



Giáo trình 2009 - Cơ sở dữ liệu

Cơ sở dữ liệu (Trường Đại học Đà Lạt)



Scan to open on Studocu

TRƯỜNG ĐẠI HỌC ĐÀ LẠT
KHOA TOÁN - TIN HỌC



CƠ SỞ DỮ LIỆU

(Bài giảng tóm tắt)



NGƯỜI BIÊN SOẠN
TẠ THỊ THU PHƯỢNG

Đà Lạt 2009

MỤC LỤC

Chương 1: Tổng quan về cơ sở dữ liệu.....	Trang 1
I. Khái niệm và định nghĩa về cơ sở dữ liệu	1
II. Mô tả cách tiếp cận cơ sở dữ liệu	2
III. Người sử dụng cơ sở dữ liệu.....	7
IV. Các tính năng của hệ quản trị cơ sở dữ liệu.....	7
V. Ngôn ngữ cơ sở dữ liệu	9
Chương 2: Mô hình thực thể - Kết hợp.....	10
I. Quá trình thiết kế cơ sở dữ liệu	10
II. Các thành phần cơ bản của Mô hình E-R.....	11
III. Các ràng buộc trên các liên kết.....	13
IV. Lược đồ thực thể - kết hợp (ERD).....	14
V. Các tính năng mở rộng của mô hình E-R.....	15
VI. Thiết kế lược đồ cơ sở dữ liệu E-R.....	16
Chương 3: Mô hình dữ liệu quan hệ.....	18
I. Các khái niệm của mô hình quan hệ.....	18
II. Chuyển lược đồ thực thể - kết hợp (ERD) sang thiết kế quan hệ.....	21
III. Đại số quan hệ	23
IV. Ngôn ngữ phép tính quan hệ.....	32
V. Khung nhìn	35
Chương 4: Ngôn ngữ SQL.....	36
I. Giới thiệu.....	36
II. Định nghĩa dữ liệu	36
III. Truy vấn dữ liệu.....	38
IV. Cập nhật dữ liệu.....	47
V. Khung nhìn (view).....	48
VI. Chỉ mục (index).....	49
Chương 5: Ràng buộc toàn vẹn.....	50
I. Các khái niệm.....	50
II. Phân loại	53
II.1 Ràng buộc toàn vẹn có bối cảnh trên một lược đồ quan hệ.....	53
II.2 Ràng buộc toàn vẹn có bối cảnh trên nhiều lược đồ quan hệ	55
III. Cài đặt ràng buộc toàn vẹn	58
Chương 6: Lý thuyết thiết kế cơ sở dữ liệu quan hệ	60
I. Mở đầu	60
II. Phụ thuộc hàm.	60
III. Các dạng chuẩn của lược đồ quan hệ	70
IV. Phân rã lược đồ quan hệ dựa vào phụ thuộc hàm	73
V. Phụ thuộc đa trị và dạng chuẩn thứ 4	78
Chương 7: Tối ưu hóa câu truy vấn.....	82
I. Mở đầu	82
II. Các chiến lược tối ưu hóa	82
III. Các quy tắc liên quan đến các phép toán.....	83
IV. Kỹ thuật tối ưu hóa.....	85
Bài Tập	87

Chương 1

TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

I. Khái niệm và định nghĩa về cơ sở dữ liệu

Ngày nay, các hệ cơ sở dữ liệu xuất hiện khắp nơi trong thế giới thực. Chúng được sử dụng để tổ chức và quản lý dữ liệu trong các lĩnh vực ứng dụng như:

- *Các hoạt động kinh doanh*: quản lý bán hàng (lẻ/sỉ), thương mại điện tử, ngân hàng và tài chính, các hệ thống đặt chỗ, lập lịch và lên kế hoạch phân công, các hệ thống phân bổ tài nguyên, ...
- *Các hoạt động khác của tổ chức*: thư viện, bệnh viện, chính phủ, trường học, phòng nghiên cứu, ...

Dữ liệu (Data): là một mô tả hình thức về những sự kiện, khái niệm, hay chỉ thị để giúp con người giao tiếp, diễn giải, xử lý hoặc thực hiện (một cách tự động).

Cơ sở Dữ liệu (Database - DB): Một tập hợp có cấu trúc của rất nhiều dữ liệu mang tính bền (*persistent*) có liên quan với nhau về một lĩnh vực cụ thể, được tổ chức đặc biệt cho việc lưu trữ, tìm kiếm và rút trích dữ liệu.

- Một CSDL biểu diễn một phần của thế giới thực (thế giới thu nhỏ).
- CSDL được thiết kế, xây dựng, và lưu trữ với một mục đích xác định, phục vụ cho một số ứng dụng và người dùng.
- Tập ngẫu nhiên của các dữ liệu không thể xem là một CSDL.

Hệ Quản trị CSDL (Database Management System - DBMS): Một phần mềm cho phép định nghĩa, xây dựng và xử lý dữ liệu.

- Định nghĩa: khai báo bộ khung dữ liệu cùng với các mô tả chi tiết về dữ liệu.
- Xây dựng: lưu trữ dữ liệu lên bộ nhớ phụ.
- Xử lý: truy vấn, cập nhật và phát sinh báo cáo.

Hệ quản trị cơ sở dữ liệu (HQT CSDL) cung cấp cách thức lưu trữ hiệu quả, an toàn đối với lượng dữ liệu lớn. Ngôn ngữ cao cấp được dùng để đặc tả các thao tác mà người dùng muốn thao tác trên dữ liệu. Bảo vệ người dùng không được biết chi tiết về cách thức lưu trữ dữ liệu.

Ứng dụng CSDL (Database application): Chương trình tương tác với một CSDL trong quá trình thi hành.

Hệ CSDL (Database System): CSDL cùng với HQT CSDL và các ứng dụng liên quan.

Mô hình hóa Dữ liệu: Để thực hiện nhiệm vụ, CSDL lưu trữ thông tin theo một cách có cấu trúc. Cấu trúc được dùng để lưu trữ thông tin được gọi là *mô hình dữ liệu (data model)*. Mô hình dữ liệu rất phổ biến là *Mô hình Quan hệ (Relational Model)*.

Một CSDL có cấu trúc chặt chẽ về mặt logic, ngữ nghĩa vốn có, có mục đích, phục vụ cho

đối tượng người sử dụng và chương trình ứng dụng cụ thể, có kích thước thay đổi, nội dung có thể thay đổi và mở rộng, tổ chức vật lý với độ phức tạp khác nhau, dùng cho nhiều ứng dụng với mục đích khác nhau.

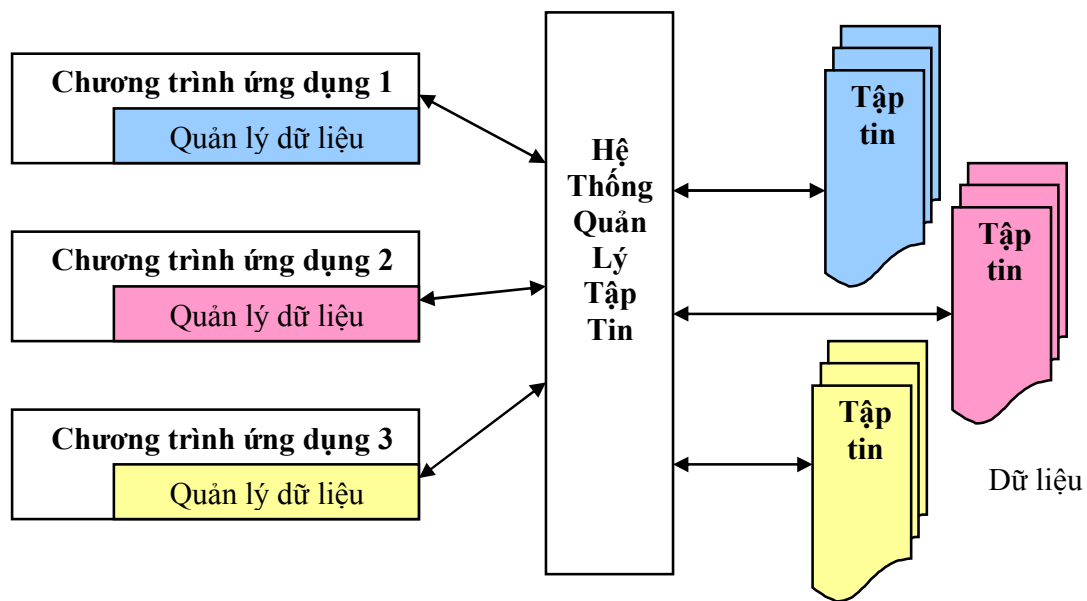
II. Mô tả cách tiếp cận cơ sở dữ liệu

Giả sử cần thực hiện một hệ thống để quản lý hàng hóa cho một công ty bán lẻ.

Dữ liệu: thông tin về hàng hóa (tên hàng, mô tả, kho).

Thao tác thêm mặt hàng mới, quản lý kho hàng, ...

Tiếp cận dựa trên hệ thống tập tin (HTTT):



Hình 1.1: Cách tiếp cận dựa trên tập tin

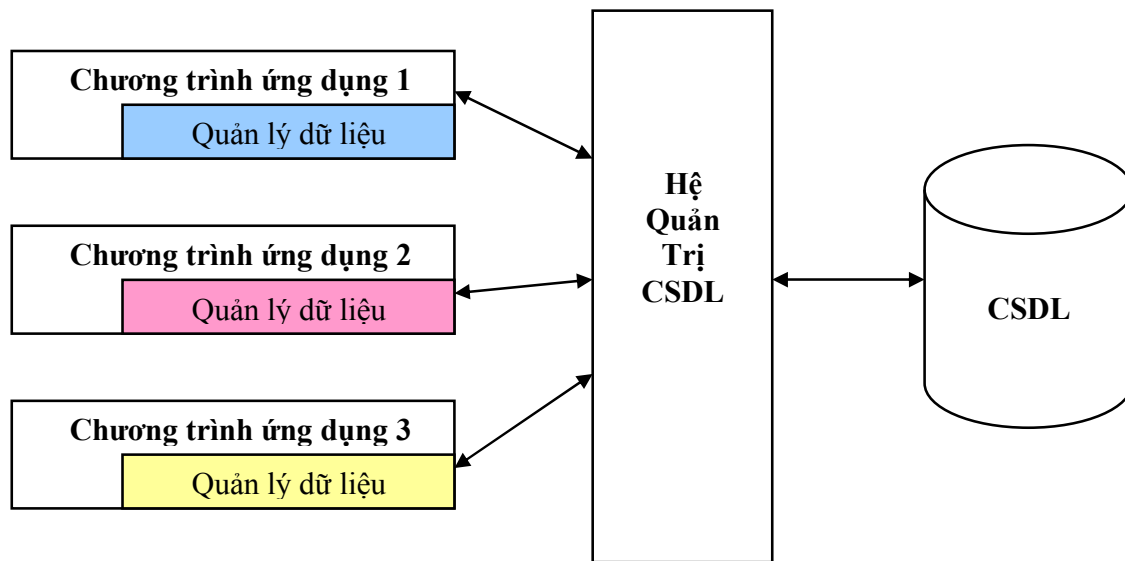
Để lưu trữ dữ liệu một cách bền vững ứng dụng phải dựa vào các tập tin. Một *hệ thống tập tin (file-based system)* là một tập các ứng dụng sử dụng tập tin để lưu dữ liệu.

Phát triển ứng dụng dựa trên hệ thống tập tin có nghĩa là ứng dụng phụ thuộc trên cấu trúc tập tin dẫn đến *không có sự độc lập chương trình - dữ liệu tách biệt ứng dụng khỏi dữ liệu mà nó đang thi hành*

Cách tiếp cận HTTT có nhiều hạn chế:

- Dữ liệu bị trùng lặp và dư thừa.
- Thiếu tính nhất quán giữa các dữ liệu.
- Khó khăn trong việc truy xuất.
- Việc chia sẻ dữ liệu bị hạn chế.
- Khó khôi phục.

Tiếp cận cơ sở dữ liệu



Hình 1.2: Cách tiếp cận cơ sở dữ liệu

Tiếp cận CSDL: tách biệt giữa dữ liệu (lớn, được cập nhật thường xuyên) và cấu trúc của dữ liệu (nhỏ, ổn định).

Trừu tượng dữ liệu (data abstraction) cho phép định nghĩa 1 đối tượng thay đổi mà không ảnh hưởng đến các chương trình sử dụng đối tượng đó thông qua một định nghĩa bên ngoài.

Một số đặc tính của CSDL

Tính tự mô tả

- Hệ CSDL không chỉ chứa bản thân CSDL mà còn chứa định nghĩa đầy đủ (mô tả) của CSDL.
- Các định nghĩa được lưu trữ trong *catalog*. Catalog chứa các thông tin về cấu trúc tập tin, kiểu và dạng thức lưu trữ của mỗi thành phần dữ liệu và những ràng buộc dữ liệu. Dữ liệu trong catalog gọi là *meta-data* (data of data).
- Các Chương trình ứng dụng (CTƯD) có thể truy xuất đến nhiều CSDL nhờ thông tin cấu trúc được lưu trữ trong catalog.

Tính độc lập: Vì định nghĩa về cấu trúc CSDL được lưu trữ trong catalog nên khi có thay đổi nhỏ về cấu trúc ta ít phải sửa lại chương trình.

Tính trừu tượng: Hệ CSDL cho phép trình bày dữ liệu ở một mức trừu tượng cho phép, nhằm che bớt những chi tiết lưu trữ thật của dữ liệu. Tính trừu tượng dữ liệu có được nhờ vào việc định nghĩa các *khung nhìn* (*view*) khác nhau về dữ liệu.

Tính nhất quán:

- Lưu trữ dữ liệu thống nhất → Tránh được tình trạng trùng lặp thông tin.

- Có cơ chế điều khiển truy xuất dữ liệu hợp lý → Tránh được việc tranh chấp dữ liệu và bảo đảm dữ liệu luôn đúng tại mọi thời điểm.

Các cách nhìn dữ liệu:

- Hệ CSDL cho phép nhiều người dùng thao tác lên cùng một CSDL.
- Mỗi người có một cách nhìn (view) khác nhau về CSDL.
- Một view là một phần của CSDL hoặc Dữ liệu tổng hợp từ CSDL.

Mô hình dữ liệu

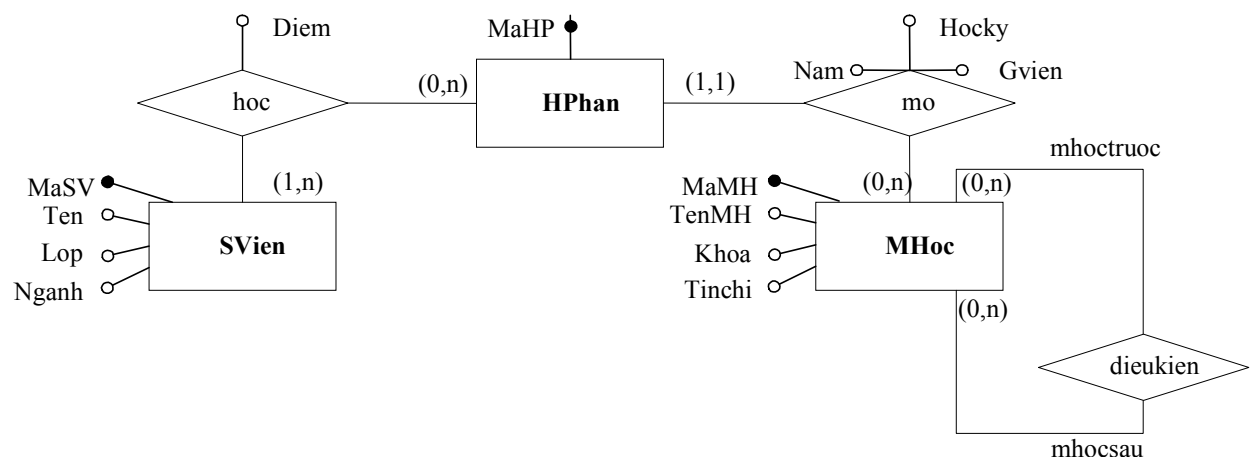
Mô hình dữ liệu (data model): là một tập hợp các khái niệm mô tả dữ liệu. Mô hình dữ liệu bao gồm:

- Các khái niệm biểu diễn dữ liệu.
- Các phép toán xử lý dữ liệu.

Mô hình dữ liệu quan hệ (relational data model): được sử dụng rộng rãi nhất hiện nay.

Mô hình mức cao: cung cấp các khái niệm gần gũi với người dùng. Mô hình phải tự nhiên và giàu ngữ nghĩa.

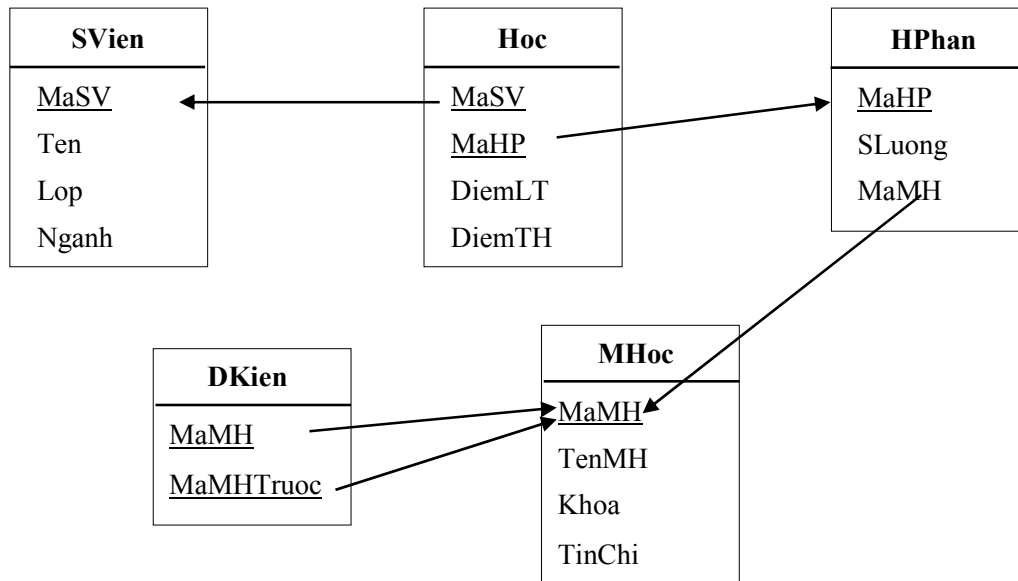
VD: mô hình thực thể kết hợp (ER), mô hình đối tượng...



Hình 1.3: CSDL dữ liệu quản lý sinh viên được mô hình hóa bằng mô hình ER

Mô hình cài đặt: đưa ra các khái niệm người dùng có thể hiểu được nhưng không quá xa với cách dữ liệu được tổ chức thật sự trên máy tính.

VD: mô hình quan hệ.



Hình 1.4: CSDL dữ liệu quản lý sinh viên được mô hình hóa bằng mô hình quan hệ

Mô hình mức thấp (mô hình vật lý): đưa ra các khái niệm mô tả chi tiết về cách thức dữ liệu được lưu trữ trong máy tính.

Lược đồ

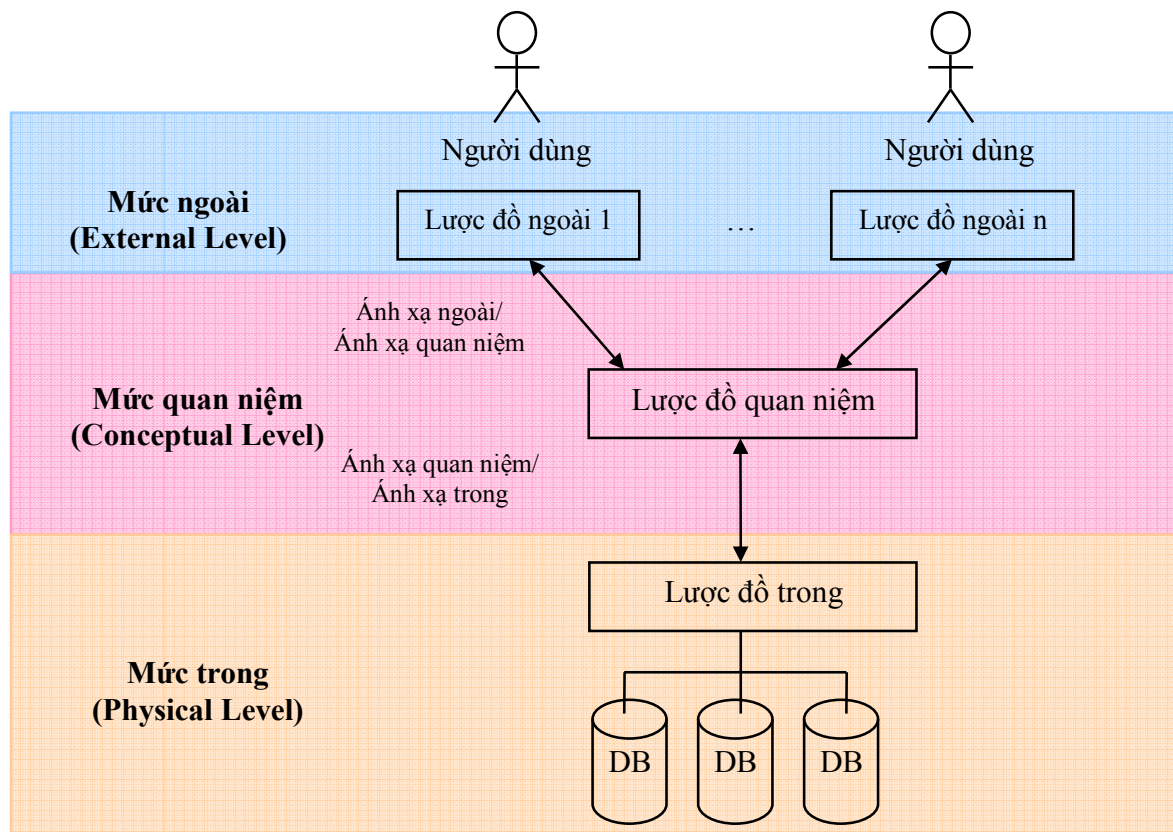
Lược đồ (schema): một mô tả về một tập dữ liệu cụ thể, bằng cách sử dụng một mô hình dữ liệu cụ thể.

Lược đồ CSDL (Database Schema): là các mô tả về cấu trúc và ràng buộc trên CSDL.

Thể hiện

Thể hiện CSDL (Database Instance): là dữ liệu hiện thời được lưu trữ trong CSDL ở một thời điểm nào đó. Một thể hiện CSDL còn được gọi là *tình trạng* của CSDL.

Kiến trúc 3 lược đồ (Kiến Trúc ANSI/SPARC)



Hình 1.5 : Kiến Trúc ANSI/SPARC

Mức ngoài (lược đồ ngoài – External schema): còn gọi là mức khung nhìn (view)

- Mô tả một phần của CSDL mà một nhóm người dùng quan tâm đến và che dấu phần còn lại của CSDL đối với nhóm người dùng đó.
- Mỗi người dùng có thể truy xuất dữ liệu theo nhiệm vụ của mình, có khung nhìn dữ liệu riêng độc lập với những người dùng khác.

Mức quan niệm (lược đồ quan niệm – Conceptual schema)

- Mô tả cấu trúc của toàn thể CSDL cho một cộng đồng người sử dụng, gồm thực thể, kiểu dữ liệu, mối liên hệ và ràng buộc.
- Che bớt các chi tiết của cấu trúc lưu trữ vật lý.
- Người dùng không phải hiểu chi tiết thể hiện vật lý.

Mức trong (lược đồ trong – Internal schema): Mô tả cấu trúc lưu trữ vật lý CSDL.

Các mức độc lập dữ liệu

- Ứng dụng cô lập khỏi cách thức dữ liệu được định nghĩa và lưu trữ.
- *Tính độc lập dữ liệu logic*: khả năng sửa đổi conceptual schema mà không ảnh hưởng đến external schema hay các chương trình ứng dụng.
- *Tính độc lập dữ liệu vật lý*: khả năng sửa đổi physical schema mà không ảnh hưởng đến conceptual (hay external) schema hoặc phải viết lại các chương trình ứng dụng.

Ví dụ

- Sắp thứ tự dữ liệu khi đưa ra kết quả: cách thức thực hiện (sắp xếp vật lý) là do HQTCSDL đảm nhận và chương trình ứng dụng không thể nhìn thấy.
- Thêm một chỉ mục vật lý để làm tăng tốc độ một ứng dụng: không nên đòi hỏi phải cập nhật chương trình ứng dụng, cái có thể thấy được là hiệu quả của việc thêm chỉ mục.
- Thêm một field vào một file: chỉ các chương trình nào truy xuất đến thông tin mới này mới cần phải thay đổi.

III. Người sử dụng cơ sở dữ liệu

Thiết kế viên (Database Designer)

- Liên hệ với người dùng để nắm bắt được những yêu cầu và đưa ra một thiết kế CSDL thỏa yêu cầu này.
- Chịu trách nhiệm về:
 - o Lựa chọn cấu trúc phù hợp để lưu trữ dữ liệu.
 - o Quyết định những dữ liệu nào cần được lưu trữ.
- Có thể là một nhóm các DBA quản lý các CSDL sau khi việc thiết kế hoàn tất.

Quản trị viên (Database Administrators - DBA)

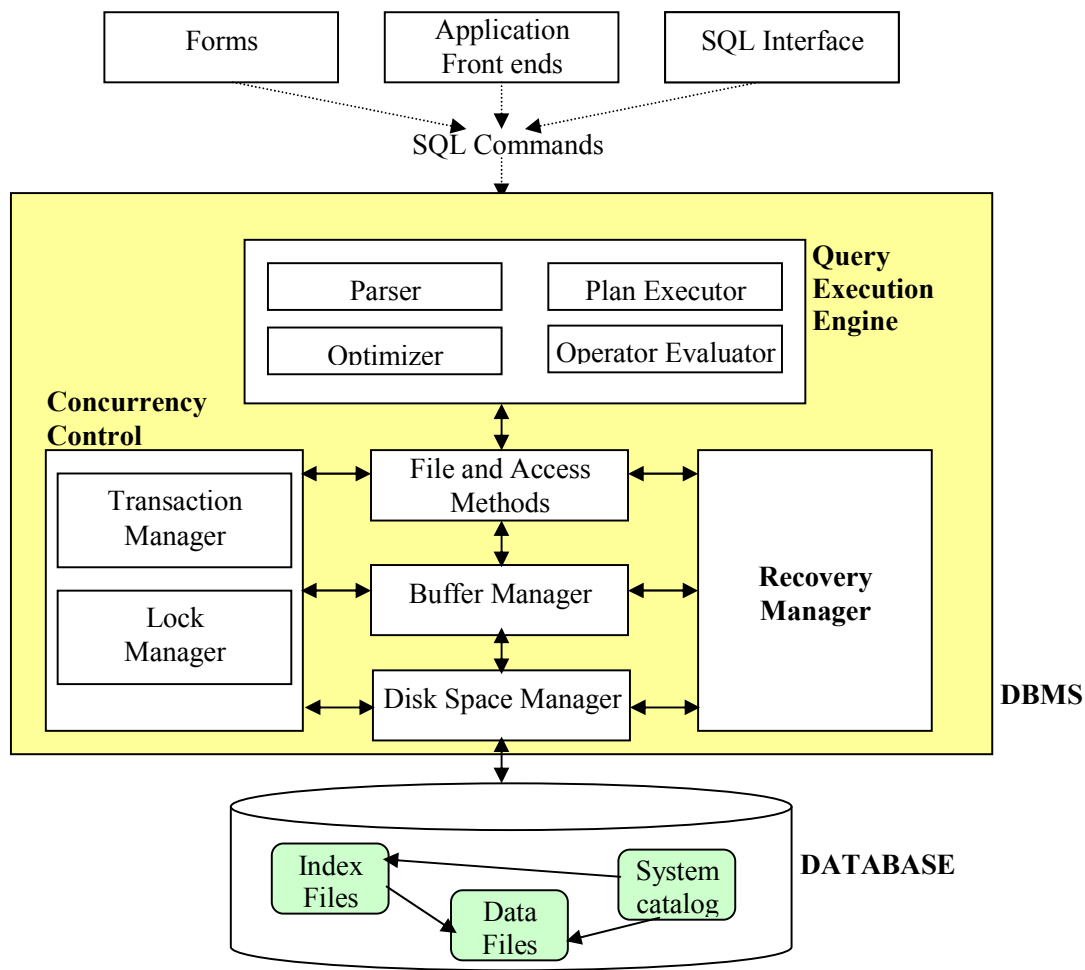
- Có trách nhiệm quản lý hệ CSDL.
- Cấp quyền truy cập CSDL.
- Điều phối và giám sát việc sử dụng CSDL.
- Thiết kế các lược đồ logic/ vật lý.
- Xử lý an toàn và chứng thực.
- Bảo đảm đáp ứng dữ liệu, khôi phục sau sự cố.
- Điều chỉnh dữ liệu phù hợp với các yêu cầu.

Người dùng cuối (End users)

- *Người ít sử dụng*: Ít khi truy cập CSDL, nhưng cần những thông tin khác nhau trong mỗi lần truy cập và dùng những câu truy vấn phức tạp. Ví dụ: Người quản lý, nhà khoa học, người phân tích, ...
- *Người sử dụng thường xuyên*: Thường xuyên truy vấn và cập nhật CSDL nhờ vào một số các chức năng đã được xây dựng sẵn. Ví dụ: Nhân viên bán hàng, thủ thư, ...
- *Lập trình viên ứng dụng CSDL* (database application programmers): phát triển chương trình cho người dùng cuối truy xuất dữ liệu thông qua việc sử dụng các NNLT và các công cụ phần mềm đưa ra bởi nhà cung cấp HQTCSDL.

IV. Các tính năng của hệ quản trị cơ sở dữ liệu

Kiến trúc của HQT CSDL



Hình 1.6 : Kiến trúc của một HQT CSDL

Các tính năng của HQT CSDL

- Kiểm soát được tính dư thừa của dữ liệu: Tích hợp các nhu cầu dữ liệu của người dùng để xây dựng một CSDL thống nhất.
- Chia sẻ dữ liệu: Trong môi trường đa người dùng, các HQT CSDL phải cho phép truy xuất dữ liệu đồng thời.
- Hạn chế những truy cập không cho phép: Từng người dùng và nhóm người dùng có một tài khoản và mật mã để truy xuất dữ liệu.
- Cung cấp nhiều giao diện: HQT cung cấp ngôn ngữ giữa CSDL và người dùng.
- Đảm bảo các ràng buộc toàn vẹn: Ràng buộc toàn vẹn (Integrity Constraints) là những qui định cần được thỏa mãn để đảm bảo dữ liệu luôn phản ánh đúng ngữ nghĩa của thế giới thực. Một số ràng buộc có thể được khai báo với HQT và HQT sẽ tự động kiểm tra. Một số ràng buộc khác được kiểm tra nhờ CTƯD.
- Khả năng sao lưu dự phòng khi gặp sự cố: Có khả năng khôi phục dữ liệu khi có sự hư hỏng về phần cứng hoặc phần mềm.

- Chuẩn hóa: Cho phép DBA định nghĩa và bắt buộc áp dụng một chuẩn thống nhất cho mọi người dùng.
- 유연 chuyên: Khi nhu cầu công việc thay đổi, cấu trúc CSDL rất có thể thay đổi, HQT cho phép thêm hoặc mở rộng cấu trúc mà không làm ảnh hưởng đến CTUD.
- Giảm thời gian phát triển ứng dụng.
- Tính khả dụng: Khi có một sự thay đổi lên CSDL, tất cả người dùng đều thấy được.

V. Ngôn ngữ cơ sở dữ liệu

Ngôn ngữ định nghĩa dữ liệu (Data Definition Language -DDL): Được dùng để định nghĩa các lược đồ mức quan niệm và lược đồ trong. DDL gồm có:

- *Ngôn ngữ định nghĩa ràng buộc (constraint definition language (CDL))* để mô tả các điều kiện mà các thể hiện CSDL phải thỏa.
- *Ngôn ngữ định nghĩa lưu trữ (storage definition language (SDL))* để tác động đến mô hình của lược đồ vật lý (một số HQT CSDL).

Ngôn ngữ thao tác dữ liệu (Data Manipulation Language -DML): Được dùng để mô tả các thao tác trên các thể hiện của CSDL.

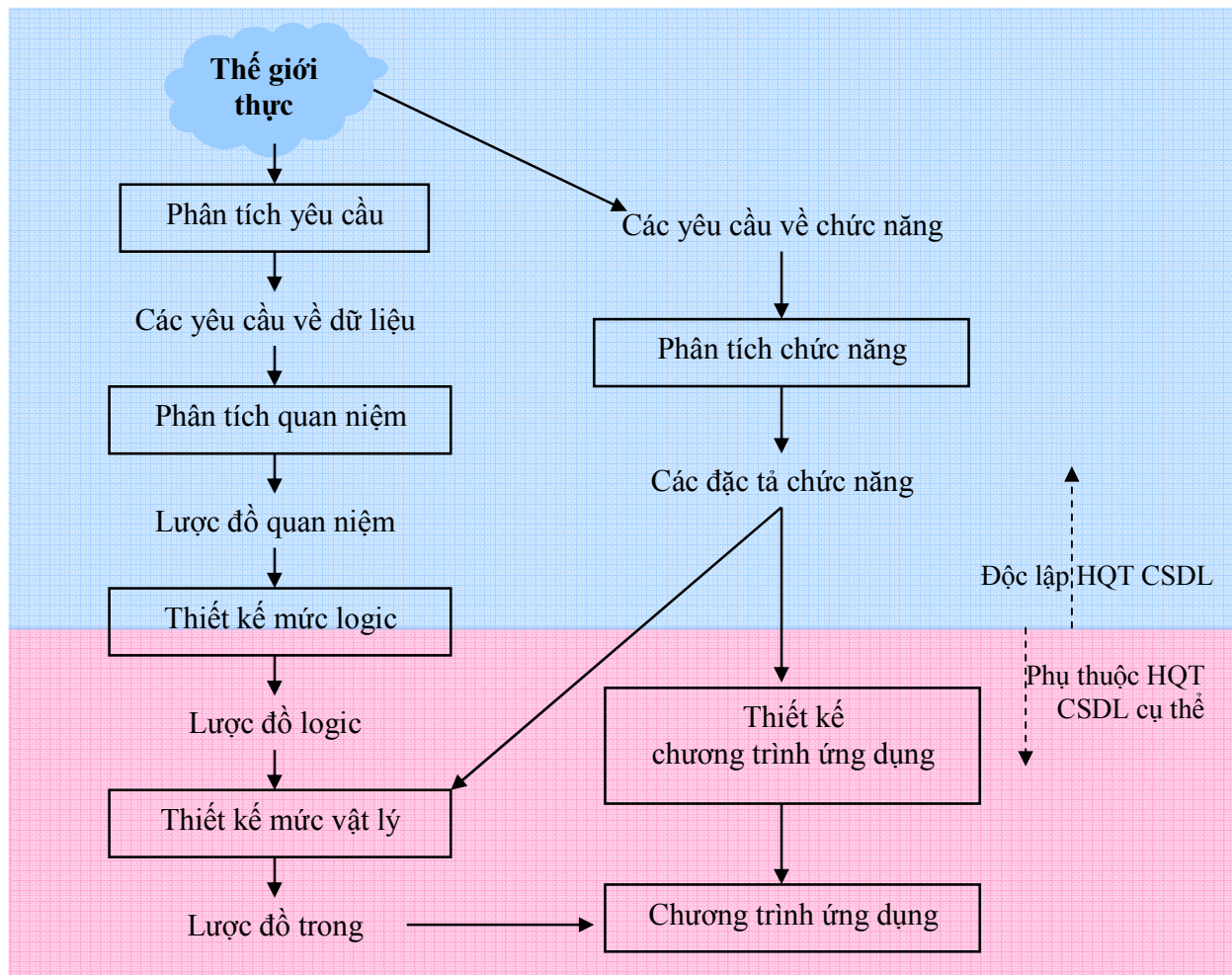
Các Cấp độ của Ngôn ngữ CSDL:

- Ngôn ngữ CSDL cấp cao còn được gọi là ngôn ngữ *khai báo (declarative), phi thủ tục (nonprocedural)*: chỉ cần mô tả dữ liệu gì được rút trích hơn là *cách thức* rút trích, được sử dụng hay nhúng (*data sublanguage*) trong các *ngôn ngữ lập trình (host language)* như C, ...*Lưu ý: SQL bao gồm cả DML và DDL!*
- Ngôn ngữ CSDL cấp thấp còn được gọi là ngôn ngữ *thủ tục (procedural)*: rút trích từng mẫu tin và xử lý riêng lẻ.

