

CƠ SỞ DỮ LIỆU

1. TỔNG QUAN VỀ CSDL
2. MÔ HÌNH THỰC THỂ KẾT HỢP
3. MÔ HÌNH DỮ LIỆU QUAN HỆ
4. NGÔN NGỮ CSDL – SQL
5. RÀNG BUỘC CSDL
6. ĐẠI SỐ QUAN HỆ
7. LÝ THUYẾT THIẾT KẾ CSDL

Tài liệu

- <https://goo.gl/ipFeKC>
- https://drive.google.com/open?id=0B_rPqUNqK0TOMGgyWURzSkpFcDg
- Sách ở thư viện Trường ĐH Sư phạm - ĐHĐN
(<http://thuvien.ued.udn.vn>)
- Lý thuyết cơ sở dữ liệu / Nguyễn Thị Ngọc Mai. - H. : Lao động và xã hội, 2007
- Nguyên lý của các hệ cơ sở dữ liệu : Sách dùng cho sinh viên các trường đại học, cao đẳng / Nguyễn Kim Anh. - H. : Đại học Quốc gia Hà Nội, 2004
- Cơ sở dữ liệu / Đỗ Trung Tuấn. - In lần thứ 2. - H. : Đại học Quốc gia Hà Nội, 2007
- Bài tập cơ sở dữ liệu / Nguyễn Xuân Huy, Lê Hoài Bắc. - Hiệu đính và xuất bản lần thứ 6. - H. : Thông tin và Truyền thông, 2012

1. TỔNG QUAN VỀ CSDL

- 1.1 Một số khái niệm
 - Dữ liệu
 - CSDL
 - HQT CSDL
- 1.2 Các vấn đề CSDL&HQT CSDL cần giải quyết

1.1 Một số khái niệm

- 1.1 Một số khái niệm
- CSDL là một tập hợp các dữ liệu có tổ chức, mô tả một lĩnh vực ứng dụng nào đó hoặc một lĩnh vực của thế giới thực.
 - Ví dụ, CSDL về đào tạo của một trường đại học có thể chứa các thông tin về:
 - Các thực thể như sinh viên, giảng viên, các học phần, khoa, lớp, phòng học.
 - Các mối quan hệ giữa các thực thể, chẳng hạn như sinh viên đăng ký một học phần, giảng viên phụ trách một học phần cho lớp nào đó, việc sử dụng các phòng học,...
- HQT CSDL là một hệ thống phần mềm được thiết kế nhằm mục đích quản trị và khai thác một số lượng lớn các tập hợp lớn dữ liệu với lượng dữ liệu có thể tăng lên rất nhanh.

What is Data?

- Collection of facts or numbers
- Can be quantitative or qualitative
- Describes a variable or set of variables
- Essentially data can be thought of as the result of observations based on things like:
 - measurements
 - statistics

Data and Information

- Data is simply facts
- Why is data important?
 - By relating different pieces of data we are able to extract valuable information by presenting it in meaningful context
- For that we need to be able to:
 - Store data → so it can be persisted
 - Structure data → so it is easier to manipulate
 - Organize data → in a meaningful way
 - Process data → to derive data value from it

Why Databases?

- Data can be stored using multiple methods such as:
 - Text files
 - Comma delimited data files
 - Spreadsheets
 - Databases
- Why database?
 - The way data is accessed
 - The way data is handled

I.2 I.1 Các khái niệm cơ bản

- Hệ quản trị cơ sở dữ liệu:
- HQT CSDL là một hệ thống phần mềm được thiết kế nhằm mục đích quản trị và khai thác một số lượng lớn các tập hợp lớn dữ liệu với khối lượng có thể tăng lên rất nhanh.
- Thực tế, nếu hệ thống thông tin hỗ trợ lưu trữ dữ liệu một cách nhất quán trong một hoặc nhiều tệp, thì có thể nói rằng, nó đã đảm bảo tính nhất quán, và có thể gọi nó là HQT CSDL.

Managing Data

- Using a database provides:
 - a standard interface for accessing data
 - multiple users with simultaneous ability to insert, update and delete data
 - changes to the data without risk of losing data and its consistency
 - the capability to handle huge volumes of data and users
 - tools for data backup, restore and recovery
 - security
 - reduce redundancy
 - data independence

Database Management Systems

- Database Management System (DBMS)
 - It is the software system that manages databases
 - Provides an interface of access to the databases
 - Provides data services to applications
 - Efficient data querying and update mechanisms
 - Data integrity -guarantees data is always right even in case of software and hardware errors
 - Others: backup, compression, security, replication, etc.
- DBMS: DB2, Oracle, SQL Server, PostgreSQL, MySQL, Interbase, FireBird...

I.2 Chức năng của một HQT CSDL

- Quản trị dữ liệu trực tiếp trong bộ nhớ ngoài
 - Quản trị các bộ nhớ đệm (buffers) trong RAM
 - Điều khiển các giao tác
 - Nhật ký
 - Hỗ trợ ít nhất một ngôn ngữ CSDL
- 4 chức năng đầu tiên chủ yếu nhằm đảm bảo tính toàn vẹn dữ liệu - Tính chất quan trọng nhất đối với CSDL. Tất nhiên, HQT CSDL thực hiện chức năng dễ thấy, đó là truy nhập/xuất.

Ví dụ về sự phá huỷ tính toàn vẹn dữ liệu

- Để chuyển 500\$ từ tài khoản A vào tài khoản B ta thực hiện các bước sau (3 và 4 có thể hoán vị cho nhau):
 - Đọc tài khoản A: $SA = 1.000\$$
 - Đọc tài khoản B: $SB = 100\$$
 - Rút từ tài khoản A 500\$: $SA = 1000\$ - 500\$$
 - Thêm 500\$ vào tài khoản B: $SB = 100\$ + 500\$$
- Chuyện gì xảy ra nếu giữa hai bước 3 và 4 xảy ra sự cố phần cứng, phần mềm, đường truyền hay mất điện? Tài khoản A bị mất 500\$ và không biết nó sẽ đi về đâu, hoặc tài khoản B được 500\$ và không biết từ đâu

I.3 Kiến trúc của một hệ quản trị cơ sở dữ liệu

- Các bộ phận cấu thành của HQT CSDL tương ứng với tổ hợp các chức năng của nó.
- Trong các HQT CSDL quan hệ hiện đại có thể phân chia thành phần bên trong nhất - nhân của HQT CSDL, gọi là DataBase Engine (DBE); trình biên dịch ngôn ngữ (thường là SQL đối với HQT CSDL quan hệ); phân hệ hỗ trợ thời gian thực thi; và tổ hợp các tiện ích. Trong vài hệ thống, các phần đó được phân chia một cách rõ ràng, và trong một số không rõ ràng, nhưng một cách luận lý (logic), sự phân chia như thế được áp dụng trong tất cả các HQT CSDL.

Kiến trúc HQT CSDL

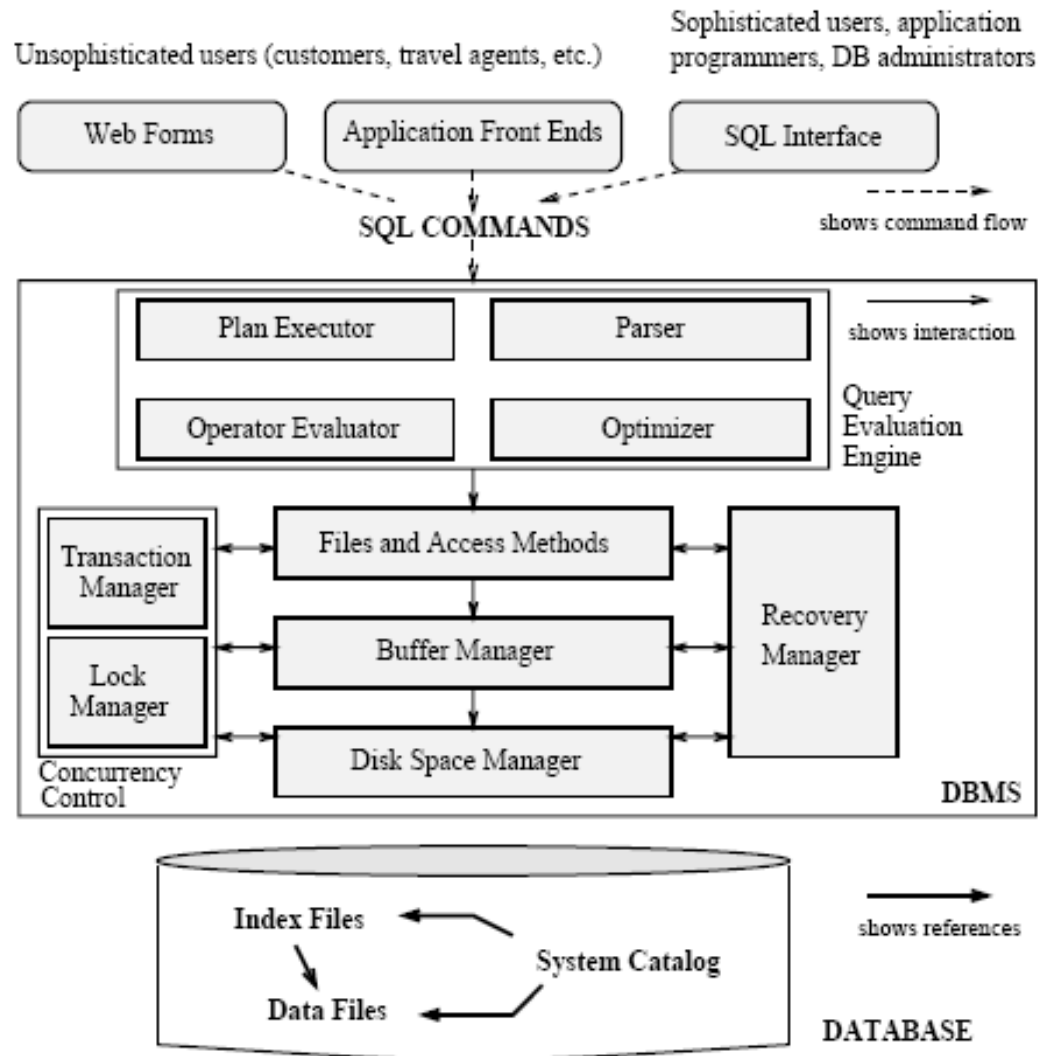


Figure 1.3 Architecture of a DBMS

2. Mô hình dữ liệu

- 2.0 Các mô hình dữ liệu
- 2.1 Mô hình thực thể kết hợp
- 2.2 Mô hình dữ liệu quan hệ
- 2.3 Chuyển mô hình thực thể kết hợp sang mô hình dữ liệu quan hệ
- 2.4 Ràng buộc toàn vẹn

2.0 Các mô hình dữ liệu

- Information Model
- Data Model
- Types of Data Models

Information Model

- Information Model
 - Abstract management of objects at a conceptual level
 - Independent of specific implementations and protocols
 - Hides all protocol and implementation details
 - Defines relationships between managed objects.
- Multiple implementations of an information model exists
 - Data models

Data Model

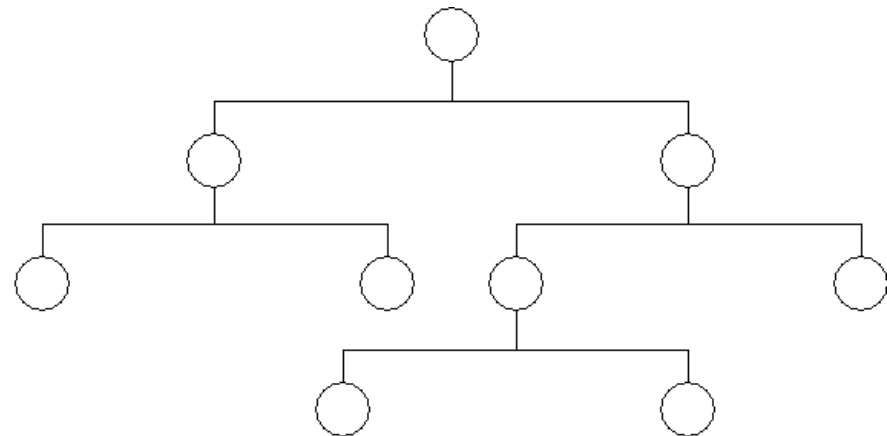
- A model is a representation of an object or concept of the real world
 - 3D model: a graphical representation of an object
 - Scale model: a replica or copy of an object in a smaller size
 - Business model: describes how a company operates
- Data Model
 - Define how data is to be represented and structured
 - It can be used to map how data from the real world is to be represented in a software system
- Characteristics of a Data Model
 - Lower level of abstraction
 - Intended for the software developer
 - Includes specific implementation and protocol details

Types of Data Models

- Extended Relational
- Entity-Relationship
- Hierarchical
- Network
- Relational
- Object-Oriented
- Object-relational
- Semantic
- Semi-structured (XML)

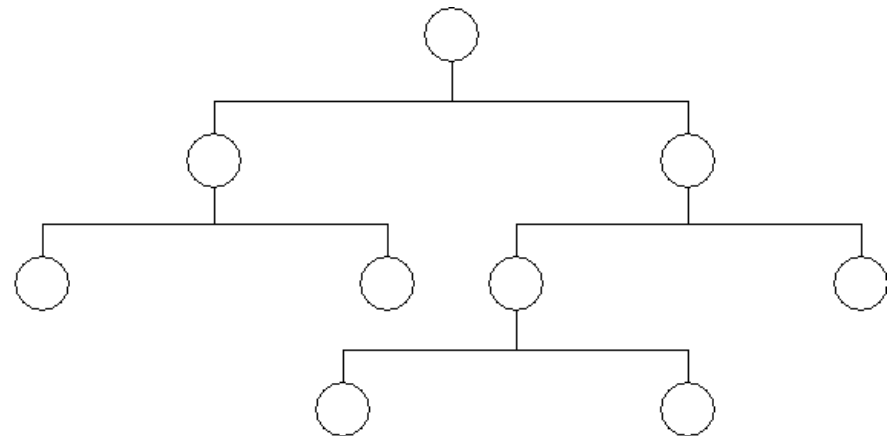
The Hierarchical Data Model – MH DL phân tầng (1)

- The Hierarchical Data Model structures data in a tree of records, with each record having one parent record and many children.
- A hierarchical database consists of the following:
 - It contains nodes connected by branches.
 - The top node is called the root.
 - If multiple nodes appear at the top level, the nodes are called root segments.



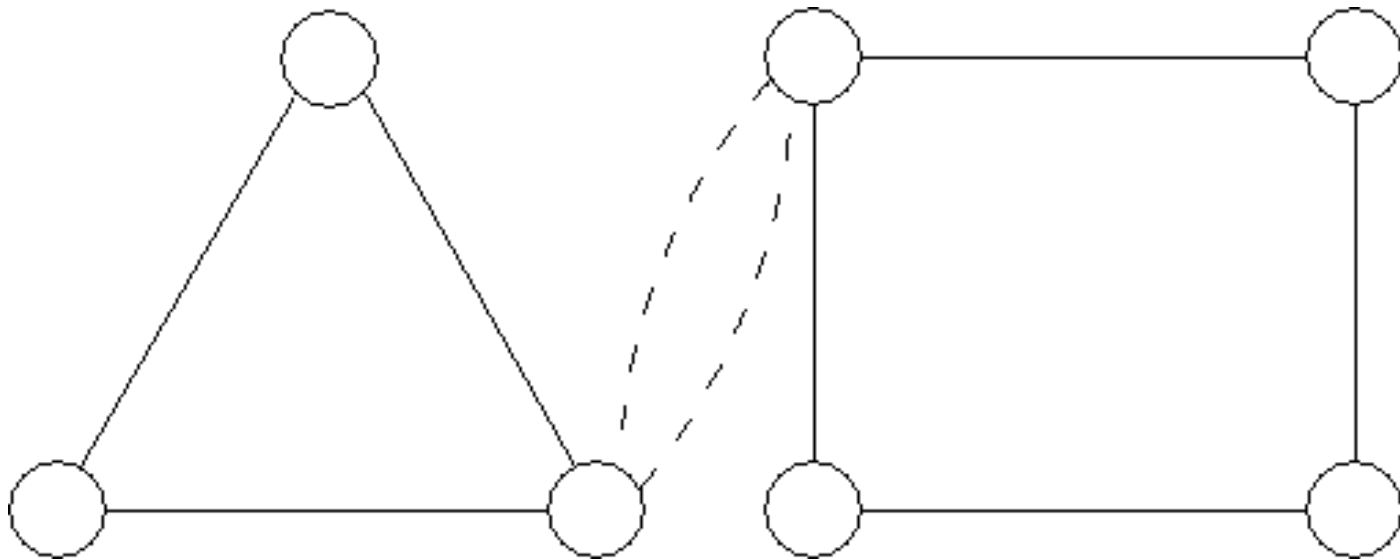
The Hierarchical Data Model – MH DL phân tầng (2)

- A hierarchical database consists of the following:
 - The parent of node n_x is a node directly above n_x and connected to n_x by a branch.
 - Each node (with the exception of the root) has exactly one parent.
 - The child of node n_x is the node directly below n_x and connected to n_x by a branch.
 - One parent may have many children.



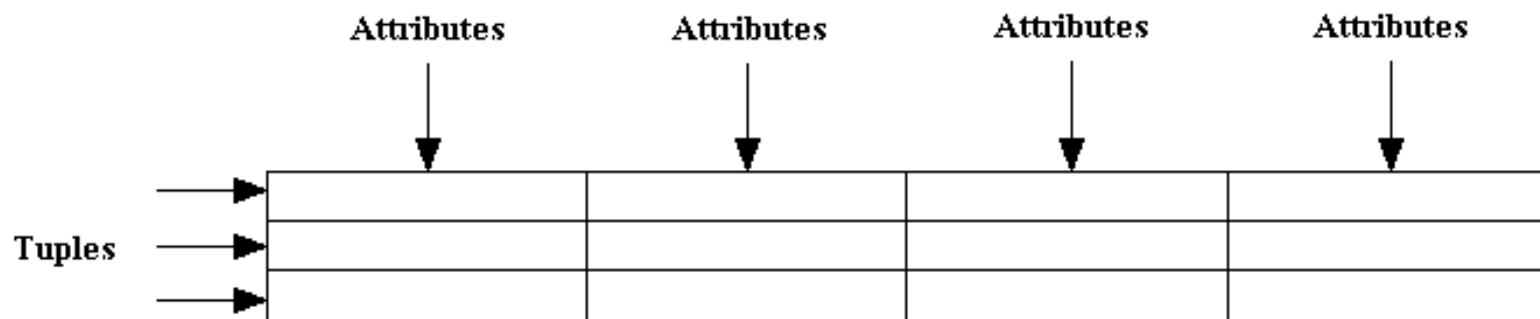
The Network Data Model – MH DL mạng

- The Network Data Model uses a lattice structure in which a record can have many parents as well as many children
- Like the The Hierarchical Data Model the Network Data Model also consists of nodes and branches, but a child may have multiple parents within the network structure instead of being restricted to just one.



The Relational Data Model – MH DL quan hệ

- The Relational Data Model has the relation at its heart, but then a whole series of rules governing keys, relationships, joins, functional dependencies, transitive dependencies, multi-valued dependencies, and modification anomalies.
- The Relation is the basic element in a relational data model.



