1. **Concepts and Architecture**

**Quản lý bằng tập tin:**

* Dữ liệu được lưu trữ trong một hoặc nhiều file riêng biệt.
* Dữ liệu sau đó được xử lý bằng các phần mềm, ứng dụng máy tính

Hạn chế:

* Dư thừa dữ liệu
* Mâu thuẫn dữ liệu
* Dữ liệu lưu trữ rời rạc
* Khó chia sẻ
* Kém hiệu quả trong truy xuất ngẫu nhiên, xử lý đồng thời,….
* Vấn đề an toàn, bảo mật

**Quản lý dữ liệu bằng CSDL:**

* Dữ liệu có thể chia sẻ
* Giảm bớt dư thừa dữ liệu
* Vấn đề không nhất quán dữ liệu trên các phiên bản khác nhau của cùng một tệp đã được giải quyết.

Các vấn đề khác:

* Cấu trúc dữ liệu cứng nhắc (Rigid data structure)
* Phụ thuộc dữ liệu vật lý
* Không hỗ trợ kiểm soát đồng thời

**DBMS**: là một software system có mục đích hỗ trợ quá trình định nghĩa, xây dựng, thao tác và chia sẻ cơ sở dữ liệu giữa những người dùng và ứng dụng khác nhau (defining, constructing, manipulating, and sharing databases).

**Database System** = Database + DBMS software

Roles in the Database Environment:

+ Database Administrator:

Cấp quyền truy cập vào cơ sở dữ liệu

Điều phối và giám sát việc sử dụng nó

Nhận tài nguyên phần mềm và phần cứng

+ Database Designers

Xác định dữ liệu được lưu trữ

Chọn cấu trúc thích hợp để biểu diễn và lưu trữ dữ liệu này

+ Application Programmers

Triển khai các thông số kỹ thuật này như một chương trình

+ End Users

Những người có công việc yêu cầu quyền truy cập vào cơ sở dữ liệu

**Kiến trúc three-schema**

**Mục tiêu của kiến trúc ba lược đồ**

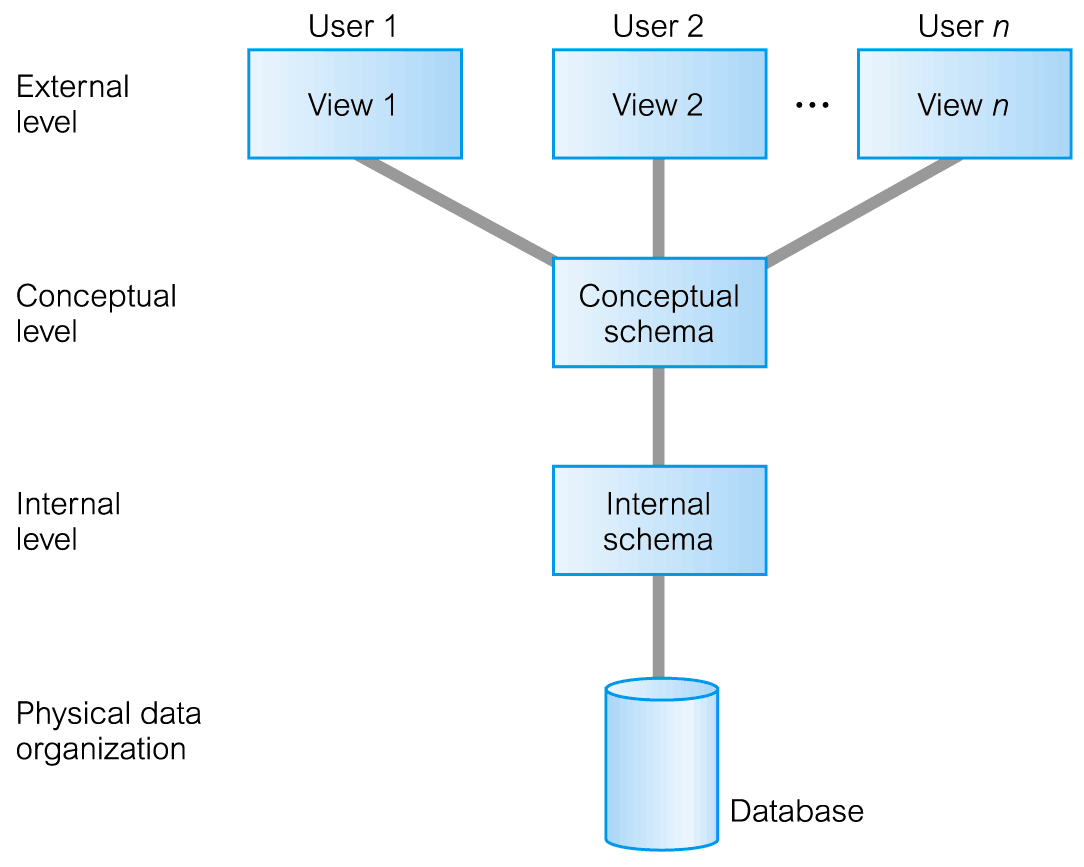
+ Tất cả người dùng sẽ có thể truy cập vào cùng một dữ liệu

+ Người dùng không cần biết chi tiết lưu trữ cơ sở dữ liệu vật lý

+ DBA sẽ có thể thay đổi cấu trúc lưu trữ cơ sở dữ liệu mà không ảnh hưởng đến chế độ xem của người dùng

+ Cấu trúc bên trong của cơ sở dữ liệu không bị ảnh hưởng bởi những thay đổi đối với các khía cạnh vật lý của lưu trữ

+ DBA sẽ có thể thay đổi cấu trúc khái niệm của cơ sở dữ liệu mà không ảnh hưởng đến tất cả người dùng



**External Level:**

+ Chế độ xem cơ sở dữ liệu của người dùng

+ Mô tả phần cơ sở dữ liệu có liên quan đến một người dùng cụ thể

**Conceptual Level**

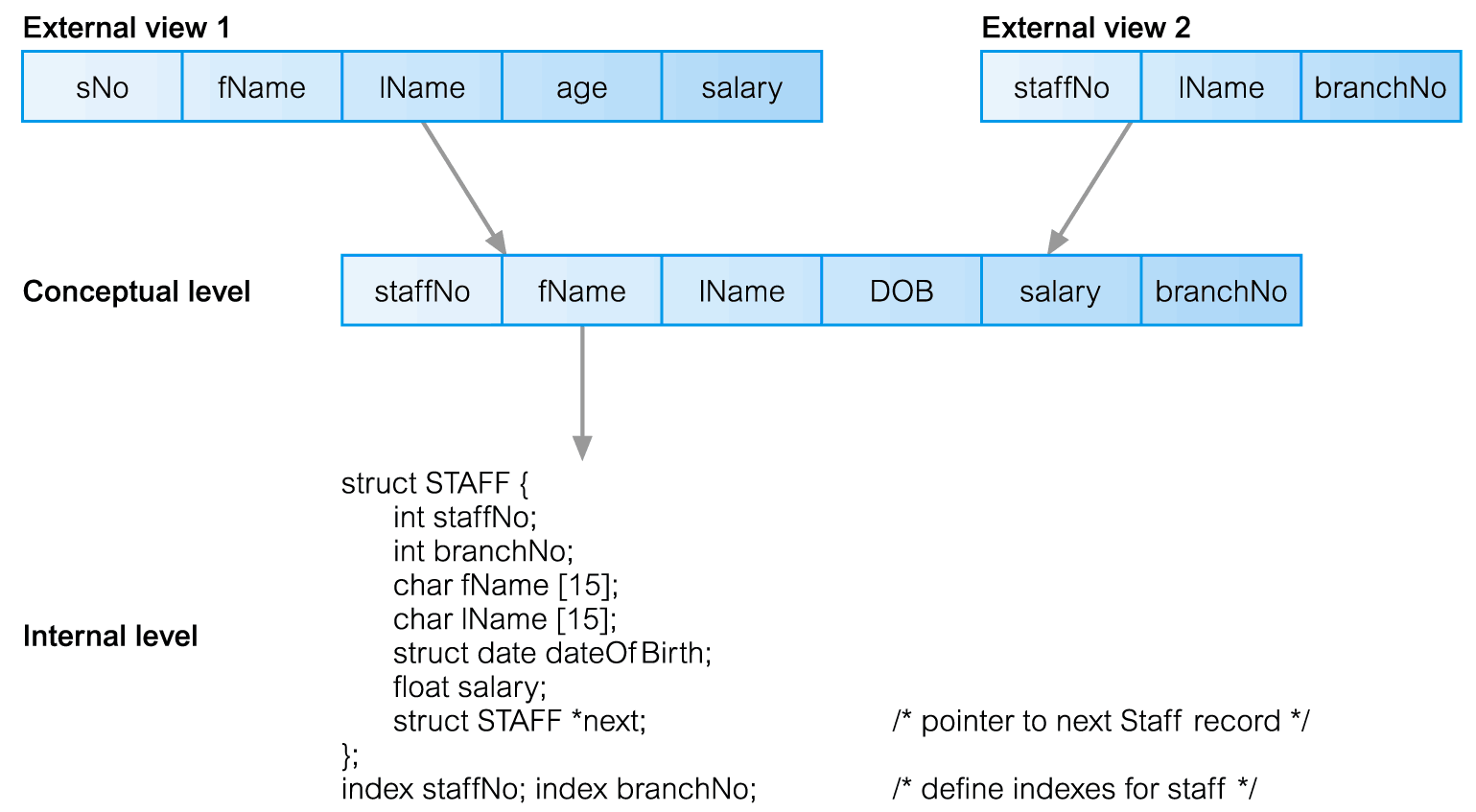
+ Chế độ xem cơ sở dữ liệu của cộng đồng