Chương 2

MÔ HÌNH DỮ LIỆU THỰC THỂ - KẾT HỢP (Entity-Relationship Model)

I. Quá trình thiết kế cơ sở dữ liệu



Hình 2.1: Các giai đoạn thiết kế cơ sở dữ liệu

Quá trình thiết kế CSDL gồm các giai đoạn:

- Phân tích yêu cầu: xác định yêu cầu dữ liệu của người dùng, CSDL phải lưu những thông tin, dữ liệu nào.
- Thiết kế CSDL mức quan niệm: mô tả dữ liệu mức quan niệm (dùng mô hình ER).
- Thiết kế CSDL mức logic: Chuyển thiết kế ER sang mô hình dữ liệu của DBMS, tinh chế và chuẩn hóa lược đồ CSDL.
- Thiết kế CSDL mức vật lý: xem xét các workloads cụ thể (có thể phải hiệu chỉnh lại thiết kế CSDL), chọn chỉ mục, ...

Thiết kế CSDL mức quan niệm: mô hình hóa thông tin thu thập được ở mức trừu tượng cao mà không phụ thuộc vào DBMS nào.

Mô hình thực thể - kết hợp (Entity – Relationship Model) được đề xuất bởi Peter Chen (1976). Mô hình này được dùng để thiết kế CSDL mức quan niệm. Nó biểu diễn trừu tượng cấu trúc của CSDL.

II. Các thành phần cơ bản của Mô hình E-R

Mô hình hóa ER (Entity – Relationship modeling) mô hình dữ liệu thông qua các khái niệm: thực thể (Entity), thuộc tính (Attribute) và mối kết hợp (Relationship).

Mô hình ER đặc tả thực thể và mối kết hợp bằng cách mô tả tính chất của thực thể và mối kết hợp gọi là *thuộc tính*, và định nghĩa các ràng buộc trên thực thể, mối kết hợp và thuộc tính.

Vì mô hình ER độc lập với mô hình dữ liệu nên sau đó nó có thể được chuyển đổi thành *mô hình logic* theo ý đồ của người thiết kế (Thông thường mô hình logic là mô hình quan hệ).

Trong Mô hình ER, thiết kế CSDL được thể hiện bằng hình ảnh, kết quả của việc mô hình hóa là một *sơ đồ thực thể - kết hợp* (Entity – Relationship Diagram - ERD)

Trong giai đoạn thiết kế quan niệm, các câu hỏi thường xuyên được đề cập là:

- Thực thể và mối kết hợp nào được dùng trong hệ thống?
- Thông tin nào về thực thể và mối kết hợp nên được lưu trữ trong CSDL?
- Các ràng buộc liên quan tới thực thể và các quy định về nghiệp vụ phải đáp ứng?

II.1 Thực thể

Thực thể (Entity) là một đối tượng tồn tại độc lập và được phân biệt với các đối tượng khác, thực thể có thể là một đối tượng cụ thể hoặc trừu tượng.

Tập thực thể (Entity set) là tập hợp các thực thể cùng loại. Các tập thực thể không nhất thiết phải tách rời nhau.

II.2 Thuộc tính

- Một thực thể được biểu diễn bởi một tập thuộc tính.
- Mỗi thuộc tính có *một miền giá trị* quy định các giá trị cho phép đối với thuộc tính đó. Miền giá trị của mỗi thuộc tính đều chứa một giá trị đặc biệt gọi là giá trị trống (giá trị rỗng) NULL.
- Các thuộc tính khác nhau có thể có chung một miền giá trị.
- Phân loại thuộc tính:
 - Thuộc tính đơn (single attribute): là thuộc tính có giá trị không thể phân chia nhỏ.

- Thuộc tính kết hợp (composite attribute): là thuộc tính giá trị của nó là kết hợp của nhiều thành phần. Ví dụ thuộc tính *địa chỉ* là kết hợp của *Số Nhà*, *Đường phố*, *Thành phố*.
- Thuộc tính đơn trị (single-valued attribute): là thuộc tính ứng với một đối tượng nó chỉ có nhiều nhất một giá trị tại thuộc tính này. Ví dụ *họ tên*, *ngày sinh* của một nhân viên.
- Thuộc tính đa trị (multi-valued attribute): là thuộc tính có thể có nhiều giá trị tương ứng với một đối tượng. Ví dụ thuộc tính *số điện thoại* của một nhân viên.
- Thuộc tính suy ra (derived attribute): là thuộc tính được tính toán từ những thuộc tính khác. Ví dụ thuộc tính *Thâm niên công tác* của một nhân viên.

II.3 Mỗi kết hợp

- **Mỗi kết hợp (Relationship)** là một sự kết hợp cụ thể của một số các thực thể.
- **Tập mỗi kết hợp (Relationship set)** là tập hợp các mỗi kết hợp cùng loại.
- *Lưu ý: giữa cùng 2 tập thực thể có thể có nhiều hơn một tập mỗi kết hợp.*
- Mỗi kết hợp cũng có thể có thuộc tính mô tả. Thuộc tính mô tả của mỗi kết hợp thông thường là thời gian, địa điểm, kết quả của hoạt động.
- **Bậc của mỗi kết hợp** là số lượng tập thực thể tham gia vào tập mỗi kết hợp. Bậc của mỗi kết hợp thường gặp là mỗi kết hợp đệ quy (unary), mỗi kết hợp bậc 2 (mỗi kết hợp nhị phân - binary), mỗi kết hợp bậc 3 (ternary), mỗi kết hợp bậc n (n-ary).
- **Vai trò (Role)** là chức năng mà thực thể đảm nhận trong một mỗi kết hợp. Một cách hình thức các role thông thường không được chỉ rõ trong mỗi kết hợp. Một tập thực thể có thể có nhiều mỗi kết hợp với các tập thực thể khác, hay với chính nó. Khi một tập thực thể có một mỗi kết hợp với chính nó thì mỗi kết nối phải có một *tên vai trò* để làm rõ nghĩa của mỗi kết hợp.
- **Bản số của mỗi kết hợp** (còn gọi là ràng buộc nối kết số lượng) là số lượng các thực thể trong một tập thực thể có thể liên quan tới một thực thể khác thông qua một mỗi kết hợp cụ thể. Đối với mỗi kết hợp nhị phân R giữa hai tập thực thể E và F, bản số của R có thể là một trong ba loại sau:
 - **1 - 1 (one – to – one):** Một thực thể trong E được kết hợp với nhiều nhất một thực thể trong F và một thực thể trong F được kết hợp với nhiều nhất một thực thể trong E.
 - **1 – nhiều (one – to – many):** Một thực thể trong E được kết hợp với nhiều thực thể trong F và một thực thể trong F được kết hợp với nhiều nhất một thực thể trong E.
 - **Nhiều – nhiều (many – to – many):** Một thực thể trong E được kết hợp với nhiều thực thể trong F và một thực thể trong F được kết hợp với nhiều một thực thể trong E.

Lưu ý: Ràng buộc nối kết số lượng của một tập mỗi kết hợp nào đó rõ ràng phụ thuộc vào thế giới thực mà nó được mô hình hóa bởi tập mỗi kết hợp.

III. Các ràng buộc trên các liên kết

III.1 Khóa

Khi mô hình hóa CSDL, việc quan trọng là chỉ ra cách thức các thực thể và các mối kết hợp phân biệt được với nhau. Về mặt khái niệm, các thực thể là riêng lẻ và các quan hệ là phân biệt nhưng từ bối cảnh CSDL sự phân biệt giữa chúng phải được diễn tả theo các thuộc tính của chúng. Để chỉ sự phân biệt này ta dùng khái niệm khóa.

Siêu khóa (Super key) là một tập hợp gồm một hay nhiều thuộc tính cho phép định danh duy nhất một thực thể trong một tập thực thể.

Khóa ứng viên (Candidate key) là super key nhỏ nhất (tức là super key mà nó không có một tập con thực sự khác rỗng nào của nó là super key).

Khóa chính (primary key): Một tập thực thể có thể có nhiều candidate key. Khi đó ta dùng thuật ngữ khóa chính để chỉ candidate key nào được chọn bởi người thiết kế CSDL như là phương tiện cơ bản để định danh các thực thể trong một tập thực thể.

III.2 Ràng buộc về sự tham gia

- Sự tham gia toàn phần (*total participation*): mọi thực thể đều phải tham gia vào mỗi kết hợp. Khi đó sự tồn tại của thực thể phụ thuộc vào mỗi kết hợp còn được gọi là sự phụ thuộc tồn tại (*existence dependency*).
- Sự tham gia một phần (*partial participation*): có thể có một số thực thể không tham gia vào mỗi kết hợp.

III.3 Sự phụ thuộc tồn tại:

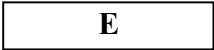

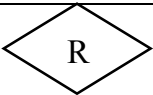
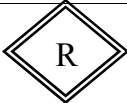
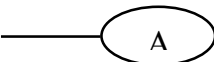
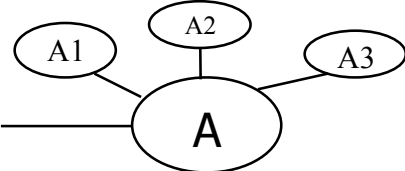

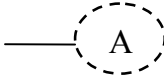
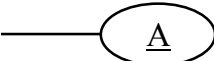
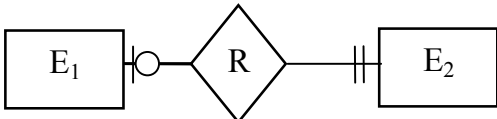
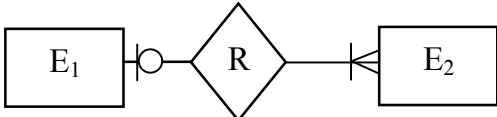
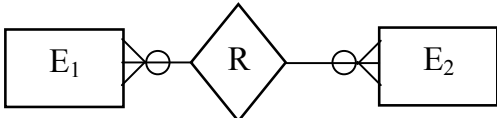
Nếu sự tồn tại của thực thể x phụ thuộc vào sự tồn tại của thực thể y, thì x đang là phụ thuộc tồn tại trên y (*existence dependent*). Về mặt thao tác, nếu y bị loại bỏ thì x cũng bị loại bỏ theo. Khi đó y là thực thể trội, còn x là thực thể phụ thuộc.

Tập thực thể yếu

- Đôi khi một thực thể chỉ có thể tồn tại khi kết hợp với một thực thể khác, và cần “sự trợ giúp” để định danh nó một cách duy nhất.
- Tập thực thể yếu mượn khóa của tập thực thể khác (gọi là thực thể mạnh/thực thể chủ) để xác định duy nhất một thực thể trong nó.
- Tập thực thể yếu (*weak entity set*) có thể có một *khóa bộ phận* (*partial key*). Khóa hoàn chỉnh của nó có được bằng cách mượn thêm khóa chính của tập thực thể chủ kết hợp với khóa bộ phận.
- Mỗi kết hợp giữa tập thực thể mạnh và tập thực thể yếu phụ thuộc tồn tại vào nó được gọi là quan hệ định danh. Quan hệ định danh này là mỗi quan hệ 1-1 hoặc 1- nhiều.

IV.Lược đồ thực thể - kết hợp (ERD)

Kí hiệu của lược đồ ER

Kí hiệu	Ý nghĩa
	Tập thực thể
	Tập thực thể yếu
	Quan hệ R
	Quan hệ định danh
	Thuộc tính
	Thuộc tính kết hợp A gồm 3 thành phần A1, A2, A3
	Thuộc tính đa trị
	Thuộc tính suy ra/dẫn xuất
	Thuộc tính khóa
	Quan hệ một – một
	Quan hệ một – nhiều
	Quan hệ nhiều - nhiều

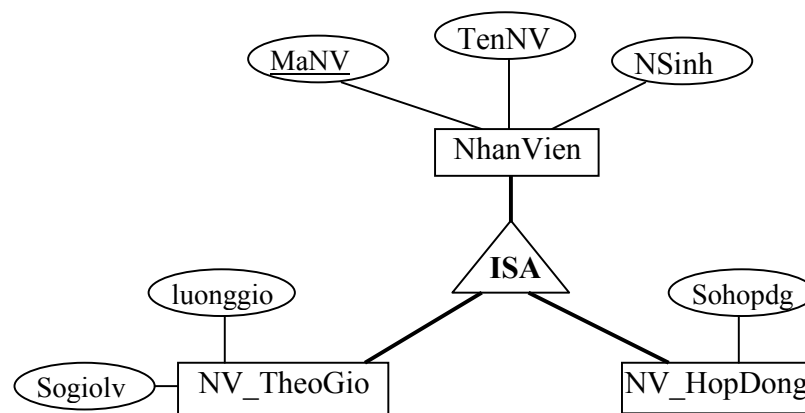
Lưu ý: có thể tồn tại nhiều hệ thống ký hiệu khác nhau của mô hình ER. Trong một lược đồ ER chỉ sử dụng nhất quán một hệ thống ký hiệu.

V. Các tính năng mở rộng của mô hình E-R

Mô hình ER mở rộng (Enhanced ER Model – EER Model) bao gồm tất cả các khái niệm của mô hình ER và bổ sung thêm các khái niệm: Lớp con (subclasses)/Lớp cha (superclasses), Chuyên biệt hóa (Specialization)/Tổng quát hóa (Generalization), sự kế thừa.

Mô hình ER mở rộng được dùng để mô hình hóa những ứng dụng phức tạp, yêu cầu sự diễn đạt về ngữ nghĩa chính xác và hoàn chỉnh.

Ví dụ: Xét sơ đồ sau:



Trong ví dụ trên, một thực thể *NhanVien* có thể được phân vào nhóm *NV_TheoGio* hoặc *NV_HopDong*. Ngoài các thuộc tính chung (*MaNV*, *TenNV*, *Nsinh*) tập thực thể *NV_TheoGio* còn có thêm hai thuộc tính *luonggio*, *Sogiolv*, còn tập thực thể *NV_HopDong* có thêm thuộc tính *Sohopdg*. *NhanVien* được gọi là lớp cha (superclass). *NV_TheoGio* và *NV_HopDong* được gọi là các lớp con (subclasses). Quan hệ giữa lớp con và lớp cha gọi là quan hệ ISA.

Sự kế thừa: lớp con được kế thừa tất cả đặc trưng (thuộc tính và mối kết hợp) của lớp cha. Ngoài ra, các lớp con cũng có thể có các mối kết hợp với các tập thực thể khác.

Chuyên biệt hóa (Specialization): là quá trình định nghĩa tập các lớp con của một lớp cha. Thông thường việc hình thành các lớp con được dựa trên các đặc tính khác biệt của các thực thể trong lớp cha.

Tổng quát hóa (Generalization): là quá trình ngược của chuyên biệt hóa. Với một số tập thực thể có những thuộc tính chung, ta sẽ rút các thuộc tính chung này hình thành tập thuộc tính của lớp cha, các tập thực với các thuộc tính còn lại được xem là các lớp con.

Sự tương quan của các lớp trong quan hệ tổng quát hoá

Sự tương quan giữa các lớp chuyên biệt với lớp tổng quát:

- Tập hợp các đối tượng của tất cả các lớp chuyên biệt phủ toàn bộ tập đối tượng của

lớp tổng quát thì gọi là toàn phần (**total**).

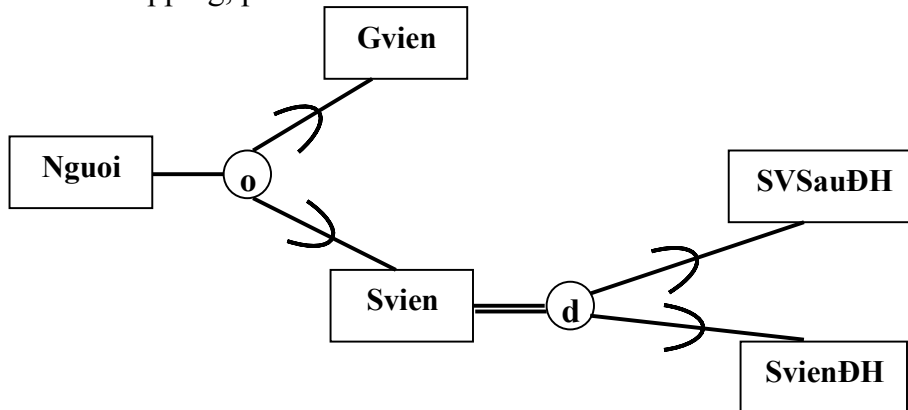
- Tập hợp các đối tượng của tất cả các lớp chuyên biệt không phủ toàn bộ tập đối tượng của lớp tổng quát thì gọi là bán phần (**partial**).

Sự tương quan giữa các lớp chuyên biệt:

- Không tồn tại một đối tượng của lớp tổng quát thuộc hai lớp chuyên biệt trở lên thì gọi là riêng biệt (**disjoint**).
- Tồn tại một đối tượng của lớp tổng quát thuộc hai lớp chuyên biệt trở lên thì gọi là chồng lấp (**overlapping**).

Do đó chúng ta có 4 kiểu specialization/generalization:

- Disjoint, total
- Disjoint, partial
- Overlapping, total
- Overlapping, partial



Kí hiệu

<p>⊙ : uninon</p> <p>○ : overlapping</p> <p>⊞ : disjoint</p>	<p>— : partial</p> <p>== : total</p> <p>Superclass ⊃ subclass</p>
--	---

VI. Thiết kế lược đồ cơ sở dữ liệu E-R.

Các bước thiết kế ER

- Xác định tập thực thể
- Xác định mối kết hợp
- Xác định thuộc tính và gắn thuộc tính cho tập thực thể và mối kết hợp
- Quyết định miền giá trị cho thuộc tính
- Quyết định thuộc tính khóa

B6. Quyết định các ràng buộc cho mỗi kết hợp (ràng buộc nối kết số lượng, sự tham gia).

Lưu ý: có nhiều lựa chọn khác nhau, đặc biệt trong các hệ thống lớn. Một số lựa chọn để biểu diễn một khái niệm thường gặp như:

- Thực thể >< thuộc tính
- Thực thể >< quan hệ
- Quan hệ nhị phân >< quan hệ bậc cao
- Nên hay không nên sử dụng quan hệ ISA

Qui tắc thiết kế

- Chính xác
- Tránh trùng lặp
- Dễ hiểu
- Chọn đúng mối quan hệ
- Chọn đúng kiểu thuộc tính

Để đảm bảo cơ sở dữ liệu được thiết tốt, lược đồ quan hệ có được sau đó nên tiếp tục được phân tích và tinh chế.

Ví dụ

Hãy thiết kế ER cho hệ quản lý sinh viên với các thông tin như sau:

- Các thông tin cần quản lý: MonHoc, SVien, GVien, Khoa.
- Mỗi MonHoc có một MaMH, Ten, Tinch.
- Mỗi Khoa có TenKhoa và thuộc một tòa nhà cụ thể (vị trí, địa điểm).
- Xác định các thuộc tính thích hợp cho Svien và Gvien.
- Mỗi sinh viên phải được quản lý bởi một khoa.
- Mỗi khoa đảm trách tối thiểu hai môn học. Mỗi môn học do duy nhất một khoa đảm trách.
- Một khoa có ít nhất 3 giáo viên. Mỗi giáo viên chỉ làm việc tại duy nhất một khoa.
- Môn học được mở ra dạy trong một học kỳ của một năm học được gọi là học phần. Một học phần có duy nhất 1 giáo viên dạy. Một giáo viên có thể dạy nhiều học phần.
- Một sinh viên có thể đăng ký học nhiều học phần và có điểm tương ứng.

Chương 3

MÔ HÌNH DỮ LIỆU QUAN HỆ

(Relational Data Model)

I. Các khái niệm của mô hình quan hệ

I.1 Các khái niệm

- Mô hình dữ liệu (Data Model) là cách liên kết giữa cách nhìn của người dùng và các bit lưu trữ trong máy tính.
- Mô hình dữ liệu gồm có:
 - Cấu trúc dữ liệu.
 - Các thao tác để rút trích và cập nhật.
 - Ràng buộc toàn vẹn.
- Trong mô hình dữ liệu quan hệ, các thông tin lưu trữ trong CSDL được tổ chức thành các quan hệ. Mỗi quan hệ gồm có lược đồ quan hệ (relation schema) và thể hiện quan hệ (relation instance) còn gọi tắt là quan hệ.
- **Lược đồ quan hệ:** mô tả cấu trúc logic của một quan hệ, bao gồm:
 - Tên của quan hệ.
 - Tập thuộc tính của quan hệ, mỗi thuộc tính có miền trị.
 - Tân từ: biểu diễn ngữ nghĩa của lược đồ.
 - Bậc của lược đồ quan hệ là số lượng thuộc tính có trong lược đồ quan hệ.

Ví dụ: Cho lược đồ quan hệ sau:

NHANVIEN(MSNV, Ho, Ten, NgaySinh, NgayVaoLam, MSCN)

- Tên quan hệ: nhanvien
- Tập thuộc tính: {MSNV, Ho, Ten, NgaySinh, NgayVaoLam, MSCN}
- Kí hiệu: tập tất cả thuộc tính của quan hệ R là R^+** , trong trường hợp trên là $NHANVIEN^+$
- Tân từ: *Mỗi nhân viên có một mã số (MANV) duy nhất, họ và tên lót (Ho), tên nhân viên (Ten), ngày sinh (NgaySinh), ngày vào làm (NgayVaoLam), mã số chi nhánh mà nhân viên đang làm việc (MSCN).*
- Bậc của lược đồ quan hệ NHANVIEN là 6.
- **Quan hệ** (hay thể hiện quan hệ): Thể hiện của quan hệ (gọi tắt là quan hệ) là một bảng (table) 2 chiều.

Ví dụ: xét một thể hiện của lược đồ quan hệ NHANVIEN như sau:

Một cột là một thuộc tính của
quan hệ nhanvien

Một dòng
là một nhân
viên

MSNV	Ho	Ten	NgaySinh	NgayVaoLam	MSCN
0001	Lê Văn	Minh	10/06/1960	02/05/1986	01
0002	Nguyễn Thị	Mai	20/04/1970	04/07/2001	01
0003	Lê Anh	Tuấn	25/06/1975	01/09/1982	02
0004	Vương Tuấn	Vũ	25/03/1960	12/01/1986	02
0005	Lý Anh	Hân	01/12/1980	15/05/2004	02
0006	Phan Lê	Tuấn	04/06/1976	25/10/2002	03
0007	Lê Tuấn	Tú	15/08/1975	15/08/2000	03

- **Lược đồ cơ sở dữ liệu quan hệ:** bao gồm tập các lược đồ quan hệ và tập các ràng buộc.
- **Các ký hiệu của mô hình quan hệ**

Ý nghĩa	Kí hiệu
Lược đồ quan hệ R bậc n	$R(A_1, A_2, \dots, A_n)$
Tập thuộc tính của R	R^+
Quan hệ (hay thể hiện quan hệ) của lược đồ quan hệ R	r
Bộ	t, u, v
Miền giá trị của thuộc tính A	DOM(A) hay MGT(A)
Giá trị tại thuộc tính A của bộ t	t.A hay t[A]

I.2 Các đặc trưng của quan hệ

- Mỗi quan hệ có một tên duy nhất trong lược đồ cơ sở dữ liệu quan hệ.
- Mỗi quan hệ là một sưu tập các bộ.
 - o Mỗi giá trị trong một bộ:
 - Hoặc là một giá trị nguyên tố.
 - Hoặc là một giá trị rỗng (NULL).
- Các thuộc tính của một quan hệ đều có tên khác nhau.
- Tất cả các giá trị của một thuộc tính có cùng một miền trị.
- Mỗi bộ trong một quan hệ là duy nhất. Không có hai bộ nào trùng nhau.
- Thứ tự các bộ trong quan hệ là không quan trọng.
- Thứ tự giữa các giá trị trong một bộ là quan trọng.

I.3 Ràng buộc trong mô hình quan hệ

- Ràng buộc toàn vẹn (Integrity Constraint): là những qui tắc, điều kiện, ràng buộc cần được thỏa mãn cho mọi thể thiện của CSDL quan hệ.

- Ràng buộc toàn vẹn (RBTV) được mô tả khi định nghĩa lược đồ quan hệ.
- RBTV được kiểm tra khi các quan hệ có thay đổi.

Khóa

Các bộ trong quan hệ r phải khác nhau từng đôi một.

a) Siêu khóa (Super key)

- Gọi S là một tập con khác rỗng các thuộc tính của R
- S là siêu khóa của R khi:

$$\forall r, \forall t_1, t_2 \in r, \quad t_1 \neq t_2 \Rightarrow t_1[S] \neq t_2[S]$$

- Siêu khóa là tập các thuộc tính dùng để xác định tính duy nhất của mỗi bộ trong quan hệ.
- Mọi lược đồ quan hệ có tối thiểu một siêu khóa.

b) Khóa (hay khóa đề nghị - Candidate key)

- Gọi K là một tập con khác rỗng các thuộc tính của R .
- K là khóa nếu thỏa đồng thời 2 điều kiện:
 - K là một siêu khóa của R , và
 - $\emptyset \neq K' \subset K, K' \neq K, K'$ không là siêu khóa (nghĩa là không có tập con thực sự khác rỗng của K là siêu khóa).

Nhận xét

- Giá trị của khóa dùng để nhận biết một bộ trong quan hệ. Hai bộ khác nhau phải khác nhau tại khóa.
- Khóa là một đặc trưng của lược đồ quan hệ, không phụ thuộc vào thể hiện quan hệ.
- Khóa được xây dựng dựa vào ý nghĩa của một số thuộc tính trong quan hệ.
- Một lược đồ quan hệ có thể có nhiều khóa.

c) Khóa chính (Primary key)

- Trong một lược đồ quan hệ tồn tại nhiều khóa đề nghị, khi cài đặt quan hệ thành bảng cần chọn một khóa để làm cơ sở nhận biết các bộ. Khóa được chọn để cài đặt được gọi là khóa chính.
- Lưu ý:
 - Các giá trị của khóa chính phải khác nhau.
 - Các thuộc tính khóa chính phải có giá trị khác NULL.
 - Các thuộc tính khóa chính thường được gạch dưới trong lược đồ quan hệ.

d) Khóa ngoại (Foreign key)

Tham chiếu

Một bộ trong quan hệ r , tại thuộc tính A nếu nhận một giá trị từ một thuộc tính B của quan hệ s , ta gọi r tham chiếu s . Khi đó, bộ được tham chiếu phải tồn tại trước.

Khóa ngoại

Xét 2 lược đồ R và S

- Gọi FK là tập thuộc tính khác rỗng của R
- FK là khóa ngoại (Foreign Key) của R khi
 - Các thuộc tính trong FK phải có cùng miền giá trị với các thuộc tính khóa chính của S .
 - Giá trị tại FK của một bộ $t \in r$:
 - * Hoặc bằng giá trị tại khóa chính của một bộ $u \in s$
 - * Hoặc bằng giá trị rỗng (NULL)

Lưu ý:

- Trong một lược đồ quan hệ, một thuộc tính vừa có thể tham gia vào khóa chính, vừa tham gia vào khóa ngoại.
- Khóa ngoại có thể tham chiếu đến khóa chính trên cùng 1 lược đồ quan hệ.
- Có thể có nhiều khóa ngoại tham chiếu đến cùng một khóa chính.
- Ràng buộc tham chiếu = Ràng buộc khóa ngoại

II. Chuyển đổi lược đồ thực thể - kết hợp (ERD) sang Lược đồ CSDL quan hệ

Thiết kế CSDL ở mức quan niệm tạo ra một mô hình ER ở mức quan niệm. Sau đó, mô hình mức quan niệm sẽ được chuyển thành mô hình quan hệ (thuộc mức logic). Mặc dù hoàn toàn có thể thiết kế mô hình quan hệ ngay từ ban đầu, tuy nhiên thông thường sẽ thuận lợi hơn nếu thực hiện thiết kế mức quan niệm bằng mô hình ER trước.

B1: Chuyển đổi tập thực thể

- Các tập thực thể (trừ tập thực thể yếu) chuyển thành các quan hệ có cùng tên và tập thuộc tính.
- Chỉ chuyển đổi các thuộc tính đơn. Đối với các thuộc tính kết hợp, chỉ tạo các thuộc tính tương ứng với các thành phần của nó.
- Các thuộc tính đa trị được chuyển theo quy tắc B6.
- Khóa chính của quan hệ là khóa của tập thực thể.

B2: Chuyển đổi tập thực thể yếu

- Tập thực thể yếu được chuyển thành một quan hệ với khóa ngoại tham chiếu tới quan hệ (thực thể) chủ xác định nó.

- Với mỗi thực thể yếu W với các tập thực thể chủ xác định nó là E_1, E_2, \dots, E_n , hãy tạo một quan hệ R tương ứng theo các bước sau:
 - Xác định các quan hệ R_1, R_2, \dots, R_n của các tập thực thể chủ E_1, E_2, \dots, E_n
 - Tên R cùng tên với thực thể yếu.
 - Khóa chính của R gồm có khóa chính của R_1, R_2, \dots, R_n cùng với khóa bộ phận của thực thể yếu.
 - Tạo một khóa ngoại trong R tham chiếu đến khóa chính của mỗi quan hệ R_1, R_2, \dots, R_n .
 - Thêm vào R các thuộc tính của tập thực thể yếu W .

B3: Chuyển đổi mỗi kết hợp một – một

Với mỗi kết hợp nhị phân **1:1** R giữa hai tập thực thể E_i và E_j :

- Xác định quan hệ tương ứng R_i và R_j của hai tập thực thể E_i và E_j .
- Chọn một trong các quan hệ trên, chẳng hạn R_i , thực hiện các thao tác sau:
 - Thêm các thuộc tính của R vào R_i .
 - Thêm các thuộc tính khóa của R_j vào R_i , và tạo một khóa ngoại tham chiếu đến R_j từ R_i .
 - Khai báo các thuộc tính khóa chính của R_j trong R_i là duy nhất.

Lưu ý:

- ✎ Có thể chọn R_i và R_j . Tốt nhất nên chọn quan hệ nào bảo đảm luôn tham gia vào mỗi kết hợp hoặc là một quan hệ sẽ tham gia nhiều nhất trong một mỗi kết hợp.
- ✎ Để rõ ràng về mặt ngữ nghĩa, có thể đổi tên thuộc tính trong quan hệ.

B4: Chuyển đổi mỗi kết hợp một – nhiều

- Chuyển đổi mỗi kết hợp nhị phân **1- nhiều** thành một khóa ngoại tham chiếu từ quan hệ ở đầu nhiều tới quan hệ ở đầu 1.
- Với mỗi kết hợp nhị phân **1- nhiều** R giữa hai tập thực thể E_i và E_j :
 - Xác định quan hệ tương ứng R_i và R_j của hai tập thực thể E_i và E_j .
 - Gọi R_j là quan hệ ở đầu nhiều, thực hiện các thao tác sau:
 - Thêm các thuộc tính của R vào R_j .
 - Thêm các thuộc tính khóa của R_i vào R_j , và tạo một khóa ngoại tham chiếu đến R_i từ R_j .

B5: Chuyển đổi mỗi kết hợp nhiều – nhiều

- Chuyển đổi mỗi kết hợp nhị phân **nhiều - nhiều** thành một quan hệ với các khóa ngoại tham chiếu tới 2 quan hệ.

- Với mỗi kết hợp nhị phân **hiều - nhiều** R giữa hai tập thực thể E_i và E_j :
 - Xác định quan hệ tương ứng R_i và R_j của hai tập thực thể E_i và E_j .
 - Tạo một quan hệ mới R đại diện cho mỗi kết hợp, trong đó:
 - R chứa các thuộc tính của mỗi kết hợp.
 - Thêm các thuộc tính khóa của R_i và R_j vào R, và tạo khóa ngoại tham chiếu đến R_i từ R và đến R_j từ R.
 - Khóa chính của R có các thuộc tính khóa chính của R_i và R_j .

B6: Chuyển đổi thuộc tính đa trị

- *Chuyển đổi một thuộc tính đa trị thành một quan hệ với khóa chính có các thuộc tính gồm thuộc tính đa trị và thuộc tính khóa chính của tập thực thể chứa thuộc tính đa trị đó.*
- Cho một thuộc tính đa trị A của tập thực thể E_i :
 - Xác định quan hệ tương ứng R_i của E_i
 - Tạo một quan hệ mới R đại diện cho thuộc tính đa trị, trong đó:
 - R chứa các thuộc tính của mỗi kết hợp.
 - Thêm các thuộc tính khóa của R_i vào R, và tạo một khóa ngoại tham chiếu đến R_i từ R.
 - Khóa chính của R bao gồm các thuộc tính khóa chính của R_i và A.

B7: Chuyển đổi mỗi kết hợp đa phân

- *Chuyển đổi mỗi kết hợp đa phân bằng cách tạo ra một quan hệ mới đại diện cho mỗi kết hợp này và tạo các khóa ngoại tham chiếu tới các thực thể có liên quan.*
- Cho một mỗi kết hợp đa phân giữa các tập thực thể E_1, E_2, \dots, E_n :
 - Xác định các quan hệ R_1, R_2, \dots, R_n ứng với các tập thực thể chủ E_1, E_2, \dots, E_n
 - Tạo một quan hệ mới R đại diện cho mỗi kết hợp đa phân, trong đó:
 - R chứa các thuộc tính của mỗi kết hợp.
 - Thêm các thuộc tính khóa của R_i ($i=1, \dots, n$) vào R, và tạo một khóa ngoại tham chiếu đến R_i từ R.
 - Khóa chính của R bao gồm các thuộc tính khóa chính của R_i ($i=1, \dots, n$).

B8: Chuyển đổi các subclasses và superclasses sang mô hình quan hệ.

Một cách tổng quát, ta có các cách sau:

1. Tạo một quan hệ riêng cho mỗi superclass và subclass. Đây là cách tổng quát nhất.

2. Chỉ tạo các quan hệ cho các subclass. Cách này chỉ phù hợp khi superclass có sự tham gia toàn phần (**total**).
3. Tạo một quan hệ và tạo thêm một thuộc tính phân loại. Thuộc tính được dùng để chỉ loại đối tượng (subclass). Cách này phù hợp khi các subclass rời nhau (**disjoint**).
4. Tạo một quan hệ và tạo nhiều thuộc tính phân loại. Thuộc tính phân loại có kiểu boolean cho mỗi subclass. Thuộc tính phân loại của một đối tượng nhận giá trị True nếu đối tượng đó có trong subclass tương ứng. Cách này phù hợp khi các subclass bị trùng lặp (**overlapping**).

III. Đại số quan hệ

III.1 Giới thiệu

- Truy vấn CSDL: trên một CSDL cho trước, người dùng đưa ra các câu hỏi (query) cho hệ cơ sở dữ liệu, hệ cơ sở dữ liệu xử lý câu hỏi và gửi các câu trả lời cho người dùng.
- Ngôn ngữ truy vấn: là một ngôn ngữ cho phép người dùng cập nhật và rút trích dữ liệu được lưu trong một CSDL.
- Ngôn ngữ truy vấn quan hệ: hai ngôn ngữ truy vấn (toán học) sau đây là cơ sở của các ngôn ngữ truy vấn được cài đặt trên các hệ cơ sở dữ liệu (ví dụ như SQL):
 - Đại số quan hệ (Relational Algebra) là ngôn ngữ mang tính thao tác (operational), rất có ưu thế trong việc biểu diễn kế hoạch thi hành truy vấn.
 - Phép tính quan hệ (Relation Calculus) là ngôn ngữ khai báo (declarative), không thủ tục (non-procedural), dựa trên ngôn ngữ logic. Cho phép người dùng mô tả cái họ muốn hơn là cách xử lý nó.
- Truy vấn được áp dụng cho các quan hệ, và kết quả nhận về là một quan hệ.
- Đại số quan hệ (ĐSQH): mọi phép toán quan hệ nhận đầu vào là một hay nhiều quan hệ và tạo ra kết quả là một quan hệ.
- Một chuỗi các phép toán ĐSQH được gọi là một *biểu thức ĐSQH (relational algebra expression)*.
- Các phép toán đại số quan hệ:
 - Phép chọn (selection): σ
 - Phép chiếu (projection): π
 - Phép toán được hình thành từ lý thuyết tập hợp toán học:
 - ▶ phép hội (Union): \cup
 - ▶ phép giao (Intersection): \cap
 - ▶ phép trừ (Difference): $-$
 - ▶ phép tích Cartesian: \times

- Phép kết (Join): \bowtie
- Phép chia: \div
- Phép đổi tên: ρ
- Phép gán: \leftarrow
- Đại số quan hệ mở rộng:
 - ▶ Phép chiếu mở rộng
 - ▶ Phép kết ngoài (OUTER JOIN)
 - ▶ Các hàm kết hợp và gom nhóm

Cơ sở dữ liệu cho các ví dụ : Cho cơ sở dữ liệu gồm các quan hệ sau:

1. NhanVien(MANV, Ho, Ten, NgaySinh, NgayVaoLam, MSCN)

Tân từ: Mỗi nhân viên có một mã số (MANV) duy nhất, họ và tên lót (Ho), tên nhân viên (Ten), ngày sinh (NgaySinh), ngày vào làm (NgayVaoLam), mã số chi nhánh mà nhân viên đang làm việc (MSCN).

2. ChiNhanh(MSCN, TenCN)

Tân từ: Mỗi chi nhánh có một mã số duy nhất (MSCN) và tên chi nhánh (TenCN).