Object-Oriented Data Model – MH DL hướng đối tượng

- A data model is a logic organization of the real world objects (entities), constraints on them, and the relationships among objects. A DB language is a concrete syntax for a data model. A DB system implements a data model.
- A core object-oriented data model consists of the following basic object-oriented concepts:
 - (1) object and object identifier
 - (2) attributes and methods
 - (3) class
 - (4) Class hierarchy and inheritance

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

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

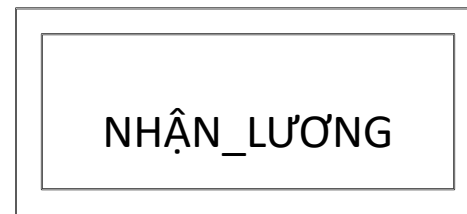
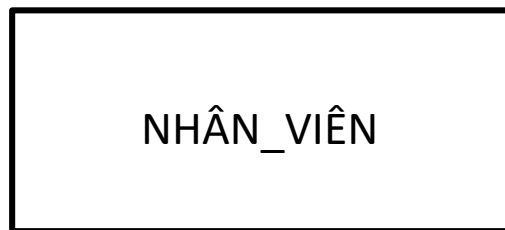
- Mô hình dữ liệu quan hệ có thể dùng để lưu trữ thông tin về lĩnh vực ứng dụng, nhưng để thiết kế CSDL cũng như các tiến trình thì không đủ vì rất phức tạp, đặc biệt trong mô tả ngữ nghĩa của lĩnh vực ứng dụng
- Mô hình thực thể - quan hệ (ER-Model) khắc phục được hạn chế đó
- Mô hình thực thể - quan hệ được thể hiện qua các sơ đồ thực thể quan hệ (sơ đồ ER)

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

- Trong mô hình ER có 3 khái niệm cơ bản: Tập thực thể, Quan hệ (mối quan hệ) và Thuộc tính
- Tập thực thể (entity-set): Thực thể là một đối tượng thực tế hoặc là tưởng tượng, và thông tin về nó được lưu trữ cũng như truy cập. Tập hợp các thực thể cùng loại gọi là tập thực thể. Các thực thể thuộc một tập thực thể không thể hoàn toàn trùng khớp nhau.
- Chú ý: Trong một số tài liệu, tập thực thể ở đây là kiểu thực thể (entity-type) hoặc thực thể (entity). Nếu “tập thực thể là thực thể”, thực thể sẽ là phiên bản của thực thể.

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

- Trong mô hình ER có 3 khái niệm cơ bản: Tập thực thể, Quan hệ và Thuộc tính
- Tập thực thể:
 - Ví dụ: Tập thực thể NHÂN_VIÊN. Có thể có rất nhiều thực thể NHÂN_VIÊN, và hai thực thể NHÂN_VIÊN không thể trùng nhau. Hai nhân viên có thể có HọTên, NgàySinh, QuêQuán, GiớiTính, PhòngBan trùng nhau, nhưng MãNhânViên thì khác nhau.
 - Trên sơ đồ ER, tập thực thể được biểu diễn bằng một hình chữ nhật có tên (nhãn) là tên của tập thực thể

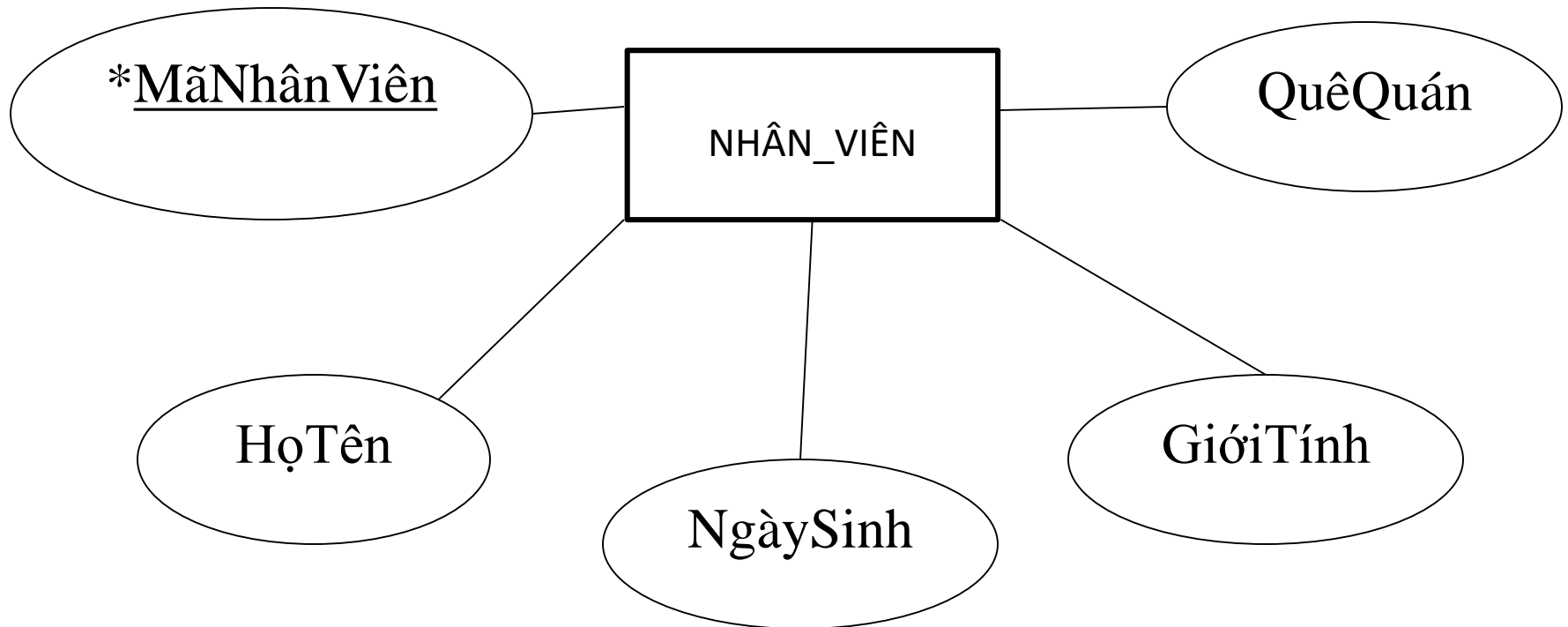


(Thực thể yếu - Weak Entity)

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

Trong mô hình ER có 3 khái niệm cơ bản:

- Thuộc tính:
 - Trên biểu đồ ER có thể biểu diễn các thuộc tính bằng các hình ô-van



2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

Trong mô hình ER có 3 khái niệm cơ bản:

- Thuộc tính:
 - Trên biểu đồ ER có thể biểu diễn các thuộc tính bằng các hình ô-van

Attribute

Họ Tên

Key attribute

*Mã Nhân Viên

Multivalued attribute



A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values

Derived attribute

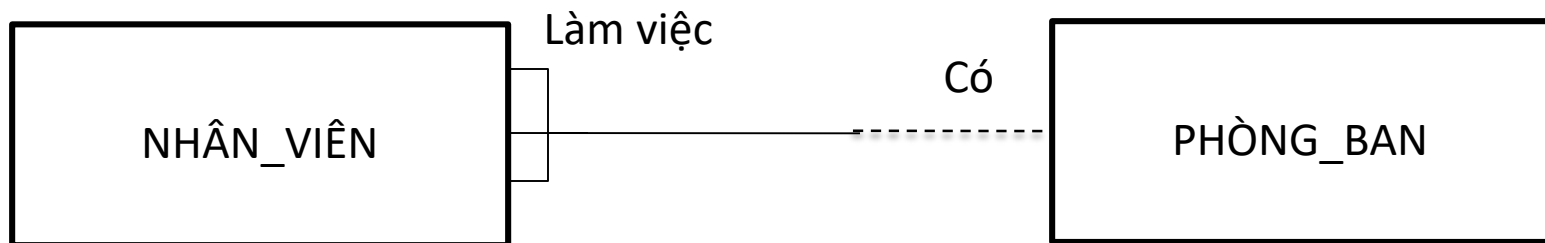


A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

Trong mô hình ER có 3 khái niệm cơ bản: Thực thể, Quan hệ và Thuộc tính

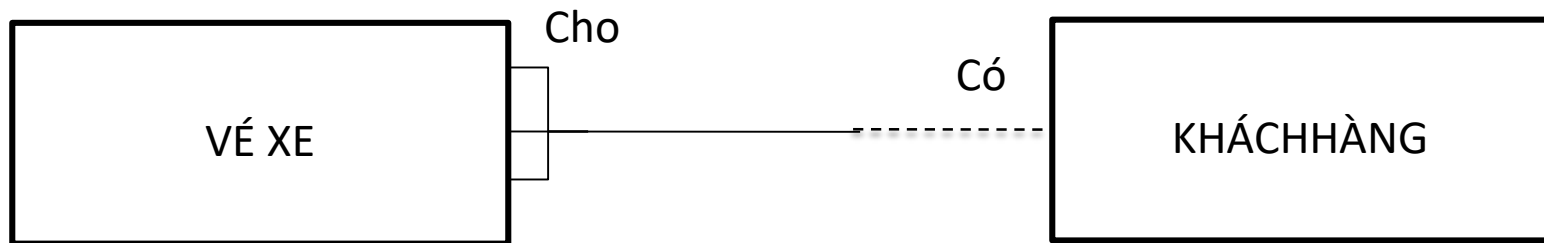
- Quan hệ: Là mối quan hệ giữa các thực thể của hai tập thực thể. Ví dụ hai tập thực thể NHÂN_VIÊN và PHÒNG_BAN. Nhiều NHÂN_VIÊN thuộc một PHÒNGBAN, hoặc trong một PHÒNGBAN có nhiều NHÂNVIÊN công tác.
- Trên sơ đồ ER, quan hệ được biểu diễn bằng hình vẽ nối hai tập thực thể có ghi chú mối quan hệ, cũng như số lượng thực thể tham gia (nhiều, một) và có nhất thiết (đường nối liền) hay không nhất thiết (đường đứt quãng)



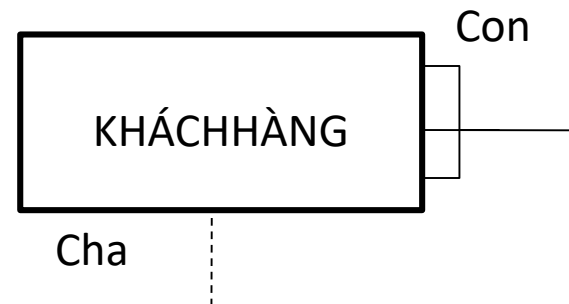
2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

Trong mô hình ER có 3 khái niệm cơ bản: Thực thể, Quan hệ và Thuộc tính

- Quan hệ (mối quan hệ):



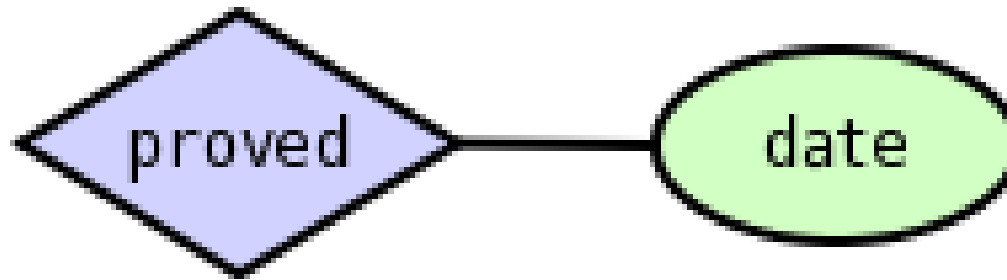
Quan hệ: Nhiều *Vé* cho một (hoặc không) *KháchHàng*, một *KháchHàng* có thể có nhiều *Vé*.
Sau đây là quan hệ “Một *Người* có thể có nhiều (hoặc không) *Con (Người)*, một *Con (Người)* thì chỉ có thể có một *Cha (Người)*”



2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

Trong mô hình ER có 3 khái niệm cơ bản:

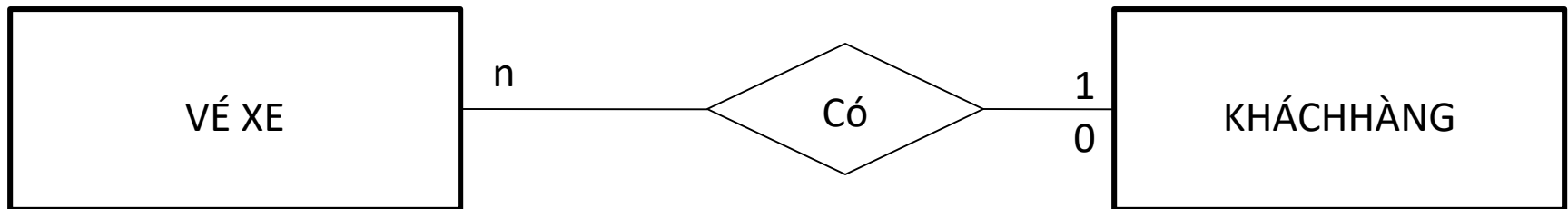
- Quan hệ:



A relationship with an attribute

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

- Trong mô hình ER có 3 khái niệm cơ bản:
- Quan hệ:
- Đôi khi người ta biểu diễn quan hệ như sau:



➤ **Thuộc tính:** Thuộc tính của thực thể là một chi tiết bất kỳ của thực thể.

Số lượng thuộc tính của một thực thể trong thực tế có thể rất lớn, nhưng trong CSDL chúng ta chỉ lưu trữ những thuộc tính cần thiết.

Ví dụ, thực thể NHÂN_VIÊN có các thuộc tính: *MãNhânViên*, *HọTên*, *NgàySinh*, *QuêQuán*, *GiớiTính*

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

Trong mô hình ER có 3 khái niệm cơ bản:

- Thuộc tính:
 - Từ định danh độc nhất (khóa) là thuộc tính, hoặc tổ hợp tối thiểu các thuộc tính, hoặc tổ hợp tối thiểu các quan hệ, hoặc tổ hợp tối thiểu các quan hệ và thuộc tính làm phân biệt một phiên bản bất kỳ của thực thể với tất cả các phiên bản khác của thực thể đó (Độc nhất nghĩa là các giá trị của nó không trùng nhau).
 - Mô hình ER chỉ dùng trong quá trình thiết kế, không dùng để lưu trữ và xử lý dữ liệu. Vì vậy, phải chuyển mô hình ER sang mô hình khác, chẳng hạn mô hình dữ liệu quan hệ. Khi đó, “dữ liệu được lưu trữ dưới dạng các bảng hai chiều”.

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

Trong mô hình ER có 3 khái niệm cơ bản:

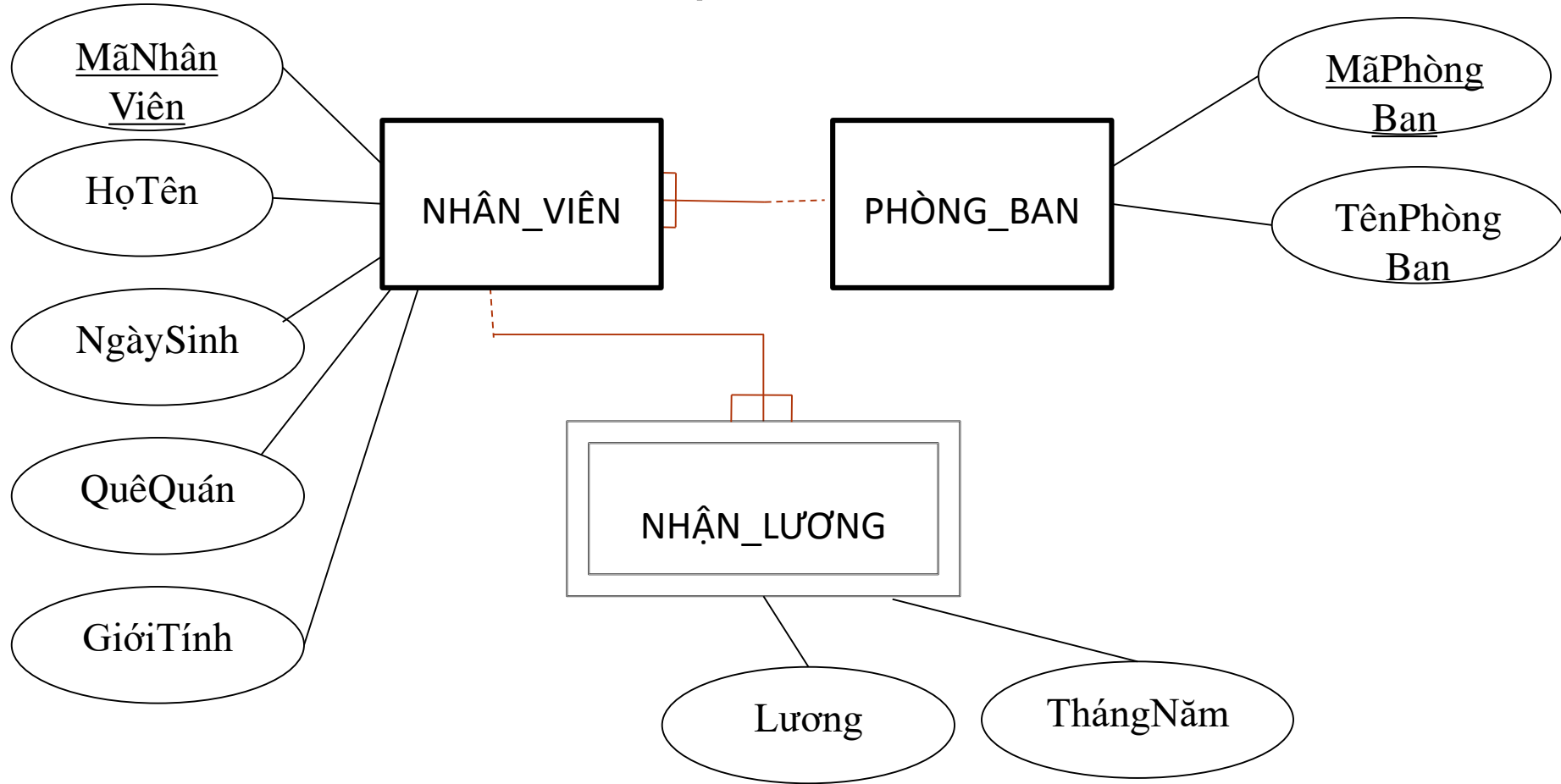
- Thuộc tính:

Ví dụ: NHÂNVIÊN (HọTên, NgàySinh, QuêQuán, GiớiTính)

- Nếu HọTên không thể trùng nhau, thì HọTên là khoá
- Nếu HọTên có thể trùng nhau, nhưng nếu không thể có hai NHÂNVIÊN có cùng HọTên và NgàySinh thì {HọTên, NgàySinh} là khoá.
- Hai nhân viên có thể có HọTên, NgàySinh, QuêQuán, GiớiTính trùng nhau, nhưng MãNhânViên luôn khác nhau, thì ta thêm thuộc tính MãNhânViên và chọn nó là khoá

2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

- Ví dụ biểu đồ thực thể - quan hệ:



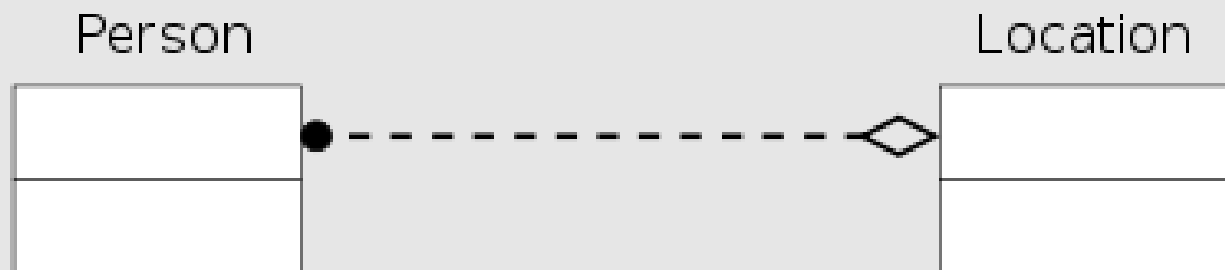
* NHẬN_LƯƠNG là thực thể yếu vì các thuộc tính của nó tự thân không thể xác định nó, khoá bao gồm: MãNhânViên và ThángNăm