+ Mô tả dữ liệu nào được lưu trữ trong cơ sở dữ liệu và mối quan hệ giữa các dữ liệu

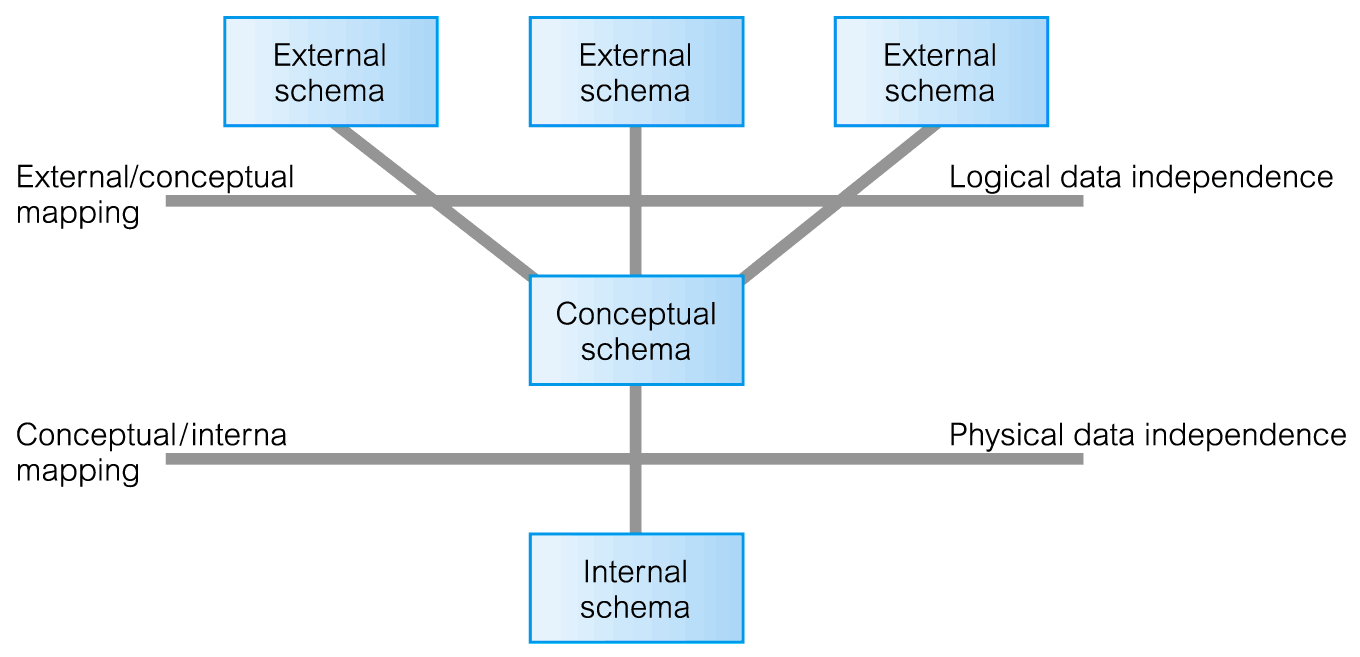
**Internal Level**

+ Biểu diễn vật lý của cơ sở dữ liệu trên máy tính.

+ Mô tả cách dữ liệu được lưu trữ trong cơ sở dữ liệu



**Độc lập dữ liệu (Data Independence):** là khả năng thay đổi lược đồ ở một cấp của hệ thống cơ sở dữ liệu mà không cần phải thay đổi lược đồ ở các cấp cao hơn tiếp theo



Database Languages:

Data Definition Language (DDL): cho phép khai báo và đặt tên các bảng, thuộc tính, mối quan hệ cùng các ràng buộc về tính toàn vẹn và bảo mật.

Data Manipulation Language (DML): cung cấp các thao tác select, insert, update, delete.

Data Control Language (DCL): xác định các hành động ko có trong DDL và DML như cấp quyền cho user, xác định lúc nào các hành động được thực hiện với database ko thể bị huỷ ngang

Data Manipulation Language (DML)

+ Procedural DML

+ Non-Procedural DML: SQL, QBE

**Data Model:** tập hợp các khái niệm mô tả dữ liệu, mối quan hệ, ràng buộc dữ liệu.

**Database Schema**: mô tả cơ sở dữ liệu trong quá trình thiết kế cơ sở dữ liệu và dự tính không có quá nhiều thay đổi.

**Schema Diagram:** lược đồ được hiện thị.

**Database State (Snapshot):** dữ liệu trong cơ sở dữ liệu tại một thời điểm cụ thể.

1. **Entity-Relationship (ER) Model**

**Các bước xây dựng database:**

* Thu thập, phân tích dữ liệu từ user -> yêu cầu dữ liệu, yêu cầu chức năng
* Thiết kế khái niệm: tạo ra lược đồ khái niệm (conceptual schema); mô tả các thực thể, mối quan hệ, ràng buộc; độc lập với lưu trữ và chi tiết thực hiện.
* Thiết kế luận lý: tạo ra database schema; ánh xạ mô hình dữ liệu tự động hoặc bán tự động với công cụ thiết kế csdl.
* Thiết kế vật lý: chỉ định cấu trúc lưu trữ nội bộ, tổ chức tệp, chỉ mục, đường dẫn truy cập và các tham số thiết kế vật lý cho tệp cơ sở dữ liệu.

ER-Model:

**Key**: 1 thực thể có thể có nhiều hơn 1 khoá

**Relationship**: có thể có nhiều hơn 1 relationship giữa các thực thể tham gia

**Recursion relationship**: Một thực thể có thể có mối quan hệ với chính nó

Phải chỉ định vai trò mà một thực thể tham gia đóng trong mỗi trường hợp mối quan hệ

**Structural constraints:**

+ cardinality ratio: 1:1, 1:M or M:1, M:N

+ participation constraint: mandatory (double line), optional (single line)

**Weak Entity Types:** Không có các thuộc tính chính của riêng chúng, được xác định bằng cách có liên quan đến các thực thể cụ thể từ một loại thực thể khác (ví dụ phòng 401 của TOÀ H6)

**Mối quan hệ xác định**: Liên quan đến một loại thực thể yếu với chủ sở hữu của nó.

Luôn luôn có 1 total participation constraint.

Các thực thể được xác định bằng sự kết hợp của:

+ Một phần khóa của loại thực thể yếu (partial key)

+ Thực thể cụ thể mà họ có liên quan đến trong loại thực thể xác định

ER Diagram: Entity type: Danh từ, Relationship type: Động từ

Alternative Dragrammatic Notations:

Problem with ER Model:

+ Ràng buộc về ngữ nghĩa:

Ví dụ tuổi của công nhân phải >= 18

+ Connection traps:

2 loại chính:

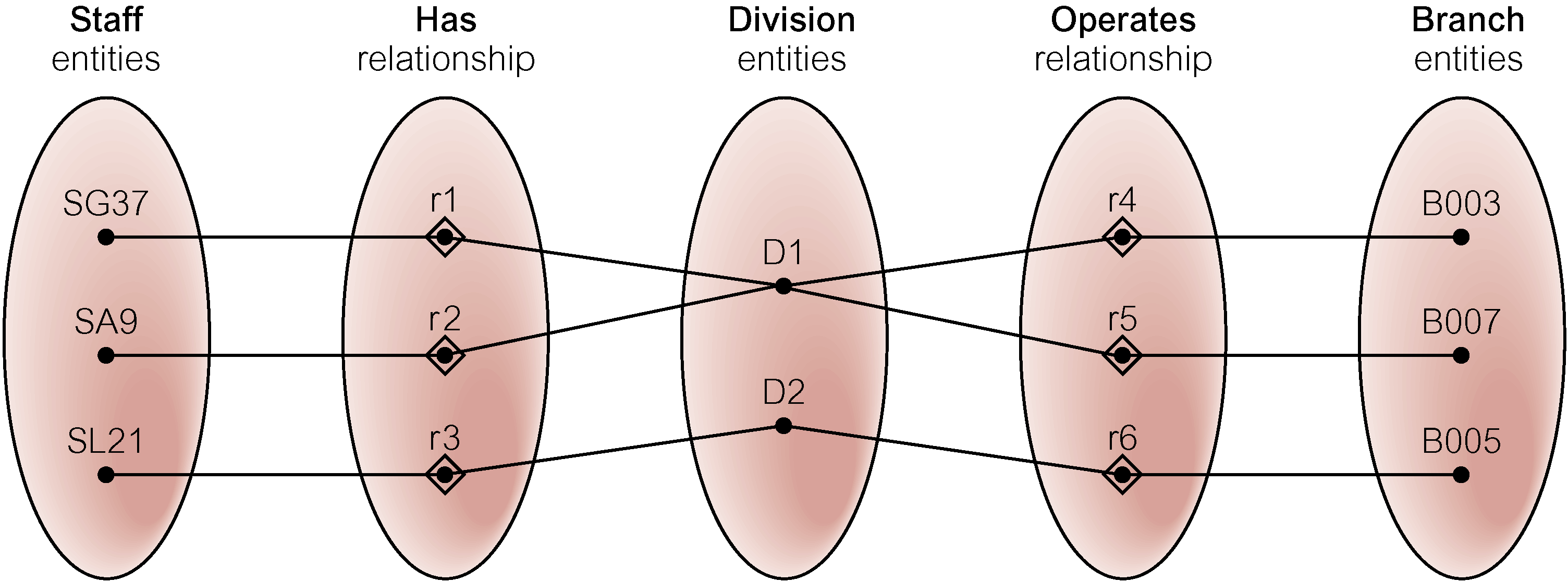
**fan traps:**

Trong đó một mô hình đại diện cho mối quan hệ giữa các loại thực thể, nhưng đường dẫn giữa các lần xuất hiện thực thể nhất định là không rõ ràng