3. KyNang(MSKN, TenKN)

Tân từ: Mỗi kỹ năng tin học có mã số kỹ năng (MSKN) để phân biệt và tên kỹ năng (TenKN).

4. NhanVienKyNang(MANV, MSKN, MucDo)

Tân từ: Thông tin về sử dụng kỹ năng tin học của nhân viên bao gồm mã số nhân viên (MANV), mã số kỹ năng (MSKN) và mức độ (MucDo) thành thạo kỹ năng đó. Mức độ thành thạo được đánh số từ 1 đến 9, số càng lớn mức độ thành thạo càng cao). Một nhân viên có thể biết nhiều kỹ năng.

III.2 Phép chọn

- Phép chọn là phép toán một ngôi. Nó được dùng để lấy ra các bộ của quan hệ r. Các bộ được chọn phải thỏa mãn điều kiện chọn P
- **Ký hiệu:** $\sigma_P(r)$

Với P là biểu thức gồm các mệnh đề có dạng :

- <tên thuộc tính> <phép so sánh> <hằng số>
- <tên thuộc tính> <phép so sánh> <tên thuộc tính>

<phép so sánh> có thể dùng là: <, >, ≤, ≥, ≠, =. Các mệnh đề điều kiện được nối lại nhờ các phép ∧, ∨, ¬

- Kết quả trả về là một quan hệ
 - Có cùng danh sách thuộc tính với r
 - Có số bộ luôn ít hơn hoặc *bằng* số bộ của r
- Phép chọn có tính chất giao hoán

$$\sigma_{p_1}(\sigma_{p_2}(r)) = \sigma_{p_2}(\sigma_{p_1}(r)) = \sigma_{p_1 \wedge p_2}(r)$$

- Ví dụ: (Q1): Cho biết các nhân viên làm việc tại chi nhánh '01'

$$\sigma_{MSCN='01'}(\text{nhanvien})$$

III.3 Phép chiếu

- Được dùng để lấy ra một vài cột của quan hệ r
- Ký hiệu:

$$\pi_{A_1, A_2, \dots, A_k}(r)$$

- Kết quả trả về là một quan hệ
 - Có k thuộc tính.
 - Có số bộ luôn **ít hơn** hoặc bằng số bộ của r.
- Phép chiếu không có tính giao hoán
- Ví dụ: (Q2): Cho biết mã số, họ tên, ngày vào làm và mã số chi nhánh của các nhân viên

$$\pi_{MANV, HO, TEN, NGAYVAOLAM, MSCN}(\text{nhanvien})$$

- Có thể mở rộng định nghĩa phép chiếu quan hệ r lên một tập con thuộc tính $X = \{A_1, A_2, \dots, A_k\} \subseteq R^+$, ký hiệu $r[X]$, như sau:

$$r[X] = \pi_{A_1, A_2, \dots, A_k}(r).$$

Phép chiếu mở rộng

- Mở rộng phép chiếu bằng cách cho phép sử dụng các phép toán số học trong danh sách thuộc tính.
- Ký hiệu: $\pi_{F_1, F_2, \dots, F_n}(E)$, trong đó:
 - E là biểu thức ĐSQH
 - F_1, F_2, \dots, F_n là các biểu thức số học liên quan đến:
 - Hằng số
 - Thuộc tính trong E
- Ví dụ (Q3): Cho biết mã số, họ tên, mã số chi nhánh và số năm công tác của các nhân viên.

$\pi_{\text{MANV, HO, TEN, MSCN, YEAR(GETDATE()) - YEAR(NGAYVAOLAM)}(\text{nhanvien})$

với GETDATE(): hàm lấy ngày hiện tại của hệ thống.

III.4 Phép gán

- Được sử dụng để nhận lấy kết quả trả về của một phép toán. Thường dùng để giữ kết quả trung gian trong chuỗi các phép toán.
- Ký hiệu \leftarrow
- Ví dụ: Tính biểu thức $\pi_{A_1, A_2, \dots, A_k}(\sigma_P(r))$

- B1: $S \leftarrow \sigma_P(r)$

- B2: $KQ \leftarrow \pi_{A_1, A_2, \dots, A_k}(S)$

III.5 Phép đổi tên

- Được dùng để đổi tên.
- Ví dụ : Xét quan hệ $R(B, C, D)$
 - $\rho_S(R)$: Đổi tên quan hệ R thành S
 - $\rho_{X,C,D}(R)$: Đổi tên thuộc tính B thành X
 - $\rho_{S(X,C,D)}(R)$: Đổi tên quan hệ R thành S và thuộc tính B thành X

III.6 Các phép toán tập hợp

- Các phép toán chuẩn trên tập hợp cũng được áp dụng trên quan hệ. Phép toán tập hợp là phép toán hai ngôi lấy hai quan hệ r và s làm input và tạo ra một quan hệ output.
- Ký hiệu:
 - Phép hội (hợp): $r \cup s$
 - Phép giao: $r \cap s$
 - Phép trừ: $r - s$

Điều kiện khả hợp (union compatible):

Hai lược đồ quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_n)$ là *khả hợp* nếu:

R và S cùng bậc (cùng số lượng thuộc tính)

và có $DOM(A_i) = DOM(B_i), 1 \leq i \leq n$

- Kết quả của \cup , \cap , và $-$ là một quan hệ có cùng tên thuộc tính với quan hệ đầu tiên R

$$r \cup s = \{ t \mid t \in r \vee t \in s \}$$

$$r \cap s = \{ t \mid t \in r \wedge t \in s \}$$

$$r - s = \{ t \mid t \in r \wedge t \notin s \}$$

Phép tích

- Được dùng để kết hợp các bộ của các quan hệ lại với nhau
- Ký hiệu: $r \times s$
- Kết quả trả về là một quan hệ q có
 - Mỗi bộ của q là tổ hợp giữa 1 bộ trong r và 1 bộ trong s
 - Nếu r có k_1 bộ và s có k_2 bộ thì Q sẽ có $k_1 \times k_2$ bộ
 - Nếu R có n thuộc tính và Q có m thuộc tính thì Q sẽ có $n + m$ thuộc tính.

Ví dụ:

r	A	B
	α	1
	β	2

s	B	C	D
	α	10	+
	β	10	+
	β	20	-
	γ	10	-

rxs	A	r.B	s.B	C	D
	α	1	α	10	+
	α	1	β	10	+
	α	1	β	20	-
	α	1	γ	10	-
	β	2	α	10	+
	β	2	β	10	+
	β	2	β	20	-
	β	2	γ	10	-

III.7 Phép kết

- Được dùng để tổ hợp 2 bộ có liên quan từ 2 quan hệ thành 1 bộ
- Ký hiệu $r \bowtie \langle \text{điều kiện kết} \rangle s$
- Cho 2 lược đồ quan hệ $R(A_1, A_2, \dots, A_n)$ và (B_1, B_2, \dots, B_m) . Kết quả của phép kết là một quan hệ q :
 - Có $n + m$ thuộc tính $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$
 - Mỗi bộ của q là tổ hợp của 2 bộ trong r và s , thỏa mãn một số <điều kiện kết> nào đó
- Phép kết $r \bowtie \langle \text{điều kiện kết} \rangle s$ được định nghĩa bởi $\sigma_{\langle \text{điều kiện kết} \rangle}(r \times s)$.
- Phân loại phép kết
 - *Phép θ -kết*: phép kết với điều kiện:

- Có dạng $A_i \theta B_j$
- A_i là thuộc tính của R , B_j là thuộc tính của S
- A_i và B_j có cùng miền giá trị
- θ là phép so sánh $\neq, =, <, >, \leq, \geq$
- *Phép kết bằng (equi-join)*
 - Chỉ sử dụng phép '=' trong <điều kiện kết>.
 - kết quả trả về có các thuộc tính kết của R và S . Thuộc tính kết của S xuất hiện *lặp lại* (dư thừa) trong quan hệ kết quả.
- *Phép kết tự nhiên (natural join)*
 - Ký hiệu $r \bowtie s$ hay $r * s$
 - Thực hiện so sánh '=' trên các thuộc tính cùng tên của hai quan hệ (các thuộc tính thuộc tập $R^+ \cap S^+ \neq \emptyset$)
 - Các thuộc tính kết dư thừa của S sẽ được tự động loại bỏ khỏi quan hệ kết quả.
 - **Lưu ý:** nếu $R^+ \cap S^+ = \emptyset$ thì phép kết tự nhiên trở thành phép tích.

• Ví dụ:

(Q4): Liệt kê các thông tin về nhân viên: Ho, Ten, NgaySinh, NgayVaoLam, TenCN

$\pi_{HO, TEN, NGAYSINH, NGAYVAOLAM, TENCN}(nhanvien \bowtie chinhanh)$

(Q5): Lập danh sách các nhân viên (Ho, Ten, TenKN, MucDo) biết sử dụng kỹ năng 'Word'

$\pi_{HO, TEN, TENKN, MUCDO}((nhanvien \bowtie nhanvienkynang) \bowtie \sigma_{TENKN = 'Word'}(kynang))$

III.8 Phép chia

- Phép chia trên hai lược đồ quan hệ $R(A_1, \dots, A_m, B_1, \dots, B_n)$ và $S(B_1, \dots, B_n)$, được ký hiệu là $r \div s$. Các thuộc tính của S là một tập thuộc tính con của R .
- Được dùng để lấy ra một số bộ trong quan hệ r sao cho thỏa với tất cả các bộ trong quan hệ s .
- Định nghĩa: Kết quả phép chia $r \div s$ là 1 quan hệ trên lược đồ $Q \equiv R - S = (A_1, \dots, A_m)$

$$r \div s = \{t \mid t \in \pi_{A_1, \dots, A_m}(r) \wedge (\forall u \in s (\exists (t, u) \in r))\}$$

(Nghĩa là $r \div s$, với các thuộc tính A_1, \dots, A_m , là một tập hợp chứa các bộ t sao cho với mọi bộ u trong s , thì có một bộ $\langle t, u \rangle$ trong r).

- Ví dụ (Q6): Cho biết họ tên nhân viên biết sử dụng tất cả các kỹ năng

$\pi_{HO, TEN, MSKN}(nhanvien \bowtie nhanvienkynang) \div \pi_{MSKN}(kynang)$

III.9 Các hàm kết hợp và gom nhóm.

Hàm kết hợp (Aggregate Functions) nhận vào tập hợp giá trị và trả về một giá trị đơn.

- *avg*: tính giá trị trung bình
- *min*: tìm giá trị nhỏ nhất
- *max*: tìm giá trị lớn nhất
- *sum* : tính tổng
- *count*: đếm số bộ

Ví dụ: (Q7) Cho biết số nhân viên của công ty

$$\mathcal{F}_{\text{count}}(\text{MANV}) (\text{nhanvien})$$

Phép gom nhóm (Grouping) \mathcal{F} trong ĐSQH:

- Được dùng để phân chia quan hệ thành nhiều nhóm dựa trên điều kiện gom nhóm nào đó.
- Kí hiệu:

$$G_1, G_2, \dots, G_n \mathcal{F} F_1(A_1), F_2(A_2), \dots, F_n(A_n)(E)$$

trong đó:

- E là biểu thức ĐSQH
- G_1, G_2, \dots, G_n là các thuộc tính gom nhóm (có thể có hoặc không)
- F_1, F_2, \dots, F_n là các hàm nhóm
- A_1, A_2, \dots, A_n là các thuộc tính tính toán trong hàm F
- Trong quan hệ kết quả, mỗi nhóm có một bộ tương ứng gồm:
 - 1. Các thuộc tính gom nhóm và
 - 2. Kết quả của các hàm kết hợp thực hiện trên nhóm đó
- Ví dụ (Q8): cho biết số nhân viên làm việc ở từng chi nhánh

$$\text{MSCN}, \text{TENCN} \mathcal{F}_{\text{COUNT}}(\text{MANV})(\text{chinhanh} \bowtie \text{nhanvien})$$

III.10 Phép kết ngoài

Các phép kết trong mục III.7 chỉ cho kết quả bao gồm các bộ thỏa điều kiện kết. Để lấy thêm các bộ không thỏa điều kiện kết vào trong kết quả, có thể sử dụng các phép kết ngoài.

Có 3 loại kết ngoài:

Left outer join: $r \bowtie\!\!\!\bowtie s$

Kết quả sẽ chứa tất cả các bộ của r khớp với các bộ của s . Với một bộ thuộc r , nếu không tìm thấy bộ nào khớp với s , thì các bộ này cũng được xuất hiện trong kết quả

cuối cùng và giá trị thuộc tính tương ứng của s sẽ được đặt là NULL.

Right outer join: $r \bowtie_r s$

Kết quả sẽ chứa tất cả các bộ của s khớp với các bộ của r . Với một bộ thuộc s , nếu không tìm thấy bộ nào khớp với r , thì các bộ này cũng được xuất hiện trong kết quả cuối cùng và giá trị thuộc tính tương ứng của r sẽ được đặt là NULL

Full outer join: $r \bowtie_{fs} s$

Tất cả các bộ của r và s đều có trong kết quả cho dù chúng có bộ khớp với quan hệ kia hay không.

Ví dụ: Xét hai quan hệ r và s

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

s	C	D
	3	1
	6	2
	7	5

$r \bowtie_r s$	A	B	C	D
	1	2	3	1
	4	5	6	2

$r \bowtie_{fs} s$	A	B	C	D
	1	2	3	1
	4	5	6	2
	7	8	9	

$r \bowtie_{fs} s$	A	B	C	D
	1	2	3	1
	4	5	6	2
			7	5

$r \bowtie_{fs} s$	A	B	C	D
	1	2	3	1
	4	5	6	2
	7	8	9	
			7	5

III.11 Các thao tác cập nhật quan hệ

Tình trạng của CSDL có thể được cập nhật bằng các thao tác: Thêm (insertion), Xóa (deletion), Sửa (updating).

Các thao tác cập nhật được diễn đạt thông qua phép toán gán:

$$R_{\text{new}} \leftarrow \text{các phép toán trên } R_{\text{old}}$$

a) *Thao tác thêm:* được diễn đạt bởi:

$$R \leftarrow R \cup E$$

trong đó R là quan hệ, E là một biểu thức ĐSQH

Ví dụ: Thêm bộ < '06', 'PhotoShop' > vào bảng KyNang

$$\text{kynang} \leftarrow \text{kynang} \cup ('06', 'PhotoShop')$$

b) *Thao tác xóa:* được diễn đạt bởi:

$$R \leftarrow R - E$$

trong đó R là quan hệ, E là một biểu thức ĐSQH

Ví dụ: Xóa thông tin sử dụng các kỹ năng của nhân viên có mã số '0001'

$$\text{nhanvienkynang} \leftarrow \text{nhanvienkynang} - \sigma_{\text{MANV}='0001'}(\text{nhanvienkynang})$$

c) *Thao tác sửa*: sử dụng phép chiếu mở rộng, được diễn đạt bởi:

$$R \leftarrow \pi_{F_1, F_2, \dots, F_n}(R)$$

trong đó R là quan hệ, F_i là biểu thức tính toán cho ra giá trị mới của thuộc tính

Ví dụ: Cập nhật cho các nhân viên có sử dụng kỹ năng 'Word' (có mskn = '01') mức độ tăng thêm một bậc

$$\text{nhanvienkynang} \leftarrow \pi_{\text{MANV}, \text{MSKN}, \text{MUCDO}+1}(\sigma_{\text{MSKN}='01'}(\text{nhanvienkynang}))$$

Tập đầy đủ các phép toán ĐSQH

Tập các phép toán $\{\sigma, \pi, \times, -, \cup\}$ được gọi là tập đầy đủ các phép toán ĐSQH. Nghĩa là các phép toán thảo luận ở trên có thể được biểu diễn qua chúng.

Ví dụ

$$R \cap S = R \cup S - ((R - S) \cup (S - R))$$

$$R \bowtie_C S = \sigma_C(R \times S)$$

IV. Ngôn ngữ phép tính quan hệ

IV.1 Giới thiệu

Đại số quan hệ là một ngôn ngữ quan hệ thủ tục, vì khi ta viết một truy vấn ta đã chỉ ra dãy toán tử để tạo ra câu trả lời, riêng tính toán quan hệ lại là phi thủ tục vì nó chỉ cung cấp một mô tả hình thức về thông tin mà ta muốn có và không chỉ ra cách thức để thu được thông tin đó.

Phép tính quan hệ gồm có 2 loại:

- Phép tính quan hệ trên bộ - Tuple Relational Calculus (giống như SQL).
- Phép tính quan hệ trên miền - Domain Relational Calculus (giống như QBE – Query By Example).

Xây dựng Phép tính quan hệ

Phép tính (Calculus) gồm có các biến, hằng, phép toán so sánh, kết nối logic, và lượng từ.

- Hằng số: 'Nghien cuu', 50000, ...
- Các kết nối logic: $\wedge, \vee, \Rightarrow, \neg, \in$
- Biến: x, y, z, ...
- Lượng từ:

- $\forall x(p(x))$: Với mọi x , $p(x)$ phải true
- $\exists x(p(x))$: Tồn tại ít nhất một x sao cho $p(x)$ true
- Vị từ (predicate): Tên các quan hệ
- Toán tử so sánh: $=, <, >, \leq, \geq$

Phép tính quan hệ trên bộ có các biến nhận các bộ của quan hệ làm giá trị.

Phép tính quan hệ trên miền: Biến biến thiên trên các *thành phần miền giá trị*.

Biểu thức trong phép tính quan hệ gọi là *công thức* (formula).

Trong phần này ta chú trọng vào phép tính quan hệ trên bộ.

IV.2 Phép tính quan hệ trên bộ

- Biểu thức phép tính quan hệ trên bộ có dạng: $\{t \mid p(t)\}$
với t là biến bộ nhận giá trị là một bộ của quan hệ trong CSDL. $p(t)$ là một công thức có liên quan đến t . $P(t)$ có giá trị TRUE hoặc FALSE phụ thuộc vào t .
- Kết quả: trả về tập các bộ t sao cho công thức $p(t)$ được đánh giá là TRUE.

Biến tự do và biến kết buộc

Một biến bộ được gọi là biến tự do (free variable) nếu như nó không được lượng hóa bởi \forall hay \exists . Ngược lại ta gọi là biến kết buộc (bound variable).

Lưu ý: trong truy vấn $\{t \mid p(t)\}$, biến t xuất hiện bên trái ' \mid ' phải là biến tự do (free variable) trong công thức $P(t)$ và tất cả các biến bộ khác phải là biến kết buộc.

Công thức

Một công thức nguyên tố (*atomic formula*) nếu nó thuộc một trong các dạng sau:

- (i). $t \in r$, trong đó t là biến bộ và r là quan hệ
- (ii). $t.A \theta s.B$, với:
 - A là thuộc tính của biến bộ t
 - B là thuộc tính của biến bộ s
 - θ là các phép so sánh $<, >, \leq, \geq, \neq, =$
- (iii). $t.A \theta c$, với
 - c là hằng số
 - A là thuộc tính của biến bộ t
 - θ là các phép so sánh $<, >, \leq, \geq, \neq, =$

Một công thức tổng quát được xây dựng từ các *atomic formula* bằng cách dùng các phép toán logic như sau:

- một *atomic formula* là một công thức.

- $\neg p, p \wedge q, p \vee q$ trong đó p, q là công thức.
- $\forall t(p(t))$ trong đó t là biến bộ.
- $\exists t(p(t))$ trong đó t là biến bộ.

Một số quy tắc biến đổi :

- (i) $P1 \wedge P2 = \neg (\neg P1 \vee \neg P2)$
- (ii) $\forall t \in r (P(t)) = \neg \exists t \in r (\neg P(t))$
- (iii) $\exists t \in r (P(t)) = \neg \forall t \in r (\neg P(t))$
- (iv) $P \Rightarrow Q = \neg P \vee Q$

Biểu thức an toàn

Một tính toán quan hệ bộ có thể tạo ra một quan hệ vô hạn. Chẳng hạn, ta viết:

$\{t \mid \neg (t \in \text{khachhang})\}$ với $\text{KhachHang}(\text{MaKh}, \text{TenKH}, \text{DiaChi})$, như vậy có vô hạn bộ không thuộc quan hệ **khachhang**. Phần lớn những bộ này chứa những trị mà chúng không hề xuất hiện trong CSDL.

Miền giá trị của công thức: Cho P là một công thức, domain của P , ký hiệu $\text{Dom}(P)$, là tập của trị xuất hiện rõ ràng trong P hoặc xuất hiện trong một hoặc nhiều quan hệ có tên xuất hiện trong P .

Một biểu thức $\{t \mid P(t)\}$ là biểu thức an toàn nếu nó thỏa tất cả các điều kiện sau :

- Mọi trị xuất hiện trong các bộ của biểu thức là những trị từ $\text{Dom}(P)$.
- Với mỗi công thức dạng $\exists s(P1(s))$, thì công thức là đúng nếu và chỉ nếu có một bộ s có trị từ $\text{Dom}(P1)$ sao cho $P1(s)$ là đúng.
- Với mỗi công thức dạng $\forall s(P1(s))$, thì công thức là đúng nếu và chỉ nếu $P1(s)$ đúng với mọi bộ s có trị từ $\text{Dom}(P1)$.

Các ví dụ

a) Cho biết các nhân viên làm việc tại chi nhánh '01'

$$\{t \mid t \in \text{nhanvien} \wedge t.\text{MSCN} = '01'\}$$

b) Tìm các nhân viên biết sử dụng kỹ năng 'Word'

$$\{n \mid n \in \text{nhanvien} \wedge \exists t(t \in \text{nhanvienkynang} \wedge t.\text{MANV} = n.\text{MANV} \\ \wedge \exists k(k \in \text{kynang} \wedge k.\text{MSKN} = t.\text{MSKN} \wedge k.\text{TENKN} = 'Word'))\}$$

c) Tìm các nhân viên biết sử dụng tất cả các kỹ năng

$$\{n \mid n \in \text{nhanvien} \wedge \forall k((k \in \text{kynang}) \Rightarrow (\exists t(t \in \text{nhanvienkynang} \wedge t.\text{MANV} = n.\text{MANV} \\ \wedge k.\text{MSKN} = t.\text{MSKN}))))\}$$

V. Khung nhìn

- Khung nhìn cũng là một quan hệ ảo, không được lưu trữ vật lý, không chứa dữ liệu. Quan hệ này được định nghĩa từ những quan hệ bảng được lưu trữ thật sự.
- Có thể truy vấn hay cập nhật thông qua khung nhìn
- Mục đích của việc sử dụng khung nhìn nhằm che dấu tính phức tạp của dữ liệu, đơn giản hóa các câu truy vấn, hiển thị dữ liệu dưới dạng tiện dụng nhất và an toàn dữ liệu.

Ví dụ : xét khung nhìn TT_NHANVIEN được định nghĩa như sau:

TT_NHANVIEN \leftarrow chinhanh \bowtie nhanvien \bowtie nhanvienkynang \bowtie kynang

(Q9) : Liệt kê tên và mức độ thành thạo của các kỹ năng mà nhân viên 'Lê Anh Tuấn' biết sử dụng.

$\pi_{TENKN,MUCDO}(\sigma_{HO = 'Lê Anh' \wedge TEN = 'Tuấn'}(TT_NHANVIEN))$

Chương 4

NGÔN NGỮ SQL

I. Giới thiệu

- Structured Query Language (SQL) là ngôn ngữ truy vấn chuẩn cho CSDL quan hệ.
- SQL là ngôn ngữ truy vấn khai báo, phi thủ tục. Người sử dụng chỉ cần đưa ra nội dung cần truy vấn, không cần nêu cách thức có được nó.
- SQL không phải là ngôn ngữ lập trình hoàn chỉnh.
- Các hệ quản trị cơ sở dữ liệu thương mại đều cài đặt chuẩn SQL với một số cú pháp, câu lệnh mở rộng cũng như các tính năng tiên tiến từ các chuẩn gần đây.
- SQL gồm có ngôn ngữ định nghĩa dữ liệu (Data Definition Language – DDL) và ngôn ngữ thao tác dữ liệu (Data Manipulation Language – DML).
- SQL sử dụng các thuật ngữ:
 - Bảng (table) ~ quan hệ (realation)
 - Dòng (Row) ~ bộ (tuple)
 - Cột (Column) ~ thuộc tính (attribute)
- SQL không phân biệt chữ hoa chữ thường.
- Không phải tất cả các HQTCSDL đều hỗ trợ tất cả các tính năng mở rộng của SQL.
- Trong chương này ta sử dụng CSDL sau để làm các ví dụ minh họa:

ChiNhanh(MSCN, TenCN)

NhanVien(MANV, Ho, Ten, NgaySinh, NgayVaoLam, MSCN)

KyNang(MSKN, TenKN)

NhanVienKyNang(MANV, MSKN, MucDo)

II. Định nghĩa dữ liệu

II.1 Kiểu dữ liệu

Các kiểu dữ liệu thường dùng trong SQL Server gồm:

Kiểu số

- Số nguyên: bit, tinyint, smallint, int, bigint
- Số thực: float, real

Kiểu chuỗi

- char, varchar, text
- nchar, nvarchar, ntext

Kiểu ngày, giờ

- Datetime
- Smalldatetime

Ngoài ra còn các kiểu dữ liệu khác.

II.2 Lệnh tạo bảng

- Chức năng: dùng để định nghĩa một bảng. Khi định nghĩa một bảng cần khai báo: tên bảng, các thuộc tính và kiểu dữ liệu tương ứng, các ràng buộc/qui định (RBTV) trên thuộc tính nếu có.
- Cú pháp

```
CREATE TABLE <Tên_bảng>
(
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],
    ...
    [<RBTV>]
)
```

Các RBTV có thể là:

- **NOT NULL**: khai báo thuộc tính không được phép nhận giá trị NULL.
- **NULL**: khai báo thuộc tính được phép nhận giá trị NULL (mặc định).
- **UNIQUE**: khai báo giá trị thuộc tính là duy nhất (không được phép nhập trùng giá trị).
- **DEFAULT**: khai báo giá trị mặc định của thuộc tính.
- **PRIMARY KEY**: khai báo khóa chính.
- **FOREIGN KEY / REFERENCES**: khai báo khóa ngoại.
- **CHECK**: khai báo biểu thức điều kiện cần kiểm tra trên thuộc tính.
- Ví dụ:

```
Create table chinhanh
(
    MSCN nchar(2) primary key,
    TENCN nvarchar(20) NOT NULL UNIQUE
)
```

```
Create table NhanVien
(MANV nchar(4) primary key,
HO nvarchar(20) NOT NULL,
TEN nvarchar(10) NOT NULL,
NGAYSINH datetime,
NGAYVAOLAM datetime,
MSCN nchar(2) references ChiNhanh(MSCN)
)
```


II.3 Lệnh sửa bảng

- Chức năng: dùng để thay đổi cấu trúc bảng hoặc thay đổi ràng buộc toàn vẹn.
- Cú pháp

- Lệnh thêm cột:

ALTER TABLE <Tên_bảng> **ADD COLUMN**
<Tên_cột> <Kiểu_dữ_liệu> [<RBTV>]

- Lệnh xóa cột:

ALTER TABLE <Tên_bảng> **DROP COLUMN** <Tên_cột>

- Lệnh thay đổi kiểu dữ liệu của cột

ALTER TABLE <Tên_bảng> **ALTER COLUMN**
<Tên_cột> <Kiểu_dữ_liệu_mới>

- Lệnh thêm ràng buộc toàn vẹn

ALTER TABLE <Tên_bảng> **ADD**
CONSTRAINT <Tên_RBTV> <RBTV>,
CONSTRAINT <Tên_RBTV> <RBTV>,
...

- Lệnh xóa RBTV

ALTER TABLE <Tên_bảng> **DROP** <Tên_RBTV>

II.4 Lệnh xóa bảng

- Chức năng: dùng để xóa cấu trúc bảng, khi đó tất cả các dữ liệu chứa trong bảng cũng bị xóa.
- Cú pháp

DROP TABLE <Tên_bảng>

III. Truy vấn dữ liệu

Truy vấn dữ liệu là thao tác rút trích dữ liệu thỏa một số điều kiện nào đó. Không giống như đại số quan hệ, SQL không tự động loại bỏ các dòng trùng trong kết quả truy vấn.

III.1 Truy vấn cơ bản

Gồm 3 mệnh đề:

SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <điều kiện>

Trong đó:

<danhsách các cột>: tên các cột cần hiển thị trong kết quả truy vấn.

<danhsách các bảng>: các bảng liên quan đến truy vấn.

<điều kiện>: là một biểu thức Boolean. Các phép toán trong điều kiện là =, <>, <, >, <=, >=, AND, OR, NOT.

a) Mệnh đề **SELECT**

- Liệt kê các thuộc tính cần hiển thị trong kết quả của một truy vấn.
- Tương ứng với phép chiếu của ĐSQH.
- Mệnh đề SELECT có thể chứa các biểu thức toán học liên quan tới các phép toán +, -, *, và / thực hiện trên các hằng số hay thuộc tính của bộ.
- Thay đổi tên thuộc tính/Đặt tên cho một biểu thức trong mệnh đề SELECT.
- Loại bỏ dòng trùng, dùng **DISTINCT** trong mệnh đề SELECT.

b) Mệnh đề **WHERE**

- Mệnh đề WHERE mô tả điều kiện mà kết quả truy vấn phải thỏa.
- Thường chứa điều kiện kết bảng và điều kiện chọn của ĐSQH.
- Các điều kiện được kết hợp với nhau bởi AND, OR, hoặc NOT.
- Các so sánh có thể áp dụng trên kết quả của các biểu thức toán học.

❖ So sánh chuỗi

- Hằng chuỗi sử dụng nháy đơn: 'TP HCM', 'THK31', ...
- Có thể sử dụng bất kỳ các phép so sánh "bình thường" để so sánh hai chuỗi: <, >, =, ...
- So sánh tương đối sử dụng LIKE, NOT LIKE và ký tự đại diện:
'%': thay thế cho một chuỗi bất kỳ (trong SQL Server)
'_': thay cho một ký tự bất kỳ (trong SQL Server)
- SQL được cài đặt trong các DBMS hỗ trợ nhiều thao tác xử lý chuỗi khác nhau:
 - Phép nối chuỗi: +
 - Các hàm: lấy chuỗi con trái (left), lấy chuỗi con phải (right), tính chiều dài của chuỗi (len), ...

❖ Kiểm tra giá trị NULL

- Những biểu thức tính toán liên quan đến giá trị NULL sẽ cho ra kết quả là NULL.
- IS NULL (bằng NULL), IS NOT NULL (khác NULL).

❖ **Điều kiện liên quan đến ngày, giờ**

- Hằng ngày tháng: 'YYYY-MM-DD' hoặc 'MM/DD/YYYY' (trong SQL Server); #DD/MM/YYYY# (trong MS Access)
- Hằng giờ: 'HH:MM:SS'
- Các hàm xử lý thời gian: getdate(), day(biến_ngày), month(biến_ngày), year(biến_ngày),

c) Mệnh đề FROM

- Sử dụng biến bộ (còn gọi là bí danh – alias) trong mệnh đề FROM
- Thuộc tính trùng tên

Ví dụ:

Q1: Hiển thị MANV, HoTen, số năm làm việc (SoNamLamViec).

```
SELECT manv, ho+' '+ten AS hoten,
        year(getdate())-year(ngayvaolam) as sonamlamviec
FROM NhanVien
```

Q2: Liệt kê các nhân viên (HoTen, TenKN, MucDo) của những nhân viên biết sử dụng 'Word'

```
SELECT    ho+' '+ten AS hoten, TenKN, mucdo
FROM      NhanVien A, nhanvien kynang B, Kynang C
WHERE     A.manv = B.manv and B.mskn = C.mskn and TenKN = 'Word'
```

Lưu ý: Các tên A, B, C trong truy vấn Q2 được gọi là bí danh.

Q3 Liệt kê các kỹ năng (TenKN, MucDo) mà nhân viên 'Lê Anh Tuấn' biết sử dụng

```
SELECT    TenKN, mucdo
FROM      NhanVien A, nhanvien kynang B, Kynang C
WHERE     A.manv=B.manv and B.mskn = C.mskn and Ho= N'Lê Anh'
        and Ten=N'Tuấn'
```

d) Mệnh đề ORDER BY

Mặc định, kết quả truy vấn trả về không được sắp xếp theo thứ tự của các thuộc tính. Người sử dụng có thể sắp xếp dữ liệu bằng cách sử dụng mệnh đề ORDER BY.

```
SELECT    ...
FROM      ...
WHERE     ...

ORDER BY tập_thuộc_tính ASC|DESC
```

- *Tập_thuộc_tính* gồm 1 thuộc tính hoặc nhiều thuộc tính và độ ưu tiên sắp xếp tính từ trái sang phải.
- ASC: sắp tăng (mặc định)
- DESC: sắp giảm

Ví dụ:

Q4: Liệt kê các thông tin về nhân viên: HoTen, NgaySinh, NgayVaoLam, TenCN (sắp xếp theo tên chi nhánh)

```
SELECT      Ho+ ' '+Ten AS hoten, NgaySinh, NgayVaoLam, TenCN
FROM        NhanVien A, ChiNhanh B
WHERE       A.mscn=B.mscn
ORDER BY    TenCN
```

III.2 Gom nhóm và Hàm kết hợp

Một số truy vấn mang tính thống kê không thể được trả lời bằng cách sử dụng dạng đơn giản của lệnh SELECT. Các truy vấn này đòi hỏi phải thực hiện tính toán kết hợp. Xem các truy vấn sau:

Q5: Với mỗi chi nhánh, hãy cho biết các thông tin sau TenCN, SoNV (số nhân viên của chi nhánh đó).

Q6: Với mỗi kỹ năng, hãy cho biết TenKN, SoNguoiDung (số nhân viên biết sử dụng kỹ năng đó).

Để tính toán các truy vấn này, cần sử dụng các hàm kết hợp. Các hàm này tính toán trên một cột của một bảng và trả về một trị.

a) Các hàm kết hợp

- COUNT
 - o COUNT(*): đếm số dòng
 - o COUNT(<Tên thuộc tính>) : Đếm số giá trị khác NULL của thuộc tính.
 - o COUNT(DISTINCT <Tên thuộc tính>) : Đếm số giá trị khác nhau và khác NULL của thuộc tính.
- SUM
- AVG
- MAX
- MIN

Lưu ý:

- (i). Các hàm COUNT, MAX và MIN có thể áp dụng cho tất cả các cột có kiểu dữ liệu bất kỳ, trong khi SUM và AVG chỉ áp dụng cho các cột kiểu số.
- (ii). Ngoại trừ COUNT(*) tất cả các hàm khác đều bỏ qua giá trị NULL.
- (iii). Sử dụng DISTINCT để loại bỏ các giá trị trùng trước khi áp dụng các hàm kết hợp.

b) Gom nhóm

Đến đây, ta đã áp dụng các hàm kết hợp trên tất cả các bộ. Nếu muốn áp dụng các hàm trên cho từng nhóm bộ ta sẽ sử dụng mệnh đề **Group By** cùng với các hàm kết hợp.

- ❖ **Mệnh đề GROUP BY** được dùng trong hàm tập hợp để gom nhóm các bộ theo (các) thuộc tính cần gom nhóm. Các bộ nằm trong cùng một nhóm phải có cùng giá trị tại (các) thuộc tính gom nhóm. Tính toán được áp dụng riêng cho từng nhóm. Có một dòng output ứng với mỗi nhóm.

Cú pháp:

```
SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <điều kiện>
GROUP BY <danh sách các cột gom nhóm>
```

Ví dụ:

```
Q5 SELECT TENCN, count(MANV) As SoNV
FROM NhanVien A, ChiNhanh B
WHERE A.mscn=B.mscn
GROUP BY TenCN
```

```
Q6 SELECT TENKN, count(*) As SoNguoiDung
FROM NhanVienKyNang A, KyNang B
WHERE A.mskn=B.mskn
GROUP BY TenKN
```

- ❖ **Mệnh đề HAVING** thường được sử dụng cùng với mệnh đề GROUP BY. Sau HAVING là biểu thức điều kiện. Biểu thức điều kiện này không tác động vào toàn bảng được chỉ ra ở mệnh đề FROM mà chỉ tác động lần lượt từng nhóm các mẫu tin đã chỉ ra trong mệnh đề group by.

- Chọn ra các nhóm thỏa điều kiện nào đó.
- Điều kiện được áp dụng cho mỗi nhóm, và nhóm nào không thỏa điều kiện sẽ bị loại.
- Điều kiện này có thể tham chiếu tới bất kỳ bảng hay biến bộ nào trong mệnh đề FROM, có thể liên quan đến:
 - Một thuộc tính gom nhóm, hoặc
 - Hàm tính toán.

Lưu ý: Điều kiện trong mệnh đề HAVING được áp dụng sau khi hình thành nhóm, trong khi điều kiện trong mệnh đề WHERE được áp dụng trước khi hình thành nhóm.