2.1. Các hệ quy ước vẽ sơ đồ ER

Chen



IDEF1X



2.1. Các hệ quy ước vẽ sơ đồ ER

Bachman



Martin / IE /
Crow's Foot

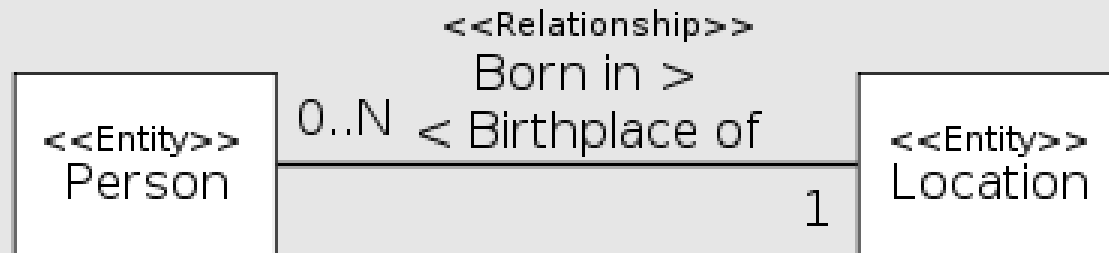


2.1. Các hệ quy ước vẽ sơ đồ ER

Min-Max / ISO

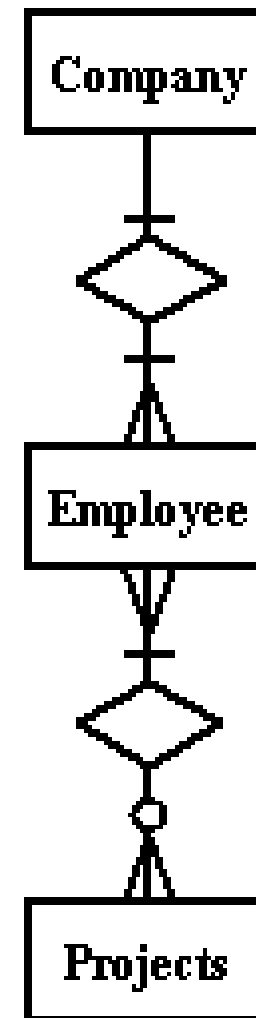
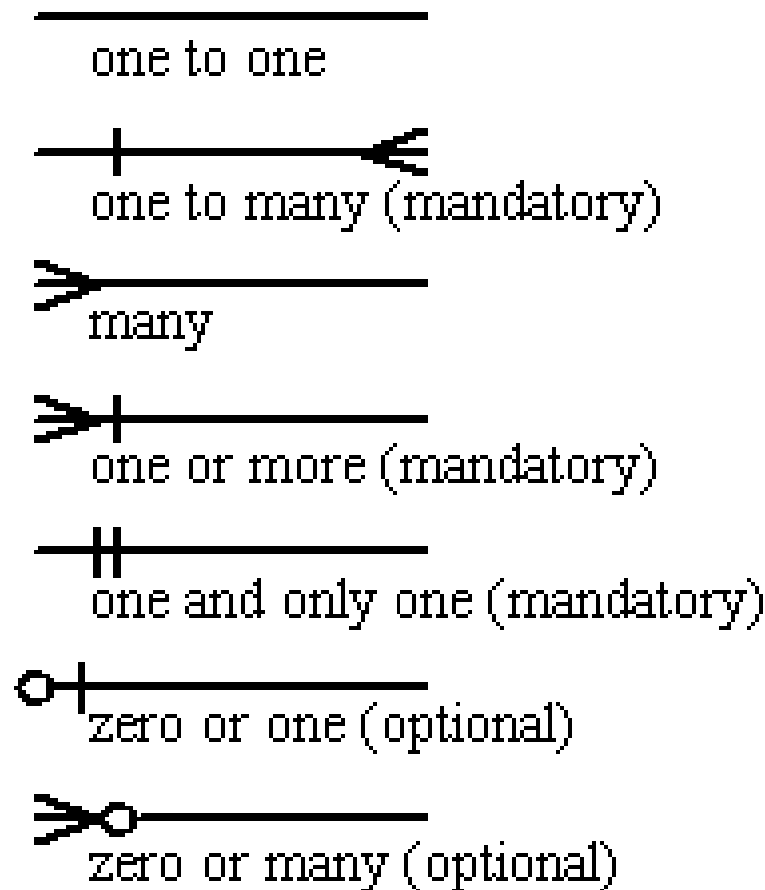


UML



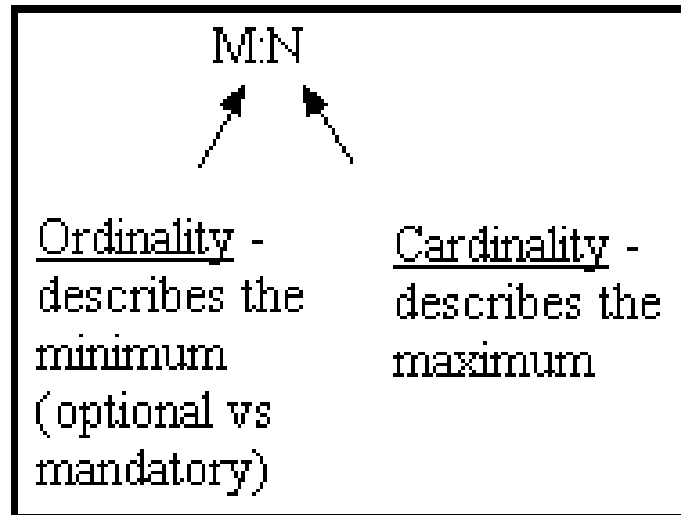
2.1. Các hệ quy ước vẽ sơ đồ ER – bản số

Information Engineering style



2.1. Các hệ quy ước vẽ sơ đồ ER – bản số

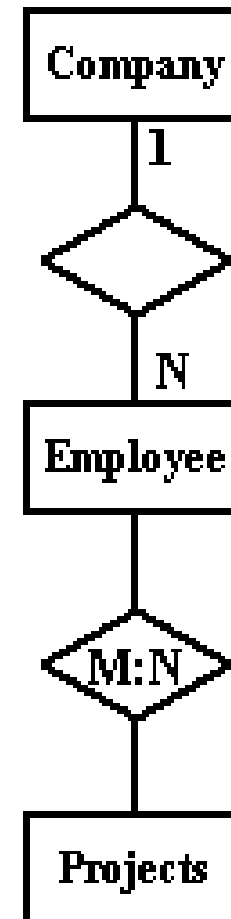
Chen style



1:N ($n=0,1,2,3\dots$)
one to zero or more

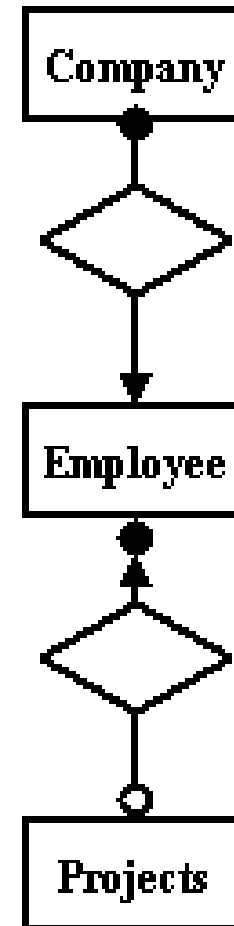
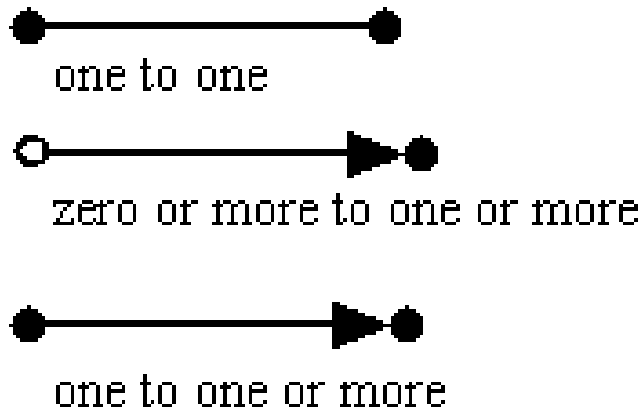
M:N (m and $n=0,1,2,3\dots$)
zero or more to zero or more
(many to many)

1:1
one to one



2.1. Các hệ quy ước vẽ sơ đồ ER – bản số

Bachman style



2.1. Các hệ quy ước vẽ sơ đồ ER – bản số

Martin style

1 - one, and only one (mandatory)

***** - many (zero or more - optional)

1...* - one or more (mandatory)

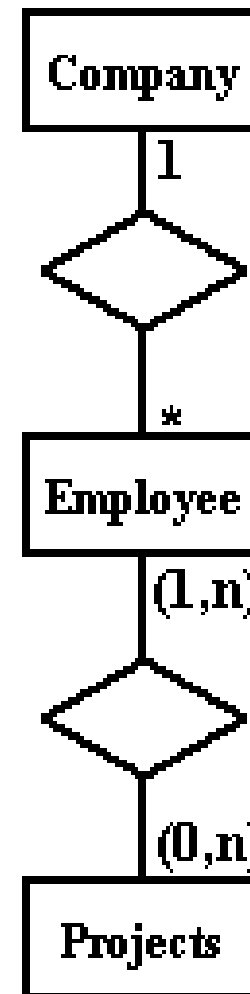
0...1 - zero or one (optional)

(0,1) - zero or one (optional)

(1,n) - one or more (mandatory)

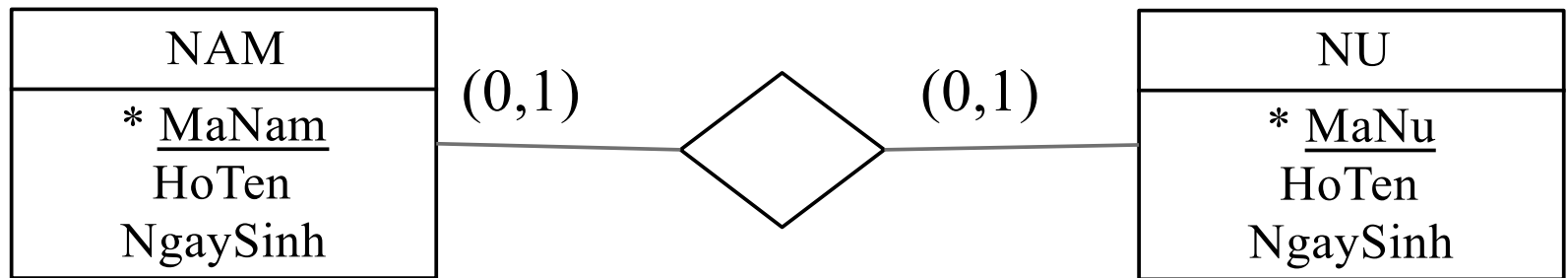
(0,n) - zero or more (optional)

(1,1) - one and only one (mandatory)



2.1. Thống nhất cách vẽ sơ đồ ER (*)

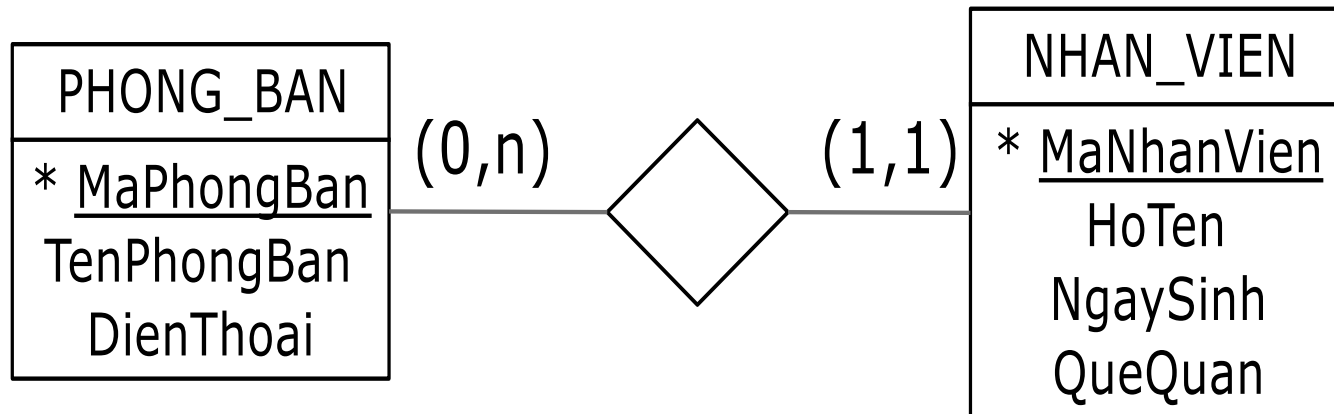
- Mỗi quan hệ Một - Một (One - One, 1 - 1, 1 - 1)
- Mỗi một nam chỉ có thể là chồng của một nữ duy nhất (hoặc không), và ngược lại, mỗi một Nữ chỉ có thể là vợ của một nam duy nhất (hoặc không)



(*) - Không biểu diễn được một số trường hợp: mỗi quan hệ tích hợp; thuộc tính đa trị; thuộc tính thứ sinh....

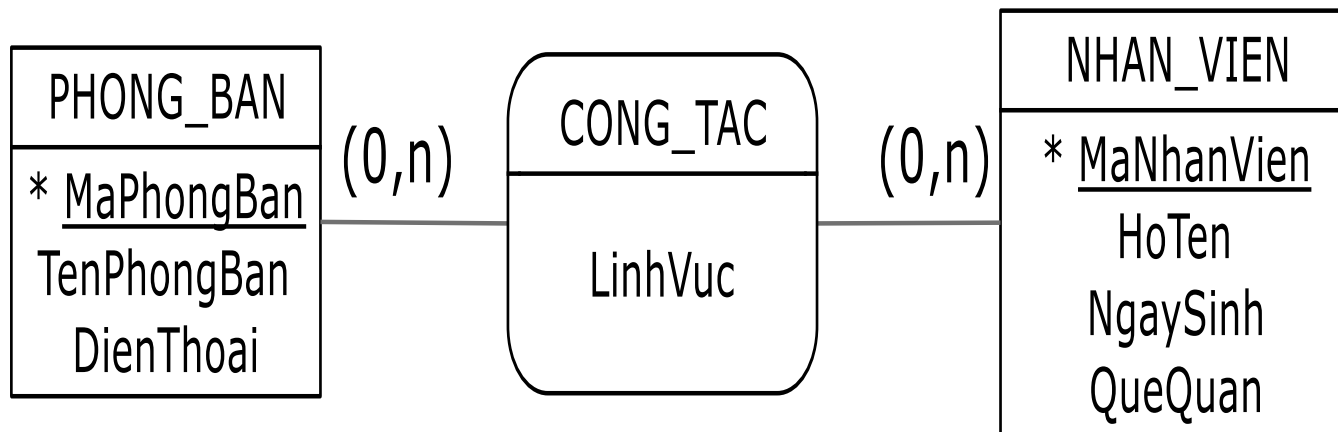
2.1. Tổng nhất các vẽ sơ đồ ER

- Mỗi quan hệ Một - Nhiều (One - Many, 1 - n, 1 - ∞)
- Mỗi phòng ban có thể có nhiều nhân viên làm việc, và mỗi nhân viên chỉ có thể cộng tác ở một phòng ban duy nhất



2.1. Tổng nhất các vẽ sơ đồ ER

- Mỗi quan hệ Nhiều - Nhiều (Many - Many, $n - n$, $\infty - \infty$)
- Mỗi phòng ban có thể có nhiều nhân viên làm việc, và mỗi nhân viên có thể cộng tác ở nhiều phòng ban khác nhau. Khi cộng tác ở một phòng ban, mỗi nhân viên được phân công một lĩnh vực công việc duy nhất



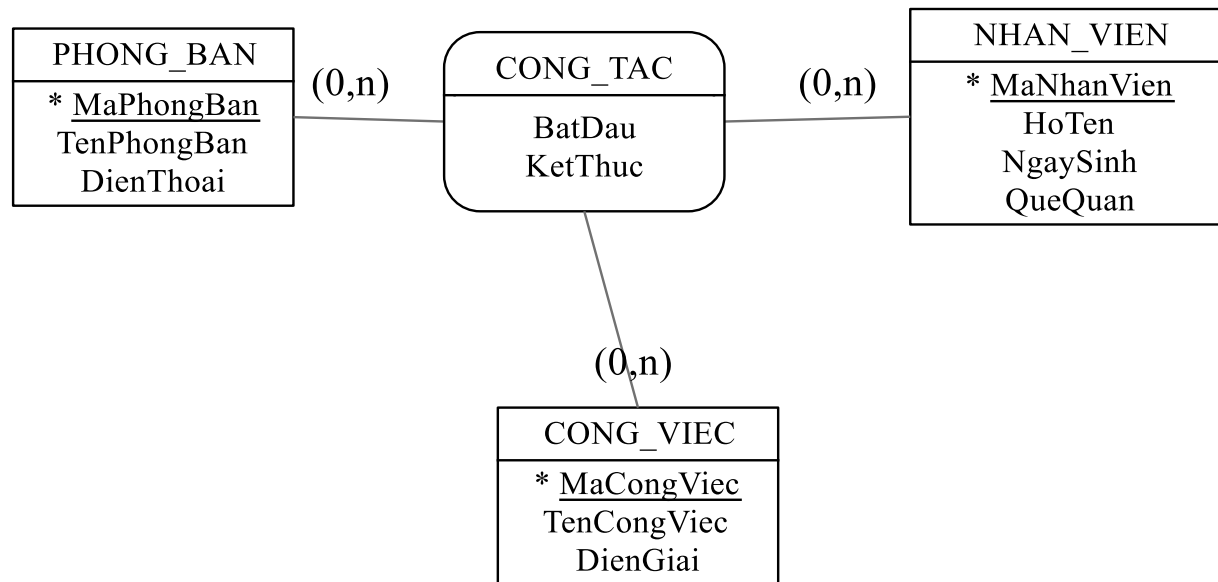
Mối quan hệ Nhiều - Nhiều (Many - Many, $n - n$, $\infty - \infty$)

- Có nhiều phòng ban và nhiều nhân viên. Mỗi phòng ban có thể có nhiều nhân viên làm việc, và mỗi nhân viên có thể cộng tác ở nhiều phòng ban khác nhau. Mỗi nhân viên có thể cộng tác ở một phòng ban nào đó nhiều lần, mỗi lần sẽ tham gia một công việc cụ thể



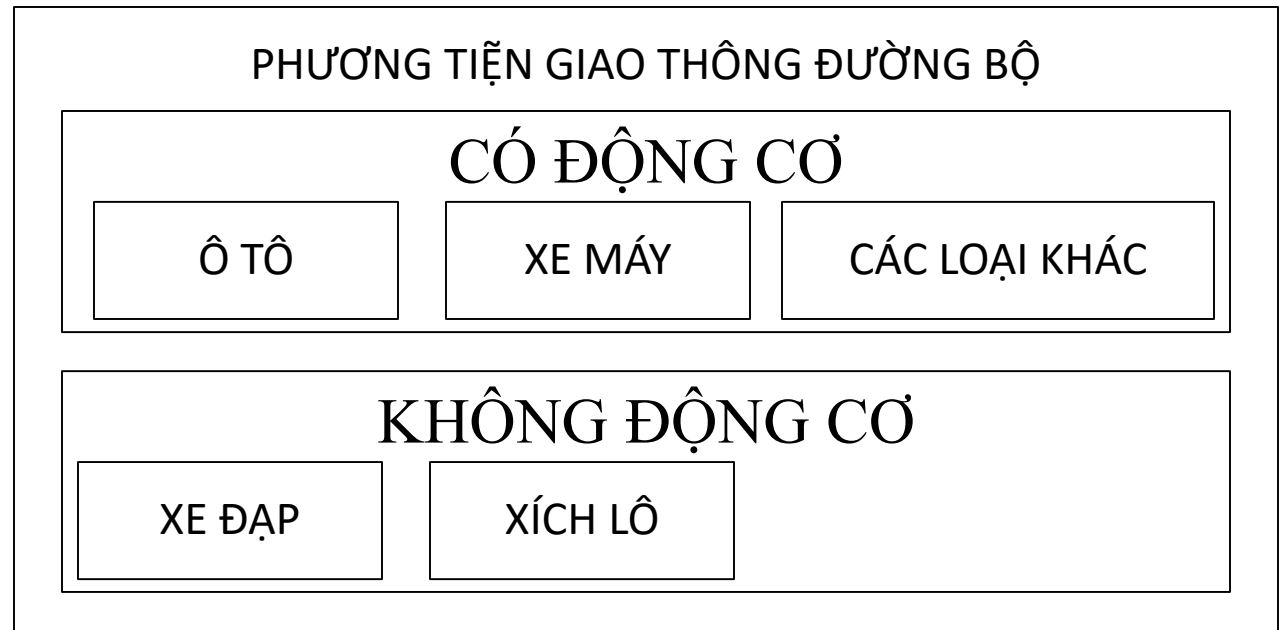
Mối quan hệ Nhiều - Nhiều (Many - Many, n - n, $\infty - \infty$)

- Có nhiều công việc, nhiều phòng ban và nhiều nhân viên. Mỗi phòng ban có thể có nhiều nhân viên làm việc, và mỗi nhân viên có thể cộng tác ở nhiều phòng ban khác nhau. Mỗi nhân viên có thể cộng tác ở một phòng ban nào đó nhiều lần, mỗi lần sẽ tham gia một công việc cụ thể và không lặp lại giữa các lần



2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

- Các phần tử phức tạp hơn của mô hình ER:
- Siêu kiểu (super type) và kiểu con (sub-type) của thực thể. Ví dụ, “Phương tiện giao thông đường bộ” là Siêu kiểu; “Ô tô”, “Xe máy”, “Xe đạp”,... là các kiểu con cấp độ thấp nhất.