Thường gặp: hai hoặc nhiều mối quan hệ 1: N xuất phát từ cùng một thực thể

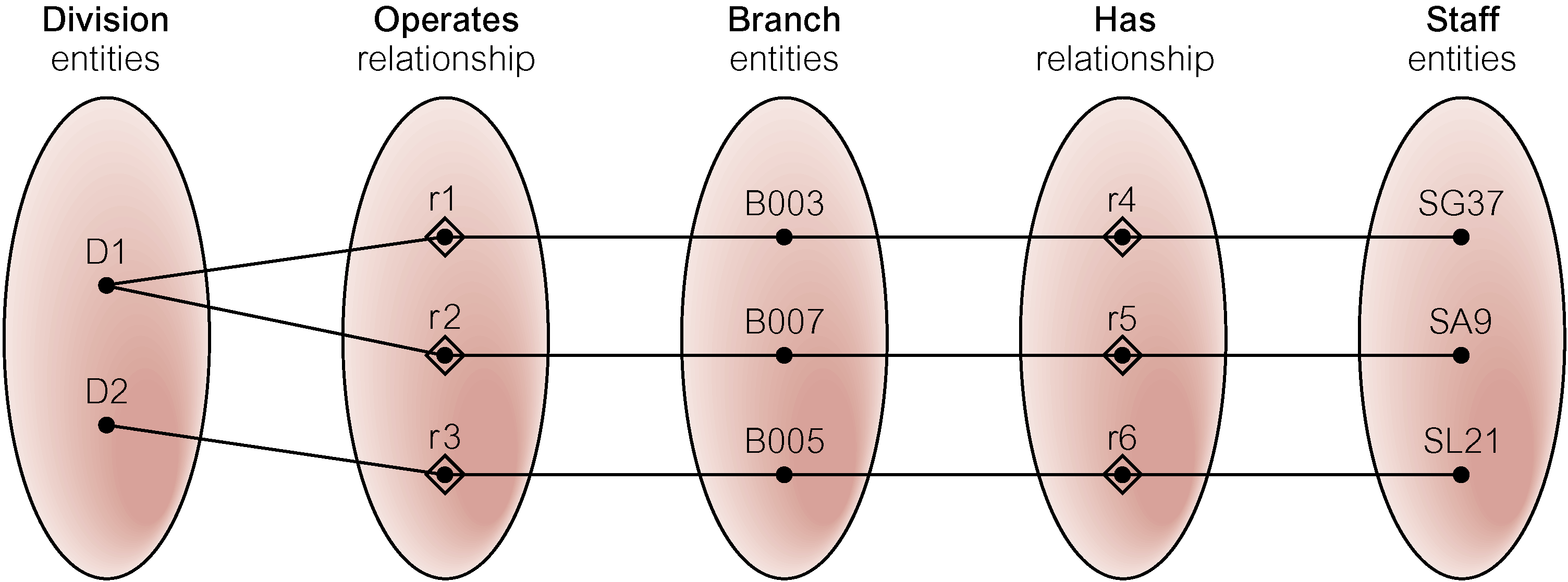
Giải quyết: Đổi vị trí





đổi thành:





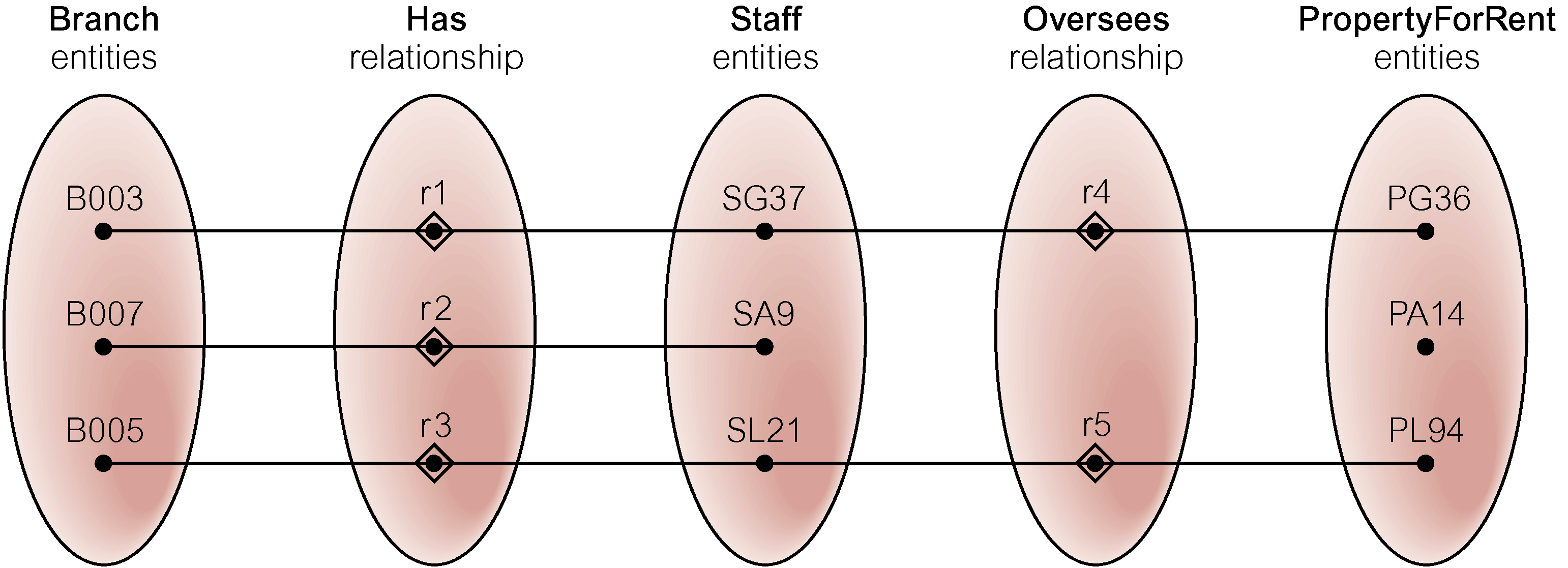
**chasm traps:**

Trường hợp mô hình gợi ý sự tồn tại của mối quan hệ giữa các loại thực thể, nhưng con đường không tồn tại giữa các lần xuất hiện thực thể nhất định

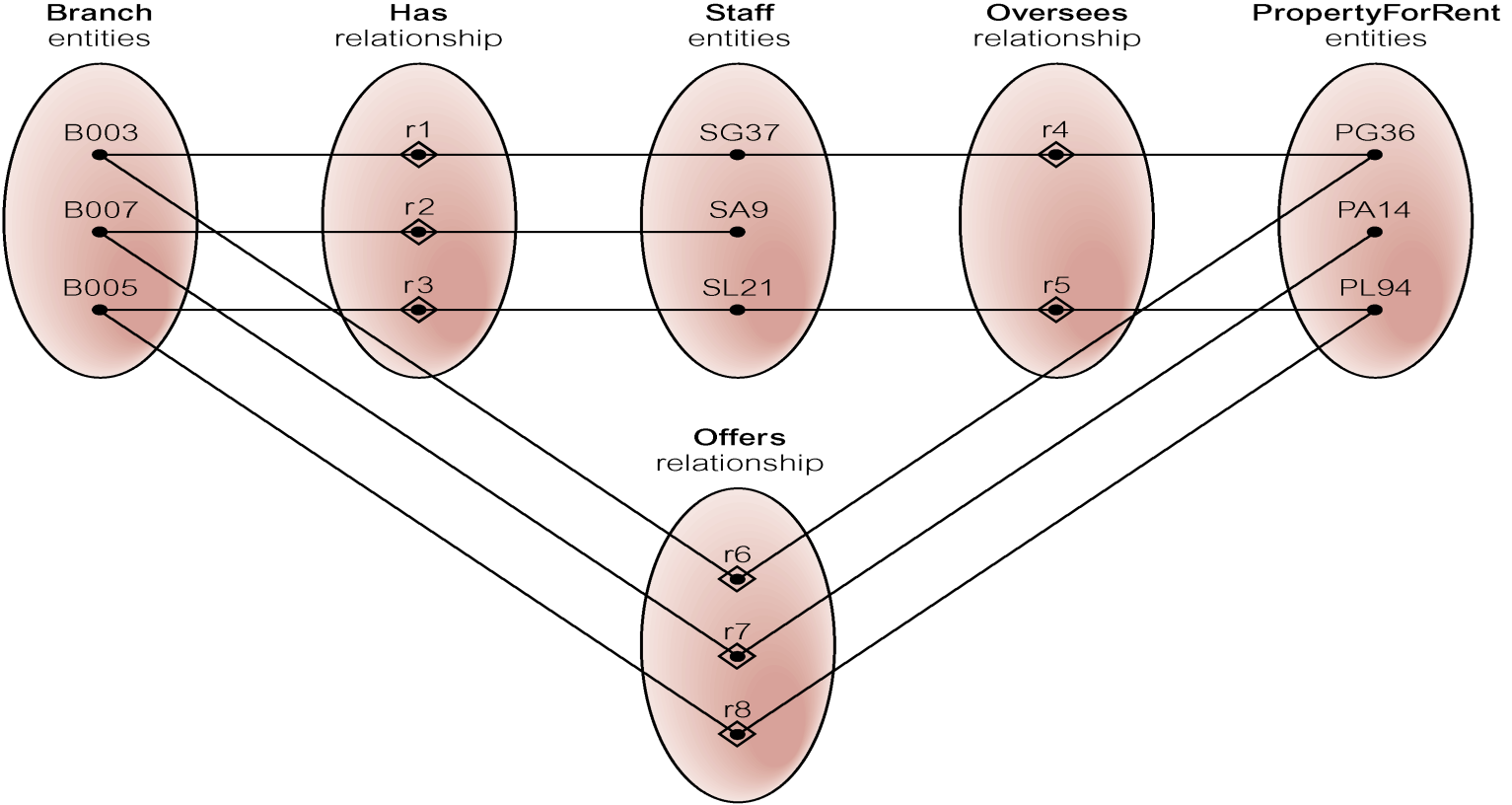
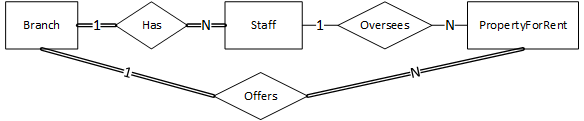
Thường gặp: tham gia tùy chọn