Cú pháp:

```
SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <điều kiện>
GROUP BY <danh sách các cột gom nhóm>
HAVING <điều kiện trên nhóm>
```

Ví dụ:

Q7: Cho biết TenKN có từ 3 nhân viên trong công ty sử dụng trở lên

```
SELECT    TENKN, count(MANV) As SoNguoiDung
FROM      NhanVienKyNang A, KyNang B
WHERE     A.mskn=B.mskn
GROUP BY  TenKN
HAVING    Count(MANV)>=3
```

III.3 Truy vấn lồng

- Để diễn đạt truy vấn phức tạp, một truy vấn có một/nhiều truy vấn con (subquery) được lồng bên trong.
- Subquery có thể được đặt ở mệnh đề SELECT, FROM, WHERE hay HAVING.
- Khi có các truy vấn lồng nhau thì câu lệnh bên ngoài xác định kết quả cuối cùng, trong khi các câu lệnh SELECT bên trong được sử dụng bởi lệnh bên ngoài (thường để lookup các giá trị cho mệnh đề WHERE).
- Các loại Subquery
 1. scalar subquery: trả về một giá trị đơn. Giá trị này sau đó dùng để so sánh.
 2. row subquery: trả về một dòng đơn có thể có nhiều cột.
 3. table subquery: trả về là một hay nhiều cột và nhiều dòng.

❖ **Subquery trong mệnh SELECT**

- Thuộc loại scalar subquery.

❖ **Subquery trong mệnh FROM**

- Sử dụng subquery như là *derived table*.
- Subquery được đặt trong dấu ngoặc và phải được gán một bí danh.
- Các cột từ subquery này có thể được dùng như các cột trong các bảng bình thường khác.

```
SELECT <danh sách các cột>
FROM R1, R2, (<truy vấn con>) AS tên_bảng
WHERE <điều kiện>
```

❖ **Subquery trong mệnh WHERE**

- Trong trường hợp này, subquery thường là table query
- Các loại kiểm tra điều kiện:
 - Kiểm tra so sánh thông thường (=, <, >, <=, >=): so sánh giá trị của một biểu thức với một giá trị đơn được tạo ra bởi một subquery.

- Kiểm tra tập hợp thành viên (IN /NOT IN): Kiểm tra xem một giá trị của một biểu thức có mặt/không có mặt trong các giá trị được tạo ra bởi một subquery.
- Kiểm tra so sánh lượng hóa (ANY, ALL): So sánh một giá trị của một biểu thức với từng giá trị thuộc tập hợp các giá trị được tạo ra bởi một subquery.

Ví dụ:

Q8 : Liệt kê MANV, HoTen, MSCN của các nhân viên giỏi ‘Excel’ nhất trong công ty

```

Select  A.manv, Ho+ ' '+Ten AS HoTen, MSCN
From    nhanvien A, nhanvienkynang B, Kynang C
Where   A.manv = B.manv and B.mskn = C.mskn and TenKN = 'Excel'
        And MucDo = (Select      Max(MucDo)
                      From        nhanvienkynang E, kynang F
                      Where       E.mskn = F.mskn and TenKN = 'Excel')
    
```

Q9: Liệt kê MANV, HoTen, MSCN của các nhân viên vừa biết ‘Word’ vừa biết ‘Excel’ .

```

Select  A.manv, Ho+ ' '+Ten AS HoTen, MSCN
From    nhanvien A, nhanvienkynang B, Kynang C
Where   A.manv = B.manv and B.mskn = C.mskn and TenKN = 'Word'
        and A.manv IN (Select    E.manv
                      From        nhanvienkynang E, kynang F
                      Where       E.mskn = F.mskn and TenKN = 'Excel')
    
```

Q10: Với từng kỹ năng, hãy liệt kê các thông tin (TenKN, MANV, HoTen, MSCN, MucDo) của những nhân viên giỏi kỹ năng đó nhất. Yêu cầu sắp kết quả tăng dần theo TenKN.

```

Select  TenKN, A.manv, Ho+ ' '+Ten AS HoTen, MSCN, B.MucDo
From    nhanvien A, nhanvienkynang B, Kynang C
Where   A.manv = B.manv and B.mskn = C.mskn
        And MucDo = (Select      Max(MucDo)
                      From        nhanvienkynang E
                      Where       E.mskn = C.mskn)
    
```

Order By TenKN

- Phân loại truy vấn lồng:
 - *Lồng phân cấp*: Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha. Khi thực hiện, câu truy vấn con sẽ được thực hiện trước. Ví dụ các truy vấn Q8, Q9.
 - *Lồng tương quan*: Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha. Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha. Ví dụ truy vấn Q10 là truy vấn lồng tương quan.

- Hàm EXISTS và NOT EXISTS: được dùng để kiểm tra kết quả của một câu truy vấn có rỗng hay không.
 - EXISTS (Query) trả về trị TRUE nếu kết quả của Query là khác rỗng, ngược lại EXISTS (Query) trả về trị FALSE.
 - NOT EXISTS (Query) trả về trị TRUE nếu kết quả của Query là rỗng (không có dòng nào), ngược lại EXISTS (Query) trả về trị FALSE.

Q11: Liệt kê các nhân viên không sử dụng được 'SPSS'

```
Select MaNV, Ho+ ' '+Ten AS HoTen, MSCN
From nhanvien A
Where NOT EXISTS (Select *
                  From nhanvienkynang E, kynang F
                  Where E.mskn = F.mskn and TenKN = 'SPSS'
                  and A.manv = E.manv)
```

Các quy tắc cú pháp của Subquery

- ▶ Mệnh đề ORDER BY không thể dùng trong subquery.
- ▶ Số lượng thuộc tính trong mệnh đề SELECT của subquery phải khớp với số lượng thuộc tính được so sánh với toán tử so sánh.
- ▶ Tên các cột trong một subquery mặc định là lấy từ mệnh đề FROM của subquery. Nếu muốn truy xuất đến một bảng xuất hiện ở cả truy vấn trong và ngoài ta phải sử dụng bí danh.

Phép toán tập hợp trong SQL

Các phép toán tập hợp: Hợp (UNION), giao (INTERSECT), Hiệu (EXCEPT) liên quan tới hai câu truy vấn riêng biệt được biểu diễn như sau:

```
(Query1) UNION [ALL] (Query2)
(Query1) INTERSECT [ALL] (Query2)
(Query1) EXCEPT [ALL] (Query2)
```

Lưu ý:

- Khi viết truy vấn phải đảm bảo tính khả hợp.
- Dùng từ khóa ALL khi cần giữ lại các dòng trùng. Mặc định các dòng trùng bị loại bỏ.
- Phép UNION được tất cả các DBMS hỗ trợ, còn các phép toán khác thì tùy vào từng DBMS. Trong trường hợp không được hỗ trợ, có thể phát biểu phép toán INTERSECT, EXCEPT qua truy vấn lồng.

Phép chia trong SQL

- Được dùng để phát biểu các câu truy vấn “tất cả”.
- Xét ví dụ: Cho 2 quan hệ R(A, B), S(B). Viết bằng SQL $R \div S$

- Chúng ta có thể diễn đạt lại: $R \div S$ là tập các giá trị a_i trong R sao cho không có giá trị b_j nào trong S làm cho bộ (a_i, b_j) không tồn tại trong R

R	A	B
	α	1
	α	2
	β	1
	γ	1
	γ	2

S	B
	1
	2

$R \div S$	A
	α
	γ

```

SELECT      a.A
FROM        R a
WHERE NOT EXISTS (SELECT *
                  FROM      S b
                  WHERE NOT EXISTS (SELECT *
                                    FROM R c
                                    WHERE c.A = a.A and c.B=b.B))
    
```

Ví dụ

Q12: Cho biết họ tên, MSCN của nhân viên biết sử dụng tất cả các kỹ năng

```

Select  Ho+ ' ' +Ten AS HoTen, MSCN
From    nhanvien A
Where   Not Exists (select *
                   From Kynang B
                   Where not exists (select *
                                    From nhanvienkynang C
                                    Where C.mskn = B.mskn and A.manv=C.manv))
    
```

Có thể phát biểu Q12 ở bằng cách dùng phép đếm như sau:

```

Select      Ho+ ' ' +Ten AS HoTen, MSCN
From        nhanvien A, nhanvienkynang B
Where       A.manv=B.manv
Group by    A.manv, Ho, Ten, MSCN
Having      count(mskn) = (select count(*)
                          From Kynang)
    
```

III.4 Một số kiểu truy vấn khác

Các loại phép kết:

INNER JOIN
LEFT OUTER JOIN
RIGHT OUTER JOIN
FULL OUTER JOIN

Cú pháp

SELECT <danh sách các cột>
FROM R1 [INNER] JOIN R2 ON <điều kiện kết>
WHERE <điều kiện>

SELECT <danh sách các cột>
FROM R1 LEFT|RIGHT|FULL [OUTER] JOIN R2 ON <biểu thức>
WHERE <điều kiện>

IV. Cập nhật dữ liệu

IV.1 Lệnh INSERT

- Dùng để thêm 1 hay nhiều dòng vào bảng
- Cú pháp:
 - *Thêm 1 dòng:*
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)
VALUES (<danh sách các giá trị>)
 - *Thêm nhiều dòng*
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)
<câu truy vấn>
- **Nhận xét**
 - Thứ tự các giá trị phải trùng với thứ tự các cột.
 - Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và NOT NULL.
 - Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV
 - Khóa chính
 - Tham chiếu
 - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị

IV.2 Lệnh DELETE

- Dùng để xóa các dòng của bảng
- Cú pháp


```
DELETE FROM <tên bảng>
[WHERE <điều kiện>]
```
- Nhận xét
 - Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE.
 - Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa.
 - Lệnh DELETE có thể gây ra vi phạm RB tham chiếu. Các cách xử lý có thể là:
 - Không cho xóa
 - Xóa luôn những dòng có giá trị đang tham chiếu đến
 - Đặt NULL cho những giá trị tham chiếu

IV.3 Lệnh UPDATE

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp


```
UPDATE <tên bảng>
SET <tên thuộc tính>=<giá trị mới>,
    <tên thuộc tính>=<giá trị mới>,
    ...
[WHERE <điều kiện>]
```
- Nhận xét
 - Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
 - Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị cập nhật
 - Lệnh UPDATE có thể gây ra vi phạm RB tham chiếu
 - Không cho sửa
 - Sửa luôn những dòng có giá trị đang tham chiếu đến

V. Khung nhìn (View)

- Cú pháp Tạo khung nhìn


```
CREATE VIEW <tên khung nhìn> AS
    <câu truy vấn>
```

- Khung nhìn là một bảng ảo có:
 - ▶ Danh sách thuộc tính trùng với các thuộc tính trong mệnh đề SELECT
 - ▶ Số dòng phụ thuộc vào điều kiện ở mệnh đề WHERE
 - ▶ Dữ liệu được lấy từ các bảng ở mệnh đề FROM
- *Truy vấn trên khung nhìn*: tuy không chứa dữ liệu nhưng có thể thực hiện các câu truy vấn trên khung nhìn
- *Cập nhật trên khung nhìn*: có thể dùng các câu lệnh INSERT, DELETE và UPDATE cho các khung nhìn được xây dựng trên 1 bảng và có khóa chính của bảng
- Không thể cập nhật dữ liệu nếu:
 - Khung nhìn có dùng từ khóa DISTINCT
 - Khung nhìn có sử dụng các hàm kết hợp
 - Khung nhìn có mệnh đề SELECT mở rộng
 - Khung nhìn được xây dựng từ bảng có RB trên cột
 - Khung nhìn được xây dựng từ nhiều bảng
- Xóa khung nhìn

DROP VIEW <tên khung nhìn>

VI. Chỉ mục (Index)

- Chỉ mục trên thuộc tính A là một cấu trúc dữ liệu làm cho việc tìm kiếm mẫu tin có chứa A hiệu quả hơn.
- Cú pháp tạo chỉ mục:

CREATE INDEX <tên chỉ mục> **ON** <tên bảng>(<tên cột>)
- Cú pháp xóa chỉ mục:

DROP INDEX <tên chỉ mục>
- Nhận xét
 - Chỉ mục giúp cho việc tìm kiếm nhanh trong trường hợp so sánh với hằng số và phép kết.
 - Làm chậm đi các thao tác thêm, xóa và sửa.
 - Tốn chi phí
 - Lưu trữ chỉ mục
 - Truy xuất đĩa nhiều
 - Nên chọn lựa cài đặt chỉ mục hợp lý.

Chương 5

RÀNG BUỘC TOÀN VỆN

I. Các khái niệm

- Ràng buộc toàn vẹn (Integrity Constraints) là những điều kiện, quy định được áp dụng trên cơ sở dữ liệu sao cho mọi cập nhật cơ sở dữ liệu (thêm, xóa, sửa) đều phản ánh đúng thể giới thực mà nó mô hình hóa.
- Nguồn gốc của ràng buộc toàn vẹn (RBTV):
 - Xuất phát từ những quy định nghiệp vụ trong hệ thống thông tin và những mô tả trong mô hình dữ liệu.
 - RBTV được mô tả dựa trên ngữ nghĩa của thể giới thực được mô hình hóa trong CSDL.
- Mục tiêu của RBTV:
 - Bảo đảm tính kết dính của các bộ phận cấu thành nên CSDL.
 - Bảo đảm tính nhất quán về mặt logic của CSDL.
 - Bảo đảm CSDL luôn biểu diễn đúng ngữ nghĩa thực tế.
- RBTV được mô tả khi định nghĩa lược đồ.
- RBTV được kiểm tra khi cập nhật các quan hệ.
- RBTV được định nghĩa bởi người thiết kế CSDL.
- RBTV được chia sẻ bởi tất cả các ứng dụng sử dụng CSDL đó.
- Một thể hiện quan hệ là hợp lệ nếu nó thỏa tất cả các RBTV được mô tả.
- Các đặc trưng của một RBTV: nội dung, bối cảnh, bảng tầm ảnh hưởng.
 - Nội dung RBTV: các phát biểu của quy tắc, điều kiện được kiểm tra trên mọi thể hiện của các quan hệ.
 - Nội dung được phát biểu bằng ngôn ngữ tự nhiên và ngôn ngữ hình thức (thông qua Đại số quan hệ, Phép tính quan hệ, Mã giả, ...).
 - Bối cảnh RBTV: một hay nhiều quan hệ cần phải sử dụng để kiểm tra RBTV. Các quan hệ này có khả năng bị vi phạm RBTV khi thực hiện các thao tác thêm, xóa, sửa dữ liệu.
 - Bảng tầm ảnh hưởng: xác định thao tác cập nhật (thêm, xóa, sửa) nào cần phải kiểm tra RBTV trên quan hệ nào.

Ví dụ

Xét lược đồ cơ sở dữ liệu sau:

- HOADON(SoHD, NgayHD, MaKH, HThucTra, TienCon)

- CTIET_HD(SoHD, MaHH, SoLG)
- MATHANG(MaHH, TenMH, QUYCACH, DVT)
- BAOGIA(MAMH, NGAYBG, DONGIA)
- PHIEUTHU(SoPT, NgayThu, SoHD, SoTien, HThucTra)

Ta có các ràng buộc toàn vẹn sau:

RB1: “Hai bộ khác nhau phải khác nhau tại giá trị khóa.”

RB2: “Hình thức thanh toán có thể là tiền mặt hoặc chuyển khoản.”

RB3: “Mỗi chi tiết hóa đơn chỉ nằm trong một hóa đơn duy nhất và có một mặt hàng duy nhất.”

Phát biểu bằng ngôn ngữ hình thức:

RB1:

$\forall t, s \in \text{hoadon}: t \neq s \Rightarrow t.\text{SoHD} \neq s.\text{SoHD}$

$\forall t, s \in \text{ctiet_hd}: t \neq s \Rightarrow t[\text{SoHD}, \text{MaHH}] \neq s[\text{SoHD}, \text{MaHH}]$

$\forall t, s \in \text{mathang}: t \neq s \Rightarrow t.\text{MaHH} \neq s.\text{MaHH}$

RB2: $\text{DOM}(\text{HthucTra}) = \{0, 1\}$

0: Tiền mặt

1: Chuyển khoản

RB3: $\forall t \in \text{ctiet_hd}, \exists! h \in \text{hoadon}: t.\text{SoHD} = h.\text{SoHD}$

$\exists! m \in \text{mathang}: t.\text{MaHH} = m.\text{MaHH}$

Lưu ý: vì CTIET_HD thể hiện cho tập thực thể yếu, một dòng trong bảng tồn tại do sự tồn tại của một dòng trong HOADON và một dòng trong MATHANG nên ràng buộc RB3 thật ra có thể được kiểm tra bởi:

RB4 : “Mỗi hóa đơn phải có ít nhất một chi tiết hóa đơn”

$\forall h \in \text{hoadon}, \exists c \in \text{ctiet_hd}: h.\text{SoHD} = c.\text{SoHD}$

RB5: $\forall c \in \text{ctiet_hd}, \exists! m \in \text{mathang}: c.\text{MaHH} = m.\text{MaHH}$.

Để đảm bảo các RBTV RB4, RB5 người ta có thể xây dựng các giao tác kiểm soát quá trình cập nhật như sau:

GIAO TÁC Thêm hoadon

Thêm t vào hoadon;

Thêm c1 và ctiet_hd;

Thêm c2 và ctiet_hd;

.

.

.

NẾU (có 1 phép thêm thất bại) \vee

$c1.SoHD \neq t.SoHD \vee$ (không có $u1 \in mathang: u1[MaHH] = c1[MaHH]$)

$c2.SoHD \neq t.SoHD \vee$ (không có $u2 \in mathang: u2[MaHH] = c2[MaHH]$)

·
·
·

THÌ

Quay lui giao tác;

NGƯỢC LẠI

Hoàn tất giao tác;

CUỐI NẾU

CUỐI GIAO TÁC

GIAO TÁC Xóa hoadon (so_hoa_don)

Với mọi $c \in ctiet_hd: c.SoHD = so_hoa_don$

Xóa c khỏi ctiet_hd

Cuối với mọi

Xóa $h \in hoadon : h.SoHD = so_hoa_don$

NẾU (có 1 phép xóa thất bại) **THÌ**

Quay lui giao tác;

NGƯỢC LẠI

Hoàn tất giao tác;

CUỐI NẾU

CUỐI GIAO TÁC

GIAO TÁC Sửa hoadon

Ghi chú: *chỉ cho phép sửa các giá trị NgayHD, MaKH, HThucTra, TienCon trên HOADON và SoLG trên CTIET_HD*

Sửa giá trị của bộ $h \in hoadon$

Sửa giá trị của một (hoặc nhiều) bộ $c \in ctiet_hd : h.SoHD = c.SoHD$

NẾU (có một phép sửa thất bại) **THÌ**

Quay lui giao tác

NGƯỢC LẠI

Hoàn tất giao tác;

CUỐI NẾU

CUỐI GIAO TÁC

Bối cảnh:

RB1: HOADON, CTIET_HD, MATHANG

RB2: HOADON

RB4: HOADON, CTIET_HD

RB5: CTIET_HD, MATHANG

Bảng tầm ảnh hưởng

(Giả sử quy định không cho phép sửa giá trị của các thuộc tính khóa) của RB1

RB1	Thêm	Xóa	Sửa
HOADON	+	-	-
CTIET_HD	+	-	-
MATHANG	+	-	-

Bảng tầm ảnh hưởng tổng hợp cho các RBTV RB1, RB2, RB4, RB5

	HOADON			CTIET_HD			MATHANG		
	T	X	S	T	X	S	T	X	S
RB1	+	-	-	+	-	-	+	-	-
RB2	+	-	+(HthucTra)						
RB4	+	-	-	-	+	-			
RB5				+	-	-	-	+	-

II. Phân loại ràng buộc toàn vẹn

RBTV trên một lược đồ quan hệ: RBTV có bối cảnh trên một quan hệ

- RBTV miền giá trị
- RBTV liên bộ
- RBTV liên thuộc tính

RBTV trên nhiều lược đồ quan hệ: RBTV có bối cảnh trên nhiều quan hệ

- RBTV tham chiếu (RBTV khóa ngoại)
- RBTV liên bộ - liên quan hệ
- RBTV liên thuộc tính – liên quan hệ
- RBTV do thuộc tính tổng hợp
- RBTV chu trình

II.1 Ràng buộc toàn vẹn có bối cảnh trên 1 quan hệ

a. RBTV miễn giá trị

Qui định các giá trị mà một thuộc tính có thể nhận

$$\text{DOM}(\text{HthucTra}) = \{0,1\}$$

$$\text{SoLG} > 0$$

b. RBTV liên bộ

- Thường là ràng buộc về sự tồn tại của một hoặc nhiều bộ có liên quan đến các bộ khác trong cùng một quan hệ
- Trường hợp đặc biệt: RB khóa chính, RB duy nhất
- Ví dụ:

THIDAU (NGAY, GIO, DOI, BANTHANG, KQUA)

RB6: “Mỗi trận đấu là cuộc thi đấu của đúng 2 đội”

$$\forall t \in \text{thidau}, \exists! s \in \text{thidau}: s \neq t \wedge t[\text{NGAY}, \text{GIO}] = s[\text{NGAY}, \text{GIO}]$$

Bối cảnh: THIDAU

Bảng tầm ảnh hưởng:

RB6	Thêm	Xóa	Sửa
THIDAU	+	+	+ (NGAY, GIO, DOI)

(Bài tập: Viết giao tác tương ứng cho RB6)

c. RBTV liên thuộc tính

- Là ràng buộc giữa các thuộc tính trong cùng một bộ của cùng một lược đồ quan hệ. Thông thường là các phụ thuộc tính toán, một suy diễn giá trị của một hay nhiều thuộc tính (còn gọi là thuộc tính suy ra) trong cùng một bộ hoặc ràng buộc liên quan đến các thuộc tính thời gian.
- Ví dụ: Xét LĐQH

PHANLOAI(MALOP, NAMHOC, HOCKY, DIEMTB, LOAI)

RB7: “Lớp xếp loại ‘Tốt’ nếu điểm trung bình chung (DIEMTB) của toàn bộ các học sinh của lớp ở mỗi học kỳ ≥ 7.5 , loại ‘Kém’ nếu $\text{DIEMTB} < 4.5$, còn lại xếp loại Trung bình”

Phát biểu hình thức

$$\forall t \in \text{phanloai} : (t[\text{DIEMTB}] \geq 7.5 \wedge t[\text{LOAI}] = \text{'Tốt'}) \vee$$

$$(t[\text{DIEMTB}] \geq 4.5 \wedge t[\text{DIEMTB}] < 7.5 \wedge t[\text{LOAI}] = \text{'Trung bình'}) \vee$$

$$(t[\text{DIEMTB}] < 4.5 \wedge t[\text{LOAI}] = \text{'Kém'})$$

Bảng tầm ảnh hưởng:

RB7	Thêm	Xóa	Sửa
PHANLOAI	+	-	+(DIEMTB, LOAI)

II.2 Ràng buộc toàn vẹn có bối cảnh trên nhiều quan hệ

a) RBTV tham chiếu

- Là ràng buộc về sự phụ thuộc tồn tại
- Ví dụ: Trong lược đồ cơ sở dữ liệu trên có các RBTV tham chiếu là:

RB5: “Mỗi chi tiết hóa đơn liên hệ với một mặt hàng duy nhất”

RB4’: “Mỗi chi tiết hóa đơn liên hệ với duy nhất một hóa đơn.”

(RB4’, RB5 là hai thành phần RBTV kết hợp nên RB3)

RB8: “Cập nhật biến động giá (nếu có) chỉ liên hệ đến từng mặt hàng mà công ty có”

$\forall t \in \text{baogia}, \exists! s \in \text{mathang}: t[\text{MaHH}] = s[\text{MaHH}]$

Có thể phát biểu : $\text{baogia}[\text{MaMH}] \subseteq \text{mathang}[\text{MaHH}]$

Bảng tầm ảnh hưởng cho R8:

RB8	Thêm	Xóa	Sửa
BAOGIA	+	-	+(MaHH)
MATHANG	-	+	+(MaHH)

b) RBTV liên bộ - liên quan hệ

- Xây ra giữa các bộ trên nhiều quan hệ khác nhau
- Ví dụ: Trong lược đồ cơ sở dữ liệu trên, các RBTV là RBTV liên bộ - liên quan hệ

RB4: “ Mỗi hóa đơn có ít nhất một chi tiết hóa đơn.”

RB9: “Mỗi mặt hàng có ít nhất một lần báo giá.”

$\forall t \in \text{mathang}, \exists s \in \text{baogia}: t[\text{MaHH}] = s[\text{MaHH}]$

Bảng tầm ảnh hưởng

RB9	Thêm	Xóa	Sửa
MATHANG	+	-	+(MaHH)
BAOGIA	-	+	+(MaHH)

Lưu ý: cần phân biệt giữa RB8 và RB9.

c) TBTV liên thuộc tính – liên quan hệ

- Là ràng buộc xảy ra giữa các thuộc tính trên nhiều quan hệ khác nhau
- Ví dụ:

RB10: “Ngày khách trả tiền không được trước ngày ghi trong hóa đơn của họ”

$\forall t \in \text{phieuthu}, \exists! s \in \text{hoadon}: t[\text{SoHD}] = s[\text{SoHD}] \wedge t[\text{NgayThu}] \geq s[\text{NgayHD}]$

Bảng tầm ảnh hưởng

RB10	Thêm	Xóa	Sửa
HOADON	-	+	+(SoHD, NgayHD)
PHIEUTHU	+	-	+(SoHD, NgayThu)

d) RBTV do thuộc tính tổng hợp

- Thuộc tính tổng hợp là thuộc tính được tính toán từ giá trị của các thuộc tính khác (bằng các hàm Sum, Count).
- Khi hệ thống có thuộc tính tổng hợp, cần kiểm soát RBTV để đảm bảo việc tính toán giá trị của các thuộc tính này.
- Ví dụ: xét hai lược đồ

HOADON(SoHD, NgayLap, MaKH, TriGiaHD)

CTIET_HD(SoHD, MaHH, SL, DonGia)

RB11: “Trị giá hóa đơn bằng tổng trị giá của các mặt hàng trong hóa đơn đó.

$$\forall h \in \text{hoadon}: h.\text{TriGiaHD} = \sum_{\substack{c \in \text{ctiet_hd} \\ \wedge c.\text{SoHD} = h.\text{SoHD}}} c.SL * c.DonGia$$

Bảng tầm ảnh hưởng

RB11	Thêm	Xóa	Sửa
HOADON	+	-	+(SoHD, TriGiaHD)
CTIET_HD	+	+	+(SoHD, SL, DonGia)

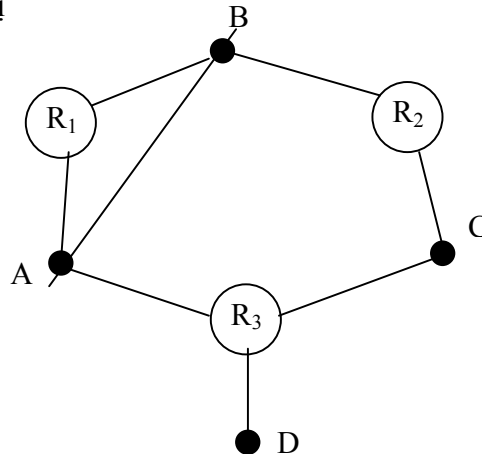
e) RBTV chu trình

- Lược đồ CSDL có thể được biểu diễn bằng đồ thị vô hướng:
 - Tập đỉnh: các quan hệ và các thuộc tính
 - Quan hệ : Tên quan hệ
 - Thuộc tính: Tên thuộc tính
 - Cạnh: đường nối một đỉnh quan hệ với một đỉnh thuộc tính trong lược đồ CSDL.
 - Lược đồ CSDL có sự hiện diện của một chu trình khi xuất hiện đường đi khép kín trên đồ thị.

Cho một thiết kế CSDL gồm các lược đồ quan hệ:

R₁(A, B) R₂(B, C) R₃(A, C, D)

Biểu diễn đồ thị



Xét $q \leftarrow \pi_{A,B}(r_2(R_2) \bowtie r_3(R_3))$ và $p \leftarrow \pi_{A,B}(r_1(R_1))$; với mọi quan hệ r_1, r_2, r_3 hợp lệ:

- (i). Nếu $q = p$ ta nói hai con đường là tương đương.
- (ii). Nếu $q \subseteq p$ hoặc $p \subseteq q$ ta nói có một con đường phụ thuộc vào con đường kia.
- (iii). Nếu không thể so sánh giữa p và q (nghĩa là $p \neq q$) ta nói hai con đường là độc lập nhau (và gọi là chu trình thật sự).

Trường hợp (i) hoặc (ii) gọi là chu trình thật sự.

Trong một chu trình thật sự:

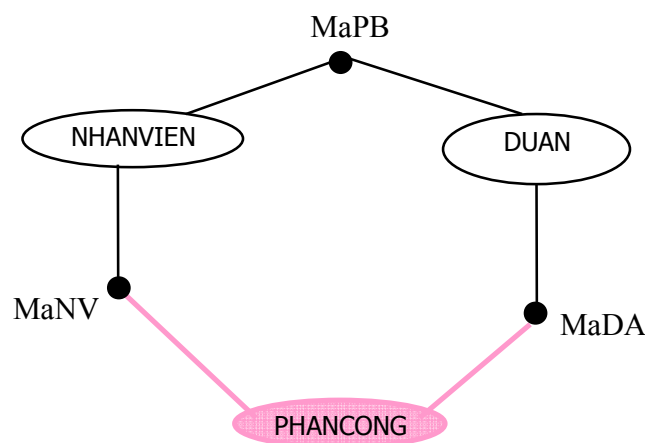
- Nếu (i) xảy ra: hủy đường đi ngắn $R_1(A, B)$ (Khi R_1 không có thuộc tính nào khác ngoài $\{A, B\}$).
- Nếu (ii) xảy ra: xác định RBTV để kiểm tra.

- **Ví dụ:** Xét Thiết kế CSDL gồm các quan hệ sau:

NhanVien(MaNV, HoTen, NgaySinh, DiaChi, DienThoai, MaPB)

DuAn(MaDA, TenDA, DiaDiem, MaPB)

PhanCong(MaNV, MaDA, ThoiGian)



RB12: “Nhân viên chỉ được phân công vào các dự án do phòng ban mình phụ trách.”

$$\pi_{\text{MaNV}, \text{MaDA}}(\text{phancong}) \subseteq \pi_{\text{MaNV}, \text{MaDA}}(\text{nhanvien} \bowtie \text{duan})$$

Bảng tầm ảnh hưởng

RB12	Thêm	Xóa	Sửa
PHANCONG	+	-	+ (MaNV, MaDA)
NHANVIEN	-	+	+ (MaNV, MaPB)
DUAN	-	+	+ (MaDA, MaPB)

III. Cài đặt ràng buộc toàn vẹn

III.1 Quy trình tìm ràng buộc toàn vẹn trên một lược đồ CSDL:

Bước 1: Phát hiện ràng buộc toàn vẹn: tìm, phân loại và phát biểu RBTV bằng ngôn ngữ tự nhiên:

- Với từng lược đồ quan hệ, hãy tìm:
 - RBTV miền giá trị
 - RBTV liên bộ
 - RBTV liên thuộc tính
- Với nhiều lược đồ quan hệ, hãy tìm
 - RBTV tham chiếu (RBTV khóa ngoại)
 - RBTV liên bộ - liên quan hệ
 - RBTV liên thuộc tính – liên quan hệ
 - RBTV do thuộc tính tổng hợp
 - RBTV chu trình

Bước 2: Phát biểu và lập bảng tầm ảnh hưởng

Với từng RBTV đã tìm thấy ở bước 1, phát biểu (biểu diễn) hình thức và lập bảng tầm ảnh hưởng cho nó, đề nghị cách xử lý khi RBTV bị vi phạm.

Bước 3: Dựa trên các bảng tầm ảnh hưởng của từng RBTV đã làm ở bước 2, lập bảng tầm ảnh hưởng tổng hợp.

III.2 Các cơ chế cài đặt và kiểm tra RBTV

- Các cơ chế kiểm tra RBTV được hầu hết các hệ QTCSDL hỗ trợ như:
 - Miền giá trị
 - Khóa chính (Primary key), duy nhất (Unique)
 - Khóa ngoại (Foreign key)
 - Ép thỏa đơn giản: check, rule, ... (chẳng hạn SQL Server): kiểm tra RBTV miền giá trị
 - Ép thỏa phức tạp: Trigger (của SQL Server): kiểm tra ràng buộc ngữ nghĩa phức tạp như liên thuộc tính, liên thuộc tính - liên quan hệ, liên bộ, liên bộ - liên quan hệ, thuộc tính tổng hợp, chu trình.

- Cần xây dựng giao tác cho các loại RBTV: Liên bộ (ngoại trừ trường hợp RBTV khóa chính, RBTV duy nhất), liên bộ - liên quan hệ
- Dùng Stored Procedure
- Nên cài đặt RBTV ở :
 - DBMS: các RBTV miền giá trị, khóa chính, duy nhất, khóa ngoại.
 - Ứng dụng (Application): các RBTV Liên thuộc tính, liên thuộc tính – liên quan hệ, liên bộ - liên quan hệ, thuộc tính tổng hợp, chu trình.

Lưu ý:

- *Sử dụng Trigger quá nhiều → hệ thống chậm chạp*
 - *Sử dụng Stored Procedure → hiệu quả cao.*
-

Chương 6

LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ

I. Mở đầu

Quá trình thiết kế cơ sở dữ liệu tích hợp các dữ liệu liên quan từ các yêu cầu về thông tin/dữ liệu, yêu cầu khai thác/xử lý, và đặc trưng của hệ quản trị để có một cấu trúc cơ sở dữ liệu hiệu quả, phù hợp với ứng dụng và môi trường triển khai.

Các tiêu chí thiết kế cơ sở dữ liệu: tính hiệu quả, tính toàn vẹn, tính an toàn, tính uyển chuyển của cấu trúc CSDL (dễ dàng thêm, bớt, sửa đổi trong tương lai), dễ cài đặt. Ngoài các tiêu chí ở trên, thiết kế cơ sở dữ liệu phải đảm bảo dữ liệu không dư thừa.

Nếu một cơ sở dữ liệu bị trùng lặp thông tin nó sẽ gây ra một số vấn đề như bất tiện khi cập nhật quan hệ, không biểu diễn được thông tin, mất thông tin.

Để có thể thiết kế một lược đồ cơ sở dữ liệu tốt, cần xem xét sự phụ thuộc dữ liệu. Trong chương này chúng ta sẽ xem xét một loại ràng buộc đặc biệt gọi là *phụ thuộc hàm*. Phụ thuộc hàm cùng với hệ thống các *dạng chuẩn* sẽ là công cụ giúp cho chúng ta đánh giá chất lượng của một thiết kế cơ sở dữ liệu và phát hiện ra các thiết kế tốt cũng như cải tiến chất lượng một thiết kế cơ sở dữ liệu.

II. Phụ thuộc hàm

- Phụ thuộc hàm (functional dependency) là một công cụ dùng để biểu diễn một cách hình thức các ràng buộc toàn vẹn (vấn đề: ràng buộc). Phương pháp biểu diễn này có rất nhiều ưu điểm, và gắn chặt với lý thuyết thiết kế cơ sở dữ liệu.
- Phụ thuộc hàm được ứng dụng trong việc giải quyết các bài toán tìm khóa, tìm phủ tối thiểu và chuẩn hóa cơ sở dữ liệu.
- Ví dụ: Xét lược đồ quan hệ K_HANG(#KH, TEN, ĐC) và thể hiện của quan hệ:

#KH	TEN	ĐC
001	An	2 HBT
002	Nam	12 NCT
003	An	10 Nguyễn Du
004	Hải	36 LHP

- Ta thấy, dùng #KH ta xác định được TEN, ĐC nhưng TEN không xác định được ĐC. Khi đó ta nói: #KH **xác định** TEN hay TEN **phụ thuộc hàm vào** #KH. #KH **xác định** ĐC hay ĐC **phụ thuộc hàm vào** #KH.
- Kí hiệu: #KH \rightarrow TEN
#KH \rightarrow ĐC

II.1 Các định nghĩa

Định nghĩa phụ thuộc hàm (Functional dependencies)

Cho lược đồ quan hệ (LĐQH) R , $\emptyset \neq X \subseteq R^+$, $\emptyset \neq Y \subseteq R^+$. Phụ thuộc hàm $X \rightarrow Y$ định nghĩa trên R nếu như với mọi quan hệ r hợp lệ (quan hệ r được gọi là hợp lệ nếu nó thỏa tất cả các RBTV được mô tả) ta có:

$$\forall t_1, t_2 \in r: t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

(ta nói: **X xác định Y** hay **Y phụ thuộc hàm vào X**).

Ký hiệu: $X \not\rightarrow Y$ nghĩa là không có sự phụ thuộc hàm giữa X và Y .

Phụ thuộc hàm tầm thường (phụ thuộc hàm hiển nhiên - Trivial Dependencies)

$X \subseteq R^+$, $Y \subseteq R^+$, phụ thuộc hàm $X \rightarrow Y$ là phụ thuộc hàm tầm thường nếu $Y \subseteq X$.