2.1. Mô hình hoá DL theo ngữ nghĩa, sơ đồ ER

- Một số phần tử phức tạp hơn của mô hình ER:
- Mức độ chính xác của liên kết. Ví dụ, một SINHVIÊN không thể đăng ký nhiều hơn 10 MÔN HỌC cho một học kỳ. Thay vì n (many), người ta sẽ ghi số 10 (số lớn nhất) trên biểu đồ ER.
- Xoá/cập nhật liên hoàn (cascade) các phiên bản của các thực thể. Ví dụ, một SINHVIÊN chỉ và phải thuộc một LỚP HỌC. Khi xoá/cập nhật một phiên bản của thực thể LỚP HỌC, thì tất cả các phiên bản của thực thể SINHVIÊN thuộc lớp học đó cũng bị xoá/cập nhật theo.
- Miền giá trị (domain). Có thể chỉ rõ những giá trị nào một thuộc tính có thể nhận được. Ví dụ, miền giá trị của thuộc tính GiớiTính = {Nam, Nữ}

Bài tập – Bài toán quản lý hoá đơn

Có nhiều nhân viên, nhiều khách hàng, nhiều hàng hoá, nhiều hoá đơn. Mỗi nhân viên có thể lập nhiều hoá đơn, mỗi hoá đơn do một nhân viên lập. Mỗi khách hàng có thể có nhiều hoá đơn, mỗi hoá đơn thuộc về một khách hàng. Trong mỗi hoá đơn có thể có nhiều mặt hàng, mỗi mặt hàng có thể có mặt trong nhiều hoá đơn. Mỗi mặt hàng nếu có mặt trong một hoá đơn nào đó thì chỉ một lần duy nhất với số lượng và đơn giá cụ thể.

Học viên tự cho các thuộc tính tối thiểu.

Vẽ sơ đồ thực thể kết hợp và chuyển sang sơ đồ cơ sở dữ liệu quan hệ.

VD: Quản lý học tập theo hệ thống tín chỉ

- Các tập thực thể: Có nhiều khoa, ngành, bộ môn, môn học, giảng viên, sinh viên, lớp sinh hoạt, lớp tín chỉ

Các mối quan hệ:

- 1 khoa -> n bộ môn
- 1 bộ môn -> 1 khoa
- 1 bộ môn -> n ngành
- 1 ngành -> 1 bộ môn
- 1 ngành -> n lớp SH
- 1 lớp SH -> 1 ngành
- 1 sinh viên -> 1 lớp SH
- 1 lớp SH -> n sinh viên
- 1 giảng viên -> n môn học
- 1 môn học -> n giảng viên
- 1 bộ môn -> n giảng viên
- 1 giảng viên -> 1 bộ môn
- 1 lớp TC -> 1 giảng viên
- 1 giảng viên -> n lớp TC
- 1 môn học -> n lớp TC
- 1 lớp TC -> 1 môn học
- 1 sinh viên -> n lớp TC
- 1 lớp TC -> n sinh viên
- Sinh viên – Lớp tín chỉ -> điểm
- 1 sinh viên -> n ngành
- 1 ngành -> n sinh viên
- 1 ngành -> n môn học
- 1 môn học -> n ngành

VD: Quản lý học tập theo hệ thống niên chế

- Các tập thực thể: Có nhiều khoa, ngành, bộ môn, môn học, giảng viên, sinh viên, lớp sinh hoạt

• Các mối quan hệ:

- 1 khoa \rightarrow n bộ môn
- 1 bộ môn \rightarrow 1 khoa
- 1 bộ môn \rightarrow n ngành
- 1 ngành \rightarrow 1 bộ môn
- 1 ngành \rightarrow n lớp SH
- 1 lớp SH \rightarrow 1 ngành
- 1 sinh viên \rightarrow 1 lớp SH
- 1 lớp SH \rightarrow n sinh viên
- 1 giảng viên \rightarrow n môn học
- 1 môn học \rightarrow n giảng viên
- 1 bộ môn \rightarrow n giảng viên
- 1 giảng viên \rightarrow 1 bộ môn
- 1 môn học \rightarrow n lớp SH
- 1 lớp SH \rightarrow n môn học
- lớp SH - môn học \rightarrow giảng viên, phòng
- 1 sinh viên \rightarrow n môn học
- 1 môn học \rightarrow n sinh viên
- Sinh viên – môn học – lần \rightarrow điểm
- 1 ngành \rightarrow n môn học
- 1 môn học \rightarrow n ngành

2.2. Mô hình dữ liệu quan hệ, sơ đồ DR

- Các khái niệm:
- Quan hệ (bảng)
- Thuộc tính (cột)
- Khoá và thuộc tính khoá
- Khoá chính
- Bản ghi (bộ giá trị, dòng)
- Trường
- Kiểu dữ liệu
- Miền giá trị
- Lược đồ quan hệ

The Relational Data Model (1)

- Proposed by E.F. Codd in 1970.
- It is mathematical model that describes data as a collection of Relations (sets) and the values of the data is defined by Domains.
- Focuses on providing better data independence
- Data are operated upon by means of a relational calculus or relational algebra

The Relational Data Model (2)

- Advantages
 - Based on a formal theoretical model and proven in practice
 - Provides logical view of the data
- It is implemented by most DBMS in the market, such as DB2.
 - There are called Relational Database Management Systems

Components of the Relational Data Model

- The relational data model has its own unique terms used to define its concepts.
 - Attribute
 - Relation
 - Tuple
 - Data type defines the set of possible values that data can assume
 - Domain is the sub-set of set of all values of the data type
 - Relation is composed by a heading and a body
 - Heading: a set of attributes
 - Body: a set of tuples
 - Attribute is composed by a name and a domain (type)
 - A tuple is a set of attribute values

Thành phần của mô hình dữ liệu quan hệ

- Mỗi ô là một trường - field, trong một số trường hợp, field được đồng nhất với column

Cột/Column/Attribute

Bảng NHAN_VIEN

*<u>MaNhanVien</u>	HoTen	NgaySinh	QueQuan	CMND
N1	Nhân Viên 1	NS 1	Q1	1
N2	Nhân Viên 1	NS 1	Q1	3
N3	Nhân Viên 2	NS 3	Q1	2
N4	Nhân Viên 4	NS 4	Q2	7
N5	Nhân Viên 5	NS 5	Q1	5
N6	Nhân Viên 6	NS 6	Q1	6

Bản ghi/
Record
Bộ/
Tuple
Dòng/
Row

Components of a Relational Database

- Concepts from Relational Data Model can be mapped to their implementation found in a Relational Database
- Relational databases store data using tables
 - A table consists of columns and rows
 - Each column has a specific data type
 - Each row features a certain value for each column

Tables (1)

- A Table is the counterpart of a Relation from the Relational
 - Data Model
- A table stores data in rows and columns
 - Rows are the same as Tuples
 - Columns are the same as Attributes
- There can be multiple tables for different types of data to reduce redundant information
 - Normalization (more on this later)

Tables (2)

- For example:
 - You want to store data about a company
 - Data about branch offices will be stored in a table
 - Employee data for specific branches will be stored in its own table
 - Product data will be stored in another table

Columns

- Columns are also known as fields
- Each field contains a specific type of information such as name, extension, position and so on
- Columns must be designated a specific data type such as DATE, VARCHAR, INTEGER and so on

The diagram illustrates a database column structure. It features a vertical stack of seven cells. The top cell is dark blue with the header 'HoTen' in white. The subsequent six cells are light blue and contain the values 'Nhân Viên 1', 'Nhân Viên 1', 'Nhân Viên 2', 'Nhân Viên 4', 'Nhân Viên 5', and 'Nhân Viên 6' respectively. To the right of the table, two orange arrows point left towards the first and fourth rows. The top arrow is labeled 'Specific information' and points to the header row. The bottom arrow is labeled 'Data type: Char' and points to the fourth row, which contains the value 'Nhân Viên 4'.

HoTen	Specific information
Nhân Viên 1	
Nhân Viên 1	
Nhân Viên 2	
Nhân Viên 4	Data type: Char
Nhân Viên 5	
Nhân Viên 6	

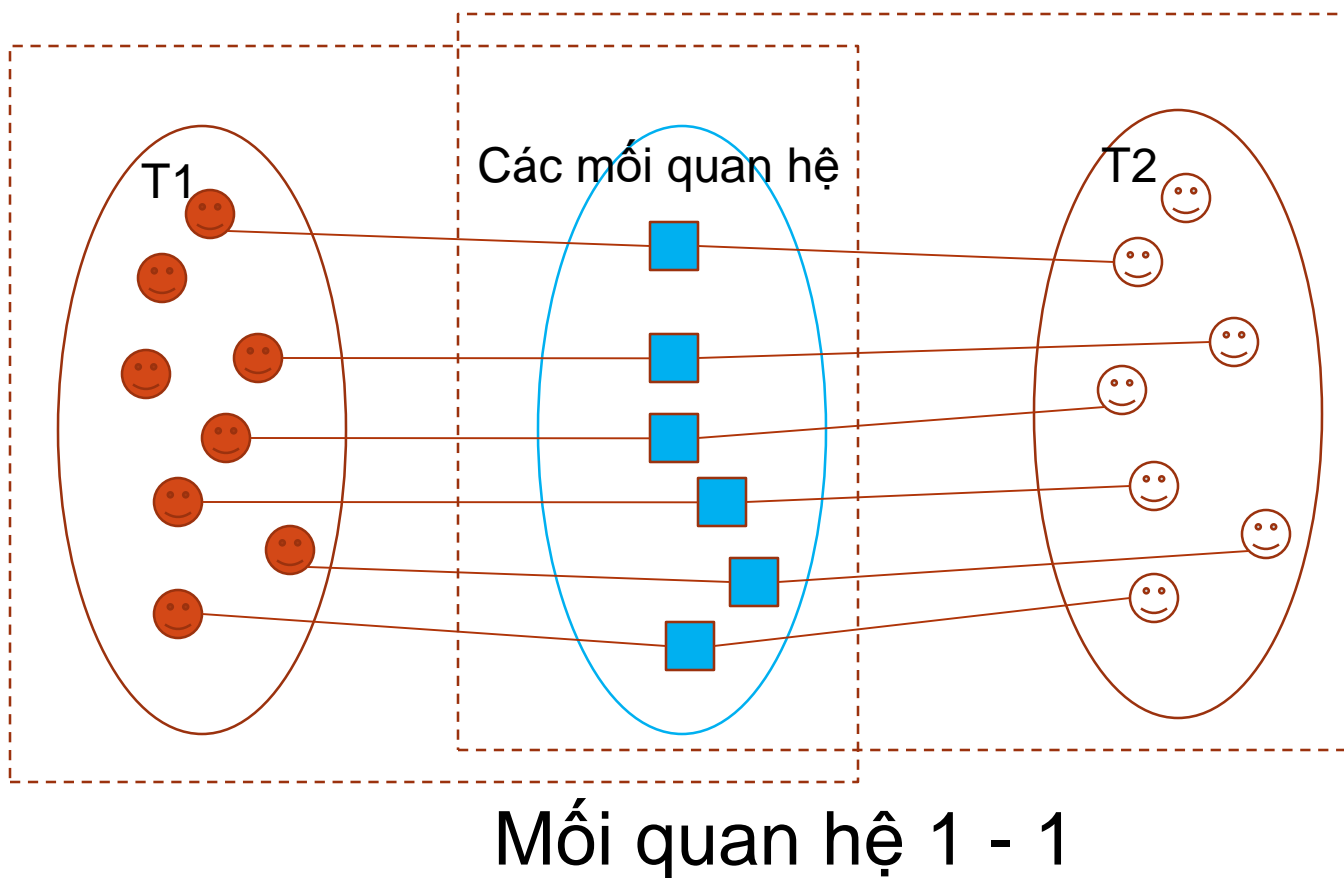
Domains and Data Types

- Data Type
 - This is counterpart of Domains in the Relational Model, which defines the smallest unit of data that can be stored
 - Columns always have a data type

HoTen	Specific information
Nhân Viên 1	
Nhân Viên 1	
Nhân Viên 2	
Nhân Viên 4	Data type: Char
Nhân Viên 5	
Nhân Viên 6	

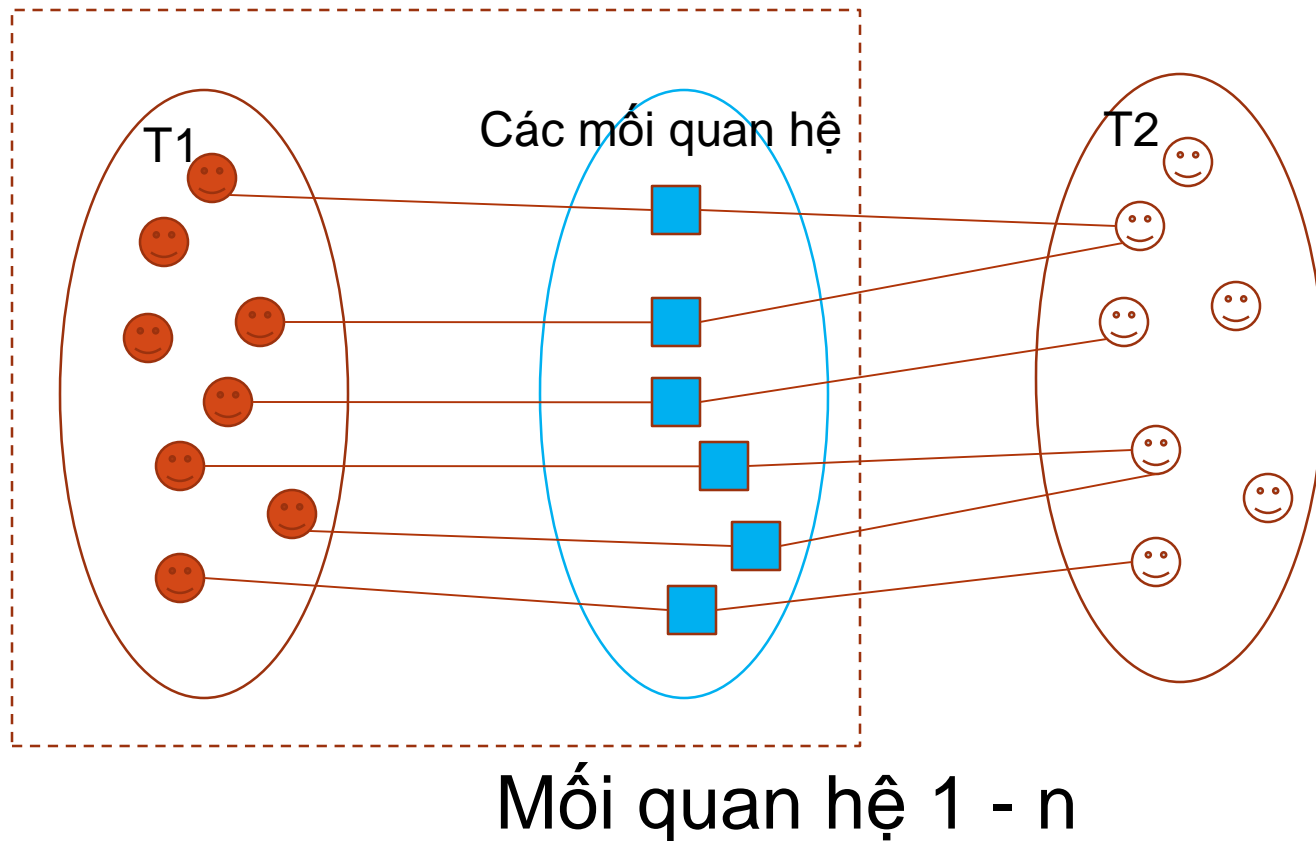
CSDL quan hệ phải mô tả:

- Các thực thể
- Các mối quan hệ giữa các thực thể thuộc một tập hoặc giữa các thực thể thuộc các tập khác nhau



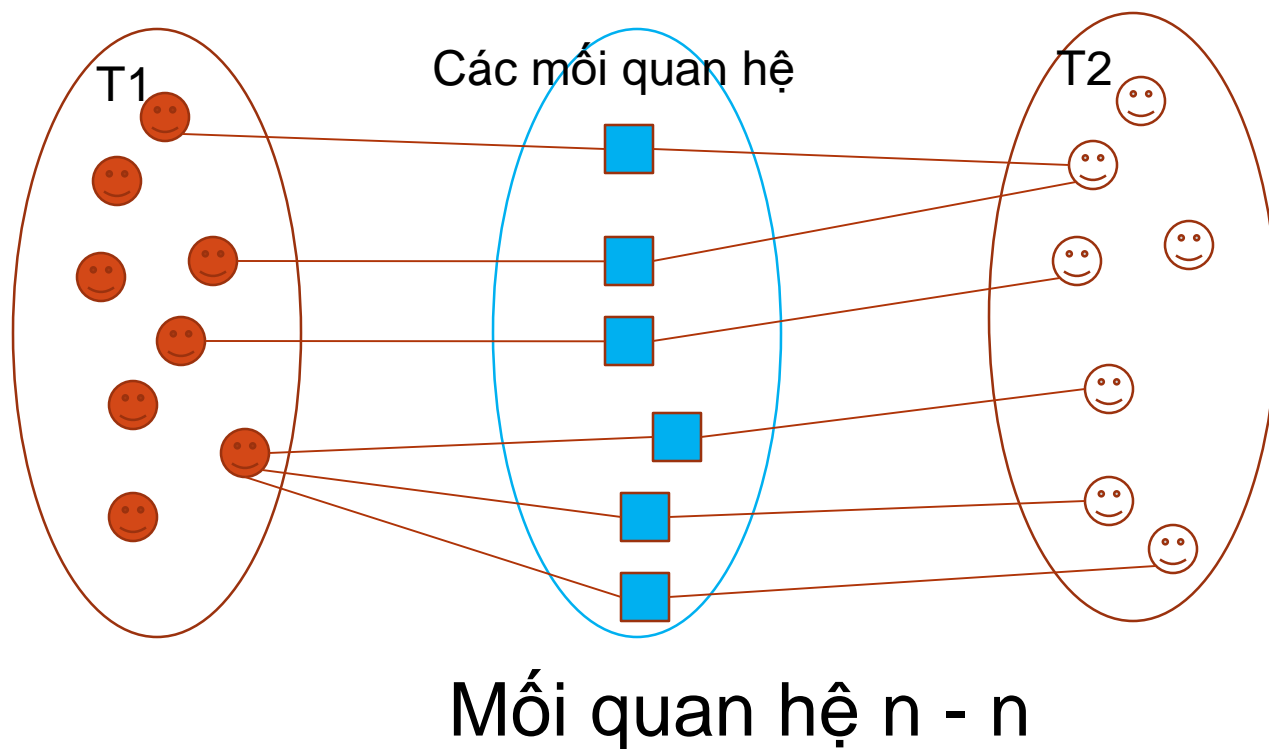
CSDL quan hệ phải mô tả:

- Các thực thể
- Các mối quan hệ giữa các thực thể thuộc một tập hoặc giữa các thực thể thuộc các tập khác nhau



CSDL quan hệ phải mô tả:

- Các thực thể
- Các mối quan hệ giữa các thực thể thuộc một tập hoặc giữa các thực thể thuộc các tập khác nhau



Ví dụ bảng dữ liệu – quan hệ

Bảng PHONG_BAN

<u>MaPhongBan</u>	TenPhongBan	DienThoai	DiaChi
P1	Phòng 1	DT 1	DC 1
P2	Phòng 2	DT 2	DC 2
P3	Phòng 3	DT 3	DC 3

Cột/Column

Bảng NHAN_VIEN

<u>MaNhanVien</u>	HoTen	NgaySinh	QueQuan
N1	Nhân Viên 1	NS 1	Q1
N2	Nhân Viên 1	NS 1	Q1
N3	Nhân Viên 2	NS 3	Q1
N4	Nhân Viên 4	NS 4	Q2
N5	Nhân Viên 5	NS 5	Q1
N6	Nhân Viên 6	NS 6	Q1

Bản ghi/
Record
Bộ/
Tuple

2.2. Mô hình dữ liệu quan hệ, sơ đồ DR

- Các tính chất của quan hệ:
 - Không quan trọng thứ tự các cột
 - Không quan trọng thứ tự các bản ghi
 - Mỗi trường chỉ chứa giá trị có tính nguyên tố
 - Các bản ghi là không trùng nhau

2.2. Mô hình dữ liệu quan hệ, sơ đồ DR

- Lược đồ quan hệ:
- NHAN_VIEN(*MaNhanVien, HoTen, NgaySinh, QueQuan)
- Khoá: Là tập hợp tối thiểu các thuộc tính cho phép phân biệt các bản ghi với nhau
- Khoá chính: Mỗi quan hệ có thể có nhiều khoá, khoá được chọn để sử dụng được gọi là khoá chính
 - Khoá chính sẽ là primary, các khoá còn lại sẽ là unique

2.3. Chuyển mô hình ER sang mô hình quan hệ bao gồm 7 bước sau:

- Bước 1: Mỗi một tập thực thể (thực thể đơn, không chứa bên trong nó các thực thể khác) được chuyển đổi thành một bảng. Tên của tập thực thể sẽ là tên của bảng.
- Bước 2: Mỗi một thuộc tính sẽ trở thành một cột (có thể) của bảng, cột có tên là tên (nhãn) của thuộc tính, và có thể có định dạng chính xác hơn. Những cột tương ứng với các thuộc tính bắt buộc, thì không thể nhận những giá trị NULL (nghĩa là bắt buộc phải có một giá trị xác định), ngược lại, những những cột tương ứng với các thuộc tính không bắt buộc thì có thể nhận giá trị NULL.

2.3. Chuyển sơ đồ ER sang sơ đồ CSDL quan hệ bao gồm 7 bước sau:

- Bước 3: Từ định danh độc nhất của tập thực thể trở thành khoá của bảng. Nếu tồn tại nhiều từ định danh độc nhất, thì từ định danh độc nhất nào thường sử dụng nhất (nhỏ nhất) sẽ được chọn trở thành khoá chính.
- Nếu trong thành phần của từ định danh độc nhất của tập thực thể có các liên kết, thì một bản sao (copy) của từ định danh độc nhất của tập thực thể đầu bên kia trong liên kết được thêm vào các cột thành phần của khoá chính. Quá trình này có thể lặp lại đệ quy. Tên của các đầu cuối của các liên kết và/hoặc tên của các thực thể được sử dụng để đặt tên cho các cột đó

2.3. Chuyển sơ đồ ER sang sơ đồ CSDL quan hệ bao gồm 7 bước sau:

- Bước 4: Các liên kết “nhiều-một” (và “một-nhiều”) trở thành các khoá ngoại. Nghĩa là tạo một bản sao của từ định danh độc nhất của liên kết “một”, và các cột tương ứng tạo thành các khoá ngoại. Các liên kết không bắt buộc tương ứng với các cột cho phép các giá trị không xác định (NULL), còn các liên kết bắt buộc – không cho phép.
- Bước 5: Tạo chỉ mục (index) cho các cột như khoá chính (chỉ mục độc nhất, unique index), khoá ngoại,...

2.3. Chuyển sơ đồ ER sang sơ đồ CSDL quan hệ bao gồm 7 bước sau:

- Bước 6 (trường hợp phức tạp): Nếu như trong mô hình ER có các kiểu con (sub-type), thì có thể chọn một trong hai phương pháp:
 - a) Các kiểu con cho vào một bảng. Tạo một bảng cho siêu kiểu (super type) ngoài cùng, còn cho các kiểu con có thể tạo các khung nhìn (view). Thêm ít nhất một cột vào bảng. Cột này chứa mã của kiểu, và trở thành một phần của khoá chính
 - b) Một bảng riêng biệt cho mỗi kiểu con. Tạo bảng cho mỗi kiểu con cấp độ thấp nhất. Các siêu kiểu được tái tạo bằng khung nhìn UNION - từ tất cả các bảng của các kiểu con chọn các cột chung là những cột của siêu kiểu.
 - Mỗi phương pháp có ưu, nhược điểm riêng.

2.3. Chuyển sơ đồ ER sang sơ đồ CSDL quan hệ bao gồm 7 bước sau:

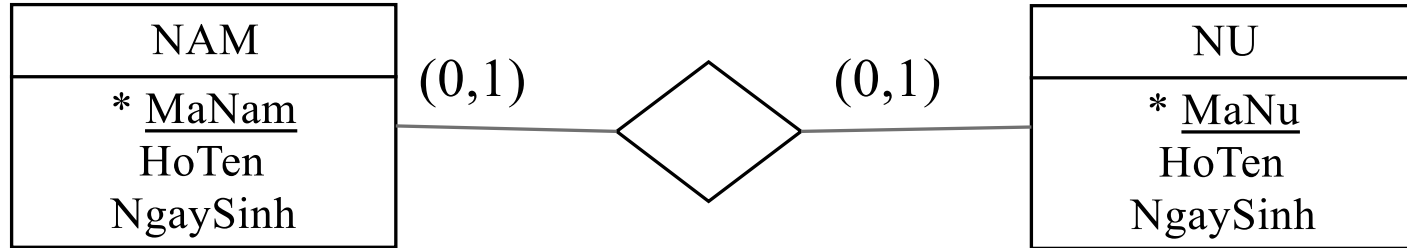
- Bước 7 (trường hợp phức tạp): Nếu như trong mô hình ER có các liên kết đặc biệt:
 - a) Miền giá trị chung. Nếu để tất cả các khoá ngoài vào một miền giá trị (có chung định dạng - format chung), thì tạo ra hai cột: Từ định danh liên kết và từ định danh tập thực thể. Cột từ định danh liên kết để phân biệt các liên kết. Cột từ định danh thực thể được dùng để lưu trữ các giá trị của từ định danh độc nhất của tập thực thể ở đầu cuối của liên kết tương ứng.
 - b) Các liên kết ngoài tường minh. Nếu như các khoá ngoài không thuộc một miền giá trị, thì tạo một cột khoá ngoài riêng lẻ cho mỗi liên kết. Các cột đó có thể chứa giá trị NULL.

2.3. Chuyển sơ đồ ER sang sơ đồ CSDL quan hệ bao gồm 7 bước sau:

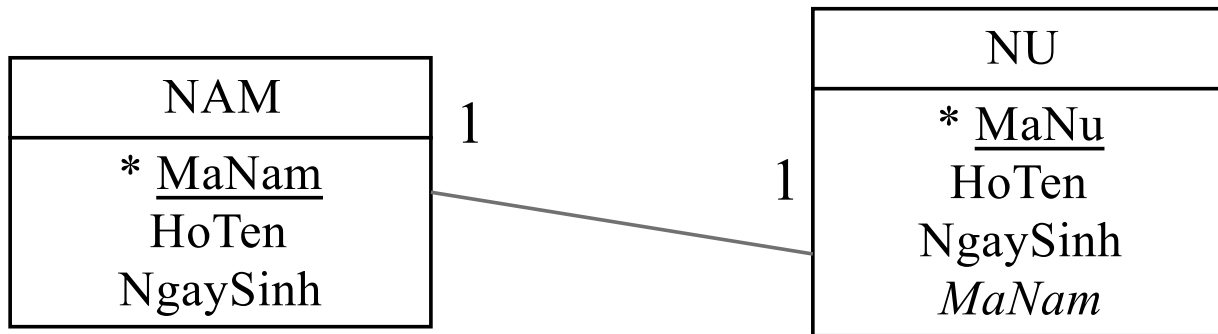
- Ví dụ, từ biểu đồ ER trước ta có các bảng trong mô hình quan hệ:
- PHONG_BAN (MãPhòngBan*, TênPhòngBan)
- NHAN_VIEN (MãNhânViên*, HọTên, MãPhòngBan, Ngày Sinh, QuêQuán, GiớiTính)
- BANG_LUONG (MãNhânViên*, ThángNăm*, Lương)
- (* = cột khoá)

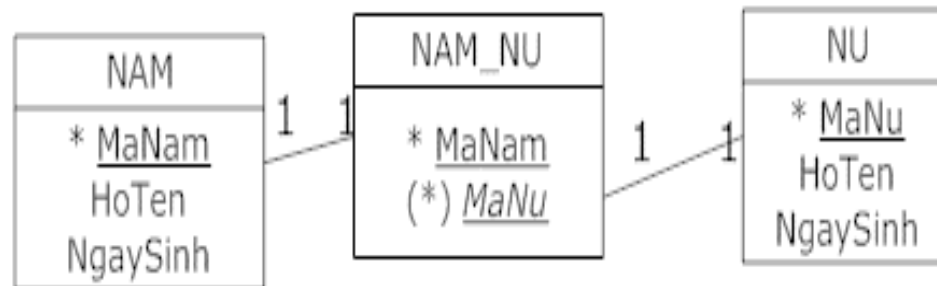
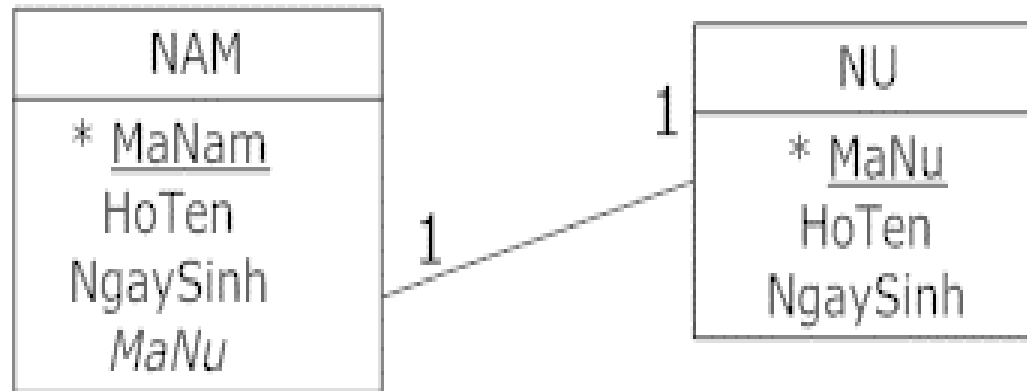
Chuyển sơ đồ TT-KH sang sơ đồ CSDL QH

- Một nam chỉ có thể là chồng (hoặc không) của một nữ, và ngược lại, một nữ chỉ có thể là vợ (hoặc không) của một nam

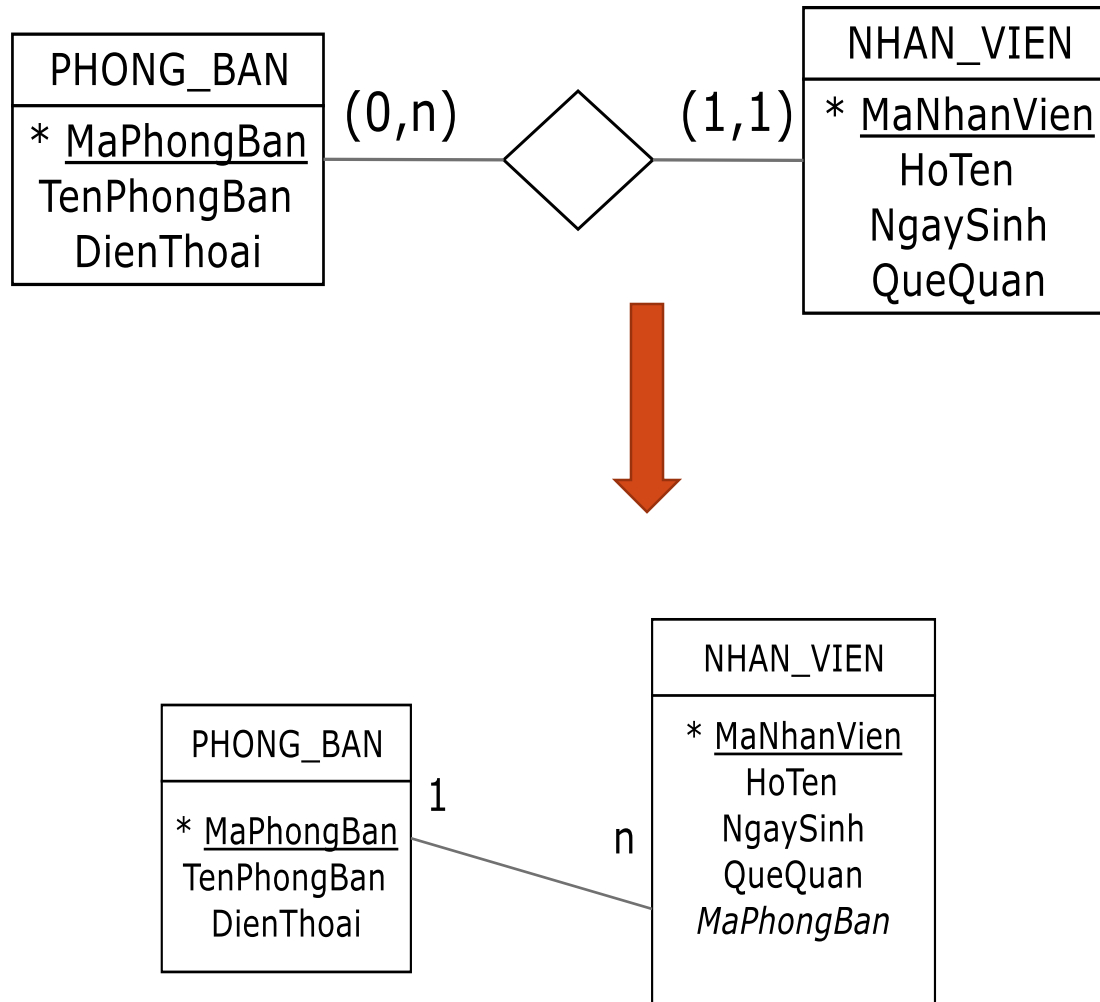


(1 trong 3 cách, chú ý giới hạn =1)





Mỗi phòng ban có thể có nhiều (hoặc không) nhân viên làm việc, ngược lại, mỗi nhân viên chỉ có thể làm việc ở một phòng ban duy nhất



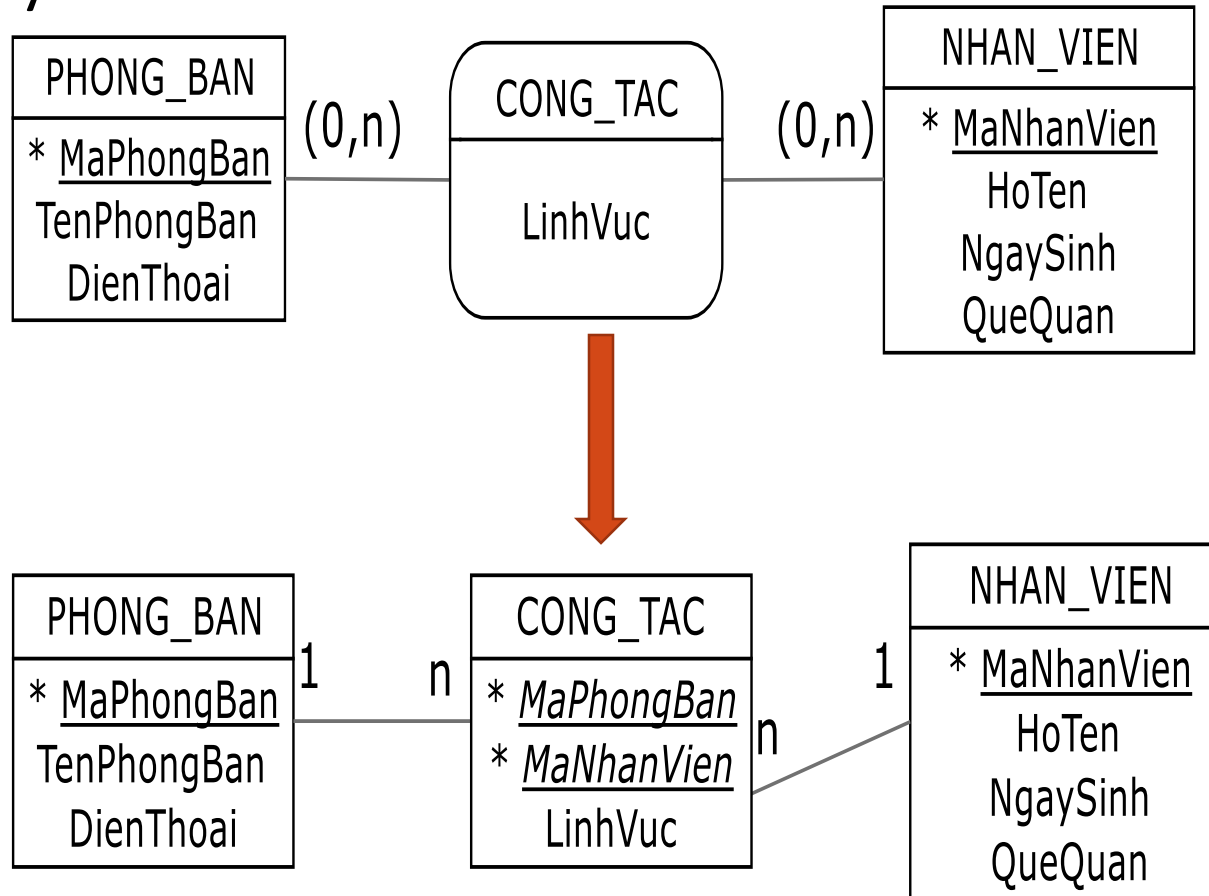
Bảng PHONG_BAN

MaPhongBan	TenPhongBan	DienThoai	DiaChi	MaNhanVien
P1	Phòng 1	DT 1	DC 1	N1
P2	Phòng 2	DT 2	DC 2	N2
P3	Phòng 3	DT 3	DC 3	N3