Giải quyết: Thêm relationship





đổi thành:



1. **EERD**
2. **Relational Data Model And Relational Mapping**

**Mapping:**

Step 1: Mapping of Regular Entity Types  
Step 2: Mapping of Weak Entity Types

- Khoá là kết hợp của khoá owner và khoá riêng phần của thực thể yếu  
Step 3: Mapping of Binary 1:1 Relationship Types  
Step 4: Mapping of Binary 1:N Relationship Types  
Step 5: Mapping of Binary M:N Relationship Types

**- Mandatory:**

E1 – 1:1 = E2

E1 – 1:N = E2

* E2 will include the prime attributes of E1 như là 1 khoá ngoại

E1 = 1:1 = E2

* A foreign key can be used in either relation

**- Optional**

E1 – 1:N – E2:

* **Thường** được biểu diễn bằng cách tạo 1 relation mới với khoá của E2

E1 – 1:1 – E2

* Tạo 1 relation mới chứa prime attributes của cả E1 và E2, và khoá là 1 trong 2 prime attribute hoặc combine.

E1 – N:M – E2

* **Luôn luôn** được biểu diễn bằng cách tạo 1 relation mới chứa prime attributes của 2 bên, và khoá là khoá kết hợp của 2 primary key.

Step 6: Mapping of Multivalued attributes

Với mỗi 1 thuộc tính đa trị A, tạo 1 relation mới. Relation này chứa 1 attribute corresponding với A, và có thêm thuộc tính khoá chính K của A. Khoá chính của relation vừa tạo là khoá kết hợp A-K.

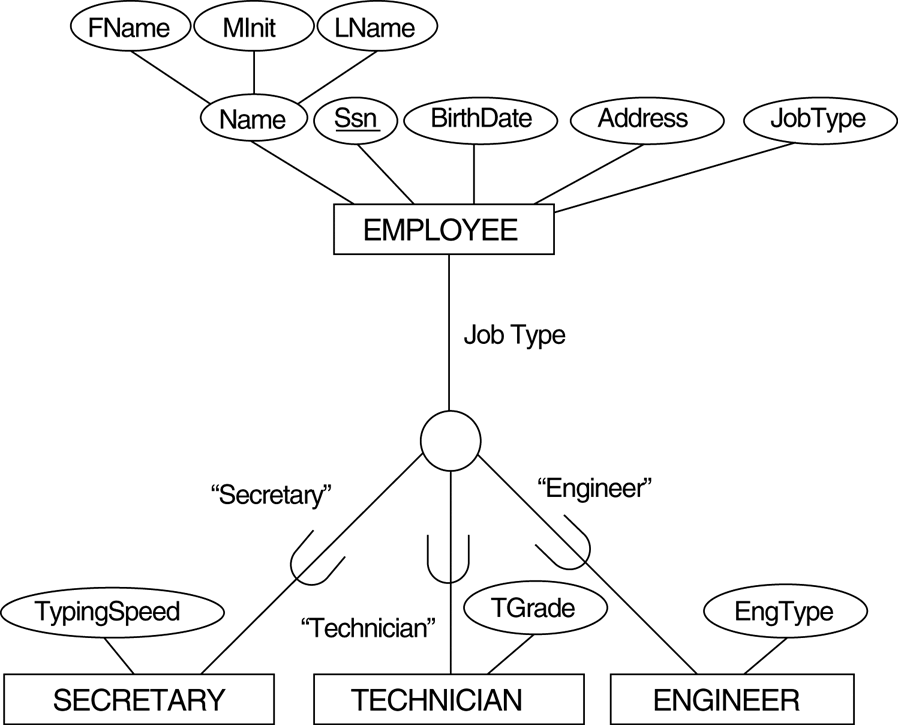
Step 7: Mapping of N-ary Relationship Types’

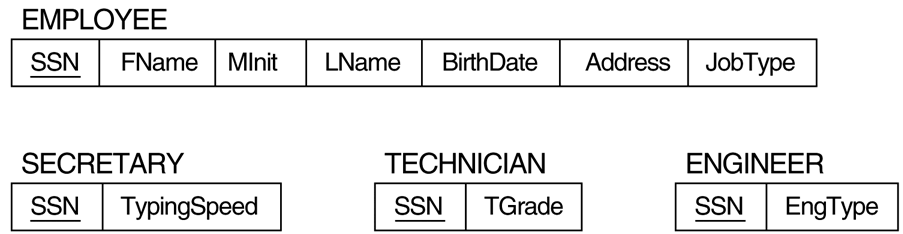
Tạo relationship mới S khoá của nó là kết hợp khoá của các thực thể tham gia.

Step 8: Options for Mapping Specialization or Generalization.

Option 8A: Multiple relations-Superclass and subclasses

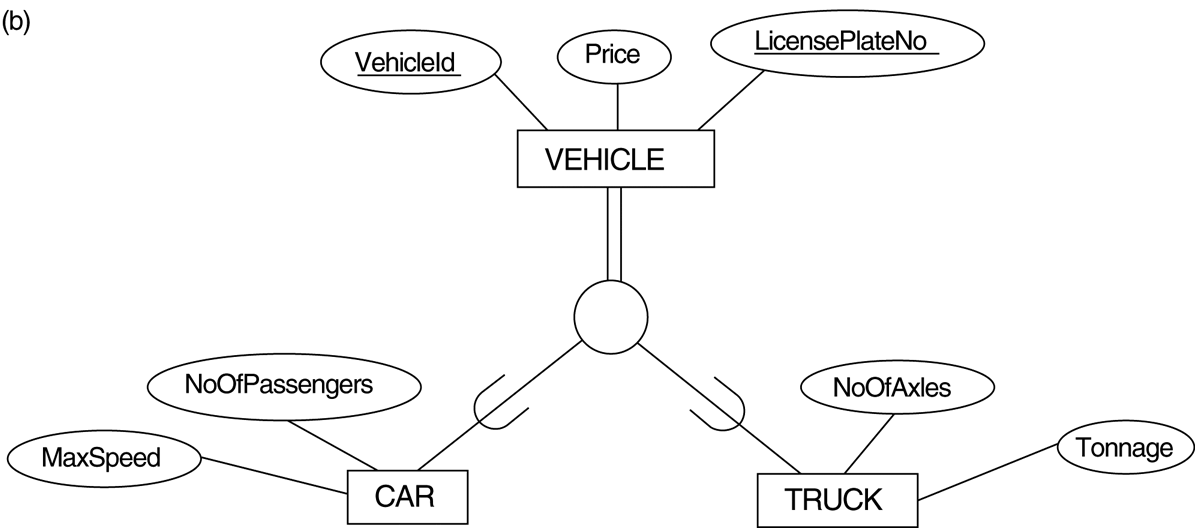
**Dùng được cho total lẫn partial, disjoint lẫn over-lapping.**

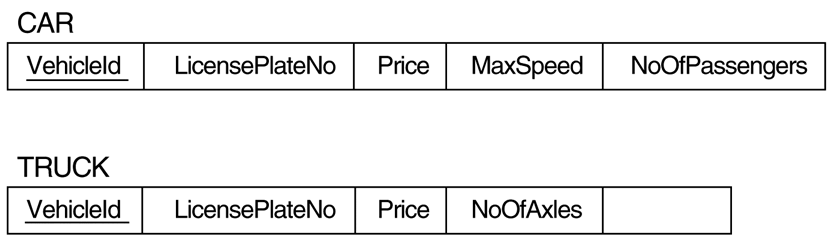




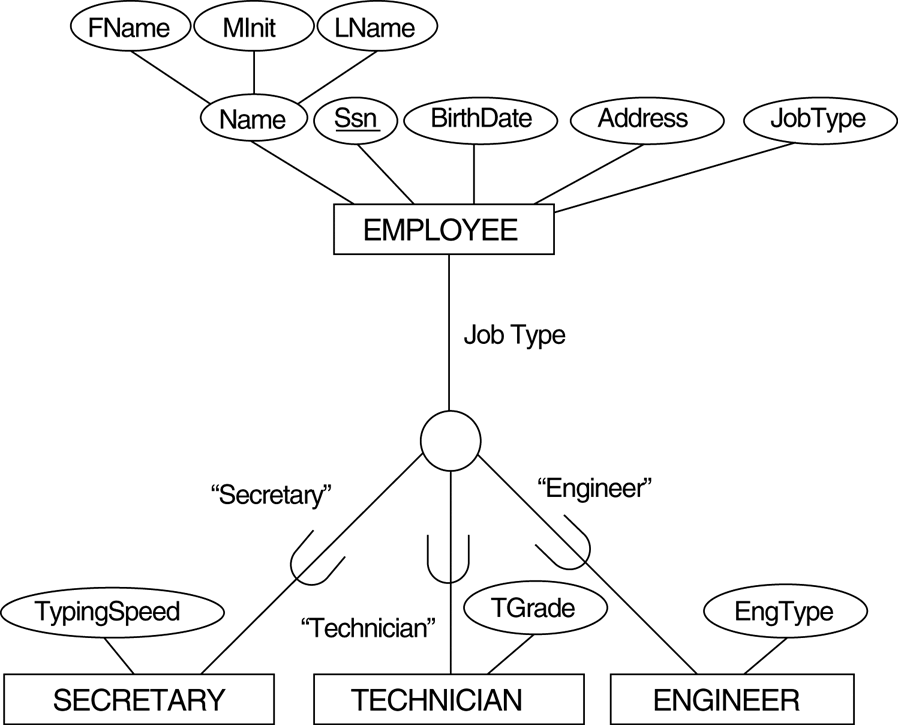
Option 8B: Multiple relations-Subclass relations only

