cơ sở dữ liệu quan hệ là một kho dữ liệu riêng biệt. Khác với hệ cơ sở dữ liệu thứ bậc và hệ cơ sở dữ liệu mạng, trong hệ cơ sở dữ liệu quan hệ, người sử dụng và người thiết kế hoàn toàn không phải quan tâm tới cấu trúc cơ sở dữ liệu. Do đó, tính độc lập về cấu trúc cơ sở dữ liệu là ưu điểm nổi bật nhất của cơ sở dữ liệu quan hệ. Hơn thế nữa, do nó giải phóng cho ta vè mặt cần phải quan tâm tới khía cạnh vật lý của cơ sở dữ liệu, nên ta có thêm thời gian quan tâm tới khía cạnh logic của cơ sở dữ liệu.

Hệ cơ sở dữ liệu quan hệ có khả năng linh hoạt rất cao. Do đó, nó đòi hỏi ít việc lập trình để truy cập cơ sở dữ liệu hơn các loại cơ sở dữ liệu khác. Một trong những ưu thế mạnh của cơ sở dữ liệu dạng này là nó dễ tạo ra một giao diện thích hợp với người sử dụng hơn các cơ sở dữ liệu khác.  
+mô hình quan hệ có nhiều ưu điểm và được nhiều người quan tâm hơn cả, bởi lẽ, mô hình dữ liệu quan hệ có tính độc lập rất cao, lại dễ dàng sử dụng. Điều quan trọng hơn cả là, mô hình quan hệ được hình thức hóa toán học tốt, do đó được nghiên cứu phát triển và cho được nhiều kết quả lý thuyết cũng như ứng dụng trong thực tiễn.  
**Câu hỏi 1.27: Trình bày khái niệm phụ thuộc hàm và các đặc tính có ích của phụ thuộc hàm cho việc chuẩn hóa 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 hỏi 1.28: Trình bày sáu luật suy diễn phổ biến nhất cho các phụ thuộc hàm.** **Câu hỏi 1.29: Trình bày các luật suy diễn trong hệ tiên đề Amstrong. Giải thích tính đúng đắn và đầy đủ của hệ tiên đề**

***Câu hỏi 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.***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.  
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**.**

**Câu hỏi 1.31: Trình bày các khái niệm: phủ của một tập các phụ thuộc hàm, sự tương đương của hai tập phụ thuộc hàm và phủ không dư thừa.**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+.   
 => F được phủ nếu mọi phụ thuộc hàm trong F có thể được suy diễn từ G.  


**Câu hỏi 1.32: Trình bày định nghĩa phủ tối thiểu của một tập phụ thuộc hàm. Có phải mọi tập phụ thuộc hàm đều có phủ tối thiểu hay không?** **Mọi tập phụ thuộc hàm đều có phủ tối thiểu .  
  
Câu hỏi 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.** một thuộc tính A là 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+.  
Cho F là một tập các phụ thuộc hàm trên lược đồ R và cho X -> Y thuộc F.   
X -> Y được gọi là tối giản trái nếu X không chứa các thuộc tính dư thừa A.   
Một phụ thuộc hàm tối giản vế trái cũng được gọi là một phụ thuộc hàm đầy đủ.  
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 và tối giản phải và Y khác rỗng.