Bảng NHAN_VIEN

MaNhanVien	HoTen	NgaySinh	QueQuan	MaPhongBan
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

Mỗi phòng ban có thể có nhiều nhân viên làm việc, và mỗi nhân viên có thể cộng tác ở nhiều phòng ban khác nhau. Khi cộng tác ở một phòng ban, mỗi nhân viên được phân công một lĩnh vực công việc duy nhất



Bảng PHONG_BAN

<u>MaPhongBan</u>	TenPhongBan	DienThoai	DiaChi	MaNhanVien
P1	Phòng 1	DT 1	DC 1	N1
P2	Phòng 2	DT 2	DC 2	N2
P3	Phòng 3	DT 3	DC 3	N3

Bảng NHAN_VIEN

<u>MaNhanVien</u>	HoTen	NgaySinh	QueQuan	MaPhongBan
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

Bảng
PHONG_BAN

<u>MaPhongBan</u>	TenPhongBan	DienThoai	DiaChi
P1	Phòng 1	DT 1	DC 1
P2	Phòng 2	DT 2	DC 2
P3	Phòng 3	DT 3	DC 3

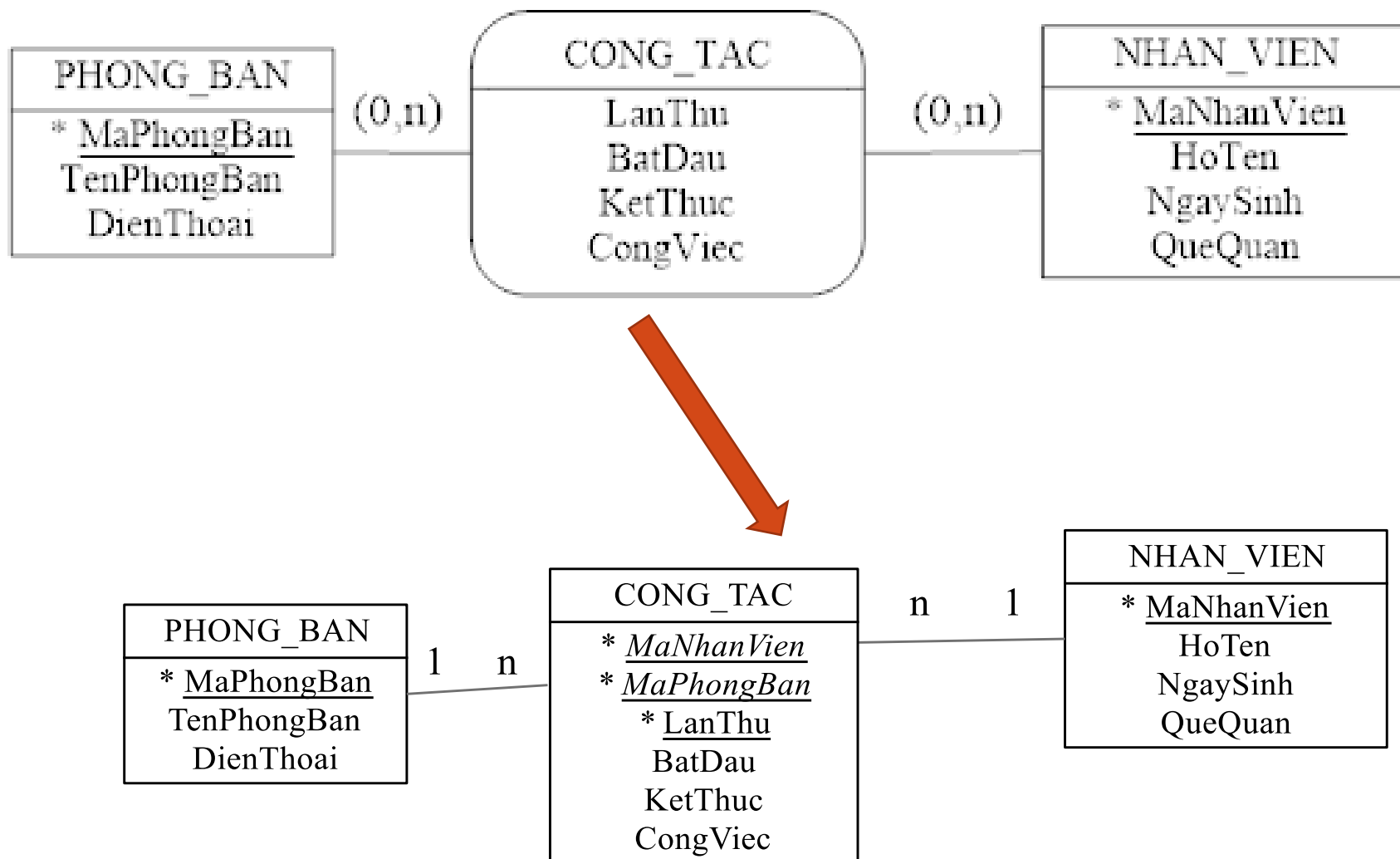
Bảng
NHAN_VIEN

<u>MaNhanVien</u>	HoTen	NgaySinh	QueQuan
N1	Nhân Viên 1	NS 1	Q1
N2	Nhân Viên 1	NS 1	Q1
N3	Nhân Viên 2	NS 3	Q2
N4	Nhân Viên 4	NS 4	Q2
N5	Nhân Viên 5	NS 5	Q2
N6	Nhân Viên 6	NS 6	Q1

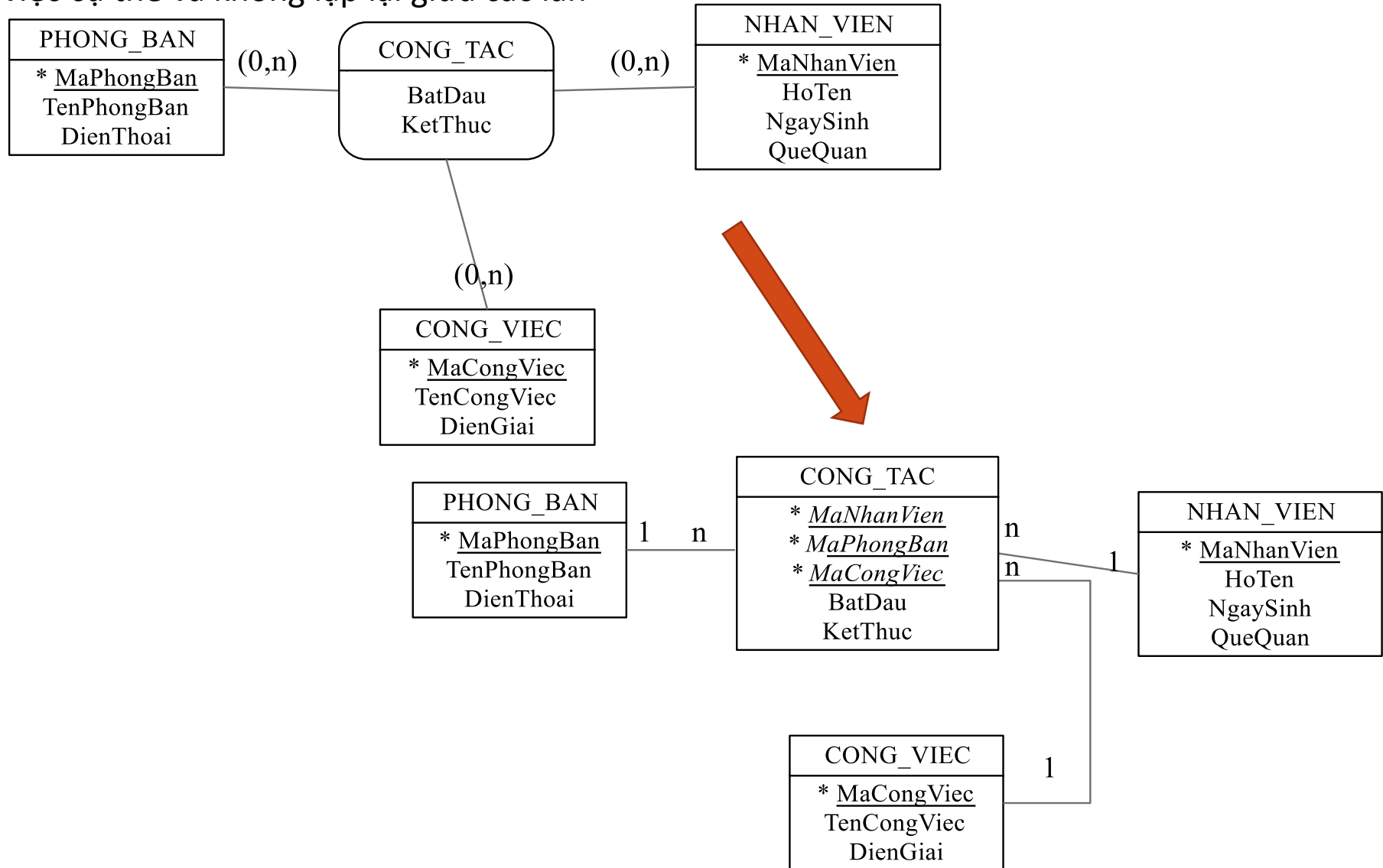
Bảng
LAM_VIEC

<u>MaNhanVien</u>	<u>MaPhongBan</u>	LinhVuc
N1	P1	Lĩnh vực 1
N1	P2	Lĩnh vực 2
N2	P1	Lĩnh vực 1
N3	P1	Lĩnh vực 1

Có nhiều phòng ban và nhiều nhân viên. Mỗi phòng ban có thể có nhiều nhân viên cộng tác, và mỗi nhân viên có thể cộng tác ở nhiều phòng ban khác nhau. Mỗi nhân viên có thể cộng tác ở một phòng ban nào đó nhiều lần, mỗi lần sẽ tham gia một công việc cụ thể



Có nhiều công việc, nhiều phòng ban và nhiều nhân viên. Mỗi phòng ban có thể có nhiều nhân viên làm việc, và mỗi nhân viên có thể cộng tác ở nhiều phòng ban khác nhau. Mỗi nhân viên có thể cộng tác ở một phòng ban nào đó nhiều lần, mỗi lần sẽ tham gia một công việc cụ thể và không lặp lại giữa các lần



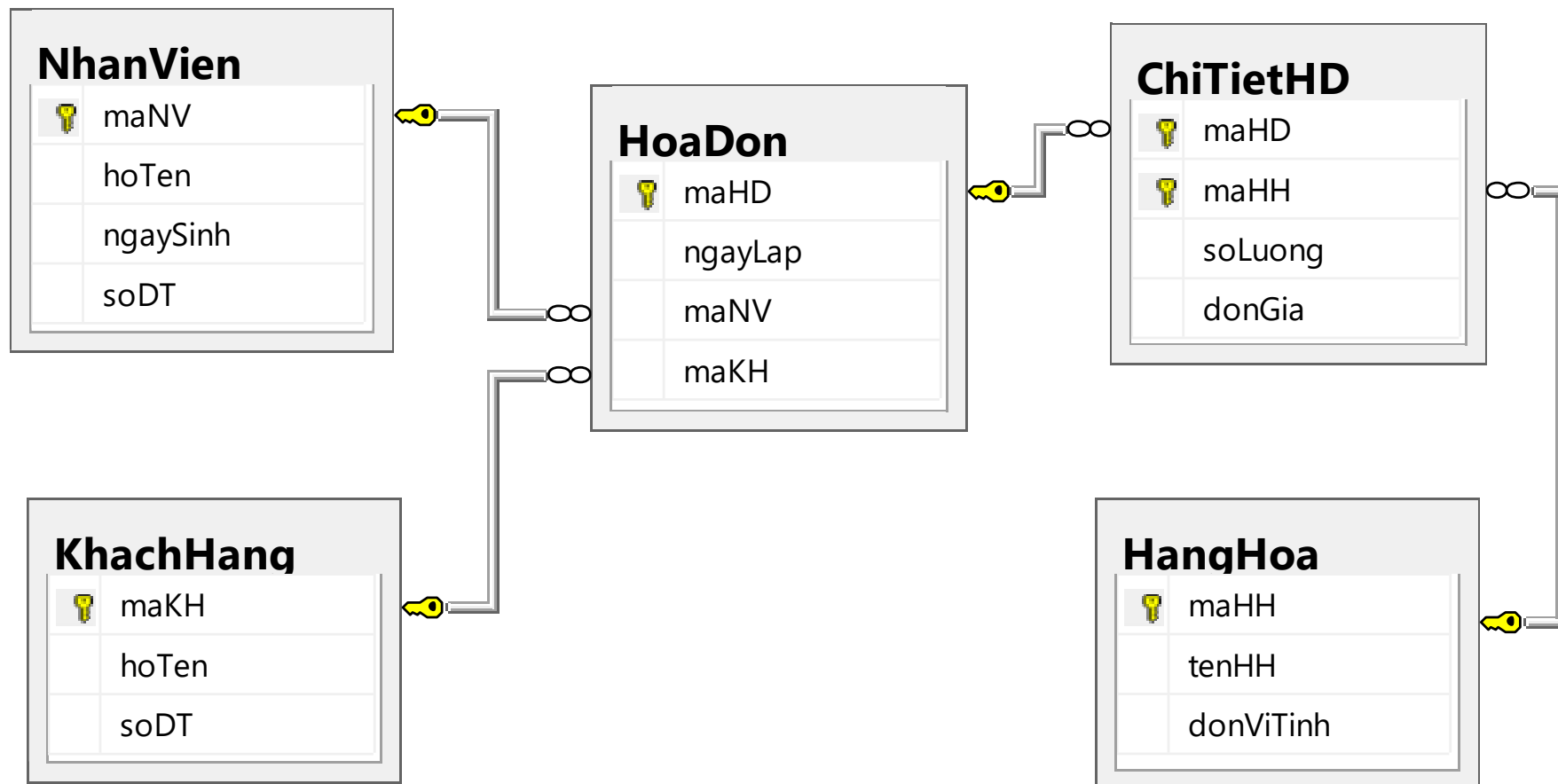
Bài tập – Bài toán quản lý hoá đơn

Có nhiều nhân viên, nhiều khách hàng, nhiều hàng hoá, nhiều hoá đơn. Mỗi nhân viên có thể lập nhiều hoá đơn, mỗi hoá đơn do một nhân viên lập. Mỗi khách hàng có thể có nhiều hoá đơn, mỗi hoá đơn thuộc về một khách hàng. Trong mỗi hoá đơn có thể có nhiều mặt hàng, mỗi mặt hàng có thể có mặt trong nhiều hoá đơn. Mỗi mặt hàng nếu có mặt trong một hoá đơn nào đó thì chỉ một lần duy nhất với số lượng và đơn giá cụ thể.

Học viên tự cho các thuộc tính tối thiểu.

Vẽ sơ đồ thực thể kết hợp và chuyển sang sơ đồ cơ sở dữ liệu quan hệ.

Bài tập – Bài toán quản lý hoá đơn



2.4 Ngôn ngữ SQL

2.4.1. DDL - data definition language

- Create database
- Create table
- Create index
- Create view
- Create procedure, function, trigger...
- Alter, drop
- Cho các bảng:
 - NhanVien(MaNV, HoTen, NgaySinh, QueQuan, BacLuong MaPB)
 - Luong(MaNV, ThangNam, SoTien)

2.4.1. DDL: Create table (VD1)

- **CREATE TABLE NhanVien(
MaNV varchar(10) NOT NULL PRIMARY KEY,
HoTen nvarchar(50) NOT NULL,
NgaySinh date,
QueQuan nvarchar(200),
BacLuong int,
MaPB varchar(10) NOT NULL)**

2.4.1. DDL: Create table (VD2)

- CREATE TABLE Luong(
MaNV varchar(10) NOT NULL,
ThangNam varchar(7) NOT NULL,
SoTien bigint NOT NULL,
PRIMARY KEY (MaNV, ThangNam))

2.4 Ngôn ngữ SQL

- 2.4.2. DML - data manipulation language
- Insert
- Update
- Delete
- Truncate....

2.4.2. DML: INSERT

- NhanVien(MaNV, HoTen, QueQuan, BacLuong, MaPB)
- Thêm mới bản ghi bao gồm các giá trị đầy đủ cho tất cả các trường:
- INSERT INTO NhanVien VALUES('NV1', N'Nguyễn Văn A', N'Đà Nẵng', 3, 'PB1')
- Thêm mới bản ghi không bao gồm các giá trị cho một số trường cho phép NULL hoặc thứ tự các giá trị khác thứ tự các cột:
- INSERT INTO NhanVien(MaNV, HoTen, BacLuong, MaPB) VALUES('NV1', N'Nguyễn Văn A', 3, 'PB1')
- INSERT INTO NhanVien(MaNV, HoTen, BacLuong, MaPB, QueQuan) VALUES('NV1', N'Nguyễn Văn A', 3, 'PB1', NULL)

2.4.2. DML: UPDATE

- Thiết lập bậc lương 4 cho nhân viên có mã NV1
- UPDATE NhanVien SET BacLuong = 4 WHERE MaNV = 'NV1'
- Thiết lập bậc lương 4, quê quán ABC cho nhân viên có mã NV1
- UPDATE NhanVien SET BacLuong = 4, QueQuan = N'ABC' WHERE MaNV = 'NV1'

2.4.2. DML: UPDATE

- Tăng 15% số tiền trong tháng 12/2011 cho các nhân viên thuộc phòng ban có mã P1
- UPDATE Luong SET SoTien = SoTien*1,15 WHERE MaNV IN (SELECT MaNV FROM NhanVien WHERE MaPB = 'P1') AND ThangNam = '12/2011'
- SQL Server: UPDATE Luong SET SoTien = SoTien*1,15 FROM NhanVien JOIN Luong ON NhanVien.MaNV = Luong.MaNV WHERE MaPB = 'P1' AND ThangNam = '12/2011'

2.4.2. DML: DELETE

- Xoá nhân viên có mã NV1 từ DS nhân viên
- `DELETE FROM NhanVien WHERE MaNV = 'NV1'`

2.4.2. DML: DELETE

- Xoá thông tin nhận lương trong tháng 12/2011 của các nhân viên thuộc phòng ban có mã P1
- `DELETE FROM Luong WHERE MaNV IN (SELECT MaNV FROM NhanVien WHERE MaPB = 'P1') AND ThangNam = '12/2011'`
- `DELETE FROM Luong FROM NhanVien JOIN Luong ON NhanVien.MaNV = Luong.MaNV WHERE MaPB = 'P1' AND ThangNam = '12/2011'`

2.4 Ngôn ngữ SQL

- 2.4.3. DQL - data Query language

SELECT (TOP, DISTINCT...)

FROM (...JOIN...)