Phụ thuộc hàm nguyên tố

$X \subseteq R^+$, $Y \subseteq R^+$, phụ thuộc hàm $X \rightarrow Y$ là phụ thuộc hàm nguyên tố nếu $\forall \emptyset \neq X' \subset X$ thì $X' \not\rightarrow Y$.

Phụ thuộc hàm nguyên tố còn được gọi là phụ thuộc hàm đầy đủ.

Gọi F là tập tất cả các phụ thuộc hàm định nghĩa trên lược đồ quan hệ R . Chúng ta qui ước rằng chỉ cần mô tả các phụ thuộc hàm *không tầm thường* trong tập F (các phụ thuộc hàm tầm thường được hiểu ngầm là cũng có mặt trong F).

Để thuận tiện cho việc tính toán ta kí hiệu XY là hợp của 2 tập hợp X và Y thay cho kí hiệu $X \cup Y$.

II.2 Hệ tiên đề Armstrong

- **Luật phản xạ (reflexive rule):** $X \subseteq R^+$ và $Y \subseteq X$ thì $X \rightarrow Y$.
- **Luật mở rộng (tăng trưởng - augmentation rule):**

Nếu $X \rightarrow Y$ và $W \subseteq R^+$ thì $WX \rightarrow WY$.

- **Luật bắc cầu (transitive rule):**

Nếu $X \rightarrow Y$ và $Y \rightarrow Z$ thì $X \rightarrow Z$.

Các luật hệ quả

- **Hệ quả 1 (Luật hợp - union rule)**

Nếu $X \rightarrow Y$ và $X \rightarrow Z$ thì $X \rightarrow YZ$.

- **Hệ quả 2 (Luật phân rã, Luật tách - decomposition rule)**

Nếu $X \rightarrow YZ$ thì $X \rightarrow Y$ và $X \rightarrow Z$.

- **Hệ quả 3 (Tính bắc cầu giả - pseudo transitive rule)**

Nếu $X \rightarrow Y$ và $WY \rightarrow Z$ thì $WX \rightarrow Z$.

Phụ thuộc hàm hệ quả

Cho LĐQH R và tập phụ thuộc hàm F . Ta nói $f: X \rightarrow Y, X \subseteq R^+, Y \subseteq R^+$ là phụ thuộc hàm hệ quả của F nếu f được suy diễn từ F dựa vào hệ tiên đề Armstrong và được kí hiệu là $F \models f$.

Bao đóng của tập phụ thuộc hàm F

Bao đóng của F , kí hiệu F^+ , là tập hợp tất cả các phụ thuộc hàm có thể suy diễn từ F :

$$F^+ = \{f \mid F \models f\}.$$

Nhận xét:

- $F \subset F^+$
- Nếu $G \subseteq F \Rightarrow G^+ \subseteq F^+$
- Các luật dẫn của hệ tiên đề Armstrong cung cấp một phương tiện tiếp cận khá tốt để tìm F^+ , tuy nhiên nếu lược đồ quan hệ R có tập phụ thuộc hàm khá lớn thì không dễ dàng kiểm soát được hết các phụ thuộc hàm suy diễn được từ F , chính vì vậy, cần có một cách tiếp cận khác hiệu quả hơn.

II.3 Bao đóng tập thuộc tính (closures of attributes set)

Định nghĩa bao đóng tập thuộc tính X dựa trên F

- Giả sử F là tập các phụ thuộc hàm định nghĩa trên $R, X \subseteq R^+$, ta gọi bao đóng của tập thuộc tính X dựa trên F , kí hiệu X_F^+ , là tập:

$$X_F^+ = \{A \in R^+ \mid f: X \rightarrow A, F \models f\} (\subseteq R^+)$$

- **Hệ quả:** $X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X_F^+$

Chứng minh:

(\Rightarrow) theo định nghĩa X_F^+ .

(\Leftarrow) Giả sử $Y \subseteq X_F^+$, chú ý rằng $X \rightarrow X_F^+$ (luật dẫn hội)

Vì $Y \subseteq X_F^+$ nên theo luật dẫn phân rã ta có $X \rightarrow Y$ tức là $f: X \rightarrow Y$ được suy diễn từ F .

Ghi chú: Từ bây giờ trở đi, để đơn giản chúng ta dùng ký hiệu X^+ thay cho X_F^+ khi tính toán chỉ liên quan đến một lược đồ quan hệ R và một tập phụ thuộc hàm F .

Thuật toán tính X^+

Dữ liệu vào: $X \subseteq R^+, F = \{f_1, f_2, \dots, f_n\}$

Dữ liệu ra: $X^+ \equiv KQ$

Thuật toán:

Bước 1: $KQ = X$;

Bước 2: **WHILE** (KQ còn thay đổi) **DO**

FOR (mỗi phụ thuộc hàm $Y \rightarrow Z \in F$) **DO**

IF $Y \subseteq KQ$ **THEN** $KQ := KQ \cup Z$

Lưu ý: Vì $X^+ \subseteq R^+$ nên số phần tử của X^+ là hữu hạn suy ra vòng lặp kết thúc sau hữu hạn bước.

Ví dụ 1:

Cho lược đồ quan hệ $R(A, B, C, D, E, G, H)$ và tập phụ thuộc hàm F

$F = \{$

- $f1: B \rightarrow A$
- $f2: DA \rightarrow CE$
- $f3: D \rightarrow H$
- $f4: GH \rightarrow C$
- $f5: AC \rightarrow D \}$

Tìm bao đóng của các tập $X = \{AC\}$ dựa trên F .

Giải:

Bước 1: $KQ = AC$

Bước 2:

- Do $f1, f2, f3, f4$ không thỏa. $f5$ thỏa vì $X^+ \supseteq AC$

$$KQ = AC \cup D = ACD$$
- Lập lại bước 2: $f1$ không thỏa, $f2$ thỏa vì $X1 \supseteq AD$:

$$\text{Nên } KQ = ACD \cup CE = ACDE$$

$$f3 \text{ thỏa vì } X2 \supseteq D \text{ nên } KQ = ACDE \cup H = ACDEH$$

$$f4 \text{ không thỏa, } f5 \text{ không xét vì đã thỏa}$$
- Lập lại bước 2: $f2, f3$ không xét vì đã thỏa; $f1, f4$ không thỏa; $f5$ không xét vì đã thỏa. Trong bước này KQ không thay đổi $\Rightarrow KQ = \{ACDEH\}$ là bao đóng của X .

Ví dụ 2: $R(A, B, C, D, E, G)$

$F = \{f1: A \rightarrow C;$

$$f_2: A \rightarrow EG;$$

$$f_3: B \rightarrow D;$$

$$f_4: G \rightarrow E \}$$

$$X = \{A, B\}; Y = \{C, G, D\}$$

Kết quả:

$$X^+ = \{ABCDEG\}$$

$$Y^+ = \{CGDE\}$$

Bài toán thành viên

Một vấn đề quan trọng khi nghiên cứu lý thuyết CSDL là khi cho trước tập các phụ thuộc hàm F và một phụ thuộc hàm $f: X \rightarrow Y$, làm thế nào để biết f có thuộc F^+ hay không? Để trả lời câu hỏi này ta có thể tính F^+ rồi kiểm tra xem f có thuộc F^+ hay không. Trong thực tế, tập F^+ là khá lớn, thường người ta không tính F^+ mà kiểm tra bằng cách tính X^+ , nếu thấy $Y \subseteq X^+$ thì $f \in F^+$.

Thuật toán xác định $f: X \rightarrow Y \in F^+$

Bước 1: Tính X^+

Bước 2: So sánh X^+ với Y nếu $X^+ \supseteq Y$ thì ta khẳng định $X \rightarrow Y \in F^+$

II.4 Phủ của tập phụ thuộc hàm

a) Định nghĩa phủ của tập phụ thuộc hàm F

- Ta nói F **phủ** G nếu $G \subseteq F^+$
- Hai tập phụ thuộc hàm F và G được gọi là tương đương (equivalent), kí hiệu $F \sim G$, nếu như F phủ G và G phủ F tức là $F^+ = G^+$.
- Kí hiệu $F \models G$ tức là: $\forall g \in G : F \models g$

Nhận xét:

- F phủ $G \Leftrightarrow F \models G$
- $F \sim G \Leftrightarrow F \models G$ và $G \models F$

Thuật toán xác định F và G có tương đương không

Bước 1: Với mỗi phụ thuộc hàm $f: X \rightarrow Y \in F$ ta kiểm tra xem $X \rightarrow Y \in G^+$

Bước 2: Với mỗi phụ thuộc hàm $g: W \rightarrow Z \in G$ ta kiểm tra xem $W \rightarrow Z \in F^+$

Bước 3: Nếu tất cả các phụ thuộc hàm được kiểm tra đều thỏa thì $F \sim G$.

Ví dụ 3: Cho lược đồ quan hệ $R(A,B,C,D,E)$ và hai tập phụ thuộc hàm:

$$F = \{A \rightarrow BC; A \rightarrow D; CD \rightarrow E\} \text{ và } G = \{A \rightarrow BCE; A \rightarrow ABD; CD \rightarrow E\}$$

- a) F có tương đương với G không?
b) F có tương đương với $G' = \{A \rightarrow BCDE\}$ không?

b) Định nghĩa phủ tối thiểu của một tập phụ thuộc hàm

❖ **Phụ thuộc hàm có vẻ trái dư thừa**

Cho F là tập các phụ thuộc hàm trên lược đồ quan hệ R , X là tập con thuộc tính khác rỗng, A là một thuộc tính, $X \rightarrow A \in F^+$. Phụ thuộc hàm $X \rightarrow A$ được gọi là phụ thuộc hàm có vẻ trái dư thừa (phụ thuộc hàm không đầy đủ) nếu có một thuộc tính $B \in X$ sao cho:

$$X \setminus B \rightarrow A \in F^+$$

Lưu ý :

- Nếu $X \rightarrow A$ là phụ thuộc hàm đầy đủ (phụ thuộc hàm nguyên tố) khi đó ta nói A phụ thuộc hàm đầy đủ vào X .
- Phụ thuộc hàm có vẻ trái một thuộc tính là phụ thuộc hàm đầy đủ

Ví dụ 4: Cho $R(A, B, C)$ và tập phụ thuộc hàm $F = \{AB \rightarrow C; B \rightarrow C\}$

Vì $AB \rightarrow C$ là phụ thuộc hàm không đầy đủ nên

$$F \sim (F \setminus \{AB \rightarrow C\}) \cup \{(AB \setminus A) \rightarrow C\} = \{B \rightarrow C\}.$$

Ví dụ 5: Cho $R(A,B,C,D)$ và tập phụ thuộc hàm $F = \{A \rightarrow BC; B \rightarrow C; AB \rightarrow D\}$. Ta có phụ thuộc hàm $AB \rightarrow D$ có vẻ trái dư thừa B . Từ đó:

$$\begin{aligned} F &\sim (F \setminus \{AB \rightarrow D\}) \cup \{A \rightarrow D\} \\ &\sim \{A \rightarrow BC; B \rightarrow C; A \rightarrow D\}. \end{aligned}$$

Ta nói F là tập phụ thuộc hàm có vẻ trái không dư thừa nếu F không chứa phụ thuộc hàm không đầy đủ.

Thuật toán loại khỏi F các phụ thuộc hàm có vẻ trái dư thừa.

Bước 1: Lần lượt thực hiện bước 2 cho các phụ thuộc hàm $X \rightarrow Y$ của F

Bước 2: Với mọi tập con thật sự $X' \subsetneq X$ của X

Nếu $X' \rightarrow Y \in F^+$ thì thay $X \rightarrow Y$ trong F bằng $X' \rightarrow Y$, thực hiện lại bước 2.

Ở ví dụ 5 phụ thuộc hàm $AB \rightarrow D$ có $A^+ = ABCD \Rightarrow A \rightarrow D \in F^+$. Trong F ta thay $AB \rightarrow D$ bằng $A \rightarrow D \Rightarrow F \sim \{A \rightarrow BC, B \rightarrow C, A \rightarrow D\}$.

❖ **Tập phụ thuộc hàm có vẻ phải một thuộc tính**

Mỗi tập phụ thuộc hàm F đều tương đương với một tập phụ thuộc hàm G mà vẻ phải của các phụ thuộc hàm trong G chỉ gồm một thuộc tính.

Ví dụ 6: Cho $F = \{A \rightarrow BC; B \rightarrow C; AB \rightarrow D\}$, dùng luật tách ta có:

$$F \sim \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow D\} \equiv G$$

G được gọi là **tập phụ thuộc hàm có vẻ phải một thuộc tính**.

❖ **Tập phụ thuộc hàm không dư thừa**

F là tập phụ thuộc hàm không dư thừa nếu không tồn tại $F' \subset F$ sao cho $F' \sim F$. Ngược lại F là tập phụ thuộc hàm dư thừa.

Ví dụ: cho $F = \{A \rightarrow BC, B \rightarrow D, AB \rightarrow D\}$ thì F dư thừa vì $F \sim F' = \{A \rightarrow BC, B \rightarrow D\}$

Thuật toán loại khỏi F các phụ thuộc hàm dư thừa:

Bước 1: Lần lượt xét các phụ thuộc hàm $X \rightarrow Y$ của F

Bước 2: Nếu $X \rightarrow Y \in (F \setminus \{X \rightarrow Y\})^+$ thì loại $X \rightarrow Y$ khỏi F .

❖ **Phủ tối thiểu (minimal cover)**

Cho LDQH R và tập phụ thuộc hàm F . Phủ tối thiểu của F , kí hiệu F_{tt} , là một tập phụ thuộc hàm thỏa đồng thời ba điều kiện sau:

1. F_{tt} chỉ gồm các phụ thuộc hàm có vẻ phải một thuộc tính và $F_{tt} \sim F$.
2. F_{tt} chỉ gồm các phụ thuộc hàm đầy đủ.
3. F_{tt} là một tập phụ thuộc hàm không thừa.

Thuật toán tìm phủ tối thiểu.

Thuật toán tìm phủ tối thiểu của một tập phụ thuộc hàm

Bước 1: Tách các phụ thuộc hàm có vẻ phải trên một thuộc tính của F thành các phụ thuộc hàm có vẻ phải một thuộc tính (dùng luật tách) \Rightarrow kết quả được tập F_1 .

Bước 2: Thay các phụ thuộc hàm không đầy đủ của F_1 bằng phụ thuộc hàm đầy đủ \Rightarrow kết quả được tập F_2 .

Bước 3: Loại khỏi F_2 các phụ thuộc hàm dư thừa \Rightarrow kết quả thu được sau bước này chính là F_{tt} .

Ví dụ 7: Cho lược đồ quan hệ $Q(A, B, C, D)$ và tập phụ thuộc F như sau:

$F = \{AB \rightarrow CD; B \rightarrow C; C \rightarrow D\}$. Hãy tính phủ tối thiểu của F .

Giải:

Bước 1: Tách các phụ thuộc hàm có vế phải trên một thuộc tính thành các phụ thuộc hàm có vế phải một thuộc tính

$$F_1 = \{AB \rightarrow C, AB \rightarrow D, B \rightarrow C, C \rightarrow D\}$$

Bước 2: Thay các phụ thuộc hàm không đầy đủ của F_1 bằng phụ thuộc hàm đầy đủ

$$F_2 = \{B \rightarrow D, B \rightarrow C, C \rightarrow D\}$$

Bước 3 Loại khỏi F_2 các phụ thuộc hàm dư thừa

Ta thấy $B \rightarrow D$ là phụ thuộc hàm thừa, vậy $F_{tt} = \{B \rightarrow C; C \rightarrow D\}$.

Ví dụ 8: Cho lược đồ quan hệ $Q(MSCD, MSSV, CD, HG)$ và tập phụ thuộc F như sau:

$$F = \{ \begin{array}{l} MSCD \rightarrow CD; \\ CD \rightarrow MSCD; \\ CD, MSSV \rightarrow HG; \\ MSCD, HG \rightarrow MSSV; \\ CD, HG \rightarrow MSSV; \\ MSCD, MSSV \rightarrow HG \end{array} \}$$

Hãy tìm phủ tối thiểu của F .

Kết quả: phủ tối thiểu của F là

$$F_{tt} = \{MSCD \rightarrow CD; CD \rightarrow MSCD; CD, HG \rightarrow MSSV; MSCD, MSSV \rightarrow HG\}$$

$$\text{hoặc } F_{tt} = \{MSCD \rightarrow CD; CD \rightarrow MSCD; CD, MSSV \rightarrow HG; MSCD, HG \rightarrow MSSV\}$$

Nhận xét:

- F có thể có nhiều phủ tối thiểu.
- Để kiểm tra các phụ thuộc hàm của F , ta chỉ cần kiểm tra các phụ thuộc hàm của F_{tt} .

II.5 Khóa của lược đồ quan hệ

a) Các định nghĩa

Siêu khóa: $R(A_1, A_2, \dots, A_n)$ là lược đồ quan hệ. R^+ là tập thuộc tính của R , F là tập phụ thuộc hàm định nghĩa trên R , $\emptyset \neq S \subseteq R^+$. **S là một siêu khóa của R nếu $S^+ = R^+$.**

Khóa: $R(A_1, A_2, \dots, A_n)$ là lược đồ quan hệ. R^+ là tập thuộc tính của R , F là tập phụ thuộc hàm trên R , $\emptyset \neq K \subseteq R^+$. **K là một khóa của R nếu thỏa hai điều kiện sau:**

- (i). $K^+ = R^+$ và

(ii). Không tồn tại $K' \subset K$ sao cho $(K')^+ = R^+$

Thuộc tính khóa: Thuộc tính A được gọi là **thuộc tính khóa** nếu $A \in K$ với K là một khóa của R. Ngược lại A được gọi là **thuộc tính không khóa**.

Lưu ý: Một lược đồ quan hệ có thể có nhiều khóa và tập thuộc tính không khóa cũng có thể bằng rỗng.

b) Các thuật toán tìm khóa

❖ Thuật toán tìm một khóa của một lược đồ quan hệ R

Bước 1: Gán $K = R^+$

Bước 2: A là một thuộc tính của K, đặt $K' = K - A$.

Nếu $(K')^+ = R^+$ thì gán $K = K'$ thực hiện lại bước 2.

Ví dụ 9:

Cho $R(A,B,C,D,E,G,H,I)$ và

$F = \{AC \rightarrow B; BI \rightarrow ACD; ABC \rightarrow D; H \rightarrow I; ACE \rightarrow BCG; CG \rightarrow AE\}$

Tìm một khóa của R.

Giải: Lần lượt loại các thuộc tính trong K theo thứ tự sau: A, B, D, E, I, ta được một khóa của lược đồ quan hệ R là $\{C,G,H\}$

❖ Thuật toán tìm tất cả khóa

Thuật toán đơn giản

Bước 1: Xác định tất cả các tập con khác rỗng của R^+ . Kết quả tìm được giả sử là các tập thuộc tính $X_1, X_2, \dots, X_{2n-1}$

Bước 2: Tìm bao đóng của các X_i ($i=1, 2, \dots, 2n-1$)

Bước 3: Siêu khóa là các X_i có bao đóng đúng bằng R^+ . Giả sử ta đã có các siêu khóa là $S = \{S_1, S_2, \dots, S_m\}$

Bước 4: Xây dựng tập các khóa của R từ tập S bằng cách xét mọi $S_i, S_j \in S$ ($i \neq j$), nếu $S_i \subset S_j$ thì ta loại S_j ($i, j=1 \dots m$), kết quả còn lại của S chính là tập tất cả các khóa cần tìm.

Ví dụ 10: Tìm tất cả các khóa của lược đồ quan hệ và tập phụ thuộc hàm như sau:

$R(A, B, C); F = \{AB \rightarrow C; C \rightarrow A\}$

$\emptyset \neq X_i \subseteq R^+$	X_i^+	Siêu khóa	Khóa
A	A		
B	B		
C	CA		
AB	ABC	AB	AB
AC	AC		
BC	ABC	BC	BC
ABC	ABC	ABC	

Vậy lược đồ quan hệ R có hai khóa là: {A,B} và {B,C}.

Thuật toán trên thì dễ hiểu, dễ cài đặt, tuy nhiên nếu tập thuộc tính R^+ khá lớn thì phép duyệt để tìm ra tập tất cả các tập con của tập R^+ là không hiệu quả. Do vậy cần thu hẹp không gian duyệt. Sau đây là thuật toán tìm khóa cải tiến theo hướng giảm số tập con thuộc tính cần duyệt.

Thuật toán cải tiến

Phân hoạch tập thuộc tính R^+ thành 3 tập con:

- Tập thuộc tính nguồn (kí hiệu là **S**) là tập tất cả các thuộc tính của R^+ mà không nằm trong vế phải của bất kỳ phụ thuộc hàm nào của F.
- Tập thuộc tính đích (kí hiệu là **T**): chứa các thuộc tính không thuộc tập thuộc tính nguồn và chỉ xuất hiện ở vế phải của các phụ thuộc hàm.
- Tập thuộc tính trung gian (kí hiệu là **M**): chứa các thuộc tính xuất hiện ở cả vế trái lẫn vế phải của các phụ thuộc hàm.

3 tập S, T, M thỏa tính chất $S \cap T = \emptyset$, $S \cap M = \emptyset$, $T \cap M = \emptyset$, $R^+ = S \cup T \cup M$.

Hệ quả: Nếu K là một khóa của R thì $S \subseteq K$ và $T \cap K = \emptyset$

Thuật toán tìm tất cả khóa của một lược đồ quan hệ

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

Output: Tất cả các khóa của lược đồ quan hệ R

Bước 1: Tạo tập thuộc tính nguồn S, tập thuộc tính trung gian M

Bước 2: Nếu $M = \emptyset$ Thì lược đồ quan hệ chỉ có duy nhất một khóa $K = S$ và kết thúc.

Ngược lại chuyển qua Bước 3

Bước 3: Tìm tất cả các tập con X_i của tập trung gian M

Gán tập siêu khóa $SK = \emptyset$

Bước 4: Tìm các siêu khóa: với mỗi tập con X_i thực hiện kiểm tra

Nếu $(S \cup X_i)^+ = R^+$ Thì

$SK_i = S \cup X_i$ và $SK = SK \cup \{SK_i\}$

Bước 5: Tìm khóa bằng cách loại bỏ các siêu khóa không tối thiểu:

$\forall SK_i, SK_j$ thuộc tập các siêu khóa SK tìm thấy ở bước 4

Nếu $SK_i \subset SK_j$ Thì Loại SK_j ra khỏi Tập siêu khóa SK.

Bước 6: SK còn lại chính là tập khóa cần tìm.

III. Các dạng chuẩn của lược đồ quan hệ

III.1 Dạng chuẩn 1 (1NF)

- Một LDQH R có dạng chuẩn 1 nếu trong R:
 - Không có thuộc tính kết hợp (là thuộc tính có giá trị trên hai kiểu khác nhau trở lên).
 - Không có thuộc tính chứa nhiều giá trị cùng lúc (thuộc tính đa trị).
- Một thiết kế CSDL gọi là ở dạng chuẩn 1NF nếu tất cả các lược đồ quan hệ của nó đều có dạng chuẩn 1NF.

III.2 Dạng chuẩn 2 (2NF)

- Một lược đồ quan hệ R ở dạng chuẩn 2 nếu R đạt chuẩn 1 và mọi thuộc tính không khóa của R đều phụ thuộc đầy đủ vào khóa.
- Một thiết kế CSDL gọi là ở dạng chuẩn 2NF nếu tất cả các lược đồ quan hệ của nó đều đạt dạng chuẩn 2NF.

Thuật toán kiểm tra dạng chuẩn 2

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

Ra: Khẳng định R đạt chuẩn 2 hay không đạt chuẩn 2.

Bước 1: Tìm tất cả khóa của R .

Bước 2: Với mỗi khóa K , tìm bao đóng của tất cả tập con thật sự S của K .

Bước 3: Nếu có bao đóng S^+ chứa thuộc tính không khóa thì R không đạt chuẩn 2.
Ngược lại thì R đạt chuẩn 2.

Ví dụ 11: Cho lược đồ quan hệ $R(A, B, C, D)$ và tập phụ thuộc hàm

$F = \{AB \rightarrow C; B \rightarrow D; BC \rightarrow A\}$. R có đạt chuẩn 2 không?

Giải:

- Tập khóa của R là $\{AB, BC\} \Rightarrow$ thuộc tính không khóa là D .
- Ta thấy $B \rightarrow D \in F \Rightarrow AB \rightarrow D$ là phụ thuộc hàm không đầy đủ $\Rightarrow D$ là thuộc tính không khóa không phụ thuộc đầy đủ vào khóa $\Rightarrow R$ không đạt dạng chuẩn 2.

III.3 Dạng chuẩn 3 (3NF)

- Một lược đồ quan hệ R ở dạng chuẩn 3 nếu mọi phụ thuộc hàm $f: X \rightarrow A \in F^+$ ($X \subseteq R^+, A \in R^+$) thỏa một trong các điều kiện sau:
 - (i) f là phụ thuộc hàm tầm thường (nghĩa là $A \in X$).
 - (ii) X là siêu khóa của R
 - (iii) A là thuộc tính khóa của R
- Một thiết kế CSDL được gọi là ở dạng chuẩn 3NF nếu như mọi lược đồ quan hệ của nó đều đạt dạng chuẩn 3.

Ví dụ 12: Xét LDQH GIAOVIEN ($\#GV, TENG, KHOA, CMON, LUONG, TO$)

$F = \{ \#GV \rightarrow TENG, KHOA, CMON$
 $TENG \rightarrow LUONG$
 $CMON \rightarrow TO \}$

Khoá của lược đồ là: $\#GV$

Vì các phụ thuộc hàm $TENG \rightarrow LUONG$ và $CMON \rightarrow TO$ không thỏa cả 3 điều kiện (i), (ii), (iii) nên LDQH GIAOVIEN không đạt 3NF.

Thuật toán kiểm tra dạng chuẩn 3

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

Ra: Khẳng định R đạt chuẩn 3 hay không đạt chuẩn 3.

Bước 1: Tìm tất cả khóa của R

Bước 2: Từ F tạo tập phụ thuộc hàm tương đương F_1 có vẻ phải một thuộc tính.

Bước 3: Nếu mọi phụ thuộc hàm $X \rightarrow A \in F_1$ với $A \notin X$ đều có X là siêu khóa hoặc A là thuộc tính khóa thì R đạt chuẩn 3 ngược lại R không đạt chuẩn 3.

Ví dụ 13: Cho lược đồ quan hệ $R(A,B,C,D)$ và $F=\{AB \rightarrow C; D \rightarrow B; C \rightarrow ABD\}$.

Hỏi R có đạt chuẩn 3 không?

Giải: Tập các khóa của R là: $K1 = \{AB\}$; $K2 = \{AD\}$; $K3 = \{C\} \Rightarrow$ mọi phụ thuộc hàm thuộc F đều có vẻ phải là thuộc tính khóa. Vậy R đạt chuẩn 3

III.4 Dạng chuẩn BC (Boyce-Codd Normal Form – BCNF)

- Một lược đồ quan hệ R ở dạng BCNF nếu mọi phụ thuộc hàm $f: X \rightarrow A \in F^+$ ($X \subseteq R^+, A \in R^+$) thỏa một trong các điều kiện sau:
 - f là phụ thuộc hàm tầm thường ($A \in X^+$).
 - X là siêu khóa của R .
- Một thiết kế CSDL được gọi là ở dạng chuẩn BCNF nếu như mọi lược đồ quan hệ của nó đều đạt dạng chuẩn BCNF.

Thuật toán kiểm tra dạng chuẩn BC

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

Ra: Khẳng định R đạt chuẩn BC hay không đạt chuẩn BC.

Nếu mọi phụ thuộc hàm không tầm thường $X \rightarrow Y \in F$ đều có $X^+ = R^+$ thì R đạt chuẩn BC ngược lại R không đạt chuẩn BC.

Ví dụ 14:

Cho quan hệ $R(A,B,C,D,E,I)$ và $F=\{ACD \rightarrow EBI; CE \rightarrow AD\}$. Hỏi R có đạt chuẩn BC không?

Thuật toán Tìm dạng chuẩn (cao nhất) của một lược đồ quan hệ

Dữ liệu vào: Lược đồ quan hệ R , tập phụ thuộc hàm F

Dữ liệu ra: Khẳng định R đạt chuẩn (cao nhất) gì?

Bước 1: Tìm tất cả khóa của R .

Bước 2: Kiểm tra xem R có đạt dạng chuẩn BC.

Nếu đúng Thì R đạt chuẩn BC và kết thúc thuật toán

Ngược lại qua bước 3.

Bước 3: Kiểm tra xem R có đạt dạng chuẩn 3.

Nếu đúng Thì R đạt dạng chuẩn 3 và kết thúc thuật toán.

Ngược lại qua bước 4.

Bước 4: Kiểm tra xem R có đạt dạng chuẩn 2.

Nếu đúng Thì R đạt dạng chuẩn 2 và kết thúc thuật toán.

Ngược lại R đạt dạng chuẩn 1.

Dạng chuẩn của một lược đồ cơ sở dữ liệu là dạng chuẩn thấp nhất trong các dạng chuẩn mà các lược đồ quan hệ đạt được.

IV. Phân rã lược đồ quan hệ dựa vào phụ thuộc hàm

Với thiết kế CSDL có chất lượng chưa tốt (chưa đạt dạng chuẩn cao) ta có thể nâng cấp bằng phương pháp phân rã (phép tách) những quan hệ chưa đạt dạng chuẩn cao thành nhiều quan hệ nhỏ hơn có chất lượng tốt hơn.

IV.1 Định nghĩa phép phân rã

Cho lược đồ quan hệ $R(A_1, A_2, \dots, A_n)$, ta gọi phép phân rã (phép tách) R thành m lược đồ con R_1, R_2, \dots, R_m , kí hiệu $\rho(R_1, R_2, \dots, R_m)$, nếu như:

$$R_1^+ \cup R_2^+ \cup \dots \cup R_m^+ = R^+$$

Phép phân rã (tách) một quan hệ thành nhiều quan hệ nhỏ không thể thực hiện một cách tùy ý mà cần đảm bảo được hai tính chất: *tính nối không mất thông tin (bảo toàn thông tin)* và *tính giữ lại phụ thuộc (bảo toàn phụ thuộc)*.

IV.2 Phân rã nối không mất (Lossless-join decomposition)

Một phân rã $\rho(R_1, R_2, \dots, R_m)$ của lược đồ quan hệ R được gọi là phân rã nối không mất thông tin (phân rã bảo toàn thông tin) nếu $\forall r$, đặt $r_1 = r[R_1^+]$, $r_2 = r[R_2^+]$, \dots , $r_m = r[R_m^+]$ (xem kí hiệu ở mục III.3, chương 3), luôn có:

$$r = r_1 \bowtie r_2 \bowtie \dots \bowtie r_m$$

Kiểm tra tính nối không mất thông tin của một phân rã:

Trường hợp 1: (LDQH R được phân rã thành 2 lược đồ con)

Cho LDQH R và tập phụ thuộc hàm F trên R. Phép phân rã lược đồ quan hệ R thành 2 lược đồ con R_1 và R_2 là phân rã nối không mất thông tin khi và chỉ khi một trong các phụ thuộc hàm sau đây thuộc F^+ :

- $R_1^+ \cap R_2^+ \rightarrow R_1^+$
- $R_1^+ \cap R_2^+ \rightarrow R_2^+$

Ví dụ 15: Cho $R(S, A, I, P)$, $R_1=(SA)$, $R_2=(SIP)$ và $F=\{S \rightarrow A, SI \rightarrow P\}$. Hỏi việc tách R thành R_1 và R_2 có gây ra mất mát thông tin không?

Trường hợp 2: (LDQH R được phân rã thành nhiều hơn 2 lược đồ con)

Sử dụng Bảng Tableau và qui trình thay thế đuôi

Input: Lược đồ quan hệ $R(A_1, A_2, \dots, A_n)$, tập phụ thuộc hàm F, phép phân rã

$$\rho=(R_1, R_2, \dots, R_k).$$

Output: Kết luận phân rã ρ có phải là phân rã bảo toàn thông tin?

Bước 1. Thiết lập bảng với $k+1$ dòng, $n+1$ cột. Cột j ứng với thuộc tính A_j ($j=1 \dots n$), hàng i ứng với lược đồ quan hệ R_i ($i=1 \dots k$). Tại vị trí hàng i , cột j ta điền ký hiệu a_j nếu $A_j \in R_i^+$, nếu không ta đặt ký hiệu b_t vào vị trí đó (với t bắt đầu bằng 1) và sau đó tăng t lên một đơn vị tính từ trái sang phải và từ trên xuống.

Bước 2. Xét lần lượt các phụ thuộc hàm trong F, áp dụng cho bảng vừa mới thành lập ở trên. Giả sử xét $(X \rightarrow Y) \in F$, chúng ta tìm những hàng giống nhau ở tất cả các thuộc tính của X, nếu thấy những hàng như vậy ta sẽ làm đồng nhất các ký hiệu của từng thuộc tính $A \in Y$ (thuộc tính ở vế phải của phụ thuộc hàm đang xét) tại các hàng đó. Khi làm cho đồng nhất các ký hiệu này, nếu một trong các ký hiệu là a_j thì đồng nhất các ký hiệu kia trở thành a_j , nếu các ký hiệu là b_k và một ký hiệu là b_l thì có thể cho chúng trở thành b_t (với $t = \min(k, l)$).

Tiếp tục áp dụng bước này cho các phụ thuộc hàm của F cho đến khi **bảng không còn thay đổi được** hoặc trong **bảng có xuất hiện một dòng $\langle a_1, \dots, a_n \rangle$** thì chuyển sang bước 3.

Bước 3. Xét bảng kết quả, nếu thấy trong bảng này có một hàng chứa toàn a_i ($i=1 \dots n$) thì kết luận đó là phép kết nối bảo toàn thông tin, ngược lại là phép kết nối mất mát thông tin.

Ví dụ 16: Cho $R(ABCDE)$ và tập phụ thuộc

$$F = \{A \rightarrow C; B \rightarrow C; A \rightarrow D; DE \rightarrow C; CE \rightarrow A\}$$

Một phân rã của R là $R_1(AD)$, $R_2(AB)$, $R_3(BE)$, $R_4(CDE)$, $R_5(AE)$.

Phân rã trên có bảo toàn thông tin không?

IV.3 Phân rã giữ lại phụ thuộc (Dependency preservation)

Tập phụ thuộc hàm hình chiếu (tập phụ thuộc hàm thu hẹp):

Cho lược đồ quan hệ R và tập phụ thuộc hàm F định nghĩa trên R , xét một phân rã $\rho = (R_1, R_2, \dots, R_k)$ của R . Thu hẹp (restriction) của F đến R_i ($i = 1, \dots, k$) là tập F_i (còn gọi là tập phụ thuộc hàm hình chiếu) được xác định như sau:

$$F_i = \{ X \rightarrow Y \in F^+ \mid XY \subseteq R_i^+ \}$$

Lưu ý: Trong F_i thông thường ta chỉ quan tâm tìm các phụ thuộc hàm không tầm thường duy diễn được từ F .

Đặt: $F' = F_1 \cup F_2 \cup \dots \cup F_k$

Phân rã $\rho = (R_1, R_2, \dots, R_k)$ của R được gọi là phân rã giữ lại phụ thuộc nếu có $F' \sim F$.

Thuật toán kiểm tra bảo toàn phụ thuộc hàm

Vào: $\rho = (R_1, R_2, \dots, R_k)$ và tập phụ thuộc hàm F của R

Ra: Kết luận phép tách ρ bảo toàn hay không bảo toàn phụ thuộc hàm

Bước 1: Với mỗi phụ thuộc hàm $X \rightarrow Y \in F$ thực hiện từ bước 2 đến bước 3:

Bước 2: $X_F^+ := X$;

Trong khi X_F^+ còn thay đổi thực hiện:

Với $i=1$ đến k thực hiện:

$$X_F^+ := X_F^+ \cup [(X_F^+ \cap R_i^+)_F^+ \cap R_i^+];$$

Nếu $Y \subseteq X_F^+$, thì $X \rightarrow Y \in (F')^+$

Bước 3: Nếu tất cả phụ thuộc $X \rightarrow Y \in F$ đều thuộc $(F')^+$ thì ta kết luận phân rã ρ bảo toàn phụ thuộc hàm ngược lại ρ không bảo toàn phụ hàm.

Ví dụ 17: Cho lược đồ quan hệ $R(A, B, C)$ và $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$. Phép phân rã $\rho = (R_1, R_2)$ tách R thành hai lược đồ quan hệ $R_1(A, B)$ và $R_2(B, C)$. Hãy tính tập phụ thuộc hình chiếu của F trên R_1 . Phép phân rã có bảo toàn phụ thuộc hàm F không?

V.4 Phân rã thành dạng BCNF nối không mất thông tin.

Qua các phần trình bày ở trên, ta thấy để thiết kế một CSDL tốt cần phải đưa về một thiết kế CSDL thỏa:

- (I) $\left\{ \begin{array}{l} \text{(i). Là BCNF} \\ \text{(ii). Là Lossless- join} \\ \text{(iii). Là Dependency preservation} \end{array} \right.$

Thuật toán phân rã R, F thành dạng chuẩn BC bảo toàn thông tin

Dữ liệu vào: lược đồ quan hệ R chưa đạt dạng chuẩn BC và tập phụ thuộc hàm F.