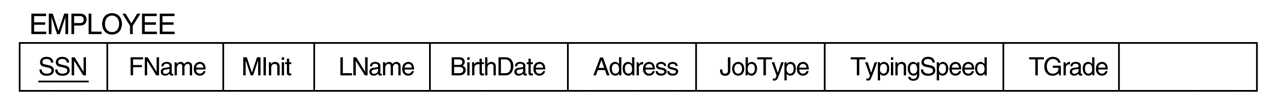
**Chỉ dùng trong trường hợp là total.**



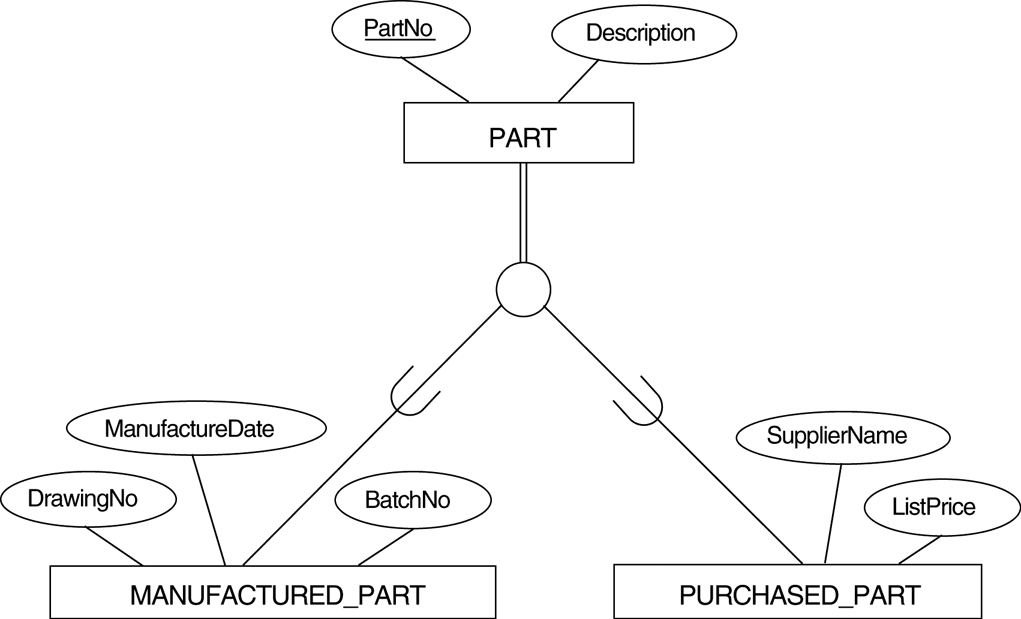


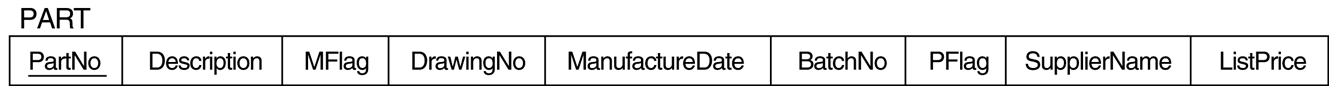
Option 8C: Single relation with one type attribute





Option 8D: Single relation with multiple type attributes





Step 9: Mapping of Union Types (Categories)

1. **Relational Algebra**

* Kết quả của một phép toán là **một quan hệ mới.**
* Một chuỗi các phép toán đại số quan hệ tạo thành một biểu thức đại số quan hệ
* Số tuple trong kết quả của một **SELECT** **nhỏ hơn (hoặc bằng)** số tuple trong quan hệ đầu vào R
* **SELECT có tính giao hoán**
* **Toán tử PROJECT remove các tuple trùng nhau** bởi vì kết quả của PROJECT phải là 1 set of tuple. (Mathematical sets không cho phép các phần tử trùng nhau)
* Số lượng tuple của kết quả phép **PROJECT LUÔN LUÔN** bé hơn hoặc bằng số lượng tuple của R. (nếu PROJECT có khoá thì bằng)
* **PROJECT không có tính giao hoán.**
* **Hội (UNION):** Duplicate tuple bị loại bỏ, phải type compatible
* **Giao (INTERSECTION):** phải type compatible
* **Hiệu (SET DIFFERENCE):** phải type compatible
* **Hội và giao có tính chất giao hoán, còn hiệu thì không.**
* **Tích đề các**: Kết quả của phép tích đề các:
  + Bậc của result relation = tổng của 2 relation tham gia
  + Số tuple bằng tích của số lượng tuple của 2 relation
  + **Không** cần “type compatible”

R và S tích đề cạc được P 🡪 |P| = |R| \* |S|; bậc P = bậc R + bậc S.

* **JOIN**: số tuple từ phép join bé hơn tích của số tuple thành phần
* **Theta-join**: General case
* **Equijoin**: only comparison operator used is =
* **Natural Join**: denoted by \*, get rid of the second attribute in an Equijoin condition
* **Complete set: select, project, hội, hiệu, tích đề các.**
* Phép chia
* Count đếm số hàng, **không loại bỏ trùng lặp.**

1. SQL

* Basic SQL:
  + Data Definition Language(DDL): Create, Alter, Drop
  + Data Manipulation Language (DML): Select, Insert, Update, Delete
  + Data Control Language (DCL): Commit, Rollback, Grant, Revoke

1. Functional Dependencies and Normalization

“Goodness” measure:

Dư thừa thông tin trong tuple

Cập nhật các điểm bất thường: sửa đổi, xóa, chèn

Giảm bớt các giá trị NULL trong tuple

Không cho phép khả năng tạo các bộ giá trị giả

* NORMALIZATION

1NF,2NF,3NF,BCNF,4NF,5NF

X 🡪 A là phụ thuộc hàm đầy đủ (full functional dependency) nếu không tồn tại Y là con X mà Y 🡪 A.

Có hai thuộc tính quan trọng của phân rã:

1) không thêm hoặc không mất kết nối tương ứng

2) bảo toàn phụ thuộc hàm

Lưu ý rằng (1) là vô cùng quan trọng và không thể mất. (2) ít nghiêm ngặt hơn và có thể bị mất

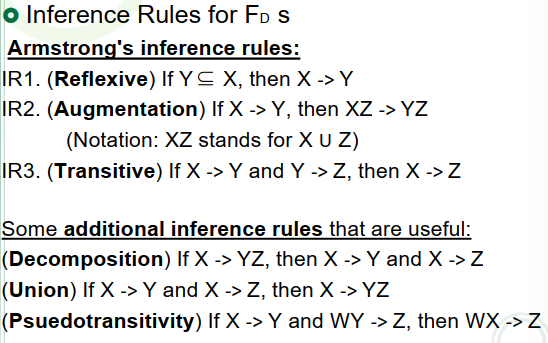
+ **Direct dependency**: Từ khoá có thể suy ra trực tiếp các trường khác

+ **Indirect dependency**: Giá trị của 1 trường không thể xác định trực tiếp bằng khoá

+ **Partial dependency:**

Composite determinant: cần có nhiều hơn 1 giá trị để xác định giá trị của trường khác, combination của các giá trị được gọi là 1 composite determinant.