WHERE

GROUP BY

HAVING

ORDER BY

- Thứ tự thực thi:

WHERE_ko_JOIN → JOIN → WHERE -> GROUP BY ->
(HAVING → SELECT → ORDER BY)/(SELECT →
ORDER BY → TOP)

2.4 Ngôn ngữ SQL

2.4.4. D(A)CL - data (ACCESS) CONTROL language

- Quyền: csdl, hqt csdl
- Quyền thao tác: Insert, update, delete, select
- Quyền thực thi: execute

- Phân quyền: GRANT quyền ON đối_tượng TO người_dùng [with grant option]
- Thu hồi quyền: REVOKE quyền ON đối_tượng FROM người_dùng

2.4 DCL: GRANT

- Phân quyền: GRANT quyền ON đối_tượng TO người_dùng [with grant option]
- Cấp tất cả các quyền thao tác trên bảng NhanVien cho người dùng UserA: GRANT INSERT, UPDATE, DELETE, SELECT ON NhanVien TO UserA

GRANT ALL ON NhanVien TO UserA

- Nếu muốn UserA có quyền cấp cho người khác: GRANT ALL ON NhanVien TO UserA WITH GRANT OPTION

GRANT INSERT, SELECT ON NhanVien TO UserA

2.4 DCL: REVOKE

- Thu hồi quyền: REVOKE quyền ON đối_tượng FROM người_dùng

REVOKE INSERT, SELECT ON NhanVien FROM UserA

Truy vấn – Query

Cho bảng NHANVIEN:

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

Cho bảng PHONGBAN:

<u>MaPhongBan</u>	TenPhongBan	DienThoai	DiaChi
P1	Phòng 1	DT 1	DC 1
P2	Phòng 2	DT 2	DC 2
P3	Phòng 3	DT 3	DC 3

Truy vấn – Query

VD1. Đưa ra danh sách tất cả nhân viên làm việc ở phòng ban P1

```
SQL: SELECT MaNV, HoTen, NgaySinh, QueQuan  
      FROM NHANVIEN WHERE MaPB = 'P1'
```

Truy vấn – Query

VD1. Đưa ra danh sách tất cả nhân viên làm việc ở phòng ban P1

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

```
SQL: SELECT MaNV, HoTen, NgaySinh, QueQuan  
      FROM NHANVIEN WHERE MaPB = 'P1'
```

Truy vấn – Query

VD1. Đưa ra danh sách tất cả nhân viên làm việc ở phòng ban P1

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

```
SQL: SELECT MaNV, HoTen, NgaySinh, QueQuan  
      FROM NHANVIEN WHERE MaPB = 'P1'
```



Kết quả:

MaNV	HoTen	NgaySinh	QueQuan
N1	Nhân Viên 1	NS 1	Q1
N2	Nhân Viên 1	NS 1	Q1
N4	Nhân Viên 4	NS 4	Q2
N6	Nhân Viên 6	NS 6	Q1

Truy vấn – Query

VD2. Đưa ra danh sách tất cả nhân viên, sắp xếp kết quả tăng dần theo MaPB và QueQuan

```
SQL: SELECT MaNV, HoTen, NgaySinh, QueQuan, MaPB  
      FROM NHANVIEN
```

```
ORDER BY MaPB, QueQuan
```


Truy vấn – Query

VD2. Đưa ra danh sách tất cả nhân viên, sắp xếp kết quả tăng dần theo MaPB và QueQuan

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

```
SQL: SELECT MaNV, HoTen, NgaySinh, QueQuan, MaPB  
      FROM NHANVIEN
```

```
ORDER BY MaPB, QueQuan
```

Truy vấn – Query

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

B1. Sắp xếp các bản ghi tăng dần theo MaPB

MaNV	HoTen	NgaySinh	QueQuan	MaPB	
N1	Nhân Viên 1	NS 1	Q1	P1	Nhóm 1
N2	Nhân Viên 1	NS 1	Q1	P1	
N4	Nhân Viên 4	NS 4	Q2	P1	
N6	Nhân Viên 6	NS 6	Q1	P1	
N3	Nhân Viên 2	NS 3	Q1	P2	Nhóm 2
N5	Nhân Viên 5	NS 5	Q1	P2	

Truy vấn – Query

B1. Sắp xếp các bản ghi tăng dần theo MaPB

MaNV	HoTen	NgaySinh	QueQuan	MaPB	
N1	Nhân Viên 1	NS 1	Q1	P1	Nhóm 1
N2	Nhân Viên 1	NS 1	Q1	P1	
N4	Nhân Viên 4	NS 4	Q2	P1	
N6	Nhân Viên 6	NS 6	Q1	P1	
N3	Nhân Viên 2	NS 3	Q1	P2	Nhóm 2
N5	Nhân Viên 5	NS 5	Q1	P2	

B2. Sắp xếp các bản ghi tăng dần theo QueQuan dựa trên kết quả của bước 1

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N6	Nhân Viên 6	NS 6	Q1	P1
N4	Nhân Viên 4	NS 4	Q2	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N5	Nhân Viên 5	NS 5	Q1	P2

Truy vấn – Query

VD2b. Đưa ra danh sách nhân viên có quê quán Q1, sắp xếp kết quả giảm dần theo MaPB và tăng dần theo họ tên

```
SQL: SELECT MaNV, HoTen, NgaySinh, MaPB  
      FROM NHANVIEN  
      WHERE QueQuan = N'Q1'  
      ORDER BY MaPB DESC, HoTen
```

MaNV	HoTen	NgaySinh	MaPB
N3	Nhân Viên 2	NS 3	P2
N5	Nhân Viên 5	NS 5	P2
N1	Nhân Viên 1	NS 1	P1
N2	Nhân Viên 1	NS 1	P1
N6	Nhân Viên 6	NS 6	P1

Truy vấn – Query

VD3. Đếm số nhân viên làm việc trong mỗi phòng ban

```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN GROUP BY MaPB
```

Truy vấn – Query

VD3. Đếm số nhân viên làm việc trong mỗi phòng ban

```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV  
FROM NHANVIEN GROUP BY MaPB
```

Bước 1:
Nhóm
theo
giá
trị
MaPB

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N4	Nhân Viên 4	NS 4	Q2	P1
N6	Nhân Viên 6	NS 6	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N5	Nhân Viên 5	NS 5	Q1	P2

Nhóm 1

Nhóm 2

MaNV	MaPB
N1	P1
N2	P1
N4	P1
N6	P1
N3	P2
N5	P2

Bước 2: Đếm cho mỗi nhóm

MaPB	SoNV
P1	4
P2	2

Truy vấn – Query

VD4. Đếm số nhân viên làm việc trong mỗi phòng ban theo quê quán

```
SQL: SELECT MaPB, QueQuan, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN GROUP BY MaPB, QueQuan
```

Truy vấn – Query

VD4. Đếm số nhân viên làm việc trong mỗi phòng ban theo quê quán

```
SQL: SELECT MaPB, QueQuan, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN GROUP BY MaPB, QueQuan
```

Bước 1: Nhóm theo giá trị MaPB

MaNV	QueQuan	MaPB	Nhóm 1
N1	Q1	P1	
N2	Q1	P1	
N4	Q2	P1	
N6	Q1	P1	Nhóm 2
N3	Q1	P2	
N5	Q1	P2	

Truy vấn – Query

VD4. Đếm số nhân viên làm việc trong mỗi phòng ban theo quê quán

```
SQL: SELECT MaPB, QueQuan, COUNT(MaNV) AS SoNV
      FROM NHANVIEN GROUP BY MaPB, QueQuan
```

Bước 1: Nhóm theo giá trị MaPB

MaNV	QueQuan	MaPB
N1	Q1	P1
N2	Q1	P1
N4	Q2	P1
N6	Q1	P1
N3	Q1	P2
N5	Q1	P2

Nhóm 1

Nhóm 2

Bước 2: Nhóm theo giá trị QueQuan cho mỗi nhóm ở kết quả của bước 1

MaNV	QueQuan	MaPB
N1	Q1	P1
N2	Q1	P1
N6	Q1	P1
N4	Q2	P1
N3	Q1	P2
N5	Q1	P2

Bước 3: Đếm cho mỗi nhóm

MaPB	QueQuan	SoNV
P1	Q1	3
P1	Q2	1
P2	Q1	2

Truy vấn – Query

VD5. Đếm số nhân viên có quê quán Q1 làm việc trong mỗi phòng ban

```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN WHERE QueQuan = 'Q1' GROUP BY MaPB
```

Truy vấn – Query

VD5. Đếm số nhân viên có quê quán Q1 làm việc trong mỗi phòng ban

```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN WHERE QueQuan = 'Q1' GROUP BY MaPB
```

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

Bước 1: Áp dụng điều kiện QueQuan = 'Q1'

MaNV	QueQuan	MaPB
N1	Q1	P1
N2	Q1	P1
N3	Q1	P2
N5	Q1	P2
N6	Q1	P1

Truy vấn – Query

VD5. Đếm số nhân viên có quê quán Q1 làm việc trong mỗi phòng ban


```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN WHERE QueQuan = 'Q1' GROUP BY MaPB
```

Bước 1: Áp dụng điều kiện QueQuan = 'Q1'

MaNV	QueQuan	MaPB
N1	Q1	P1
N2	Q1	P1
N3	Q1	P2
N5	Q1	P2
N6	Q1	P1

Bước 2: Nhóm theo MaPB

MaNV	MaPB
N1	P1
N2	P1
N6	P1
N3	P2
N5	P2



Bước 3: Đếm cho mỗi nhóm

MaPB	SoNV
P1	3
P2	2

Truy vấn – Query

VD6. Đếm số nhân viên trong mỗi phòng ban. Có đưa ra thông tin phòng ban.

```
SQL: SELECT n.MaPB, TenPhongBan, DienThoai, DiaChi, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN n JOIN PHONGBAN p ON n.MaPB = p.MaPhongBan  
      GROUP BY n.MaPB, TenPhongBan, DienThoai, DiaChi
```

<u>MaPhongBan</u>	TenPhongBan	DienThoai	DiaChi
P1	Phòng 1	DT 1	DC 1
P2	Phòng 2	DT 2	DC 2
P3	Phòng 3	DT 3	DC 3

Truy vấn – Query

VD6. Đếm số nhân viên trong mỗi phòng ban. Có đưa ra thông tin phòng ban.

```
SQL: SELECT n.MaPB, TenPhongBan, DienThoai, DiaChi, COUNT(MaNv) AS SoNV
      FROM NHANVIEN n JOIN PHONGBAN p ON n.MaPB = p.MaPhongBan
      GROUP BY n.MaPB, TenPhongBan, DienThoai, DiaChi
```

<u>MaPhongBan</u>	TenPhongBan	DienThoai	DiaChi
P1	Phòng 1	DT 1	DC 1
P2	Phòng 2	DT 2	DC 2
P3	Phòng 3	DT 3	DC 3

B1. Nối 2 bảng NHAN VIEN và PHONG BAN

<u>MaNV</u>	<u>MaPB</u>	TenPhongBan	DienThoai	DiaChi
N1	P1	Phòng 1	DT 1	DC 1
N2	P1	Phòng 1	DT 1	DC 1
N3	P2	Phòng 2	DT 2	DC 2
N4	P1	Phòng 1	DT 1	DC 1
N5	P2	Phòng 2	DT 2	DC 2
N6	P1	Phòng 1	DT 1	DC 1

Truy vấn – Query

VD6. Đếm số nhân viên trong mỗi phòng ban. Có đưa ra thông tin phòng ban.

B2, 3, 4, 5. Nhóm theo MaPhongBan, TenPhongBan, DienThoai, DiaChi dựa trên kết quả bước 1:

<u>MaNV</u>	<u>MaPB</u>	TenPhongBan	DienThoai	DiaChi
N1	P1	Phòng 1	DT 1	DC 1
N2	P1	Phòng 1	DT 1	DC 1
N4	P1	Phòng 1	DT 1	DC 1
N6	P1	Phòng 1	DT 1	DC 1
N3	P2	Phòng 2	DT 2	DC 2
N5	P2	Phòng 2	DT 2	DC 2

B6. Đếm cho mỗi nhóm:

<u>MaPB</u>	TenPhongBan	DienThoai	DiaChi	SoNV
P1	Phòng 1	DT 1	DC 1	4
P2	Phòng 2	DT 2	DC 2	2

Truy vấn – Query

VD6-Cách 2. Đếm số nhân viên trong mỗi phòng ban. Có đưa ra thông tin phòng ban.

Bước 1:Nhóm theo giá trị MaPB

MaNV	MaPB
N1	P1
N2	P1
N4	P1
N6	P1
N3	P2
N5	P2

Bước 2: Đếm cho mỗi nhóm

MaPB	SoNV
P1	4
P2	2

B3. Nối bảng kết quả của bước 2 với PHONG_BAN:

<u>MaPhongBan</u>	TenPhongBan	DienThoai	DiaChi	SoNV
P1	Phòng 1	DT 1	DC 1	4
P2	Phòng 2	DT 2	DC 2	2

Truy vấn – Query

VD7. Đưa ra danh sách các phòng ban có nhiều hơn 3 nhân viên

```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN GROUP BY MaPB HAVING COUNT(MaNv) > 3
```

Truy vấn – Query

VD7. Đưa ra danh sách các phòng ban có nhiều hơn 3 nhân viên

```
SQL: SELECT MaPB, COUNT(MaNV) AS SoNV  
FROM NHANVIEN GROUP BY MaPB HAVING COUNT(MaNV) > 3
```

Bước 1:
Nhóm
theo
giá
trị
MaPB

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N4	Nhân Viên 4	NS 4	Q2	P1
N6	Nhân Viên 6	NS 6	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N5	Nhân Viên 5	NS 5	Q1	P2

Nhóm 1

Nhóm 2

MaNV	MaPB
N1	P1
N2	P1
N4	P1
N6	P1
N3	P2
N5	P2

Bước 2: Đếm cho mỗi nhóm

MaPB	SoNV
P1	4
P2	2

Truy vấn – Query

VD7. Đưa ra danh sách các phòng ban có nhiều hơn 3 nhân viên

```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV  
      FROM NHANVIEN GROUP BY MaPB HAVING COUNT(MaNv) > 3
```

Bước 2: Đếm cho mỗi nhóm

MaPB	SoNV
P1	4
P2	2

Bước 3: Áp dụng điều kiện COUNT(MaNv) > 3

MaPB	SoNV
P1	4

Truy vấn – Query

VD8. Đưa ra danh sách các phòng ban có nhiều hơn 2 nhân viên có quê quán Q1 làm việc

```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV FROM NHANVIEN  
      WHERE QueQuan = 'Q1' GROUP BY MaPB HAVING COUNT(MaNv) > 2
```

Truy vấn – Query

VD8. Đưa ra danh sách các phòng ban có nhiều hơn 2 nhân viên có quê quán Q1 làm việc


```
SQL: SELECT MaPB, COUNT(MaNv) AS SoNV FROM NHANVIEN  
      WHERE QueQuan = 'Q1' GROUP BY MaPB HAVING COUNT(MaNv) > 2
```

Bước 1: Áp dụng điều kiện QueQuan = 'Q1'

MaNV	QueQuan	MaPB
N1	Q1	P1
N2	Q1	P1
N3	Q1	P2
N5	Q1	P2
N6	Q1	P1

Bước 2: Nhóm theo MaPB

MaNV	MaPB
N1	P1
N2	P1
N6	P1
N3	P2
N5	P2



Bước 3: Đếm cho mỗi nhóm

MaPB	SoNV
P1	3
P2	2

Truy vấn – Query

VD8. Đưa ra danh sách các phòng ban có nhiều hơn 2 nhân viên có quê quán Q1 làm việc

Bước 3: Đếm cho mỗi nhóm

MaPB	SoNV
P1	3
P2	2

Bước 4: Áp dụng điều kiện $\text{COUNT}(\text{MaNV}) > 2$

MaPB	SoNV
P1	3

Truy vấn – Query

VD9. Tính điểm trung bình của mỗi sinh viên

Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

MaLopTC	MaSV	DiemTK
LTC1	SV1	6
LTC3	SV1	7
LTC2	SV2	8
LTC3	SV2	9
LTC2	SV1	8

Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB

FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH INNER JOIN
BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC

GROUP BY MaSV

Truy vấn – Query

VD9. Tính điểm trung bình của mỗi sinh viên

Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

MaLopTC	MaSV	DiemTK
LTC1	SV1	6
LTC3	SV1	7
LTC2	SV2	8
LTC3	SV2	9
LTC2	SV1	8

Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB

FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH INNER JOIN
BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC

GROUP BY MaSV

Truy vấn – Query

VD9. Tính điểm trung bình của mỗi sinh viên

Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

MaLopTC	MaSV	DiemTK
LTC1	SV1	6
LTC3	SV1	7
LTC2	SV2	8
LTC3	SV2	9
LTC2	SV1	8

Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB

FROM MON_HOC AS m, LOP_TIN_CHI AS l, BANG_DIEM AS d WHERE m.MaMH = l.MaMH
AND l.MaLopTC = d.MaLopTC

GROUP BY MaSV

Truy vấn – Query

B1. **MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH**

I.MaLopTC	I.MaMH	m.SoTC
LTC1	M1	2
LTC3	M2	3
LTC2	M2	3

B2. **MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC**

I.MaLopTC	d.MaSV	d.DiemTK	m.SoTC
LTC1	SV1	6	2
LTC3	SV1	7	3
LTC2	SV2	8	3
LTC3	SV2	9	3
LTC2	SV1	8	3

Truy vấn – Query

B3. GROUP BY d.MaSV (trên cơ sở kết quả của B2)

MaSV	DiemTK	SoTC
SV1	6	2
SV1	7	3
SV1	8	3
SV2	8	3
SV2	9	3

B4. $\text{SUM}(d.\text{DiemTK} * m.\text{SoTC}) / \text{SUM}(m.\text{SoTC})$ cho mỗi nhóm

MaSV	DiemTB
SV1	7.1
SV2	8.5

Truy vấn – Query

VD10. Tính điểm trung bình của mỗi sinh viên thuộc lớp sinh hoạt LSH1

Có các bảng: MON_HOC, LOP_SH, SINH_VIEN, LOP_TIN_CHI, BANG_DIEM

MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3

maLopSH	tenLopSH
L1		LSH1
L2		LSH2

MaSV	maLopSH
SV1		L1
SV3		L2
SV2		L1

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

MaLopTC	MaSV	DiemTK
LTC1	SV1	6
LTC2	SV3	9
LTC3	SV1	7
LTC2	SV2	8
LTC3	SV2	9
LTC2	SV1	8

Truy vấn – Query

VD10. Tính điểm trung bình của mỗi sinh viên thuộc lớp sinh hoạt LSH1

Có các bảng: MON_HOC, LOP_SH, SINH_VIEN, LOP_TIN_CHI, BANG_DIEM

Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB

FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH INNER JOIN
BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC INNER JOIN SINH_VIEN AS s ON d.MaSV =
s.MaSV INNER JOIN LOP_SH AS lsh ON s.MaLopSH = lsh.MaLopSH

WHERE tenLopSH = 'LSH1'

GROUP BY MaSV

Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB

FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH INNER JOIN
BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC INNER JOIN SINH_VIEN AS s ON d.MaSV =
s.MaSV INNER JOIN LOP_SH AS lsh ON s.MaLopSH = lsh.MaLopSH

GROUP BY MaSV

HAVING tenLopSH = 'LSH1'

Truy vấn – Query

VD11. Đưa ra danh sách các sinh viên có điểm trung bình ≥ 8.5

Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

```
Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB  
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH  
INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC  
GROUP BY MaSV  
  
HAVING SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC)  $\geq 8.5$ 
```

Truy vấn – Query

VD11. Đưa ra danh sách các sinh viên có điểm trung bình ≥ 8.5

Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

Truy vấn – Query

VD11. Đưa ra danh sách các sinh viên có điểm trung bình ≥ 8.5
Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH
INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC
GROUP BY MaSV

HAVING SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) ≥ 8.5

B5. Áp dụng điều kiện $\text{SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC)} \geq 8.5$ cho bảng kết quả của bước 4

MaSV	DiemTB
SV1	7.1
SV2	8.5



MaSV	DiemTB
SV2	8.5

Truy vấn – Query

VD12. Đưa ra danh sách các sinh viên lớp L1 có điểm trung bình học kỳ 1 năm học 2014-2015 ≥ 8.5

Có 4 bảng: SINH_VIEN, LOP_TIN_CHI, MON_HOC và BANG_DIEM

Truy vấn – Query

VD12. Đưa ra danh sách các sinh viên lớp L1 có điểm trung bình học kỳ 1 năm học 2014-2015 ≥ 8.5

Có 4 bảng: SINH_VIEN, LOP_TIN_CHI, MON_HOC và BANG_DIEM

```
Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH
INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC INNER JOIN
SINH_VIEN AS s ON d.MaSV = s.MaSV
WHERE l.hocKy = 1 AND l.namHoc = '2014-2015' AND s.maLopSH = 'L1'
GROUP BY MaSV
HAVING SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC)  $\geq 8.5$ 
```

```
Truy vấn: SELECT d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH
INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC
INNER JOIN SINH_VIEN AS s ON d.MaSV = s.MaSV
WHERE l.hocKy = 1 AND l.namHoc = '2014-2015'
GROUP BY MaSV, s.MaLopSH
HAVING SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC)  $\geq 8.5$  AND s.maLopSH = 'L1'
```