Dữ liệu ra: một phân rã sao cho mỗi lược đồ quan hệ con đều đạt chuẩn BC nói không mất thông tin.

Tìm phụ thuộc hàm $X \rightarrow Y \in F$ có X không là siêu khóa, thực hiện các bước sau:

B1. Tách R thành R_1 và R_2 theo quy tắc sau:

$$R_1^+ = XY;$$

$$R_2^+ = R^+ \setminus Y;$$

B2. Nếu R_1 chưa là BCNF đối với tập phụ thuộc hàm chiếu F_1 của F trên R_1 thì thực hiện lại bước B1 cho R_1 .

B3. Nếu R_2 chưa là BCNF đối với tập phụ thuộc hàm chiếu F_2 của F trên R_2 thì thực hiện lại bước B1 cho R_2 .

Lưu ý: ở bước B2 và B3, thuật toán không cần phải tính toàn bộ F^+ mà chỉ cần tìm được một phụ thuộc hàm cần thiết để tiếp tục phân rã đệ qui.

Ví dụ 18: Cho $R(S, D, I, M)$ $F = \{SI \rightarrow D; SD \rightarrow M\}$ hãy phân rã R thành các lược đồ con đạt chuẩn BC bảo toàn thông tin.

Giải:

Khóa của R là $\{S, I\} \Rightarrow$ phụ thuộc hàm $SD \rightarrow M$ không thỏa điều kiện của dạng chuẩn BC vì SD không là siêu khóa \Rightarrow R không đạt BCNF.

Dùng phụ thuộc hàm $SD \rightarrow M$ để phân rã R thành R_1 và R_2 như sau:

$$R_1(S, D, M); F_1 = \{SD \rightarrow M\}$$

$$R_2(S, D, I); F_2 = \{SI \rightarrow D\}$$

Ta thấy R_1 và R_2 đều đạt dạng chuẩn BC, hơn nữa phân rã này giữ lại phụ thuộc hàm.

Không phải bất kỳ phân rã BCNF nào cũng là phân rã giữ lại phụ thuộc (mặc dù theo cách phân rã như trên thì nó luôn luôn là lossless-join). Trong trường hợp không thể đạt được đủ 3 tiêu chuẩn trong I (thông thường chỉ đạt được các tiêu chuẩn (i), (ii)) nên đưa thiết kế về dạng chuẩn yếu hơn là 3NF, tức là:

- (II) {
- (i). Là 3NF
 - (ii). Là Lossless- join
 - (iii). Là Dependency preservation

IV.5 Phân rã thành dạng chuẩn 3 vừa nội không mất vừa giữ lại phụ thuộc hàm

Thuật Toán phân rã R, F thành dạng chuẩn 3, nội không mất thông tin, bảo toàn phụ thuộc hàm

Dữ liệu vào: lược đồ quan hệ R chưa đạt dạng chuẩn 3 và tập phụ thuộc hàm F.

Dữ liệu ra: một phân rã sao cho mỗi lược đồ quan hệ con đều đạt chuẩn 3 vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm.

Bước 1: Tìm phủ tối thiểu F_{tt} của F.

$i:=1$.

Bước 2: Với mỗi phụ thuộc hàm $X \rightarrow A \in F_{tt}$ thực hiện:

Nếu XA chứa trong một R_j ($j = 1, \dots, i-1$) **thì** $F_j = F_j \cup \{ X \rightarrow A \}$

Ngược lại tạo $R_i(XA)$; $F_i = \{ X \rightarrow A \}$; $i:=i+1$.

Bước 3: **Nếu** có một lược đồ con chứa khóa K của R **thì** kết thúc thuật toán

Ngược lại tạo một lược đồ con từ một khóa K của R.

Ví dụ 19: Cho lược đồ $R(C, T, H, R, S, G)$ và tập phụ thuộc hàm

$F = \{ C \rightarrow T; \quad HR \rightarrow C; \quad TH \rightarrow R; \quad CS \rightarrow G; \quad HS \rightarrow R \}$.

Hãy phân rã R thành các lược đồ con đạt dạng chuẩn 3 vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm.

Giải

- $F = \{ C \rightarrow T, HR \rightarrow C, TH \rightarrow R, CS \rightarrow G, HS \rightarrow R \}$ đã tối thiểu.
- Áp dụng thuật toán phân rã dạng chuẩn 3 ta được các lược đồ:
 $R_1(CT), \quad R_2(HRC), \quad R_3(THR), \quad R_4(CSG), \quad R_5(HSR)$
- Khóa của R là $\{HS\}$ và được chứa trong R_5 nên thuật toán kết thúc.

Lưu ý

- Nên tránh phân rã nếu lược đồ đã ở dạng chuẩn mong muốn.
- Nên xem xét tổ hợp các lược đồ quan hệ con thành lược đồ lớn hơn nếu lược đồ lớn hơn vẫn đạt dạng chuẩn mong muốn.
- Một kết quả phân rã bảo toàn phụ thuộc hàm sẽ có giá trị hơn kết quả phân rã không bảo toàn phụ thuộc hàm.
- Không có thuật toán phân rã lược đồ R thành các lược đồ con ở dạng chuẩn BC vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm. Tuy nhiên, vẫn tồn tại lược đồ

R được phân rã thành các lược đồ con ở dạng chuẩn BC vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm.

- Với một lược đồ R có thể có nhiều kết quả phân rã khác nhau.
- Luôn luôn có thể phân rã một lược đồ R không đạt dạng chuẩn 3 thành các lược đồ con đạt dạng chuẩn 3, nối không mất và giữ lại phụ thuộc.

V. Phụ thuộc đa trị (multivalued dependency) và dạng chuẩn 4

Như trong các phần trên đã xem xét, ta thấy rằng giữa dữ liệu có mối quan hệ với nhau và đó là phụ thuộc hàm. Nhưng phụ thuộc hàm không phải là kiểu phụ thuộc dữ liệu duy nhất. Khái niệm phụ thuộc hàm trong trường hợp tổng quát không đủ biểu diễn hết các loại phụ thuộc tồn tại trong quan hệ. Trong thực tế còn có nhiều loại phụ thuộc dữ liệu nữa. Chẳng hạn, mỗi cặp vợ chồng (tên cha, tên mẹ) không phải xác định duy nhất tên một đứa con mà có thể là một hoặc một số con. Mối quan hệ đó trong CSDL gọi là phụ thuộc đa trị

Giả sử Cho $Q(XYZ)$ là một lược đồ quan hệ với $\emptyset \neq X \subseteq Q^+$, $\emptyset \neq Y \subseteq Q^+$, $Z = Q^+ - (X \cup Y)$. Nói rằng X xác định đa trị Y, kí hiệu $X \twoheadrightarrow Y$, hoặc có một phụ thuộc đa trị của Y vào X nếu mỗi giá trị x của X xác định duy nhất một tập giá trị $\{y_1, y_2, \dots\}$ của Y và tập giá trị này của Y không phụ thuộc vào các giá trị z của Z.

V.1. Định nghĩa

$X \twoheadrightarrow Y$ là một phụ thuộc đa trị được định nghĩa trên $Q(XYZ)$, với $X \neq \emptyset$, $Y \neq \emptyset$ và $Z \neq \emptyset$ nếu như với mọi quan hệ q hợp lệ ta có:

$\forall t_1, t_2 \in q$ sao cho: $t_1.X = t_2.X$ thì:

$\exists t_3, t_4 \in q$ sao cho: $t_3.X = t_4.X = t_1.X = t_2.X$

và $t_3.Y = t_1.Y$ và $t_3.Z = t_2.Z$

và $t_4.Y = t_2.Y$ và $t_4.Z = t_1.Z$

quan hệ q:

X	Y	Z	
a	b ₁	c ₁	Bộ t1
a	b ₂	c ₂	Bộ t2
a	b ₁	c ₂	Bộ t3
a	b ₂	c ₁	Bộ t4

Ví dụ 1:

Xét lược đồ quan hệ NHÂN_VIÊN(Mã_NV, HọTên, Con_NV, BậcLương).
Ta có phụ thuộc đa trị MãNV \twoheadrightarrow Con_NV.

Mã_NV	HọTên	ConNV	Bậc lương
NV001	Nguyễn Văn A	Nguyễn Văn A1	2.34
NV001	Nguyễn Văn A	Nguyễn Văn A2	2.34
NV002	Nguyễn Văn B	Nguyễn Văn B1	2.67

GIANG_DAY(ChuyênĐề, Sách, GV, SV) Phụ thuộc đa trị ChuyênĐề \twoheadrightarrow Sách được định nghĩa trên GIANG_DAY.

ChuyênĐề	Sách	GV	SV
CSDL	S1	Thủy	A
CSDL	S2	Hoa	B
CSDL	S1	Hoa	B
CSDL	S2	Thủy	A

Phụ thuộc đa trị hiển nhiên:

$X \twoheadrightarrow Y$ là phụ thuộc đa trị hiển nhiên trên Q nếu $X \cup Y = Q^+$

Xét quan hệ CD_SACH(ChuyênĐề, Sách), phụ thuộc đa trị ChuyênĐề \twoheadrightarrow Sách, và Sách \twoheadrightarrow ChuyênĐề là hiển nhiên trên CD_SACH

ChuyênĐề	Sách
CSDL	S1
CSDL	S2
TK CSDL	S2
TK CSDL	S3
TK CSDL	S4

Hệ luật dẫn cho phụ thuộc đa trị

Cho $Q(Q^+)$ và $X, Y, W, V \subseteq Q^+$

- **(MVD1) Luật bù:** Nếu $X \twoheadrightarrow Y$ thì $X \twoheadrightarrow Q^+ - (X \cup Y)$
- **(MVD2) Luật tăng trưởng:** Nếu $X \twoheadrightarrow Y$ và $V \subseteq W$ thì $WX \twoheadrightarrow VY$
- **(MVD3) Luật truyền:** Nếu $X \twoheadrightarrow Y$ và $Y \twoheadrightarrow Z$ thì $X \twoheadrightarrow (Z - Y)$
- **(FD_MVD1)** Nếu $X \rightarrow Y$ thì $X \twoheadrightarrow Y$
- **(FD_MVD2)** Nếu $X \twoheadrightarrow Y$ và $W \rightarrow Z$, với $Z \subseteq Y$, $W \cap Y = \emptyset$ thì $X \rightarrow Z$

Tập các luật dẫn trên, bổ sung 3 luật dẫn trong hệ tiên đề Armstrong là tập luật dẫn đầy đủ.

- **Luật bắc cầu giả:** Nếu $X \twoheadrightarrow Y$ và $WY \twoheadrightarrow Z$ thì $WX \twoheadrightarrow (Z - WY)$
- **Luật hội:** Nếu $X \twoheadrightarrow Y$ và $X \twoheadrightarrow Z$ thì $X \twoheadrightarrow (YZ)$
- **Luật phân rã**
 Nếu $X \twoheadrightarrow Y$ và $X \twoheadrightarrow Z$ thì :
 $X \twoheadrightarrow (Y \cap Z)$
 $X \twoheadrightarrow (Y - Z)$
 $X \twoheadrightarrow (Z - Y)$

Phụ thuộc đa trị hệ quả và suy dẫn phụ thuộc đa trị

Tương tự phụ thuộc hàm, ta cũng có khái niệm phụ thuộc đa trị hệ quả và phụ thuộc đa trị suy dẫn từ một tập phụ thuộc **D**

Ký hiệu:

$$D^+ = \{d \mid D \models d\} = \{d \mid D \models d \text{ nhờ tập luật dẫn hợp lệ và đầy đủ}\}$$

V.2 Tập phụ thuộc của X dựa trên D

- Bài toán thành viên đối với phụ thuộc đa trị

Cho $d: X \twoheadrightarrow Y$ là một phụ thuộc đa trị, kiểm tra $\mathbf{D} \models d$?

- Định nghĩa Tập phụ thuộc của X dựa trên \mathbf{D} , ký hiệu $PT(X)_D$

$$PT(X)_D = \{Y \subseteq Q^+ \mid D \models (X \twoheadrightarrow Y)\}$$

$PT(X)_D$ là cơ sở để giải quyết bài toán thành viên đối với phụ thuộc đa trị.

Lưu ý:

- Khác với phụ thuộc hàm, ta không thể phân rã tùy ý về phải của phụ thuộc đa trị, vì vậy, mỗi phần tử của $PT(X)_D$ là 1 tập thuộc tính
- Số lượng phần tử của $PT(X)_D$ có thể sẽ rất lớn. Thuật toán tìm $PT(X)_D$ phải chú ý đến việc cho ra tập $PT(X)_D$ tối thiểu.

Thuật toán tính $PT(X)_D$

Vào: $Q(Q^+)$, \mathbf{D} , X ; Ra: $PT(X)_D$

0. $G_X := \emptyset$

1. Dùng luật bù và luật tăng trưởng để xác định G_X :

$$G_X := \{A_i\}, A_i \in X$$

$$\forall d: X' \twoheadrightarrow Y' \in \mathbf{D} \text{ với } X' \subseteq X \text{ thì: } G_X := G_X \cup \{Y', Q^+ - (X' \cup Y')\}$$

2. Tối thiểu G_X bằng cách áp dụng luật phân rã:

$$\forall Y_1, Y_2 \in G_X \text{ thì:}$$

$$\text{Nếu } Y_1 \cap Y_2 \neq \emptyset \text{ thì:}$$

$$\text{thay } Y_1, Y_2 \text{ trong } G_X \text{ bằng } (Y_1 \cap Y_2), (Y_1 - Y_2), (Y_2 - Y_1) \\ (\text{chỉ thay vào các tập khác } \emptyset)$$

Cuối Nếu

Cuối \forall

(Tập G_X sau bước 2 gọi là $CTT(G_X)$)

$$PT(X)_D = CTT(G_X)$$

3. Phân nhỏ các tập hợp trong $PT(X)_D$ bằng cách sử dụng luật truyền:

$$\forall d: W \twoheadrightarrow Z \in \mathbf{D} \text{ thì:}$$

$$\text{Nếu } \exists \{Y_1, Y_2, \dots, Y_k \in PT(X)_D\} \text{ sao cho } W \subseteq Y = Y_1 \cup Y_2 \cup \dots \cup Y_k \text{ thì}$$

$$(\text{Ta có: } X \twoheadrightarrow Y \text{ (do định nghĩa } PT(X)_D))$$

$$Y \twoheadrightarrow Z (\text{luật tăng trưởng})$$

$$\text{Nếu } \neg \exists Y'_1 Y'_2 \dots Y'_m \in PT(X)_D \text{ sao cho } (Z - Y) = Y'_1 \cup Y'_2 \cup \dots \cup Y'_m$$

$$\text{Thì } PT(X)_D := CTT(PT(X)_D \cup \{Z - Y\})$$

Cuối Nếu

Cuối \forall

V.3 Phụ thuộc đa trị tiềm ẩn trong Q

Tính chất chiếu của phụ thuộc đa trị:

Cho $Q(Q^+)$ và $d: X \twoheadrightarrow Y$ định nghĩa trên Q . Nếu có Z sao cho $X \subseteq Z \subseteq Q^+$ thì

$X \twoheadrightarrow (Y \cap Z)$ là một phụ thuộc đa trị định nghĩa trên $Q(Z)$.

Không có tính chất ngược lại (tính phản chiếu):

Nếu $d: X \rightarrow \rightarrow Y$ định nghĩa trên $Q'(Q'^+)$ với $Q'^+ \subseteq Q^+$ thì không nhất thiết d phải được định nghĩa trên Q .

Định nghĩa phụ thuộc đa trị tiềm ẩn

Cho $d': X' \rightarrow \rightarrow Y'$ là một phụ thuộc đa trị tiềm ẩn trong Q nếu:

- (i). d' được định nghĩa trên $Q'(Q'^+)$ với $Q'^+ \subseteq Q^+$
- (ii). $\neg \exists d$ định nghĩa trên Q sao cho d' có được bằng cách chiếu d lên Q'^+ .

Ví dụ 2:

Cho quan hệ PHỤ_ĐẠO(ChuyênĐề, SV, TrợGiảng, SốGiờ),
với $D = \{SV, \text{TrợGiảng} \rightarrow \text{SốGiờ}; SV, \text{TrợGiảng} \rightarrow \rightarrow \text{ChuyênĐề}\}$
Thể hiện sau của PHỤ_ĐẠO thỏa D :

ChuyênĐề	SV	TrợGiảng	SốGiờ
LT_C	Hoa	Minh	0
LT_C	Hùng	Minh	12
LT_C	Lan	Minh	15
LT_C	Hoa	Nhân	0
LT_C	Hùng	Nhân	0
LT_C	Lan	Nhân	12
LT_Pas	Lan	Nhân	12
LT_Pas	Dung	Nhân	6

Nhận xét:

Trong Q không có phụ thuộc đa trị $\text{ChuyênĐề} \rightarrow \rightarrow SV$ và $\text{ChuyênĐề} \rightarrow \rightarrow \text{TrợGiảng}$ Tuy nhiên, hai phụ thuộc đa trị trên tồn tại trong quan hệ PHỤ_ĐẠO_2 (ChuyênĐề, SV, TrợGiảng).

V.4 Dạng chuẩn 4 (4NF)

Định nghĩa: Q ở dạng chuẩn 4 nếu và chỉ nếu Q ở dạng chuẩn BC và với mỗi phụ thuộc đa trị $(X \rightarrow \rightarrow Y)$ không hiển nhiên được định nghĩa trên Q thì $\forall B \in Q^+$ ta luôn có $X \rightarrow B$ là một phụ thuộc hàm thuộc D^+ . Hay nói cách khác, X chứa một khóa của Q .

Chương 7

TỐI ƯU HÓA CÂU TRUY VẤN

I. Mở đầu.

- Các ngôn ngữ bậc cao nói chung khi xử lý trong máy đều rất tốn nhiều thời gian.
- Tối ưu câu hỏi: là quá trình biến đổi câu hỏi sang một câu hỏi tương đương (nghĩa là cho cùng một kết quả) để giảm thiểu thời gian tính toán.
- Việc tối ưu không nhất thiết phải trên mọi khả năng có thể của các cách cài đặt của các câu hỏi. Chúng ta có thể phải cân nhắc giữa giải thuật phức tạp hoặc/và tốn kém không gian lưu trữ với việc tiết kiệm thời gian xử lý.
- Giả sử để xử lý 1000 bản ghi mất 1 giây \rightarrow để xử lý 4.850.000 bản ghi chúng ta mất 4.850 giây: đây quả là một thời gian khó chấp nhận cho việc chờ đợi kết quả của một câu truy vấn!
- Khi tối ưu hóa xử lý thông tin người ta ưu tiên tối ưu hóa về thời gian hơn là tối ưu hóa lưu trữ dữ liệu, thậm chí “hy sinh” luôn cả dạng chuẩn để đạt được tốc độ xử lý nhanh hơn.
- Dung lượng đĩa cứng ngày nay đã đạt được một mức độ khá lớn, tuy nhiên cũng có khả năng thiếu không gian tính toán cho máy nếu chúng ta không lưu tâm đến việc tối ưu hóa truy vấn, đặc biệt là câu truy vấn có sử dụng phép tích đề-các và kết.
- Xét $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_n)$:
 - Giả sử R và S đều có 10 thuộc tính, mỗi thuộc tính có kích thước khoảng 5 byte (điều này ít có ở thực tế) và 1000 bản ghi.
 - $R \times S$: sẽ cho ra quan hệ trung gian này 20 thuộc tính, 1.000.000 bản ghi và chiếm hết 1/10 Giga bytes.
 - Thêm vào đó, thời gian truy cập CSDL và truy xuất đĩa để tạo ra số lượng bản ghi đó không phải là ít.
- Sự “bùng nổ” về số lượng và “bành trướng” về kích thước (chiều dài) bản ghi \rightarrow cần phải tối ưu câu hỏi trước khi đưa vào máy tính toán.

II. Các chiến lược tối ưu hóa.

II.1 Các nguyên tắc để tối ưu hóa câu hỏi.

- Cho quan hệ $R(A, B)$ có u bộ và quan hệ $S(C, D)$ có v bộ.
- Câu hỏi: “Lấy giá trị của thuộc tính A sao cho $B=C$ và $D=50$ ”
- Trả lời bằng đại số quan hệ:

$$\pi_A(\sigma_{(B=C) \wedge (D=50)}(R \times S))$$

$$\pi_A(\sigma_{B=C}(R \times \sigma_{D=50}(S)))$$

$$\pi_A(R \bowtie_{B=C} (\sigma_{D=50}(S)))$$

Việc biến đổi câu hỏi thành câu hỏi tương đương như ví dụ nêu trên là một minh họa cho việc giảm bớt thời gian trả lời câu hỏi bằng cách giảm bớt số lần cần truy nhập tới bộ nhớ thứ cấp dựa trên nguyên tắc thực hiện phép chọn càng sớm càng tốt. Trình tự thực hiện các phép tính sẽ đóng một vai trò quan trọng quá trình tổ chức câu hỏi.

II.2 Sáu chiến lược tổng quát cho việc tối ưu câu hỏi(J. D. Ullman):

1. **Thực hiện phép chọn càng sớm càng tốt:** giảm bớt kích cỡ của kết quả trung gian và do vậy chi phí phải trả cho việc truy nhập bộ nhớ thứ cấp cũng như lưu trữ của bộ nhớ chính sẽ nhỏ đi.
2. **Tổ hợp những phép chọn xác định với phép tích Đề-các thành phép kết.**
3. **Tổ hợp dãy các phép toán quan hệ một ngôi như phép chọn, phép chiếu.**
4. **Tìm các biểu thức con chung trong một biểu thức.** Nếu kết quả của một biểu thức con chung (tức là biểu thức xuất hiện nhiều hơn một lần) là một quan hệ không lớn và nó có thể được đọc từ bộ nhớ thứ cấp với ít thời gian thì nên tính toán trước biểu thức đó chỉ một lần. Nếu biểu thức con chung có liên quan tới một phép kết nối thì trong trường hợp tổng quát không thể thay đổi được nó bằng cách "đẩy" phép chọn vào trong.
5. **Tiền xử lý các quan hệ/bảng:**
 - Mục tiêu: xử lý nhanh các phép toán có liên quan đến hai quan hệ (phép toán hai ngôi).
 - Giải pháp: sắp xếp trước các bộ giá trị theo thứ tự vật lý và sắp xếp logic - tức là thiết lập các bảng chỉ mục (*Index*) cho các bản ghi.
6. **Đánh giá trước khi thực hiện tính toán:**
 - Mục tiêu: xác định được tổng chi phí phải trả cho các cách khác nhau khi thực hiện câu hỏi.

III. Các quy tắc liên quan đến các phép toán.

Biểu thức tương đương:

Hai biểu thức E1 và E2 được gọi là *tương đương (Equivalent)*, viết tắt là $E1 = E2$, nếu chúng biểu diễn cùng một ánh xạ, nghĩa là, nếu chúng ta thay thế cùng các quan hệ cho tên các lược đồ tương ứng ở hai biểu thức E1 và E2, thì chúng sẽ cho ra cùng một kết quả.

➤ Quy tắc liên quan tới phép kết và tích Đề-các:

1. **Quy tắc giao hoán của phép kết nối và tích Đề-các**
 - $E1 \bowtie E2 = E2 \bowtie E1$ (Tính giao hoán của kết).
 - $E1 \times E2 = E1 \times E2$ (Tính giao hoán của tích Đề-các).
2. **Quy tắc kết hợp của phép kết nối và tích Đề-các.**

- $(E1 \bowtie E2) \bowtie E3 = E1 \bowtie (E2 \bowtie E3)$
- $(E1 \bowtie E2) \times E3 = E1 \bowtie (E2 \times E3)$

➤ **Qui tắc liên quan tới chọn và phép chiếu:**

3. *Dãy các phép chiếu*

$$\pi_{A1, A2, \dots, An}(\pi_{A1, A2, \dots, Am}(R)) = \pi_{A1, A2, \dots, An}(R), \text{ với } n \leq m$$

4. *Dãy các phép chọn*

$$\sigma_{p1}(\sigma_{p2}(R)) = \sigma_{p2}(\sigma_{p1}(R)) = \sigma_{p1 \wedge p2}(R)$$

5. *Giao hoán phép chọn và phép chiếu*

$$\pi_{A1, A2, \dots, Ak}(\sigma_P(R)) = \sigma_P(\pi_{A1, A2, \dots, Ak}(R))$$

6. *Giao hoán phép chọn và phép tích Đề-các*

$$\sigma_C(R) \times S = \sigma_C(R \times S)$$

7. *Giao hoán phép chọn và một phép hợp*

$$\sigma_C(R \cup S) = \sigma_C(R) \cup \sigma_C(S)$$

8. *Giao hoán phép chọn và một phép trừ*

$$\sigma_C(R - S) = \sigma_C(R) - \sigma_C(S)$$

9. *Giao hoán phép chiếu và phép tích Đề-các*

$$\pi_{A1, A2, \dots, An}(R \times S) = \pi_{A1, A2, \dots, AK}(R) \times \pi_{AK+1, \dots, An}(S)$$

10. *Giao hoán phép chiếu với phép hợp*

$$\pi_{A1, A2, \dots, An}(R \cup S) = \pi_{A1, A2, \dots, An}(R) \cup \pi_{A1, A2, \dots, An}(S)$$

➤ **Các biểu thức tương đương khác:**

(B1) Kết tự nhiên tương đương với dãy phép tích đề-các, phép chọn và phép chiếu:

$$R(A,B) \bowtie S(B,C) = \pi_{A,B,C}(\sigma_{R.B=S.B}(R \times S))$$

(B2) Phép the-ta kết tương đương với dãy phép tích đề-các và phép chọn với điều kiện the-ta

$$R \bowtie_C S = \sigma_C(R \times S)$$

(B3) Phép giao tương đương với phần bù của hội hai phần hai phần bù của hai quan hệ

$$R \cap S = R \cup S - ((R - S) \cup (S - R))$$

(B4) Phép bù của một quan hệ tương đương với tích đề-các của các phép chiếu trên từng thuộc tính của quan hệ trừ đi các bộ giá trị đã có trong thể hiện của quan hệ.

$$\neg Q(A_1, A_2, \dots, A_n) = (Q(A_1) \times Q(A_2) \times \dots \times Q(A_n) - Q(A_1, A_2, \dots, A_n))$$

(B5) Thương của hai quan hệ tương đương với hiệu của các quan hệ trung gian sau:

$$Q1(A,B) \div Q2(A) = \pi_B(Q1) - \pi_B(((\pi_B(Q1) \times \pi_A(Q2)) - Q1))$$

IV. Kỹ thuật tối ưu hóa

Thuật toán tổng quát

- **Input:** Sơ đồ cú pháp câu hỏi bằng ngôn ngữ đại số quan hệ.
- **Output:** Sơ đồ cú pháp tối ưu.

Bắt đầu thuật toán

Bước 1: Áp dụng phép biến đổi tương đương theo 5 luật được nêu ở trên.

Bước 2: Áp dụng luật 4 biến đổi dãy các phép chọn tương đương: tách phép chọn thành các phép chọn con.

Bước 3: Đối với mỗi phép chọn, áp dụng 5,6,7, và 8 nhằm đẩy các phép chọn xuống càng sâu càng tốt.

Bước 4: Đối với mỗi phép chiếu áp dụng luật 3, 9, 10 nhằm đẩy các phép chiếu xuống càng sâu càng tốt.

Bước 5:

- Tập trung các phép chọn nhằm áp dụng luật 4
- Áp dụng luật 3 để loại bỏ các phép chiếu vô ích.
- Tập trung các phép chọn với tích Đề -các, nếu được thì chuyển thành phép kết tự nhiên hay the-ta kết bằng cách áp dụng luật 3 và 6.

Kết thúc thuật toán.

Nhận xét:

1. Thuật giải trên nhằm chủ yếu làm giảm khối lượng dữ liệu trung gian chứ không chỉ ra thứ tự thực hiện các phép kết.
2. Thuật giải này không cho ta kết quả tối ưu mà nó chỉ đưa ra một giải pháp tốt.
3. Các phép biến đổi chỉ dựa trên các phép toán cơ bản là hội, trừ, giao, tích Đề-các, chia, chiếu, và chọn, mà chúng ta có thể thực hiện các phép biến đổi dựa trên các phép toán khác nữa.

Ví dụ: Cho lược đồ cơ sở dữ liệu sau:

1. **SÁCH** (Tên-sách, Tác-giả, Nhà_XB, Mã-sách)
2. **NHÀ-XUẤT-BÁN** (Nhà_XB, Địa-chỉ, Thành-phố)
3. **ĐỘC-GIẢ** (Tên-ĐG, Đchỉ-ĐG, Tphố-ĐG, Số-thẻ)
4. **MƯỢN-SÁCH** (Số-thẻ, Mã-sách, Ngày-mượn)

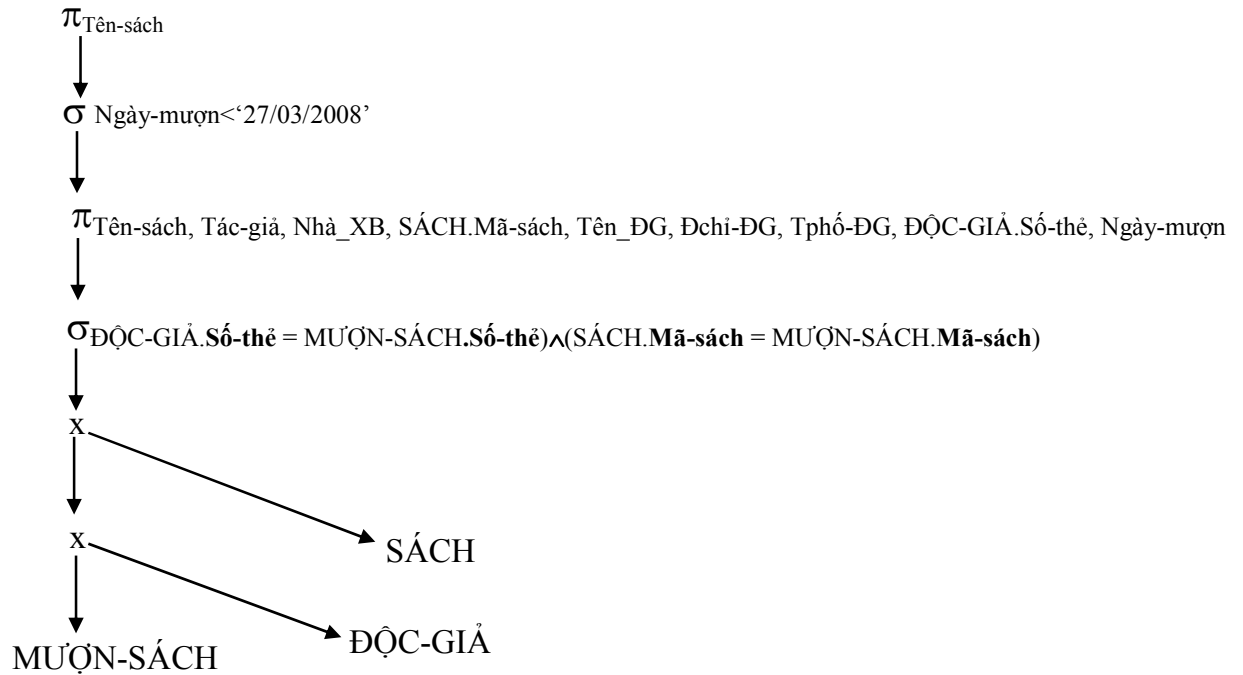
và một khung nhìn:

$$TD_MU\O\N \leftarrow S\A\CH \bowtie MU\O\N_S\A\CH \bowtie Đ\O\C_GI\A$$

Khung nhìn trên tương đương với:

$$TD_MU\O\N \leftarrow \pi_A(\sigma_C(S\A\CH \times Đ\O\C_GI\A \times MU\O\N_S\A\CH))$$

Sơ đồ cú pháp của câu hỏi trên như sau:



Hãy tối ưu câu truy vấn trên.

Bài Tập Chương 2 & 3 THIẾT KẾ CƠ SỞ DỮ LIỆU

Yêu cầu chung:

1. *Phân tích và thiết kế quan niệm: vẽ lược đồ ER hoặc EER.*
 2. *Thiết kế logic: Chuyển ERD/EERD sang lược đồ cơ sở dữ liệu theo mô hình quan hệ, có phát biểu tân từ; chỉ rõ khóa chính và khóa ngoại (nếu có) của các lược đồ quan hệ (có thể biểu diễn theo dạng Relationship (MS Access) hoặc Diagram); Xác định thứ tự cập nhật dữ liệu của các bảng. Phát biểu các ràng buộc ngữ nghĩa và lập bảng tầm ảnh hưởng tổng hợp; Chọn kiểu dữ liệu.*
- 1) Một thư viện muốn xây dựng một cơ sở dữ liệu để quản lý sách mượn. Thư viện yêu cầu cần quản lý những thông tin sau đây:
 - Sách của thư viện được phân chia theo thể loại để bạn đọc dễ dàng tra cứu. Mỗi thể loại sách có mã số thể loại (MATL) và dùng để phân biệt giữa các thể loại, tên gọi của thể loại (TENTL).
 - Đối với mỗi quyển sách, thư viện cần lưu mã sách (MASH) dùng để phân biệt các quyển sách, tên sách (TENSH) là tên (tựa đề) của sách, họ tên tác giả (TACGIA), thể loại (MATL), nhà xuất bản (MANXB), năm xuất bản (NAMXB).
 - Đối với mỗi nhà xuất bản, thư viện cần biết mã nhà xuất (MANXB), tên nhà xuất bản, địa chỉ (DCHINXB) và số điện thoại.
 - Đối với mỗi người mượn (thư viện gọi là bạn đọc) thư viện cần biết mã bạn đọc, họ tên bạn đọc, địa chỉ và số điện thoại.
 - Mỗi thể loại bao gồm nhiều quyển sách khác nhau, nhưng mỗi quyển sách chỉ thuộc về một thể loại. Mỗi nhà xuất bản sản xuất nhiều quyển sách khác nhau, nhưng mỗi quyển sách chỉ thuộc về một nhà xuất bản. Mỗi bạn đọc có thể mượn nhiều quyển sách khác nhau, mỗi quyển sách có thể được mượn bởi nhiều bạn đọc ở những thời điểm khác nhau. Khi bạn đọc mượn sách, thư viện cần biết ngày mượn, giờ mượn và ngày trả, giờ trả.
 - 2) Hoạt động cho thuê băng, đĩa (CD, DVD) ở một cửa hàng được mô tả như sau:
 - Băng/đĩa ở cửa hàng được phân thành nhiều thể loại, mỗi thể loại có một mã số để phân biệt và có tên gọi của thể loại đó.
 - Mỗi băng/đĩa có duy nhất một mã số và thuộc về một thể loại. Mỗi cuốn băng hoặc đĩa có tên gọi, số lượng bản mà cửa hàng có và đơn giá cho thuê.
 - Để tiện theo dõi, cửa hàng lưu lại thông tin của khách hàng thuê. Các khách hàng được phân biệt nhau bởi mã khách hàng. Mỗi khách hàng có có tên gọi và địa chỉ liên lạc.
 - Các lượt thuê băng đĩa của khách hàng được phân biệt bằng mã số lượt thuê (LUOT). Khách hàng phải đặt một số tiền cọc, cửa hàng sẽ ghi nhận ngày bắt đầu thuê.