Partial dependency: nếu giá trị của một thuộc tính không phụ thuộc vào toàn bộ định thức tổng hợp mà chỉ phụ thuộc vào một phần của nó, thì mối quan hệ được gọi là phụ thuộc một phần

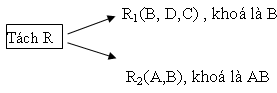


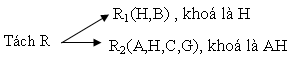
Normalization:

Prime attribute:

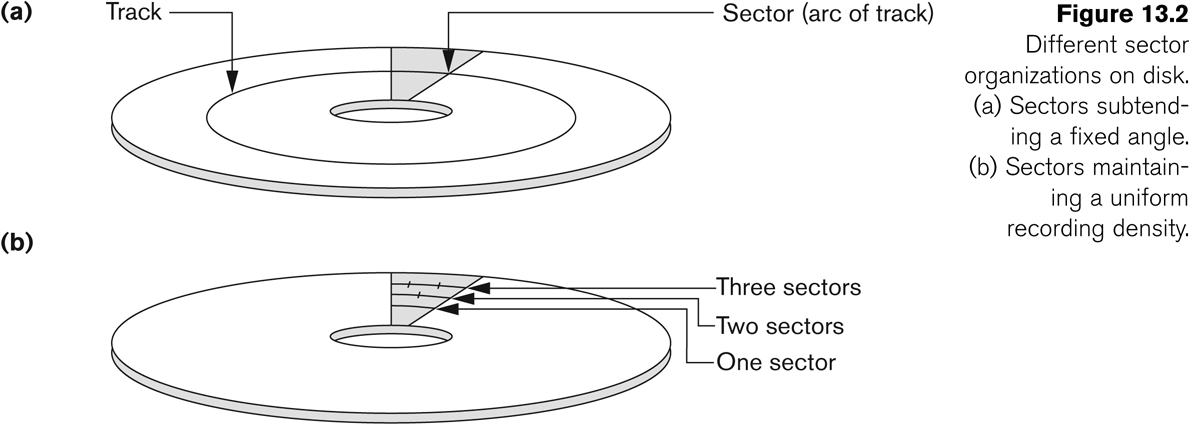
Non-prime attribute:

* **1NF**: Loại bỏ các thuộc tính phức hợp, thuộc tính đa trị, và các relation lồng nhau
  + Mỗi ô bảng chứa một giá trị duy nhất
  + Mỗi tuple là duy nhất
* **2NF**: (**KHOÁ ĐƠN => LÀ 2NF)**
  + Là 1NF và
  + Các thuộc tính không khoá phải phụ thuộc hàm đầy đủ vào khoá chính
  + ***Một quan hệ ở dạng chuẩn 2NF nếu thoả mãn 1 trong các điều kiện sau:*** 
    - ***Khoá chính chỉ gồm một thuộc tính***
    - ***Bảng không có các thuộc tính không khoá***
    - ***Tất cả các thuộc tính không khoá phụ thuộc hoàn toàn vào tập các thuộc tính khoá chính***
  + ***Ví dụ***
    - ***Ví dụ1: Cho quan hệ R = (ABCD), khoá là AB và tập phụ thuộc hàm F = {AB -> C, AB -> D}là quan hệ đạt chuẩn 2NF.***
    - ***Ví dụ2: Cho quan hệ R = (ABCD), khoá là AB và tập phụ thuộc hàm F = {AB -> C, AB -> D, B -> DC} là quan hệ không đạt chuẩn 2NF vì có phụ thuộc hàm***
    - ***B -> DC là phụ thuộc hàm bộ phận (phụ thuộc hàm không đầy đủ) vào khoá. Khi đó ta đưa về dạng chuẩn 2NF như sau:***



* **3NF**:
  + Là 2 NF và
  + Các thuộc tính không khoá phải phụ thuộc trực tiếp vào khoá chính
  + ***Ví dụ***
    - ***Ví dụ 1: Cho quan hệ R = (ABCDGH, khoá là AB và tập phụ thuộc hàm F = {AB -> C, AB -> D, AB -> GH} là quan hệ đạt chuẩn 3NF.***
    - ***Ví dụ 2: Cho quan hệ R = (ABCDGH) , khoá là AB và tập phụ thuộc hàm F = {AB -> C, AB -> D, AB -> GH, G -> DH} là quan hệ không đạt chuẩn 3NF vì có phụ thuộc hàm G -> DH là phụ thuộc hàm gián tiếp vào khoá. Khi đó ta đưa về dạng chuẩn 3NF như sau:*** **
* **BCNF**:
  + Là 3NF và
  + Không có thuộc tính khoá mà phụ thuộc hàm vào thuộc tính không khoá.
  + ***Ví dụ***
    - ***Ví dụ 1: Cho quan hệ R = (ABCDGH, khoá là AB và tập phụ thuộc hàm F = {AB -> C, AB -> D, AB -> GH} là quan hệ đạt chuẩn BCNF.***
    - ***Ví dụ 2: Cho quan hệ R = (ABCDGH) , khoá là AB và tập phụ thuộc hàm F = {AB -> C, AB -> D, AB -> GH, H -> B} là quan hệ không đạt chuẩn BCNF vì có thuộc tính khoá B phụ thuộc hàm vào thuộc tính không khoá H. Khi đó ta đưa về dạng chuẩn BCNF như sau:*** 

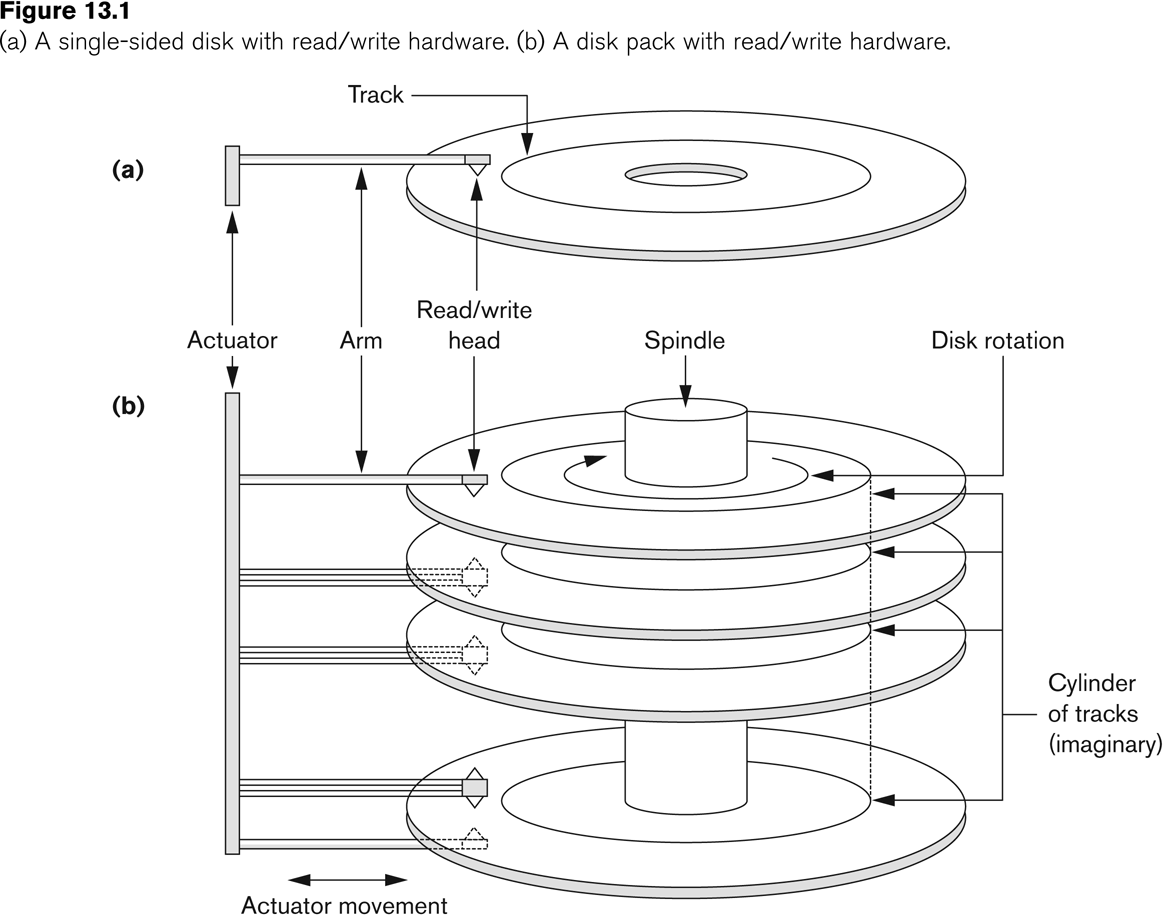
1. **Data Storage..**

* Thiết bị lưu trữ thứ cấp được ưu tiên cho khả năng lưu trữ cao và chi phí thấp.
* Dữ liệu được lưu trữ dưới dạng vùng từ hóa trên bề mặt đĩa từ.
* Disks divided in to tracks(những hình tròn đồng tâm), track divided into smaller block or sectors.
* 
* A physical disk block address consists of:

• a cylinder number

• the track number or surface number (within the cylinder)

• and block number (within track)