- Một khách hàng mỗi lần thuê có thể mượn nhiều băng đĩa, khách hàng cũng có thể không trả hết các cuốn băng đã thuê cùng lúc.
- 3) Một trung tâm ngoại ngữ cần có một hệ thống quản lý thông tin, hệ thống này được mô tả như sau: Trung tâm có đào tạo nhiều loại ngoại ngữ khác nhau: Anh, Hoa, Hàn, Nhật, ... Khi thấy có nhu cầu, trung tâm sẽ mở các lớp học. Thông tin về lớp học bao gồm: mã lớp, tên lớp, học phí, ngày khai giảng dự kiến. Một lớp học chỉ dạy một loại ngoại ngữ nhất định. Khi học viên đến đăng ký học thì nhân viên tư vấn sẽ xếp các học viên này vào các lớp phù hợp, nhận tiền học phí từ học viên và ghi biên lai thu tiền cho họ. Để thuận lợi cho học viên, trung tâm cho phép học viên đóng học phí nhiều lần. Mỗi lần thu học phí hệ thống cần ghi nhận lại ngày thu, nhân viên thu và số tiền. Kết thúc mỗi lớp học, học viên được tổ chức thi, hệ thống lưu lại điểm cho các học viên. Ngoài ra hệ thống cần lưu lại thông tin về giáo viên dạy các lớp. Một lớp học chỉ có một giáo viên dạy. Thông tin của các giáo viên bao gồm: mã giáo viên, tên giáo viên, ngày sinh.
- 4) Một trung tâm tin học muốn xây dựng một hệ thống quản lý thông tin, hệ thống này được mô tả như sau: Trung tâm có nhiều chương trình đào tạo. Thông tin về chương trình đào tạo gồm mã chương trình, tên chương trình, thời lượng dạy, bằng cấp. Một chương trình đào tạo có nhiều môn học. Thông tin về môn học bao gồm: mã môn học, tên môn học, số tiết, học phí. Học viên có thể đăng ký học từng môn học hoặc học cả chương trình đào tạo. Hệ thống cần ghi nhận lại điểm của từng môn học mà học viên này đã học và có điểm. Điểm bao gồm hai phần đó là điểm lý thuyết và điểm thực hành. Khi học một môn học hay một chương trình đào tạo học viên có thể đóng học phí nhiều lần. Hệ thống cần ghi lại đầy đủ các lần đóng học phí, số tiền đóng và ngày đóng để tiện theo dõi và quản lý.
- 5) Một khách sạn muốn tin học hoá quản lý việc cho thuê mướn phòng. Sau đây là phân tích:
- Các phòng của khách sạn được phân biệt với nhau qua mã phòng (MAPH). Mỗi phòng có sức chứa tối đa số người cụ thể (SO_NGUOI), có đặc điểm (DACDIEM) mô tả của phòng, và giá thuê phòng (GIA_PHONG) trong một ngày.
 - Ngoài các vật dụng tối thiểu, khách sạn có thể trang bị một số tiện nghi khác cho các phòng như: điện thoại, tivi, tủ lạnh. Các loại tiện nghi được phân biệt bằng mã số loại tiện nghi (LOAI_TN). Mỗi loại tiện nghi có tên gọi (TEN_TN).
 - Với Mỗi loại tiện nghi, khách sạn có thể mua một số lượng lớn và dùng số thứ tự (STT) để phân biệt các vật dụng trong cùng loại tiện nghi. Một vật dụng có thể được sắp xếp trang bị cho nhiều phòng khác nhau nhưng trong một ngày một vật dụng chỉ trang bị cho một phòng.
 - Khi khách đến thuê phòng, khách sạn sẽ lưu lại họ tên của khách thuê phòng (HOTEN). Giả sử rằng họ tên các khách thuê cùng một phòng trong một ngày luôn luôn khác nhau. Ngày bắt đầu (NGAYBD) và ngày dự kiến kết thúc (NGAYKT) thuê phòng, và ngày trả phòng thực sự (NGAYTRA). Giả sử rằng không có trường

hợp khách trả phòng và thuê lại chính phòng đó trong cùng một ngày. Số tiền thuê phòng được chia đều cho số khách thuê trong cùng phòng.

- Khách thuê phòng có thể sử dụng thêm các dịch vụ (gọi điện thoại đường dài, thuê xe, giặt ủi, ...). Mỗi loại dịch vụ có một mã số để phân biệt. Ngày thực hiện dịch vụ (NGAYDV) cho khách và số tiền khách thuê phải trả cho dịch vụ (TIENDV) cũng được ghi nhận lại. Nếu trong cùng một ngày khách thuê phòng sử dụng một dịch vụ nhiều lần thì tiền dịch vụ được cộng dồn lại thành một lần. Các dịch vụ được tính riêng đối với từng khách. Nếu là dịch vụ chung cho một số khách thì sẽ tính tiền cho một vị khách đại diện nào đó.

6) Một đài truyền hình hàng năm có tổ chức cuộc thi “Tiếng hát truyền hình”. Sau đây là mô tả của việc tổ chức giải:

- Mỗi cuộc thi được tổ chức vào một năm (NAM), có một chủ đề (CHUDE), thời điểm bắt đầu (BATDAUGIAI) và thời điểm kết thúc (KTHUCGIAI).
- Một cuộc thi có nhiều vòng thi, mỗi vòng thi của một cuộc thi được xác định bằng số vòng với ý nghĩa: vòng = 1: vòng sơ khảo, vòng = 2: vòng chung khảo, vòng = 3: vòng chung kết xếp hạng. Mỗi vòng thi có thời điểm bắt đầu (BATDAUV), thời điểm kết thúc (KTHUCV), và số lượng thí sinh tối đa được tham dự vòng tiếp theo (SLUONG_TOIDA), với vòng chung kết xếp hạng không cần xác định SLUONG_TOIDA.
- Mỗi thí sinh khi đến đăng ký dự thi sẽ có một mã (MATS) duy nhất để nhận biết, có họ tên (HOTEN), Giới tính (GTINH), địa chỉ (DCHI) và ngày tháng năm sinh (NTNS). Thí sinh đã đạt giải nhất, giải nhì hoặc giải ba của các cuộc thi trước đó không được phép đăng ký dự thi.
- Mỗi thí sinh được tham dự tại vòng thi nào của cuộc thi sẽ có số điểm (DIEM) ở vòng thi đó. Cuối mỗi vòng thi, ban giám khảo sẽ căn cứ vào số lượng thí sinh tối đa được vào vòng trong để quyết định một thí sinh có được thi tiếp vòng sau hay không. Nếu là vòng thi chung kết xếp hạng thì không có vòng sau. Đối với những thí sinh vắng thi vòng thi nào thì điểm vòng thi đó sẽ để trống, và đồng thời thí sinh không được phép tham dự các vòng sau.
- Cuối giải, mỗi thí sinh đăng ký dự thi đều được ghi nhận lại kết quả đạt được. Ban thư ký sẽ ghi nhận lại vòng thi cao nhất (VongCaoNhat) mà mỗi thí sinh đạt được, điểm số (DIEM) và hạng (HANG) ứng với vòng đó. Đối với những thí sinh bỏ cuộc thì điểm và hạng sẽ bỏ trống.

7) Kết quả khảo sát hoạt động nghiệp vụ ở một công ty như sau:

- Công ty gồm nhiều phòng ban. Thông tin của phòng ban gồm có mã phòng ban, tên phòng ban, đặc tả. Mỗi nhân viên trong công ty được quản lý bởi một phòng ban duy nhất và mỗi phòng ban phải có ít nhất 4 nhân viên. Mỗi nhân viên gồm có mã nhân viên dùng để định danh, tên nhân viên, địa chỉ, ngày sinh, ngày vào công tác, lương cơ bản và chuyên môn. Trong đó, một nhân viên có thể có nhiều chuyên môn khác nhau. Khi lưu thông tin về chuyên môn của nhân viên cần lưu lại thông tin về trình độ (văn bằng) và ngày nhận được trình độ đó. Mỗi phòng ban có một nhân viên trong phòng ban đó làm trưởng phòng. Hệ thống chỉ

lưu lại nhân viên quản lý hiện tại của phòng ban và ngày nhân viên này được bổ nhiệm.

- Công ty có nhiều dự án. Mỗi dự án cần lưu các thông tin: mã số dự án, tên dự án, ngày ký dự án, địa chỉ dự án (bao gồm số nhà, tên đường và thành phố) và giá trị của dự án. Mỗi phòng ban thực hiện một số dự án khác nhau và một dự án chỉ do một phòng ban duy nhất phụ trách
 - Công ty đã có danh mục vật tư sử dụng trong các dự án. Mỗi vật tư có mã vật tư và tên vật tư. Mỗi dự án cần lưu thông tin về các vật tư và số lượng sẽ sử dụng trong dự án.
 - Công ty mua vật tư từ các nhà cung cấp. Mỗi nhà cung cấp cần lưu thông tin: mã số nhà cung cấp, tên nhà cung cấp, số điện thoại liên lạc, các vật tư mà nhà cung cấp đó có thể cung cấp và giá bán hiện tại của vật tư đó.
 - Mỗi vật tư trong một dự án có thể được mua từ nhiều nhà cung cấp khác nhau, cần lưu số lượng mua từ mỗi nhà cung cấp. Ví dụ dự án X cần mua 100kg vật tư A, công ty có thể mua 30 kg từ nhà cung cấp 1, 20 kg từ nhà cung cấp 2, và 50 kg từ nhà cung cấp 3.
 - Khi thanh toán cho nhà cung cấp, công ty sẽ lập một phiếu chi. Thông tin trên phiếu chi cần lưu: số phiếu chi, ngày chi, số tiền chi, chi cho nhà cung cấp nào. Mỗi phiếu chi chỉ thanh toán cho một dự án. Mỗi dự án có thể thanh toán nhiều lần.
 - Một nhân viên có thể không tham gia hoặc tham gia một hay nhiều dự án. Một dự án phải có ít nhất một nhân viên. Ngoài lương cơ bản nêu trên, khi tham gia vào dự án nhân viên được trả lương theo dự án. Hệ thống lưu lại thu nhập hiện tại của nhân viên (thu nhập hiện tại bằng tổng của lương cơ bản và lương tham gia các dự án tại thời điểm hiện tại của nhân viên đó).
- 8) Để tin học hóa nghiệp vụ quản lý kho vật tư và nguyên vật liệu phục vụ sản xuất hàng hoá một đơn vị sản xuất đã có phân tích nhu cầu về dữ liệu như sau:
- Vật tư được phân thành nhiều loại, mỗi loại vật tư có một mã số để phân biệt và một tên gọi.
 - Mỗi vật tư có một mã số để phân biệt, tên vật tư, đơn vị tính, số lượng tồn.
 - Đơn vị sản xuất mua vật tư từ các nhà cung cấp. Mỗi nhà cung cấp có một mã số duy nhất, tên nhà cung cấp, địa chỉ và số điện thoại.
 - Khi vật tư về nhập kho, kế toán kho lập phiếu nhập kho. Trên phiếu nhập kho ghi rõ số phiếu nhập kho, nhà cung cấp, ngày nhập kho. Phần chi tiết phiếu nhập kho liên quan đến một hoặc nhiều vật tư gồm các thông tin như mã số và tên vật tư, đơn vị tính, số lượng nhập.
 - Trong quá trình sản xuất, các tổ sản xuất có thể lập phiếu yêu cầu vật tư cho mặt hàng đang được sản xuất. Các thông tin trong phiếu yêu cầu bao gồm số phiếu yêu

cầu, ngày yêu cầu, tổ sản xuất, tên mặt hàng. Một phiếu yêu cầu có thể liên quan đến một hoặc nhiều vật tư gồm các thông tin như mã số vật tư, tên vật tư, đơn vị tính, số lượng xuất. Một phiếu yêu cầu chỉ yêu cầu xuất vật tư cho một mặt hàng.

- Cuối mỗi ngày, kế toán kho sẽ tổng hợp số lượng tồn của từng vật tư để có yêu cầu nhập vật tư từ nhà cung nếu cần thiết.

9) Một công ty kinh doanh bất động sản cần lưu thông tin:

- Công ty có nhiều văn phòng đặt tại nhiều thành phố. Mỗi văn phòng có một mã số duy nhất, một địa điểm và số điện thoại liên lạc.
- Mỗi văn phòng có nhiều nhân viên. Mỗi nhân viên có mã số nhân viên và tên nhân viên. Một nhân viên chỉ làm tại một văn phòng. Một văn phòng luôn có một trưởng văn phòng. Trưởng văn phòng phải là một trong số nhân viên làm việc tại văn phòng đó.
- Công ty cần lưu danh sách bất động sản. Thông tin về bất động sản gồm có: mã bất động sản dùng để phân biệt và vị trí. Vị trí bất động sản bao gồm: tên đường, thành phố. Mỗi bất động sản chỉ được lưu thông tin tại một văn phòng. Một văn phòng có thể chưa có bất động sản.
- Mỗi bất động sản có một hoặc nhiều chủ sở hữu. Thông tin về người chủ sở hữu gồm có mã chủ sở hữu, tên chủ sở hữu, và số điện thoại. Một người có thể là chủ của nhiều bất động sản. Trong trường hợp nhiều người đồng sở hữu một bất động sản cần lưu thêm phần trăm sở hữu của từng người trong bất động sản đó.

10) Dưới đây là mô tả của hệ thống quản lý việc sử dụng điện trong một công ty điện lực tại một thành phố.

- Công ty có nhiều chi nhánh. Mỗi chi nhánh có tên chi nhánh duy nhất và địa điểm duy nhất. Mỗi chi nhánh quản lý việc phân phối điện từ nhiều trạm điện. Một trạm điện chỉ thuộc một chi nhánh. Mỗi trạm điện được đặt tên dựa vào địa danh nơi đặt trạm. Tên trạm có thể trùng giữa các trạm trong các chi nhánh khác nhau, nhưng trong một chi nhánh không có trạm trùng tên.
- Mỗi khách hàng có thể thuê bao nhiều điện kế. Một điện kế chỉ thuộc một khách hàng. Thông tin về khách hàng cần có mã khách hàng (khóa), tên, địa chỉ, số điện thoại. Một điện kế sử dụng điện từ một trạm điện. Tất cả điện kế của một khách hàng chỉ sử dụng điện từ các trạm do một chi nhánh quản lý. Số của điện kế được đặt không trùng nhau trong toàn thành phố.
- Mỗi định kỳ (hàng tháng), nhân viên ghi điện sẽ ghi chỉ số của điện kế. Số kwh một điện kế sử dụng trong tháng mới nhất là hiệu giữa chỉ số mới và chỉ số tháng trước. Tất cả các chỉ số hàng tháng đều được lưu giữ. Mỗi chỉ số được ghi cần kèm thêm thông tin là tên nhân viên ghi điện.

11) Dưới đây là mô tả hệ thống được sử dụng để quản lý việc mượn sách trong thư viện.

- Các tài liệu cho đọc giả mượn có các thuộc tính là mã tài liệu (khóa), tên tài liệu (tựa đề). Tài liệu gồm 2 loại: sách và báo, tạp chí.
 - Mỗi tựa đề sách cần được biết do tác giả nào viết. Thông tin về tác giả gồm có mã tác giả, tên tác giả, năm sinh. Một tác giả viết nhiều sách, một sách có thể đồng tác giả.
 - Mỗi tựa đề sách có nhiều lần xuất bản (tái bản). Thông tin về một lần xuất bản gồm có: lần xuất bản, năm xuất bản, khổ giấy, số trang, nhà xuất bản, giá, có hoặc không kèm đĩa CD. Lần xuất bản được đánh số 1, 2, 3, ... cho mỗi tựa đề sách, do đó có sự trùng nhau giữa các tựa đề sách khác nhau.
 - Mỗi lần xuất bản một tựa đề sách, thư viện nhập vào nhiều cuốn sách. Mỗi cuốn sách này được quản lý riêng dựa vào số thứ tự được đánh số từ 1, 2, 3, ... trong số các cuốn sách cùng tựa đề và cùng lần xuất bản. Khi cho đọc giả mượn, thông tin ghi trong thẻ đọc giả phải xác định chính xác cuốn nào. Thông tin về mỗi cuốn sách còn có thêm tình trạng hiện tại của sách (tốt, rách, mất trang, ...).
 - Khác với việc cho mượn sách, việc cho mượn báo, tạp chí không cần chỉ chính xác tờ nào trong số các tờ cùng tựa đề và cùng một lần xuất bản. Tuy nhiên trong số này (cùng tựa đề và cùng một lần xuất bản), đọc giả mỗi lần chỉ có thể mượn 1 tờ.
 - Mỗi tựa đề báo, tạp chí cần các thông tin: năm bắt đầu phát hành, định kỳ (hàng ngày, hàng tuần hay hàng tháng), nhà xuất bản; đối với mỗi kỳ xuất bản cần biết số lượng tờ thư viện nhập về, số lượng tờ còn lại trong thư viện hiện tại (thuộc tính này được tính từ số tờ thư viện nhập về trừ đi số tờ đang có đọc giả mượn).
 - Thông tin về đọc giả gồm số thẻ đọc giả, ngày cấp thẻ, tên, nghề nghiệp, phái. Mỗi lần đọc giả có thể mượn nhiều sách cũng như báo, tạp chí, thông tin cần lưu là ngày mượn và ngày trả cho từng tài liệu mượn.
- 12) Sản phẩm của một công ty đồ gỗ (thuộc tính của sản phẩm gồm mã, đặc tả, và giá) có ít nhất là 3 thành phần (thuộc tính gồm mã thành phần, đặc tả, đơn vị tính). Mỗi thành phần được sử dụng trong một hoặc nhiều sản phẩm. Lưu ý rằng các thành phần cũng có thể được sử dụng để lắp ghép thành một thành phần khác. Các nguyên liệu thô cũng được xem như là thành phần. Chúng ta cần biết mỗi thành phần tham gia với số lượng bao nhiêu trong cấu thành một thành phần khác hoặc một sản phẩm.

Bài Tập Chương 3 & 4

TRUY VẤN DỮ LIỆU

- I. Cho 2 lược đồ quan hệ $Q_1(A, B)$, $Q_2(B, C)$. Giả sử rằng thể hiện $q_1(Q_1)$ và $q_2(Q_2)$ tương ứng là:

q_1		q_2	
A	B	B	C
ao	1	1	c1
a	2	2	c2
ao	3	3	c3
		4	c4

- a) Hãy kiểm tra thể hiện của 2 quan hệ sau:

$$\prod_C (\sigma_{A='ao'}(q_1 \bowtie q_2))$$

$$\prod_C (\sigma_{A='ao'}(q_1 \times q_2))$$

- b) Trong trường hợp nào để q_1, q_2 là hai quan hệ bất kỳ, ta có hai quan hệ tạo thành ở câu a) là như nhau.
c) Tìm thể hiện của:

$$\prod_C (\prod_B (\sigma_{A='ao'}(q_1)) \times q_2)$$

- II. Coi cơ sở dữ liệu quan hệ sau:

Nhân viên (TÊN NV, ĐƯỜNG PHỐ, THÀNH PHỐ)

Làm việc (TÊN NV, TÊN CTY, LƯƠNG)

Công ty (TÊN CTY, THÀNH PHỐ)

Quản lý (TÊN NV, TÊN_QL)

Viết mỗi câu truy vấn từ cơ sở dữ liệu trên theo dạng:

- i. Ngôn ngữ đại số quan hệ
- ii. Ngôn ngữ phép tính quan hệ

- (a) Tìm tên của tất cả các nhân viên làm việc cho công ty Bitis.
- (b) Tìm tên nhân viên và thành phố cư ngụ của tất cả các nhân viên làm việc cho công ty Bitis.
- (c) Tìm tên nhân viên, địa chỉ đường phố và thành phố cư ngụ của tất cả các nhân viên làm việc cho công ty Bitis và có tiền lương trên 1000000.
- (d) Tìm tên của tất cả các nhân viên cư ngụ tại cùng một thành phố với thành phố nơi mà các nhân viên đang làm việc.
- (e) Tìm tên của tất cả các nhân viên sống tại cùng một thành phố và cùng đường phố với người quản lý của họ.
- (f) Tìm tên của tất cả các nhân viên trong CSDL mà họ không làm việc cho công ty Bitis.

- (g) Tìm tên của tất cả các nhân viên mà họ kiếm được nhiều tiền hơn mọi nhân viên làm việc tại công ty Sony.
- (h) Giả sử rằng các công ty có thể được tọa lạc tại vài thành phố. Tìm tất cả những công ty tọa lạc tại mọi thành phố trong đó công ty Panasonic tọa lạc.

III. Xét lược đồ CSDL quan hệ “**Thực tập**” như sau:

SINHVIENT(#SV, HOTEN, NS, QUE, HOCLUC)

Tân từ: Quan hệ này chứa thông tin về các sinh viên trong một lớp của một trường đại học. Mỗi sinh viên có mã số (#SV) để phân biệt với sinh viên khác, họ tên sinh viên (HOTEN), năm sinh (NSINH), quê quán (QUE), học lực thể hiện qua điểm trung bình (HOCLUC).

DETAI(#DT, TENDT, CN, KP)

Tân từ: quan hệ này chứa thông tin về các đề tài nhà trường quản lý. #DT là mã số đề tài, TENDT: tên đề tài, CN: họ và tên chủ nhiệm đề tài, KP: kinh phí cấp cho đề tài(triệu đồng).

SV_DT(#SV, #DT, NTT, KM, KQ)

Tân từ : quan hệ sinh viên đề tài chứa thông tin về tình hình thực tập của các sinh viên theo các đề tài. #SV: mã số sinh viên, #DT: mã số đề tài mà sinh viên đó tham gia. NTT: nơi thực tập để triển khai đề tài (tỉnh), KM: khoảng cách từ nơi thực tập đến trường, KQ: kết quả thực tập theo đề tài đã chọn. Giả thiết là một sinh viên có thể tham gia nhiều đề tài, mỗi đề tài sinh viên đó thực tập tại một địa điểm.

- 1) Vẽ lược đồ E-R cho cơ sở dữ liệu trên.
- 2) Dùng ngôn ngữ đại số quan hệ biểu diễn các truy vấn sau:
 - a) Cho thông tin về những sinh viên sinh trước năm 1983, quê ở Hải Phòng?
 - b) Cho danh sách các tỉnh có sinh viên đến thực tập?
 - c) Cho biết các địa điểm thực tập xa trường (KM>100) của đề tài số 7?
 - d) Cho thông tin về việc thực tập tại NhaTrang của các sinh viên
 - e) Cho danh sách các sinh viên thực tập tại quê nhà?
 - f) Cho thông tin về các đề tài có sinh viên thực tập?
 - g) Cho biết mã của các đề tài không có sinh viên nào tham gia?
 - h) Cho biết mã của những đề tài có kinh phí 1.5 triệu và những đề tài có kinh phí trên 2 triệu.
 - i) Cho biết mã số và Họ tên của những sinh viên dưới 20 tuổi, thực tập khá (có điểm kết quả trên 7).
 - j) Cho biết mã của những đề tài có địa bàn thực tập ít ra là như đề tài 1.
 - k) Cho danh sách những đề tài được triển khai thực tập ở tất cả các tỉnh có sinh viên thực tập.
 - l) Cho danh sách những sinh viên thực tập theo đề tài có kinh phí lớn hơn một phần năm tổng kinh phí cấp cho các đề tài.

- m) Cho danh sách các sinh viên có điểm học tập cao hơn điểm thực tập trung bình của đề tài mã số 4.

Lưu ý: Với các bài tập (IV → VIII), Tất cả các câu truy vấn đều viết bằng SQL, các câu có đánh dấu (*) sinh viên viết thêm theo dạng đại số quan hệ.

IV. Cho một lược đồ cơ sở dữ liệu của một thư viện như sau:

- Thedocgia(mathe, hotendg, diachi)
 - Sach(masach, tensach, namxb, tennxb, tentacgia, soluong)
 - Phieumuon(maphieu, ngaymuon, masach, mathedocgia, ngaytra)
1. (*) Liệt kê các phiếu mượn trong ngày 20/11/2007
 2. (*) Liệt kê các tên sách thuộc nhà xuất bản “Giáo dục” hiện có trong thư viện.
 3. (*) Liệt kê các đọc giả hiện đang còn giữ sách có tên “Đồ Họa Máy Tính” chưa trả cho thư viện.
 4. (*) Hãy liệt kê tên các sách do tác giả “Nam Cao” viết được đọc giả Nguyễn Văn An mượn trong ngày 20/11/2007.
 5. (*) Cho biết có bao nhiêu sách do tác giả “Nam Cao” viết có trong thư viện.
 6. Cho biết tác giả nào có nhiều sách nhất.
 7. Cho biết tên sách nào có số lượng còn ít nhất trong thư viện.
 8. (*) Liệt kê các cuốn sách chưa có ai mượn lần nào.
 9. Liệt kê tên các đọc giả và tên sách mà các đọc giả này đang giữ đã đến kỳ hạn trả cho thư viện. Biết rằng quy định là sau khi mượn 10 ngày thì phải trả sách cho thư viện.
 10. (*) Liệt kê tên và địa chỉ liên lạc của các đọc giả hiện đang còn nợ sách của thư viện.
 11. Liệt kê tất cả các cuốn sách được nhiều nhất trong tháng 11/2007.
 12. Liệt kê các đọc giả quen thuộc của thư viện. Biết rằng các đọc giả quen thuộc là các đọc giả đến mượn sách ít nhất 10 lần.
 13. Liệt kê các tác giả nào được yêu thích nhất trong năm 2002. Biết rằng tác giả được yêu thích nhất là các tác giả có sách được mượn nhiều nhất.

V. Cho cơ sở dữ liệu gồm các quan hệ sau:

1. NhanVien(MANV, Ho, Ten, NgaySinh, NgayVaoLam, MSCN)

Tân từ: Mỗi nhân viên có một mã số (MANV) duy nhất, họ và tên lót (Ho), tên nhân viên (Ten), ngày sinh (NgaySinh), ngày vào làm (NgayVaoLam), mã số chi nhánh mà nhân viên đang làm việc (MSCN).

2. ChiNhanh(MSCN, TenCN)

Tân từ: Mỗi chi nhánh có một mã số duy nhất (MSCN) và tên chi nhánh (TenCN).

3. KyNang(**MSKN**, TenKN)

Tân từ: Mỗi kỹ năng tin học có mã số kỹ năng (MSKN) để phân biệt và tên kỹ năng (TenKN).

4. NhanVienKyNang(**MANV**, **MSKN**, MucDo)

Tân từ: Thông tin về sử dụng kỹ năng tin học của nhân viên bao gồm mã số nhân viên (MANV), mã số kỹ năng (MSKN) và mức độ (MucDo) thành thạo kỹ năng đó. Mức độ thành thạo được đánh số từ 1 đến 9, số càng lớn mức độ thành thạo càng cao). Một nhân viên có thể biết nhiều kỹ năng.

Vẽ sơ đồ ER và viết các câu truy vấn sau:

1. **Truy vấn lựa chọn trên nhiều bảng**

- Hiển thị MANV, HoTen (Ho + Ten as HoTen), số năm làm việc (SoNamLamViec).
- Liệt kê các thông tin về nhân viên: HoTen, NgaySinh, NgayVaoLam, TenCN (sắp xếp theo tên chi nhánh).
- Liệt kê các nhân viên (HoTen, TenKN, MucDo) của những nhân viên biết sử dụng 'Word'.
- (*) Liệt kê các kỹ năng (TenKN, MucDo) mà nhân viên 'Lê Anh Tuấn' biết sử dụng.

2. **Truy vấn lồng**

- Liệt kê MANV, HoTen, MSCN, TenCN của các nhân viên có mức độ thành thạo về 'Excel' cao nhất trong công ty.
- (*) Liệt kê MANV, HoTen, TenCN của các nhân viên vừa biết 'Word' vừa biết 'Excel' (dùng truy vấn lồng).
- Với từng kỹ năng, hãy liệt kê các thông tin (MANV, HoTen, TenCN, TenKN, MucDo) của những nhân viên thành thạo kỹ năng đó nhất.
- Liệt kê các chi nhánh (MSCN, TenCN) mà mọi nhân viên trong chi nhánh đó đều biết 'Word'.
- (*) Cho biết nhân viên biết sử dụng tất cả các kỹ năng.

3. **Truy vấn gom nhóm dữ liệu.**

- (*) Với mỗi chi nhánh, hãy cho biết các thông tin sau TenCN, SoNV (số nhân viên của chi nhánh đó).
- (*) Với mỗi kỹ năng, hãy cho biết TenKN, SoNguoiDung (số nhân viên biết sử dụng kỹ năng đó).
- Cho biết TenKN có từ 3 nhân viên trong công ty sử dụng trở lên.
- Cho biết TenCN có nhiều nhân viên nhất.
- Cho biết TenCN có ít nhân viên nhất.
- (*) Với mỗi nhân viên, hãy cho biết số kỹ năng tin học mà nhân viên đó sử dụng được.
- Cho biết HoTen, TenCN của nhân viên biết sử dụng nhiều kỹ năng nhất.

4. Cập nhật dữ liệu

- a) (*) Thêm bộ <'06', 'PhotoShop'> vào bảng KyNang
- b) (*) Thêm các bộ sau vào bảng NhanVienKyNang
<'0001', '06', 3>; <'0005', '06', 2>
- c) (*) Cập nhật cho các nhân viên có sử dụng kỹ năng 'Word' có mức độ tăng thêm một bậc.
- d) Tạo bảng mới **NhanVienChiNhanh1**(MANV, HoTen, SoKyNang) (dùng lệnh *Create table*)
- e) Thêm vào bảng trên các thông tin như đã liệt kê của các nhân viên thuộc chi nhánh 1 (dùng câu lệnh **Insert Into** cho nhiều bộ).

VI. Cho lược đồ CSDL của một cửa hàng bán linh kiện máy tính như sau:

- PhanLoai(maloai, tenloai)
- NhaCungCap(mancc, tenncc, diachi, dienthoai)
- KhachHang(makh, tenkh, diachikh, dienthoai)
- MatHang(mamh, ten, donvitinh, maloai)
- BangBaoGia(mancc, mamh, ngay, giaban)
- CungCap(mancc, mamh, ngay, soluong)
- GiaBan(mamh, ngay, gia)
- HoaDonBan(mahoadon, makh, mamh, ngaylap, soluong)

Vẽ sơ đồ ER và viết các câu truy vấn sau:

1. (*) Liệt kê tất cả các mặt hàng thuộc loại có tên là "Thiết bị I/O".
2. (*) Liệt kê tất cả các nhà cung cấp ở thành phố "Hồ Chí Minh".
3. Liệt kê những khách hàng có tên bắt đầu bằng chữ "H".
4. Liệt kê đầy đủ thông tin của các nhà cung cấp đã từng cung cấp mặt hàng có tên là "CD LG 52X".
5. (*) Liệt kê những nhà cung cấp không cung cấp mặt hàng có tên là "CD LG 52X".
6. (*) Liệt kê mã số và tên của các mặt hàng hiện không có nhà cung cấp nào cung cấp.
7. (*) Liệt kê những khách hàng nào (tên khách hàng) mua nhiều hơn một lần trong một ngày (có từ hai hóa đơn trở lên) trong khoảng thời gian từ 01/01/2007 đến 31/10/2007.
8. Liệt kê đầy đủ các thông tin của các nhà cung cấp bán mặt hàng có tên là "CPU PIII – 933EB" với giá rẻ nhất.
9. (*) Liệt kê tất cả các khách hàng đã có giao dịch với cửa hàng.
10. (*) Liệt kê tất cả các khách hàng thân thuộc trong năm 2007. Biết rằng khách hàng thân thuộc được đánh giá theo mỗi năm và là khách hàng có nhiều hơn 10 lần mua trong năm đó.
11. Liệt kê tất cả các khách hàng tiềm năng: là những khách hàng có số tiền mua từ đầu năm đến nay nhiều hơn 50 triệu đồng.

12. Liệt kê những nhà cung cấp nào cung cấp hàng cho cửa hàng nhiều nhất trong năm 2007. Biết rằng tiêu chuẩn đánh giá dựa trên tổng giá trị của hàng hóa.
 13. Liệt kê các mặt hàng có số lượng được bán nhiều nhất trong năm 2007.
 14. Tính tổng số tiền lời của năm 2007.
 15. Cho biết loại mặt hàng nào bán có lời nhất trong năm 2007.
 16. Liệt kê những mặt hàng (mã mặt hàng, tên và số lượng hiện có trong cửa hàng) đang cần được nhập thêm. Biết rằng mỗi khi số lượng của mỗi mặt hàng trong cửa hàng còn ít hơn 10 đơn vị thì cần phải tiến hành nhập thêm hàng.
 17. Liệt kê những ngày “nhập siêu” của năm 2007 trong đó có hiển thị số tiền chênh lệch giữa thu và chi. Biết rằng những ngày “nhập siêu” là những ngày có số tiền chi cho nhà cung cấp lớn hơn số tiền thu được do bán hàng cho khách.
- VII. Cho lược đồ CSDL như sau:
- Khoa(MaKhoa, TenKhoa)
 - MonHoc (MaMh, TenMh, SoTC)
 - CTDaoTao(MaKhoa, KhoaHoc, MaMh)
 - SinhVien(MaSV, HoTen, MaKhoa, KhoaHoc)
 - DangKy(MaSV, MaMh, DiemThi)

Giải thích:

- Khoa gồm các thông tin: mã khoa (MaKhoa), tên khoa (TenKhoa);
- MonHoc: môn học gồm có: mã môn học (MaMh), tên môn học (TenMh), số tín chỉ (SoTC);
- CTDaoTao: chương trình đào tạo gồm có: khoa (MaKhoa), khóa học (KhoaHoc), môn học (MaMh);
- SinhVien: mã sinh viên (MaSV), họ tên (HoTen), thuộc khóa (KhoaHoc) của một khoa (MaKhoa);
- DangKy: sinh viên (MaSV) học môn học (MaMh) có điểm thi (DiemThi)

Vẽ sơ đồ ER và viết các câu truy vấn sau:

1. (*) Hãy liệt kê các sinh viên thuộc khoa “công nghệ thông tin” khóa 2006
2. (*) Cho biết hiện tại khoa “Công nghệ thông tin” có bao nhiêu sinh viên.
3. Cho biết họ tên các sinh viên thuộc khoa có mã khoa “TOAN” có nhiều hơn 3 môn dưới điểm trung bình.
4. Cho biết sinh viên của khoa nào và khóa học nào phải học nhiều môn nhất trong toàn trường.
5. Cho biết những môn học nào hiện nay có nhiều sinh viên đang học nhất (là những môn học mà các sinh viên đã đăng ký nhưng chưa thi).
6. Cho biết những môn học nào thuộc khoa “Toán - Tin”, khóa 2006 hiện nay chưa có sinh viên nào đăng ký học.

7. Cho biết những môn học nào được dạy nhiều hơn một khoa.
8. Cho biết những môn học nào là những môn học chung cho toàn trường của khóa 2006. Biết rằng môn học chung cho toàn trường là môn học được dạy cho tất cả các khoa.
9. Liệt kê những sinh viên thuộc khoa “CNTT”, khóa 2006 có số tín chỉ tích lũy cao nhất.
10. Hãy cho biết điểm trung bình tích lũy của sinh viên có tên “Nguyễn Văn Tài”.
11. Liệt kê những sinh viên thuộc khoa “CNTT”, khóa 2006 có điểm trung bình tích lũy cao nhất.
12. Cho biết sinh viên nào thuộc khoa “Toán - Tin”, khóa 2004 có điểm môn “Cơ sở dữ liệu” cao nhất trong khóa đó.
13. Cho biết sinh viên nào thuộc khoa “CNTT”, khóa 2006 hiện nay chưa tích lũy được tín chỉ nào

VIII. Cho cơ sở dữ liệu gồm các quan hệ sau:

1. CongNhan(**MACN**, Ho, Ten, Phai, NgaySinh, MaTSX)

Tân từ: Mỗi công nhân có một mã số (MACN) duy nhất, họ và tên lót (Ho), tên nhân viên (Ten), giới tính (Phai), ngày sinh (NgaySinh), mã số tổ sản xuất mà công nhân đang làm việc (MaTSX).

2. ToSanXuat(**MaTSX**, TenTSX)

Tân từ: Mỗi tổ sản xuất có một mã số duy nhất (MaTSX) và tên tổ sản xuất (TenTSX).

3. SanPham(**MaSP**, TenSP, DVT, TienCong)

Tân từ: Mỗi sản phẩm có một mã số (MaSP) để phân biệt, tên sản phẩm (TenSP), đơn vị tính (DVT), tiền công (TienCong) làm ra một sản phẩm.

4. ThanhPham(**MACN, MaSP, Ngay**, SoLuong)

Tân từ: Hàng ngày đơn vị quản lý sẽ ghi nhận số lượng sản phẩm mỗi công nhân làm được.

Vẽ sơ đồ ER và viết các câu truy vấn sau:

1. Liệt kê các công nhân theo tổ sản xuất gồm các thông tin: TenTSX, HoTen, NgaySinh, Phai (xếp thứ tự tăng dần của tên tổ sản xuất, Tên của công nhân).
2. Liệt kê các thành phẩm mà công nhân ‘Nguyễn Trường An’ đã làm được gồm các thông tin: TenSP, Ngay, SoLuong, ThanhTien (xếp theo thứ tự tăng dần của ngày).
3. Liệt kê các nhân viên không sản xuất sản phẩm ‘Bình gốm lớn’.
4. Liệt kê thông tin các công nhân có sản xuất cả ‘Nồi đất’ và ‘Bình gốm nhỏ’.
5. Thống kê Số lượng công nhân theo từng tổ sản xuất.

6. Tổng số lượng thành phẩm theo từng loại mà mỗi nhân viên làm được (Ho, Ten, TenSP, TongSLThanhPham, TongThanhTien).
7. Tổng số tiền công đã trả cho công nhân trong tháng 1 năm 2006
8. Cho biết sản phẩm được sản xuất nhiều nhất trong tháng 2/2006
9. Cho biết công nhân sản xuất được nhiều '**Chén**' nhất.
10. Tiền công tháng 2/2006 của công nhân viên có mã số 'CN002'
11. Liệt kê các công nhân có sản xuất từ 3 loại sản phẩm trở lên.
12. Cập nhật giá tiền công của các loại bình gốm thêm 1000.
13. Thêm bộ <'CN006', 'Lê Thị', 'Lan', 'Nữ', 'TS02' > vào bảng CongNhan.

Bài Tập Chương 5

RÀNG BUỘC TOÀN VỆN

- 1) Với các cơ sở dữ liệu đã thiết kế ở bài tập chương 2 & chương 3, ngoại trừ các RBTV khóa chính, khóa ngoại, và miền giá trị, hãy liệt kê tất cả các RBTV ngữ nghĩa khác (nếu có) và lập bảng tầm ảnh hưởng tổng hợp của các cơ sở dữ liệu này.
- 2) Cho cơ sở dữ liệu quản lý đơn đặt hàng, và hoá đơn giao hàng của một công ty thương mại gồm các lược đồ quan hệ:

KHACHHANG(MAKH, TENKH, DCHIKH, PHONE, CONGNO)

Tân từ: Mã số khách hàng(MAKH) xác định duy nhất họ tên khách hàng (TENKH), địa chỉ khách hàng(DCHIKH)), số phone khách hàng (PHONE)và tình trạng công nợ của khách hàng (CONGNO). CONGNO: Công nợ với khách hàng (CONGNO > 0: khách hàng nợ công ty, CONGNO < 0: công ty nợ khách hàng)

HANGHOA(MAHH, TENHH, QUICACH, DVTINH)

Tân từ: Mã số hàng hóa (MAHH) xác định duy nhất tên hàng hóa(TENHH), qui cách của hàng hóa (QUI CÁCH) và đơn vị tính của hàng hóa (DVTINH).

DATHANG(SODH, MAHH, SLMUA, NGAYDH, MAKH)

Tân Từ: Một đơn đặt hàng có thể có nhiều mặt hàng. Mã số đơn đặt hàng (SODH) xác định duy nhất ngày đặt hàng (NGAYDH) và mã số khách hàng đặt mua hàng. Mã số đơn đặt hàng cùng với mã số hàng hoá mà khách đặt mua xác định duy nhất số lượng đặt mua (SLMUA)cho hàng hóa.

HOADON (SOHD, NGAYHD, SODH, TRIGIAHD, NGAYXUAT)

Tân từ: Mã số hoá đơn (SOHD) xác định duy nhất một hoá đơn bán hàng. Mã số hóa đơn xác định duy nhất ngày viết hóa đơn (NGAYHD), mã số đơn đặt hàng (SODH), tổng trị giá đơn đặt hàng (TRIGIAHD) và ngày xuất (NGAYXUATT) kho giao hàng cho khách. Một hoá đơn bán hàng có thể gồm nhiều mặt hàng. Hóa đơn bán hàng theo yêu cầu của một đơn hàng có mã số SOHD, và ngược lại mỗi đơn đặt hàng sẽ được giải quyết chỉ trong một hóa đơn. Do điều kiện khách quan, có thể công ty không giao đầy đủ các mặt hàng cũng như số lượng từng mặt hàng yêu cầu trong đơn đặt hàng nhưng không bao giờ vượt ngoài yêu cầu.

CTIET_HD(SOHD, MAHH, GIABAN, SLBAN)

Tân từ: Mã số hóa đơn bán hàng (SOHD) và mã số hàng hóa (MAHH) xác định duy nhất giá bán (GIABAN) của hàng hóa (như vậy cùng một hàng hóa nhưng có thể có giá bán khác nhau tùy theo khách hàng) và số lượng bán cho khách (SLBAN). Số lượng bán cho khách không được phép vượt quá số lượng mà khách yêu cầu mua trong đơn đặt hàng.

PHIEUTHU(SOPT, NGAYPT, MAKH, SOTIEN)

Tân từ: Mã số phiếu thu (SOPT) xác định duy nhất một phiếu thu. Mã số phiếu thu xác định duy nhất ngày phát hành phiếu thu (NGAYPT), mã số khách hàng (MAKH) trả tiền

và số tiền thu của khách hàng (SOTIEN). Khách hàng có thể trả tiền không theo một hóa đơn bán hàng nào cả và cũng có thể trả tiền trước khi nhận hàng xem như tiền đặt cọc hay tiền ký quỹ.

Yêu cầu

- Xây dựng lược đồ ER cho cơ sở dữ liệu trên.
- Xác định khoá chính, khoá ngoại cho từng lược đồ quan hệ của lược đồ cơ sở dữ liệu trên.
- Xác định và phân loại (RBTV miễn giá trị, liên thuộc tính, liên bộ, tham chiếu, liên thuộc tính liên quan hệ, liên bộ liên quan hệ, thuộc tính tổng hợp, chu trình) những RBTV có thể có của cơ sở dữ liệu này, phát biểu chúng một cách chặt chẽ và nêu bảng tầm ảnh hưởng.

3) Cho một lược đồ CSDL sau:

KhachHang(MaKhach, Ten, DiaChi)

HangHoa(MaHang, Ten, DonViTinh, DacTa)

HoaDonNhap(MaKhach, MaHang, NgayNhap, SoLuong, DonGia)

HoaDonXuat(MaKhach, MaHang, NgayXuat, SoLuong, DonGia)

Hãy chỉ ra những vi phạm về ràng buộc dữ liệu trong các bảng dữ liệu sau:

KhachHang

MaKhach	Ten	DiaChi
KH01	Intel	NewYork
	HP	Washington
KH03	IBM	
KH04	Microsoft	Washington

HangHoa

MaHang	Ten	DonViTinh	DacTa
H01	CPU		
H02	Software		

HoaDonNhap

MaKhach	MaHang	NgayNhap	SoLuong	DonGia
KH01	H01	01/01/2007	100	
KH02	H03	15/01/2009	30	
KH03	H02	15/01/2007	20	
KH04	H01		30	250

HoaDonXuat

MaKhach	MaHang	NgayXuat	SoLuong	DonGia
KH01	H02	17/01/2007	15	
KH03	H02	20/01/2007	10	

4) Cho lược đồ CSDL gồm các quan hệ sau:

1. PHANXUONG(MAPX, TENPX, DTHOAI_PX, DCHI_PX, SL_SPPX)

Tên từ: Mỗi phân xưởng sản xuất của công ty có một mã số (MAPX) để phân biệt với các phân xưởng khác, tên gọi (TENPX), số điện thoại (DTHOAI_PX) và địa chỉ liên lạc (DCHI_PX). Mỗi phân xưởng có số lượng sản phẩm tối đa mà phân xưởng có thể sản xuất (SL_SPPX).

2. SANPHAM(MASP, TEN_SP, DVT, TON_TT)

Tân từ: Mỗi sản phẩm có một mã sản phẩm (MASP) dùng để phân biệt với các sản phẩm khác, có một tên sản phẩm (TEN_SP), đơn vị tính (DVT) và một số lượng quy định tồn kho tối thiểu cho phép (TON_TT).

3. PX_SP(MAPX, MASP)

Tân từ: Quan hệ phân xưởng sản phẩm quy định phân xưởng nào (MAPX) sản xuất sản phẩm nào (MASP).

4. KEHOACH_SX(MAKH, NGAYLAP_KH, THANGNAM, TRANGTHAI)

Tân từ: Mỗi kế hoạch sản xuất của công ty (MAKH) được lập vào ngày lập kế hoạch (NGAYLAP_KH) cho một tháng năm cụ thể nào đó (THANGNAM). Tại một thời điểm kế hoạch thuộc một trong 3 trạng thái (TRANGTHAI): chưa duyệt, đã duyệt, hoàn thành.

5. CHITIET_KH(MAKH, MAPX, MASP, SOLUONG_KH)

Tân từ: Mỗi chi tiết của kế hoạch quy định số lượng sản phẩm (MASP, SOLUONG_KH) cần sản xuất cho một phân xưởng (MAPX). Mỗi kế hoạch sản xuất (MAKH) phải có ít nhất một chi tiết kế hoạch có liên quan.

Yêu cầu:

- 1) Lập sơ đồ ER cho cơ sở dữ liệu trên.
- 2) Xác định khóa chính, khóa ngoại của các quan hệ trong lược đồ CSDL.
- 3) Ngoại trừ các ràng buộc suy ra từ khóa chính và khóa ngoại, hãy xác định 3 ràng buộc toàn vẹn thuộc 3 loại khác nhau và phát biểu một cách chặt chẽ các ràng buộc đó bằng ngôn ngữ hình thức.

Bài Tập Chương 6

LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU

I. Phụ thuộc hàm, Phủ tối thiểu, Khóa

- 1) Cho lược đồ quan hệ $R(A, B, E, G, H, I)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H \}$
 Dùng hệ tiên đề Armstrong và các hệ quả chứng minh rằng $AB \rightarrow GH$
- 2) Cho lược đồ quan hệ $R(A, B, C, D, E, G, H)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A \}$
 Dùng hệ tiên đề Armstrong và các hệ quả chứng minh rằng $AB \rightarrow E, AB \rightarrow G$
- 3) Cho lược đồ quan hệ $R(A, B, C, D)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ A \rightarrow B, BC \rightarrow D \}$. Phụ thuộc hàm nào trong dãy sau được suy từ F bằng các luật dẫn của hệ tiên đề Armstrong: $C \rightarrow D, A \rightarrow D, AD \rightarrow C, BC \rightarrow A, B \rightarrow CD$.
- 4) Cho quan hệ r như sau:

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_1	b_2	c_2	d_2	e_1
a_2	b_1	c_3	d_3	e_1
a_2	b_1	c_4	d_3	e_1
a_3	b_2	c_5	d_1	e_1

 Tìm tập phụ thuộc hàm F thỏa mãn r .
- 5) Cho lược đồ quan hệ $R(A, B, C, D, E, G, H)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ A \rightarrow D, AB \rightarrow DE, CE \rightarrow G, E \rightarrow H \}$.
 a) Tính $(AB)^+, (AC)^+, (ABE)^+, (ABC)^+$
 b) Tìm các khóa đề nghị của R .
- 6) Cho lược đồ quan hệ $R(A, B, C, D)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ AB \rightarrow C, D \rightarrow B, C \rightarrow ABD \}$. Tìm các khóa đề nghị của R
- 7) Cho lược đồ quan hệ $R(A, B, C, D, E, G)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, CG \rightarrow BD, CE \rightarrow CG \}$.
 a) Tìm phủ tối thiểu của F .
 b) Tìm các khóa đề nghị của R .
- 8) Cho lược đồ quan hệ $R(A, B, C, D)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ AB \rightarrow C, A \rightarrow D, BD \rightarrow C \}$. Tìm phủ tối thiểu của F .
- 9) Cho lược đồ quan hệ $R(A, B, C)$ và tập phụ thuộc hàm trên R sau:
 $F = \{ AB \rightarrow C, A \rightarrow B \}$. Tìm phủ tối thiểu của F .
- 10) Cho lược đồ quan hệ $R(A, B, C, D, E)$ và tập phụ thuộc hàm F trên R sau:
 $F = \{ A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A \}$.
 a) Tìm phủ tối thiểu của F .
 b) Tìm các khóa đề nghị đối với R .

11) Cho lược đồ quan hệ $R(A, B, C, D, E)$ và tập phụ thuộc hàm F trên R sau:

$$F = \{ A \rightarrow BC, A \rightarrow D, CD \rightarrow E \}.$$

a) $CD \rightarrow E$ có là phụ thuộc hàm đầy đủ không?

b) Xem : $G = \{ A \rightarrow BCE, A \rightarrow BD, CD \rightarrow E \}$

$$G' = \{ A \rightarrow BCD, CD \rightarrow E \}$$

G có tương đương với F ?

G' có tương đương với F ?

c) Tìm phủ tối thiểu của F .

d) Tìm các khóa đề nghị đối với R .

12) Cho $R(A, B, C)$ và các tập phụ thuộc hàm $G = \{ AB \rightarrow C; A \rightarrow B; B \rightarrow C; A \rightarrow C \}$, $F = \{ AB \rightarrow C; A \rightarrow B; B \rightarrow C \}$. F có tương đương với G không?

13) Cho lược đồ CSDL Kehoach(NGAY, GIO, PHONG, MONHOC, GIAOVIEN)

$$F = \{ \text{NGAY, GIO, PHONG} \rightarrow \text{MONHOC}; \text{MONHOC, NGAY} \rightarrow \text{GIAOVIEN};$$

$$\text{NGAY, GIO, PHONG} \rightarrow \text{GIAOVIEN}; \text{MONHOC} \rightarrow \text{GIAOVIEN} \}$$

a) Tính $\{ \text{NGAY, GIO, PHONG} \}^+; \{ \text{MONHOC} \}^+$

b) Tìm phủ tối thiểu của F

c) Tìm tất cả các khóa của Kehoach.

14) Cho lược đồ CSDL

$$R(\text{TENTAU}, \text{LOAITAU}, \text{MACHUYEN}, \text{LUONGHANG}, \text{BENCANG}, \text{NGAY})$$

$$F = \{ \text{TENTAU} \rightarrow \text{LOAITAU}; \text{MACHUYEN} \rightarrow \text{TENTAU}, \text{LUONGHANG};$$

$$\text{TENTAU, NGAY} \rightarrow \text{BENCANG}, \text{MACHUYEN} \}$$

a) Hãy tìm phủ tối thiểu của F

b) Tìm tất cả các khóa của R

15) Xác định phủ tối thiểu của các tập phụ thuộc hàm sau:

a) $R_1(A, B, C, D, E, G, H)$

$$F_1 = \{ A \rightarrow H, AB \rightarrow C, BC \rightarrow D, G \rightarrow B \}$$

b) $R_2(A, B, C, S, X, Y, Z)$

$$F_2 = \{ S \rightarrow A; AX \rightarrow B; S \rightarrow B; BY \rightarrow C; CZ \rightarrow X \}$$

c) $R_3(A, B, C, D, E, G, H, I, J)$

$$F_3 = \{ BG \rightarrow D; G \rightarrow J; AI \rightarrow C; CE \rightarrow H; BD \rightarrow G; JH \rightarrow A; D \rightarrow I \}$$

d) $R_4(A, B, C, D, E, G, H, I, J)$

$$F_4 = \{ BH \rightarrow I; GC \rightarrow A; I \rightarrow J; AE \rightarrow G; D \rightarrow B; I \rightarrow H \}$$

II. Các dạng chuẩn và chuẩn hóa lược đồ CSDL

1) Cho biết dạng chuẩn của các lược đồ quan hệ sau:

a) $R(\text{ABCDEG}); F = \{ A \rightarrow BC; C \rightarrow DE; E \rightarrow G \}$

b) $R(\text{ABCDEGH}); F = \{ C \rightarrow AB; D \rightarrow E; B \rightarrow G \}$

c) $R(\text{ABCDEGH}); F = \{ A \rightarrow BC; D \rightarrow E; H \rightarrow G \}$

- d) $R(ABCDEG); F = \{AB \rightarrow C; C \rightarrow B; ABD \rightarrow E; G \rightarrow A\}$
 e) $R(ABCDEFGHI); F = \{AC \rightarrow B; BI \rightarrow ACD; ABC \rightarrow D; H \rightarrow I; ACE \rightarrow BCG; CG \rightarrow AE\}$
 f) $R(A,B,C,D); F = \{CA \rightarrow D; A \rightarrow B\}$
 g) $R(S,D,I,M); F = \{SI \rightarrow D; SD \rightarrow M\}$
 h) $R(N,G,P,M,GV); F = \{N,G,P \rightarrow M; M \rightarrow GV\}$
 i) $R(S,N,D,T,X); F = \{S \rightarrow N; S \rightarrow D; S \rightarrow T; S \rightarrow X\}$
- 2) Kiểm tra tính nổi không mất thông tin của phân rã sau:
 $R(ABCDE); \rho(R1(AD); R2(AB); R3(BE); R4(CDE); R5(AE))$ với tập phụ thuộc hàm
 $F = \{A \rightarrow C; B \rightarrow C; C \rightarrow D; DE \rightarrow C; CE \rightarrow A\}$
- 3) Cho lược đồ quan hệ $R(A,B,C,D)$ và tập phụ thuộc hàm $F = \{A \rightarrow B; B \rightarrow C; A \rightarrow D; D \rightarrow C\}$ và một lược đồ CSDL $\mathcal{C} = \{Q1(AB); Q2(AC); Q3(BD)\}$
- a) \mathcal{C} có nổi không mất thông tin đối với R ?
 b) \mathcal{C} có bảo toàn phụ thuộc hàm?
- 4) Cho lược đồ CSDL: Kehoach(NGAY, GIO, PHONG, MONHOC, GIAOVIEN)
 $F = \{NGAY, GIO, PHONG \rightarrow MONHOC; MONHOC, NGAY \rightarrow GIAOVIEN; NGAY, GIO, PHONG \rightarrow GIAOVIEN; MONHOC \rightarrow GIAOVIEN\}$
- a) Xác định dạng chuẩn cao nhất của **Kehoach**
 b) Nếu Kehoach chưa đạt dạng chuẩn 3, hãy phân rã **Kehoach** thành lược đồ CSDL dạng chuẩn 3 vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin.
 c) Nếu **Kehoach** chưa đạt dạng chuẩn BC, hãy phân rã KeHoach thành lược đồ CSDL dạng BC
- 5) Cho lược đồ quan hệ $R(A, B, C, D)$ và tập phụ thuộc hàm $F = \{A \rightarrow B; B \rightarrow C; D \rightarrow B\}$
 $\mathcal{C} = \{Q1(A, C, D); Q2(B, D)\}$
- a) Xác định các F_i
 b) Lược đồ CSDL \mathcal{C} có đạt dạng chuẩn BC? Nếu không có thể phân rã tiếp các Q_i của \mathcal{C} để nâng cấp \mathcal{C} thành dạng chuẩn BC?
- 6) Giả sử ta có lược đồ quan hệ $R(C,D,E,G,H,K)$ và tập phụ thuộc hàm F như sau;
 $F = \{CK \rightarrow H; C \rightarrow D; E \rightarrow C; E \rightarrow G; CK \rightarrow E\}$
- a) Từ tập F , hãy chứng minh $EK \rightarrow DH$
 b) Tìm tất cả các khóa của R .
 c) Xác định dạng chuẩn của R .
 d) Hãy tìm cách phân rã R thành một lược đồ CSDL đạt dạng chuẩn BCNF. Tìm tập phụ thuộc hàm và khóa cho mỗi lược đồ quan hệ con.
 e) Phân rã ở câu d) có là phân rã giữ lại phụ thuộc không? Tại sao?
- 7) Cho lược đồ quan hệ $R(S,I,D,M)$
 $F = \{SI \rightarrow DM; SD \rightarrow M; D \rightarrow M\}$

- a) Tính bao đóng D^+ , SD^+ , SI^+
 - b) Tìm phủ tối thiểu của F
 - c) Tìm tất cả các khóa của R
 - d) Xác định dạng chuẩn cao nhất của R
 - e) Nếu R chưa đạt dạng chuẩn 3, hãy phân rã R thành lược đồ CSDL dạng chuẩn 3 vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin.
 - f) Nếu R chưa đạt dạng chuẩn BCNF, hãy phân rã R thành lược đồ CSDL dạng BCNF
 - g) Kiểm tra phép tách R thành các lược đồ con (SID, SIM) có bảo toàn thông tin?
 - h) Kiểm tra phép tách R thành các lược đồ con (SID, SIM) có bảo toàn phụ thuộc hàm?
- 8) Cho lược đồ quan hệ $R(W, A, Z, Y, Q, P)$; $F = \{W \rightarrow AYQP; A \rightarrow Z; YQP \rightarrow A\}$
- Một phân rã của R thành $\rho(R1(A, Z); R2(W, Y, Q, P); R3(Y, Q, P, A))$. Hãy kiểm tra tính kết nối không mất thông tin của phân rã này.
- 9) Cho lược đồ quan hệ $R(\text{Môn}, \text{GiảngViên}, \text{Giờ giảng}, \text{Phòng}, \text{SinhViên}, \text{Hạng})$ với $F = \{M \rightarrow GV; G, P \rightarrow M; G, GV \rightarrow P; M, SV \rightarrow H; G, SV \rightarrow P\}$
- Trong đó M: môn, GV: giáo viên, G: giờ giảng, P: phòng, SV: sinh viên, H: hạng.
- Một phân rã của R là $\rho(R1(M, G, P); R2(M, GV); R3(M, SV, H))$. Kiểm tra xem lược đồ cơ sở dữ liệu từ phân rã trên có bảo toàn thông tin đối với F?
- 10) Phân rã các lược đồ sau thành dạng BCNF
- a) $R(S, D, I, M)$ $F = \{S, I \rightarrow D; S, D \rightarrow M\}$
 - b) $R(A, B, C, D)$ $F = \{A \rightarrow B; B \rightarrow C; D \rightarrow B\}$
 - c) $R(C, S, Z)$ $F = \{C, S \rightarrow Z; Z \rightarrow C\}$
- 11) Phân rã các lược đồ quan hệ sau thành dạng 3NF vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin
- a) $Q(A, B, C)$; $F = \{A \rightarrow B; A \rightarrow C; B \rightarrow A; C \rightarrow A; B \rightarrow C\}$
 - b) $Q(\text{MSCD}, \text{MSSV}, \text{CD}, \text{HG})$
 $F = \{\text{MSCD} \rightarrow \text{CD};$
 $\text{CD} \rightarrow \text{MSCD};$
 $\text{CD}, \text{MSSV} \rightarrow \text{HG};$
 $\text{MSCD}, \text{HG} \rightarrow \text{MSSV};$
 $\text{CD}, \text{HG} \rightarrow \text{MSSV};$
 $\text{MSCD}, \text{MSSV} \rightarrow \text{HG}\}$
 - c) $Q(A, B, C, D)$, $F = \{AB \rightarrow C; C \rightarrow B\}$
- 14) Cho lược đồ quan hệ $R(A, B, C, D, E, K, L, N, I, J)$ và tập phụ thuộc hàm:
 $F = \{AB \rightarrow C; A \rightarrow DE; B \rightarrow K; K \rightarrow LN; D \rightarrow IJ\}.$

- a) Xét tập phụ thuộc hàm $G = \{ AB \rightarrow C; BD \rightarrow EK; AD \rightarrow LN; A \rightarrow I; N \rightarrow J \}$, F có tương đương với G ? Tại sao?

Các câu hỏi dưới đây đều được tính dựa trên tập phụ thuộc hàm F .

- b) Tìm mọi khoá của R .
- c) Tìm dạng chuẩn cao nhất của R ?
- d) Xét một phân rã của R gồm các lược đồ quan hệ sau :
- $R1(A, D, E) ; R2(B, K) ; R3(B, L, N) ; R4(A, I, J) ; R5(A, B, C)$
- a) Phân rã này có phải là phân rã nổi không mất thông tin? tại sao?
- b) Phân rã này có phải là phân rã giữ lại phụ thuộc? tại sao?
- c) Các lược đồ quan hệ trong phân rã này ở dạng chuẩn nào?
- e) Nếu R không ở dạng chuẩn 3NF, hãy tìm một phân rã ở dạng chuẩn 3NF thỏa tính chất nổi không mất thông tin và giữ lại phụ thuộc. Có nhận xét gì về dạng chuẩn của các lược đồ quan hệ trong kết quả của phân rã này?

15) Cho lược đồ quan hệ của hệ quản lý tuyển sinh đại học:

TuyểnSinh(SBD, HọTS, TênTS, GiớiTính, NgàySinh, ĐịaChỉ, KhuVực, ĐốiTượng, MãNgành, TênNgành, MãMôn, TênMôn, Điểm)

và tập phụ thuộc hàm F như sau:

$F = \{ SBD \rightarrow HọTS, TênTS, GiớiTính, NgàySinh, ĐịaChỉ, KhuVực, ĐốiTượng, MãNgành;$
 $MãNgành \rightarrow TênNgành;$
 $MãMôn \rightarrow TênMôn;$
 $SBD, MãMôn \rightarrow Điểm$
 $\}$

- a) Tìm tất cả các khóa của lược đồ quan hệ **TuyểnSinh**.
- b) Xác định dạng chuẩn của quan hệ **TuyểnSinh**. Nếu dùng lược đồ này để cài đặt thì tốt không? Vì sao?
- c) Hãy tìm cách phân rã (chuẩn hoá) lược đồ **TuyểnSinh** thành dạng 3NF và đặt tên phù hợp cho các lược đồ con trong kết quả phân rã.

16) Cho một phần cơ sở dữ liệu của hệ quản lý thẻ ATM như sau:

ThẻATM(SốThẻ, MãKH, TênKH, SốCMND, SốPin, Số_Tiền_Có, Giờ_Rút, Ngày_Rút, Số_Máy_ATM, Số_Tiền_Rút)

và tập phụ thuộc hàm F như sau:

$F = \{ \text{MãKH} \rightarrow \text{TênKH}, \text{SốCMND}, \text{SốPin};$

$\text{SốThẻ} \rightarrow \text{MãKH}, \text{Số_Tiền_Có};$

$\text{SốThẻ}, \text{Ngày_Rút}, \text{Giờ_Rút} \rightarrow \text{Số_Máy_ATM}, \text{Số_Tiền_Rút};$

$\}$

Diễn giải: Mỗi thẻ ATM có một số (**SốThẻ**) duy nhất để phân biệt với những thẻ ATM khác, được cấp cho một khách hàng duy nhất (**MãKH**) và có một số tiền hiện có trong tài khoản (**Số_Tiền_Có**). Mỗi khách hàng có một mã khách hàng để phân biệt, có họ tên (**TênKH**), số chứng minh nhân dân (**SốCMND**), số pin (**SốPin**) để thực hiện giao dịch rút tiền. Mỗi lần khách hàng rút tiền tại máy ATM ngân hàng sẽ ghi lại các thông tin về ngày rút (**Ngày_Rút**), giờ rút (**Giờ_Rút**), số máy ATM thực hiện giao dịch này (**Số_Máy_ATM**) và số tiền khách hàng đã rút (**Số_Tiền_Rút**) trong lần giao dịch đó.

- Tìm tất cả các khóa của **ThẻATM**.
- Lược đồ **ThẻATM** đạt dạng chuẩn mấy (cao nhất). Nếu dùng lược đồ này để cài đặt thì cơ sở dữ liệu có những hạn chế gì?
- Hãy tìm cách phân rã (chuẩn hoá) **ThẻATM** thành dạng BCNF nói không mất và giữ lại phụ thuộc; đặt tên phù hợp cho các lược đồ con trong kết quả phân rã; xác định khóa cho từng lược đồ con.

17) Cho lược đồ quan hệ

PhiếuGiao(MSPG, NgàyGiao, MSNV, SOHD, NgàyHD)

Tân từ: mỗi phiếu giao hàng có một mã phiếu (MSPG) duy nhất. Mỗi phiếu giao do một nhân viên lập và giao theo 1 hợp đồng (SOHD), mỗi hợp đồng có nhiều phiếu giao. NgàyGiao là ngày giao hàng cho khách, NgàyHD là ngày ký hợp đồng.

- Xác định khóa của quan hệ trên.
- Xác định dạng chuẩn cao nhất mà lược đồ quan hệ trên đạt được.
- Nếu lược đồ quan hệ trên chưa đạt dạng chuẩn 3, hãy tiến hành phân rã để lược đồ đạt ít nhất là dạng chuẩn 3.

BÀI TẬP TỔNG HỢP

I. Cho một lược đồ cơ sở dữ liệu C dùng để quản lý hoạt động sửa chữa, bảo trì xe của một gara xe hơi. Lược đồ cơ sở dữ liệu C gồm các lược đồ quan hệ như sau:

R1: Tho(MATHO, TENTHO, NHOM, NHOM_TRUONG)

Tân từ: Mỗi người thợ đều có mã số là MATHO để nhận diện. Mỗi thợ chỉ có một tên (TENTHO) và chỉ thuộc một nhóm (NHOM). Nhóm trưởng (NHOM_TRUONG) của mỗi nhóm là một trong số những người thợ của nhóm đó.
 $MGT(MATHO) = MGT(NHOM_TRUONG)$

R2: Cong_viec(MACV, NOIDUNGCV)

Tân từ: Dịch vụ sửa chữa xe được chia nhỏ thành nhiều công việc để dễ dàng tính toán chi phí với khách hàng. Mỗi công việc đều có mã riêng (MACV) và nội dung của công việc được mô tả qua NOIDUNGCV.

R3: Hop_dong(SOHD, NGAYHD, MAKH, TENKH, DCHI, SOXE, TRIGIAHD, NG_GIAO_DK, NG_NGTHU)

Tân từ: Mỗi hợp đồng sửa xe ký kết với khách hàng đều có mã số (SOHD) để phân biệt. NGAYHD là ngày ký hợp đồng. Mỗi khách hàng có một mã số (MAKH), một tên (TENKH) và một địa chỉ (DCHI) để theo dõi công nợ. SOXE là số đăng bộ của xe đem đến sửa chữa, số này do phòng CSGT đường bộ cấp (nếu xe đổi chủ thì xem như một xe khác). Khách hàng ký hợp đồng chính là chủ xe sửa chữa. Một khách hàng có thể ký nhiều hợp đồng sửa chữa nhiều xe khác nhau hoặc hợp đồng sửa chữa nhiều lần của cùng một xe nhưng trong cùng một ngày. Những công việc sửa chữa cho một đầu xe chỉ ký hợp đồng một lần. TRIGIAHD là tổng trị giá của hợp đồng. NG_GIAO_DK là ngày dự kiến phải giao trả xe cho khách. NG_NGTHU là ngày nghiệm thu thật sự sau khi đã sửa chữa xong để thanh lý hợp đồng.

R4: Chitiet_HD(SOHD, MACV, TRIGIA_CV, MATHO, KHOANTHO)

Tân từ: Mỗi hợp đồng sửa xe có thể gồm nhiều công việc. MACV là mã số của từng công việc. TRIGIA_CV là chi phí về vật tư, phụ tùng, thiết bị, công thợ ... đã tính toán với khách. Mỗi công việc của hợp đồng sẽ giao cho một người thợ phụ trách (MATHO) và một người thợ có thể tham gia vào nhiều công việc của một hay nhiều hợp đồng khác nhau. KHOANTHO là số tiền giao khoán lại cho người thợ sửa chữa.

R5: Phieu_thu(SOPH, NGAYPH, SOHD, MAKH, HOTEN, SOTIENTHU)

Tân từ: Khách hàng (MAKH) có thể thanh toán tiền của một hợp đồng (SOHD) làm nhiều lần trước hoặc sau khi nghiệm thu (trong cùng ngày hoặc khác ngày). Mỗi lần thanh toán đều có số phiếu để phân biệt (SOPH), NGAYPH là ngày phát hành phiếu và SOTIENTHU là số tiền thanh toán. HOTEN là họ tên của người mang tiền đến thanh toán (có thể khác với tên của khách hàng đứng ra ký hợp đồng)

Yêu cầu:

- 1) Xác định tập hợp F gồm tất cả các phụ thuộc hàm suy ra từ tập của các lược đồ quan hệ (không cần liệt kê các phụ thuộc hàm hiển nhiên). Xác định khóa cho từng lược đồ quan hệ.
- 2) Lược đồ cơ sở dữ liệu C ở dạng chuẩn mấy (cao nhất). Nếu chất lượng thiết kế của cơ sở dữ liệu chưa tốt, hãy dùng thuật toán phân rã để nâng cấp lược đồ cơ sở dữ liệu trên.
- 3) Mô tả tất cả các ràng buộc toàn vẹn của lược đồ cơ sở dữ liệu C. Lập bảng tầm ảnh hưởng tổng hợp.
- 4) Vẽ sơ đồ ER của hệ thống.
- 5) Dùng ngôn ngữ SQL để thực hiện những yêu cầu sau:
 - a) Cho biết danh sách những người thợ hiện không tham gia vào một hợp đồng sửa chữa nào.
 - b) Cho biết danh sách những hợp đồng hiện đã thanh lý (đã giao tra xe cho khách) nhưng chưa được thanh toán đầy đủ.
 - c) Cho biết người thợ nào thực hiện nhiều công việc nhất.
 - d) Cho biết người thợ nào thực hiện tổng giá trị công việc (tổng số tiền) cao nhất.

Lưu ý: Các thuộc tính đều được xem như thuộc tính đơn.

II. Cho lược đồ CSDL gồm các lược đồ quan hệ sau:

1. NUOC (MAN, TENN)

Tân từ: Mỗi nước có một mã nước (MAN) duy nhất và một tên nước (TENN).

2. TINH (MAT, TENT, MAN)

Tân từ: Mỗi tỉnh (thành phố) có một mã tỉnh thành phố (MAT) duy nhất, một tên tỉnh thành phố (TENT) thuộc về một nước (MAN).

3. QUAN(MAQ, MAT, TENQ, TONG_SO_DT)

Tân từ: Mỗi quận có một mã quận (MAQ) để phân biệt với các quận khác trong cùng một tỉnh (MAT) và một tên quận (TENQ) và tổng số điện thoại được lắp trong quận (TONG_SO_DT).

4. KHACHHANG (MAKH, TENKH, DIACHI, LOAIKH)

Tân từ: Mỗi khách hàng có một mã khách hàng (MAKH) duy nhất, một tên khách hàng (TENKH), địa chỉ khách hàng (DIACHI), và thuộc một loại khách hàng (LOAIKH). Loại khách hàng gồm có “cá nhân” và “doanh nghiệp”.

5. DIENTHOAI (SODT, MAKH, LOAIDT, MAQ, MAT, SOHD)

Tân từ: Mỗi điện thoại có một số điện thoại (SODT) để phân biệt với các điện thoại khác, thuộc về một khách hàng (MAKH), loại điện thoại (LOAIDT). Loại điện thoại gồm có “vô tuyến” và “dây cáp”. Một số điện thoại nằm ở một quận của một tỉnh, được lắp đặt theo một hợp đồng (SOHD).

6. DỊCH VỤ (MADV, TENDV)

Tân từ: Mỗi dịch vụ có một mã dịch vụ (MADV) duy nhất và một tên dịch vụ (TENDV).

7. ĐĂNG KÝ (SODT, MADV)

Tân từ: Mỗi số điện thoại có thể đăng ký nhiều dịch vụ, một dịch vụ cũng được nhiều số điện thoại đăng ký.

8. KHẢ NĂNG ĐĂNG KÝ (LOAIKH, MADV)

Tân từ: Mỗi loại khách hàng được cho phép đăng ký một số dịch vụ.

Yêu cầu:

- 1) Vẽ lược đồ ER
- 2) Xác định các khóa chính, khóa ngoại của các lược đồ quan hệ trên.
- 3) Không kể các ràng buộc suy từ khóa chính của các lược đồ quan hệ, hãy phát biểu một cách chặt chẽ tất cả các ràng buộc toàn vẹn của các lược đồ trên.
- 4) Thực hiện các câu truy vấn sau bằng ngôn ngữ đại số quan hệ (câu a, b, c, d, e) và ngôn ngữ SQL:
 - a) Danh sách tên các khách hàng lắp điện thoại “Dây cáp” của tỉnh có tên là “Tiền Giang”.
 - b) Danh sách các điện thoại thuộc loại “Vô tuyến”.
 - c) Cho biết tên và địa chỉ của các khách hàng có đăng ký dịch vụ có tên là “Dịch vụ 108”.
 - d) Danh sách các khách hàng có đăng ký dịch vụ có tên là dịch vụ “Dịch vụ 108” thuộc loại khách hàng “cá nhân”.
 - e) Liệt kê tất cả các tỉnh (mã tỉnh, tên tỉnh) hiện tại có lắp hơn 1000 điện thoại.
 - f) Cho biết số lượng điện thoại thuộc “TPHCM”.
 - g) Cho biết số lượng điện thoại của từng tỉnh (thành phố).
 - h) Cho biết số điện thoại đăng ký nhiều dịch vụ nhất.
 - i) Cho biết số lượng khách hàng theo từng loại.
 - j) Cho biết tổng số điện thoại của từng loại.

TÀI LIỆU THAM KHẢO

Tiếng việt

- [1] Vũ Đức Thi, *Cơ sở dữ liệu - Kiến thức và thực hành*, NXB Khoa học Thống kê, 1997.
- [2] Trần Thành Trai, *Nhập môn cơ sở dữ liệu*, Nxb Trẻ, 1996.
- [3] Đỗ Trung Tuấn, *Cơ sở dữ liệu*, NXB Giáo dục, 1998.
- [4] ULLMAN, J.D., *Nguyên lý các hệ cơ sở dữ liệu & cơ sở tri thức: Thiết kế các hệ cơ sở dữ liệu*, tập II (sách dịch), NXB Thống kê, 1999 (sách dịch).
- [5] Lê Tiên Vương, *Nhập môn Cơ sở dữ liệu quan hệ*, NXB Khoa học và Kỹ thuật, 1994.

Tiếng Anh

- [6] Elmasri, R. and Shamkant B. Navathe, *Fundamentals of Database Systems*, Fourth Edition, Addison-Wesley, 2004.
- [7] Maier, D. , *The Theory of Relational Databases*, Computer Science Press, 1983.
- [8] Ramakrishnan, R. and Gehrke, J., *Database Management Systems*, Third Edition, McGraw Hill, 2003
- [9] Silberschatz, A., Henry F. Korth and Sudarshan, S., *Database System Concepts*, Fourth Edition. McGraw Hill, 2004.
- [10] Sumathi, S. and Esakkirajan, S., *Fundamentals of Relational Database Management Systems*, Springer-Verlag, 2007