* 
* **Record**: Fixed and variable length records
  + Record chứa field có giá trị của 1 loại cụ thể.
* **Blocking**: Chứa 1 số lượng của record trong 1 block trên đĩa.
  + **Blocking factor (bfr):** số record trên 1 block.
* **Spanned Record**: Record vượt quá kích thước của 1 hoặc nhiều block do đó span(kéo dài) 1 số block.
* Tệp là một chuỗi các bản ghi (record), trong đó mỗi bản ghi là một tập hợp các data values (hoặc data item)
* A file can have fixed-length records or variable-length records.
* File records can be unspanned or spanned
* **Unordered Files:**
  + Also called a heap or a pile file.
  + New record được chèn vào cuối file.
  + **Linear search (O(n/2))**
  + Chèn record khá hiệu quả
  + Việc đọc các bản ghi theo thứ tự của một trường cụ thể **yêu cầu sắp xếp** file records.
* **Ordered files:**
  + Called a sequential file.
  + File records are kept sorted by the values of an ordering field.
  + Chèn khá khó khăn: phải chèn đúng vị trí.
  + Người ta thường giữ một tệp tin tràn (hoặc giao dịch) không có thứ tự riêng biệt cho các bản ghi mới để cải thiện hiệu quả chèn; tệp này được hợp nhất định kỳ với tệp được sắp xếp chính.
  + **Binary search (O((log2)n))**
  + Đọc các records khá hiệu quả.
* **Indexes as Access Paths**
* Chỉ mục single level là một file bổ trợ giúp tìm kiếm bản ghi trong tệp dữ liệu hiệu quả hơn.
  + Chỉ mục thường được chỉ định trên một trường của tệp (mặc dù nó có thể được chỉ định trên một số trường)
  + Một dạng của chỉ mục là một tệp gồm các mục nhập <field value, pointer to record> được sắp xếp theo giá trị trường
  + Chỉ mục được gọi là đường dẫn truy cập trên trường
  + **Primary Index:**
    - Được xác định trên một tệp dữ liệu **có thứ tự**
    - **Tệp dữ liệu được sắp xếp thứ tự trên một trường khóa**
    - Includes one index entry for each block in the data file; the index entry has the key field value for the first record in the block, which is called the **block anchor**
    - A similar scheme can use the last record in a block
    - A primary index **is a nondense (sparse) index**, since it includes an entry for each disk block of the data file and the keys of its anchor record rather than for every search value
  + **Clustering Index**
    - Được xác định trên một tệp dữ liệu **có thứ tự**
    - **Tệp dữ liệu được sắp xếp thứ tự trên một trường không khóa** => distinct value for each record.
    - Includes one index entry for each distinct value of the field; the index entry points to the first data block that contains records with that field value
    - Là **nondense index**, chèn và xoá tương đối đơn giản với Clustering Index.
  + Secondary Index
    - A secondary index provides a secondary means of accessing a file for which some primary access already exists.
    - Chỉ mục thứ cấp có thể nằm trên trường là khóa ứng viên và có giá trị duy nhất trong mọi bản ghi hoặc không phải khóa có giá trị trùng lặp.
    - Là **dense index.**
* Multi-level Indexes

1. Database Security

* 3 basic concepts:
  + Authentication (Xác thực)
  + Authorization (Uỷ quyền)
  + Access Control (Kiểm soát truy cập)
* Threats to databases
  + Loss of integrity (Mất tính toàn vẹn) - invalid or corrupted data.
  + Loss of availability - data or system cannot be accessed.
  + Loss of confidentiality (Mất tính bảo mật) - secrecy, data critical to the organization.
* To protect databases, four kinds of countermeasures can be implemented:
  + Access control
  + Inference control
  + Flow control
  + Encryption
* Two types of database security mechanisms:
  + Discretionary security mechanisms (Tuỳ ý)
  + Mandatory security mechanisms (Bắt buộc)
* Cơ chế bảo mật của DBMS phải bao gồm các điều khoản để hạn chế quyền truy cập vào cơ sở dữ liệu nói chung
  + Chức năng này được gọi là **access control** và được xử lý bằng cách tạo tài khoản người dùng và mật khẩu để kiểm soát quá trình đăng nhập bởi DBMS
* Vấn đề bảo mật liên quan đến cơ sở dữ liệu là kiểm soát quyền truy cập vào **cơ sở dữ liệu thống kê** (statistical database), được sử dụng để cung cấp thông tin thống kê hoặc tóm tắt các giá trị dựa trên các tiêu chí khác nhau.
  + Các biện pháp đối phó với vấn đề bảo mật **cơ sở dữ liệu thống kê** được gọi là các **biện pháp kiểm soát suy luận (inference control**

**measures)**

**Các cách giải quyết:**

* **Kiểm soát truy cập (access control):** dùng tài khoản và mật khẩu để kiểm soát truy cập. User là ai, được phép làm gì trong hệ thống.
* **Kiểm soát suy luận (inference control):** kiểu soát việc truy cập vào dữ liệu thống kê (statistical database). Dữ liệu X bí mật, nhưng có 1 hàm và dữ liệu Y có thể tính được X lại công khai 🡪 suy ra được dữ liệu X.
* **Kiểm soát lưu lượng (flow control):** ngăn chặn thông tin rò rỉ theo nhiều đường đến người không được phép truy cập. Các kênh bí mật làm rò rỉ thông tin gọi là **covert channels**. Dòng thông tin từ file A sang file B. File A bí mật nhưng ai đó có quyền xem file A lại chép sang file B là file ai cũng có quyền truy cập sẽ làm lộ thông tin.
* **Mã hoá (encryption):** bảo vệ các dữ liệu nhạy cảm khi được truyền qua mạng.

DBMS hỗ trợ các hệ thống con gồm hệ thống uỷ quyền và hệ thống an toàn để chống lại các truy cập bất hợp lệ. Gồm 2 cơ chế:

* **Cơ chế bảo mật tuỳ quyền (Discretionary security mechanisms):** user có quyền gì được làm quyền đó.
* **Cơ chế bảo mật bắt buộc (Mandatory security mechanisms):** không dựa vào hệ thống quản lý quyền.
* Hỗ trợ kiểm soát truy cập (access control).

**DBA:** người quản trị cơ sở dữ liệu (Database administrator).

* Chịu trách nhiệm: gán quyền cho các user cần truy cập hệ thống, phân loại người dùng và dữ liệu phù hợp.
* Chịu trách nhiệm về bảo mật tổng thể của hệ thống.
* Có tài khoản DBA trong DBMS có thể gọi là tài khoản hệ thống hay superuser. Tài khoản cho phép: tạo tài khoản, gán quyền, huỷ quyền, tuỳ chỉnh mức độ bảo mật (điều chỉnh truy cập bắt buộc).

Hệ sơ sở dữ liệu có trách nhiệm lưu lại tất các các truy cập cơ sở dữ liệu của các user trong một session (system log).

**Discretionary Access Control: Chủ sở hữu có toàn quyền, có thể gán và thu hồi quyền từ người khác.**

**Quy tắc chung:**

* User có thể bảo vệ cái thuộc sở hữu của mình.
* Người sở hữu có thể gán quyền truy cập cho những người khác.
* Người sở hữu có thể định nghĩa các loại truy cập và gán cho các user khác.

Cơ chế chính thông qua granting (gán quyền) và revoking (thu hồi quyền).

**Các loại quyền:**

* **Mức tài khoản (account level):** các quyền liên quan tới định nghĩa bảng, định nghĩa cơ sở dữ liệu, không gắn với 1 bảng cụ thể nào. (CREATE SCHEMA/TABLE/VIEW, ALTER, DROP, MODIFY, SELECT nói chung là truy vấn cơ sở dữ liệu).
* **Mức quan hệ (relation level) hoặc mức bảng (table level):** các quyền gắn với 1 bảng cụ thể (các quyền truy vấn trên 1 bảng xác định – SELECT …), 1 dòng/cột cụ thể (SELECT, UPDATE, DELETE, INSERT, REFERENCES lên từng bảng).

Để hiện thực cơ chế gán, huỷ quyền dùng mô hình ma trận. Mỗi hàng là **subject** (users, accounts, programs), mỗi cột là **object** (relations, records, columns, views, operations). Điểm giao nhau sẽ xác định xem có quyền như thế nào của subject với object.

**Để điều khiến gán và thu hồi quyền, thực hiện theo cơ chế:**

* Owner có tất cả các quyền.
* Người sở hữu có thể truyền quyền (**pass**) cho các user khác thông qua **granting** (gán quyền) lệnh là **GRANT**.
* Có thể lấy lại (**take back**) các quyền đó qua câu lệnh **REVOKE**.

Cơ chế bảo vệ dữ liệu thông qua view: A có bảng dữ liệu, muốn cho B xem vài thuộc tính 🡪 tạo view chỉ chứa các thuộc tính đó và cho B xem.

Giả sử A gán quyền cho B, B gán quyền cho C. Nếu A thu hồi quyền của B thì C cũng bị thu hồi tuy nhiên có thể tốn nhiều chi phí và một vài lỗi. Một vài DBMS sẽ tự động thu hồi, một vài thì user phải tự xử lý (tự tìm C để xoá).

Giải pháp chống lan truyền quyền mất kiểm soát:

* Theo chiều ngang: A gán cho B với hệ số 2 🡪 B chỉ gán tối đa quyền cho thêm 2 user nữa, 2 user sau cũng không có quyền lan truyền quyền.
* Theo chiều dọc: số cấp tối đa được gán quyền.

Lệnh gán quyền: **GRANT** <TÊN\_BẢNG> **TO** <USER>;

**GRANT** <LỆNH> **ON** <TÊN\_BẢNG> (<DANH SÁCH CỘT>) **TO** <USER>;

Lệnh thu hồi quyền: **REVOKE** <LỆNH> **ON** <TÊN\_BẢNG> **FORM** <USER>;

Điểm yếu của **Discretionary Access Control:**

* Không ngăn chặn được dòng thông tin đi từ đối tượng này sang đối tượng khác vì đối xử các đối tượng là như nhau.

**Trojan Horse:** A có file A, B có file B, B không được đọc file A. B cho phép A đọc file B. B tạo ra 1 chương trình có gắn Trojan Horse. Khi A chạy chương trình, cùng lúc Trojan Horse sẽ chạy, dùng user A truy cập file A, ghi dữ liệu từ A sang B 🡪 dòng thông tin chảy từ A sang B.

**Mandatory Access Control: No read-up & no write-down.**

* Phân loại user dựa trên level, phân loại dữ liệu dựa theo độ nhạy cảm. Level càng thấp thì mức độ càng quan trọng.
* Chỉ được đọc dữ liệu mức độ thấp hơn hoặc bằng, chỉ được phép ghi lên các mức độ cao hơn hoặc bằng.
* Yếu điểm: khó áp dụng thực tế do việc phân loại chi tiết từng dữ liệu rất khó, chỉ thích hợp cho môi trường quân đội.

**Mô hình Bell – LaPudula:** Chi dữ liệu thành 4 cấp: Top secret (TS), secret (S), confidential (C), unclassified (U). TS > S > C > U. Gán user cho từng cấp này.

**Multilevel Relation:**

* Mỗi thuộc tính kèm theo 1 cấp độ, cuối mỗi hàng sẽ có thêm TC lưu thuộc tính phân loại chung của hàng đó (là cấp độ cao nhất trong các thuộc tính của hàng).
* Một user cấp X khi truy xuất dữ liệu chỉ thấy được dữ liệu cấp thấp hơn hoặc bằng, các giá trị còn lại sẽ là null.
* Mỗi user tuỳ cấp độ sẽ nhận được giá trị khác nhau của cùng 1 câu truy vấn.
* Tính chất quan hệ đa mức: No read-up, no write-down.
* Ràng buộc toàn vẹn thực thể: các khoá ko được phép null.

1. Mục tiêu của kiến trúc lược đồ 3 lớp (three-schema architecture) là:

Hỗ trợ tính độc lập dữ liệu luận lý và vật lý

2. Các bước trong quá trình thiết kế một cơ sở dữ liệu:

Thu thập và phân tích yêu cầu, thiết kế ý niệm, thiết kế luận lý, thiết kế vật lý

3. Trong ERD/EERD, phát biểu đúng là:

Bậc của một kiểu mối liên kết (relationship type) là số lượng các kiểu thực thể (entity type) tham gia vào kiểu mối liên kết đó

4. Shared file approach, phát biểu đúng

Vấn đề cấu trúc dữ liệu không uyển chuyển (rigit data structure) xuất hiện và cần phải giải quyết

5. Phát biểu đúng

Ánh xạ các thuộc tính đa trị luôn tạo ra một quan hệ mới

Các thuộc tính dẫn xuất không nên cho phép người dùng nhập dữ liệu trực tiếp

Sai: Có thể tồn tại một kiểu mối liên kết giữa một kiểu thực thể và một kiểu mối liên kết khác

6. Ràng buộc nào có thể bị vi phạm khi thực hiện tác vụ DELETE

Ràng buộc tham chiếu (Referential constraint)

7. Việc thiết kế ERD/EERD nằm ở giai đoạn:

Thiết kế ý niệm (Conceptual design)

8. Chọn phát biểu SAI: Khái niệm View trong CSDL quan trọng là vì:

View có thể cải thiện tốc độ thực thi các câu truy vấn

View hỗ trợ độc lập dữ liệu (data independence)

View cho phép schema được thay đổi mà không cần phải biên dịch lại các ứng dụng sẵn có

View hỗ trợ quản lý truy cập (access control) bằng việc cho phép người dùng chỉ được truy xuất một phần dữ liệu của cơ sở dữ liệu

9. Cho quan hệ EMP(ssn, name, dno) và ba câu truy vấn sau:

Q1: SELECT name

FROM EMP

WHERE dno IN (10,20);

Q2: SELECT name

FROM EMP

WHERE dno = 10 OR dno = 20;

Q3: SELECT name

FROM EMP

WHERE dno = 10;

UNION

SELECT name

FROM EMP

WHERE dno = 20;

Đáp án: Câu truy vấn Q1 và Q2 tương đương

10. Cơ chế bảo mật trong đó hệ thống tự xác định các hành vi (đọc, ghi) cho phép của một chủ thể (subject) trên một đối tượng dữ liệu dựa sự phân lớp bảo mật của các chủ thể và đối tượng dữ liệu

Mandatory Access Control (MAC)

11. Nhiệm vụ của người quản trị cơ sở dữ liệu (DBA) là:

Quản lý việc truy xuất vào cơ sở dữ liệu .. người dùng

12. Chọn phát biểu đúng về mô hình dữ liệu quan hệ (RELATION DATA MODEL)

A. Bậc (degree) của một quan hệ (relation) là số lượng bộ dữ liệu (tuple) có trong quan hệ

B. Siêu khoá của quan hệ cũng là khoá của quan hệ

C. Thứ tự các bộ dữ liệu (tuple) là quan trọng và phải khác biệt

D. ABC đúng

E. ABC sai

13. Các bước chính trong quá trình xử lý câu truy vấn viết bởi ngôn ngữ SQL là:

Phân tích cú pháp và chuyển đổi, tối ưu hoá câu truy vấn, …. code và phiên dịch, thực thi câu truy vấn

14. Cơ sở dữ liệu của các hệ thống thông tin phải đối mặt với mối nguy hại (threats) nào sau đây:

Loss of integrity (Mất tính toàn vẹn) - invalid or corrupted data.

Loss of availability (Mất tính sẵn sàng) - data or system cannot be accessed.

Loss of confidentiality (Mất tính bí mật) - secrecy, data critical to the organization.

15. Ưu điểm của mô hình quan hệ thực thể (ER Model) là:

Yêu cầu của người dùng được mô tả một cách hình thức và rõ ràng

Tương đối dễ hiểu đối với người dùng thông thường

Mô hình quan hệ thực thể độc lập với các hệ quản trị csdl

16. Đặc điểm của một quan hệ có cấu trúc tốt (well-structured) là:

Không có dư thừa dữ liệu

Không chứa giá trị NULL

Có một vài bất thường khi người dùng thực hiện thao tác cập nhật dữ liệu

Tất cả các câu trên đều sai

17. Company

Q1: from Employee E S where E.superSSN = S.SSN

Q2: from (Employee E LEFT OUTER JOIN Employee S ON E.superSSN = S.SSN

Nếu có ít nhất 1 hàng trong bảng Employee thoả E.SuperSSN = NULL thì kết quả của Q1 luôn khác Q2

18. Toàn vẹn dữ liệu (data integrity) là:

Dữ liệu lưu trữ trong csdl chính xác và nhất quán

19. R có m tuples và n attributes, S có p,q

Phép R join S có tối đa mp và tối thiểu 0 tuple

20.

Hội, giao, hiệu phải tương thích kiểu

Join không cần

21. Chức năng nào sau đây KHÔNG THUỘC về mô hình dữ liệu mức ý niệm(conceptual)

Cho phép CSDL được hiện thực trong một DBMS

Biểu diễn các loại dữ liệu được lưu trữ trong CSDL

Biểu diễn những mối quan hệ của dữ liệu

Đóng vai trò như một công cụ giao tiếp giữa người dùng và người thiết kế CSDL

22. Những ràng buộc toàn vẹn được định nghĩa cho Specialization trong EERD là:

Completeness và Disjointness

23. Chức năng KHÔNG PHẢI là của một DBA:

Transaction management

24. Phát biểu ĐÚNG về shared subclass và category

Shared subclass thừa kế mọi thuộc tính và quan hệ từ các lớp cha của nó

Category chỉ thừa kế thuộc tính và quan hệ của một trong các lớp cha của nó

25. Trong ERD/EERD, phát biểu ĐÚNG

Bậc của một kiểu mối liên kết (relationship type) là số lượng các kiểu thực thể (entity type) tham gia vào kiểu mối liên kết đó

Khoá của một thực thể có thể là một thuộc tính phức hợp (composite attribute)

Sai: Một kiểu thực thể có thể có nhiều thuộc tính khoá nhưng ta chỉ gạch dưới một khoá được chọn làm khoá chính

26. Phát biểu ĐÚNG về thực thể yếu

Kiểu thực thể yếu có thể có nhiều hơn một kiểu thực thể mạnh cùng tham gia vào mối liên kết định dannh (Identifying Relationship) của mình

27. Mối quan hệ giữa lớp con (sub-class) và lớp cha (super-class) là:

MQH IS-A

28. Phát biểu ĐÚNG

Hệ csdl bao gồm hệ quản trị csdl và csdl

29. Phát biểu ĐÚNG

Hai khái niệm tổng quát hoá và cụ thể hoá là hai khái niệm trái ngược nhau, nhưng trong một số trường hợp có thể được sử dụng thay thế lẫn nhau tuỳ theo cách nhìn của người dùng.

30. Phần mềm nào có các chức năng như hỗ trợ người dùng định nghĩa và bảo trì dữ liệu, hỗ trợ ngôn ngữ truy vấn, xuất các báo cáo và quản lý giao tác

Hệ quản trị csdl

31. Chọn phát biểu ĐÚNG

Một quan hệ chỉ có duy nhất một khoá chính (primary key)

**Note về bài tập DataStorage:**

Số bản ghi: r (number of records)

Kích thước bản ghi: R = tổng size các field

Hệ số phân khối: Bfr (blocking factor) = floor(B/R)

Số khối: b (Number of file blocks) = ceil(r/Bfr)

Hệ số phân khối chỉ mục: Bfr(i) = floor(B/R(i)) với R(i) = block pointer + size trường dữ liệu key

Tìm kiếm binary search: ceil(log2(b(i))) + 1, b(i) = ceil(b/Bfr(i))

Tìm kiếm tuyến tính: b/2 ceil

Áp dụng:

Cho kích thước 1 khối B = 1024 byte

PB = 6 byte, PR = 7 byte, r = 50000

field: orderNo 13 byte, … 1, 4, 1, 13, 13, 35 => R = 13 + … + 35 = 80 byte

Bfr = floor(1024/80) = 12, b = ceil(50000/12) = 4166.6 = 4167

Bfr(i) = floor(1024/13+6) = 53

b(i) = ceil(4167/53) = 79

search: log2(79) = 6.3 ~ 7, 7+1=8 (áp dụng với trường dữ liệu key)

search tuyến tính = 4167 / 2 ceil = 2084

Thường xuyên truy xuất theo trường nào đó (orderDate) => đánh chỉ mục thứ cấp thưa.