Truy vấn – Query

VD13. Đưa ra danh sách n = 2 sinh viên có điểm trung bình cao nhất
Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

```
Truy vấn: SELECT TOP 2 d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB  
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH  
INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC  
GROUP BY MaSV  
  
ORDER BY SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) DESC
```

Truy vấn – Query

VD13. Đưa ra danh sách n = 2 sinh viên có điểm trung bình cao nhất
Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

```
Truy vấn: SELECT TOP 2 d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB  
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH  
INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC  
GROUP BY MaSV  
  
ORDER BY SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) DESC
```

Truy vấn – Query

VD13. Đưa ra danh sách n = 2 sinh viên có điểm trung bình cao nhất
Có 3 bảng: LOP_TIN_CHI, MON_HOC và BANG_DIEM

Truy vấn: SELECT TOP 2 d.MaSV, SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l ON m.MaMH = l.MaMH
INNER JOIN BANG_DIEM AS d ON l.MaLopTC = d.MaLopTC
GROUP BY MaSV

ORDER BY SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) DESC

B5. Sắp xếp giảm dần theo SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) dựa trên bảng kết quả của bước 4

MaSV	DiemTB
SV1	7.1
SV3	9.2
SV2	8.5
SV5	8.5
SV4	8.3



MaSV	DiemTB
SV3	9.2
SV2	8.5
SV5	8.5
SV4	8.3
SV1	7.1

B6. Lấy n = 2 bản ghi đầu tiên

MaSV	DiemTB
SV3	9.2
SV2	8.5
SV5	8.5

Truy vấn lồng với ANY, ALL; [NOT] IN, EXISTS

- Thường tham gia vào WHERE hoặc HAVING
- WHERE *expression* [NOT] IN (*subquery*)

Hoặc

WHERE *expression comparison_operator* [ANY | ALL] (*subquery*)

Hoặc

WHERE [NOT] EXISTS (*subquery*)

Truy vấn lồng với IN, NOT IN

- **SELECT MaHDX, NgayHDX
FROM HD_XUAT
WHERE MaNV IN (SELECT MaNV FROM NHAN_VIEN
WHERE MaPB = 'P1')
ORDER BY NgayHDX DESC**
- **SELECT MaHDX, NgayHDX
FROM HD_XUAT h JOIN NHAN_VIEN n ON h.MaNV =
n.MaNV WHERE MaPB = 'P1'
ORDER BY NgayHDX DESC**

Truy vấn lồng với toán tử so sánh

- Toán tử so sánh: =, >, >=, <, <=, != hoặc <>

NHAN_VIEN(MaNV, HoTen, NgaySinh, QueQuan, MaPB)

- **SELECT * FROM NHAN_VIEN WHERE MaPB = 'P1'
AND NgaySinh < ALL(SELECT NgaySinh FROM
NHAN_VIEN WHERE MaPB = 'P2')**
- **SELECT * FROM NHAN_VIEN WHERE MaPB = 'P1'
AND NgaySinh < ANY(SELECT NgaySinh FROM
NHAN_VIEN WHERE MaPB = 'P2')**

Truy vấn lồng với EXISTS, NOT EXISTS

- Toán tử so sánh: =, >, >=, <, <=, != hoặc <>

```
SELECT *  
FROM NHAN_VIEN  
WHERE EXISTS (SELECT * FROM HD_XUAT WHERE  
MaNV = NHAN_VIEN.MaNV)
```

- SELECT *
FROM NHAN_VIEN n JOIN HD_XUAT h ON h.MaNV =
n.MaNV

Truy vấn lồng như là biểu thức

```
SELECT MaNV, HoTen, (YEAR(getdate()) -  
YEAR(NgaySinh)) AS TuiNV, (YEAR(getdate()) -  
(SELECT AVG(YEAR(NgaySinh)) FROM NHAN_VIEN))  
AS TuiTrungBinh, (YEAR(NgaySinh) - (SELECT  
AVG(YEAR(NgaySinh)) FROM NHAN_VIEN)) AS  
ChenhLech  
FROM NHAN_VIEN
```

Truy vấn lồng như là bảng ảo

```
SELECT m.MaMH, TenMH, TongNhap, ISNULL(TongXuat, 0) AS TongXuat,  
(TongNhap - ISNULL(TongXuat, 0)) AS TongCon, DonViTinh
```

```
FROM MAT_HANG AS m JOIN  
(SELECT MaMH, SUM(SoLuong) AS TongNhap FROM HD_NHAP JOIN CT_NHAP  
ON HD_NHAP.MaHDN = CT_NHAP.MaHDN WHERE NgayHD <= '30/08/2015'  
GROUP BY MaMH)
```

```
AS NHAP ON m.MaMH = NHAP.MaMH  
LEFT OUTER JOIN
```

```
(SELECT MaMH, SUM(SoLuong) AS TongXuat FROM HD_XUAT JOIN CT_XUAT  
ON HD_XUAT.MaHDX = CT_XUAT.MaHDX WHERE NgayHD <= '30/08/2015'  
GROUP BY MaMH)
```

```
AS XUAT ON NHAP.MaMH = XUAT.MaMH
```

Truy vấn lồng đồng bộ

- synchronized subquery, correlated subquery
- **SELECT ***
FROM NHAN_VIEN
WHERE EXISTS (SELECT * FROM HD_XUAT WHERE
MaNV = NHAN_VIEN.MaNV)

Các hàm thống kê cơ bản

- **SUM(E), SUM(DISTINCT E)**
- **COUNT(E), COUNT(*), COUNT(DISTINCT E)**
- **AVG(E)**
- **MIN(E)**
- **MAX(E)**
- **STDEV(E), STDEV(DISTINCT E)**

Phép nối chéo

- **A CROSS JOIN B**
- **FROM A, B**
 - Chính là tích đề-cát giữa A và B
- **A JOIN B ON A.CộtA = B.CộtB...**
- **FROM A, B WHERE A.CộtA = B.CộtB...**
 - Chính là A CROSS JOIN B sau đó áp dụng điều kiện $A.CộtA = B.CộtB...$

Truy vấn – Query: Phép nối tự nhiên

- **A NATURAL JOIN B**

A INNER JOIN B ON A.CộtCùngTên_1 = B.CộtCùngTên_1
AND ... A.CộtCùngTên_n = B.CộtCùngTên_n

- Ví dụ:

SELECT v.MaSV, HoTen, NgaySinh, DiemTB FROM
VIEW_DTB AS v NATURAL JOIN SINHVIEN

Tương đương với:

SELECT v.MaSV, HoTen, NgaySinh, DiemTB FROM
VIEW_DTB AS v INNER JOIN SINHVIEN AS s ON
v.MaSV = s.MaSV

- Khi nào có thể dùng NATURAL JOIN?

Các phép nối ngoài

a) Nối ngoài trái

Biểu diễn: $A \text{ LEFT OUTER JOIN } B \text{ ON } A.\text{cotA} = B.\text{cotB}$

Kết quả: Kết quả phép nối trong $A \text{ JOIN } B \text{ ON } A.\text{cotA} = B.\text{cotB}$ hợp với tập các bản ghi của A không thoả mãn điều kiện nối kết hợp NULL cho các cột của B

b) Nối ngoài phải

Biểu diễn: $A \text{ RIGHT OUTER JOIN } B \text{ ON } A.\text{cotA} = B.\text{cotB}$

Kết quả: Kết quả phép nối trong $A \text{ JOIN } B \text{ ON } A.\text{cotA} = B.\text{cotB}$ hợp với tập các bản ghi của B không thoả mãn điều kiện nối kết hợp NULL cho các cột của A

c) Nối ngoài đầy đủ

Biểu diễn: $A \text{ FULL OUTER JOIN } B \text{ ON } A.\text{cotA} = B.\text{cotB}$

Kết quả: Kết quả phép nối trong $A \text{ JOIN } B \text{ ON } A.\text{cotA} = B.\text{cotB}$ hợp với tập các bản ghi của A không thoả mãn điều kiện nối kết hợp NULL cho các cột của B hợp với tập các bản ghi của B không thoả mãn điều kiện nối kết hợp NULL cho các cột của A

Các phép nối ngoài

a) Nối ngoài trái

Biểu diễn: **A LEFT OUTER JOIN B ON A.cotA = B.cotB**

Kết quả: Kết quả phép nối trong **A JOIN B ON A.cotA = B.cotB** hợp với tập các bản ghi của A không thoả mãn điều kiện nối kết hợp NULL cho các cột của B

VD: **A LEFT OUTER JOIN B ON A.C1 = B.C1**

A

C1	C2
a	2
b	2
c	1

B

C1	C3
a	5
d	4
b	2
a	1

A.C1	C2	B.C1	C3
a	2	a	5
a	2	a	1
b	2	b	2
c	1	NULL	NULL

Các phép nối ngoài

b) Nối ngoài phải

Biểu diễn: A **RIGHT OUTER JOIN** B ON A.cotA = B.cotB

Kết quả: Kết quả phép nối trong **A JOIN B ON A.cotA = B.cotB** hợp với tập các bản ghi của B không thoả mãn điều kiện nối kết hợp NULL cho các cột của A

VD: A **RIGHT OUTER JOIN** B ON A.C1 = B.C1

A

C1	C2
a	2
b	2
c	1

B

C1	C3
a	5
d	4
b	2
a	1

A.C1	C2	B.C1	C3
a	2	a	5
a	2	a	1
b	2	b	2
NULL	NULL	d	4

Các phép nối ngoài

c) Nối ngoài đầy đủ

Biểu diễn: **A FULL OUTER JOIN B ON A.cotA = B.cotB**

Kết quả: Kết quả phép nối trong **A JOIN B ON A.cotA = B.cotB** hợp với tập các bản ghi của A không thoả mãn điều kiện nối kết hợp NULL cho các cột của B hợp với tập các bản ghi của B không thoả mãn điều kiện nối kết hợp NULL cho các cột của A

VD: **A FULL OUTER JOIN B ON A.C1 = B.C1**

A

C1	C2
a	2
b	2
c	1

B

C1	C3
a	5
d	4
b	2
a	1

A.C1	C2	B.C1	C3
a	2	a	5
a	2	a	1
b	2	b	2
c	1	NULL	NULL
NULL	NULL	d	4

Các phép nối ngoài

c) Nối ngoài đầy đủ - ví dụ

SELECT

coalesce(a.MaSV,b.MaSV) as MaSV,
coalesce(a.HoTen, b. HoTen) as HoTen,
coalesce(a.Lop, b.Lop) as Lop,
coalesce(a.Active,0) as Active

FROM SINH_VIEN a FULL OUTER JOIN OldSINH_VIEN b ON a.MaSV = b. MaSV

Coalesce – Trả lại giá trị không NULL đầu tiên

Bảng
OldSINH
_VIEN

MaNV	HoTen	NgaySinh	QueQuan	Lop	Active
S1	Sinh Viên 1	NS 1	Q1	L1	1
S3	Sinh Viên 2	NS 3	Q1	L2	NULL
S4	Sinh Viên 4	NS 4	Q2	L1	1
S5	Sinh Viên 5	NS 5	Q1	NULL	1
S6	Sinh Viên 6	NS 6	NULL	L1	1

Các phép nối ngoài

c) Nối ngoài đầy đủ - ví dụ

Bảng
SINH_VI
EN

MaNV	HoTen	NgaySinh	QueQuan	Lop	Active
S1	Sinh Viên 1	NS 1	Q1	L1	1
S2	Sinh Viên 1	NS 1	Q1	L1	0
S3	Sinh Viên 2	NS 3	Q1	L2	1
S6	Sinh Viên 6	NS 6	Q1	L1	1

Bảng
OldSINH
_VIEN

MaNV	HoTen	NgaySinh	QueQuan	Lop	Active
S1	Sinh Viên 1	NS 1	Q1	L1	1
S3	Sinh Viên 2	NS 3	Q1	L2	NULL
S4	Sinh Viên 4	NS 4	Q2	L1	1
S5	Sinh Viên 5	NS 5	Q1	NULL	1
S6	Sinh Viên 6	NS 6	NULL	L1	1

Các phép nối ngoài

B1. SINH_VIEN a FULL OUTER JOIN OldSINH_VIEN b ON a.MaSV = b. MaSV

a.MaSV	a.HoTen	a.Lop	Active	b.MaSV	b.HoTen	b.Lop
S1	Sinh Viên 1	L1	1	S1	Sinh Viên 1	L1
S3	Sinh Viên 2	L2	1	S3	Sinh Viên 2	L2
S6	Sinh Viên 6	L1	1	S6	Sinh Viên 6	L1
S2	Sinh Viên 1	L1	0	NULL	NULL	NULL
NULL	NULL	NULL	NULL	S4	Sinh Viên 4	L1
NULL	NULL	NULL	NULL	S5	Sinh Viên 5	NULL

Các phép nối ngoài

B2. Thực hiện mệnh đề SELECT:

MaSV	HoTen	Lop	Active
S1	Sinh Viên 1	L1	1
S3	Sinh Viên 2	L2	1
S6	Sinh Viên 6	L1	1
S2	Sinh Viên 1	L1	0
S4	Sinh Viên 4	L1	0
S5	Sinh Viên 5	NULL	0

Truy vấn – Query

VD14. Đưa ra danh sách các môn học chưa được tổ chức lớp tín chỉ lần nào.

Có 3 bảng: LOP_TIN_CHI, MON_HOC

MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3
M3	Môn học 3	3

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

Truy vấn – Query

VD14. Đưa ra danh sách các môn học chưa được tổ chức lớp tín chỉ lần nào.
Có 2 bảng: LOP_TIN_CHI, MON_HOC

MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3
M3	Môn học 3	3

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

Truy vấn:

Cách 1: SELECT m.MaMH, m.TenMH, m.SoTC

FROM MON_HOC AS m **LEFT OUTER JOIN** LOP_TIN_CHI AS ltc ON m.MaMH = ltc.MaMH
WHERE ltc.MaMH IS NULL

Cách 2 - **Sử dụng truy vấn lồng (nested, sub)**: SELECT MaMH, TenMH, SoTC

FROM MON_HOC WHERE **MaMH** NOT IN (SELECT **MaMH** FROM LOP_TIN_CHI)

Truy vấn – Query: VD14 (Cách 1)

MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3
M3	Môn học 3	3

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

B1. MON_HOC AS m LEFT OUTER JOIN LOP_TIN_CHI AS ltc ON m.MaMH = ltc.MaMH

ltc.MaLo pTC	ltc.MaMH	ltc.MaGV	m.MaMH	m.TenMH	m.SoTC
LTC1	M1	GV1	M1	Môn học 1	2
LTC3	M2	GV2	M2	Môn học 2	3
LTC2	M2	GV2	M2	Môn học 2	3
NULL	NULL	NULL	M3	Môn học 3	3

B2. Áp dụng điều kiện ltc.MaMH IS NULL lên kết quả của bước 1

MaMH	TenMH	SoTC
M3	Môn học 3	3

Truy vấn – Bí danh - alias

- VD10. Đưa ra danh sách tất cả nhân viên có cùng ngày sinh

MaNV	HoTen	NgaySinh	QueQuan	MaPB
N1	Nhân Viên 1	NS 1	Q1	P1
N2	Nhân Viên 1	NS 1	Q1	P1
N3	Nhân Viên 2	NS 3	Q1	P2
N4	Nhân Viên 4	NS 4	Q2	P1
N5	Nhân Viên 5	NS 5	Q1	P2
N6	Nhân Viên 6	NS 6	Q1	P1

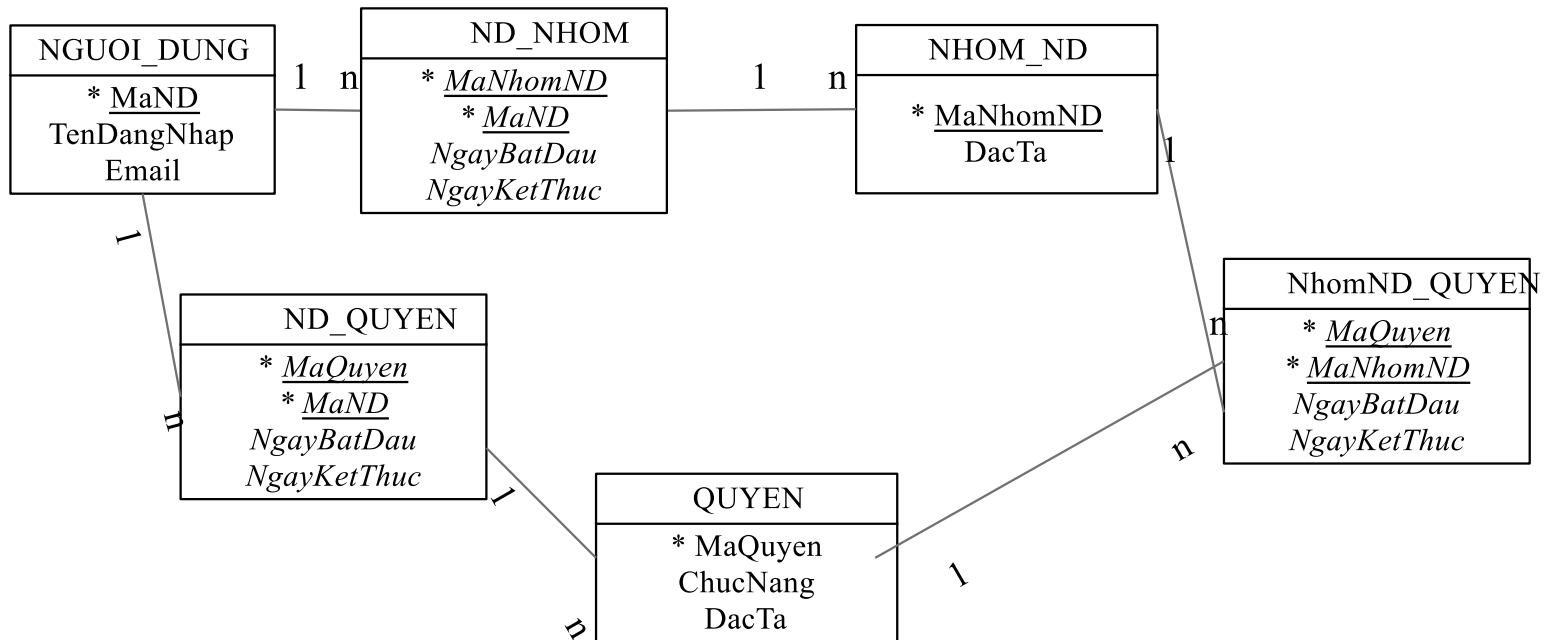
```
SQL: SELECT a.MaNV, a.HoTen, b.MaNV, b.HoTen, a.NgaySinh
      FROM NHANVIEN a JOIN NHANVIEN b on a.NgaySinh = b.NgaySinh
      WHERE a.MaNV < b.MaNV
```

Khung nhìn – View (Virtual view)

```
CREATE VIEW VIEW_DTB AS SELECT d.MaSV,  
SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB  
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l  
ON m.MaMH = l.MaMH INNER JOIN BANG_DIEM AS d  
ON l.MaLopTC = d.MaLopTC GROUP BY MaSV
```

Khung nhìn – View (Virtual view)

- Sử dụng:
 - `SELECT * FROM VIEW_DTB;`
 - `SELECT MaSV, DiemTB FROM VIEW_DTB WHERE DiemTB >= 8.5;`
 - `SELECT TOP 2 FROM VIEW_DTB ORDER BY DiemTB DESC;`
 - `SELECT v.MaSV, HoTen, NgaySinh, DiemTB FROM VIEW_DTB AS v NATURAL JOIN SINHVIEN`
 - `SELECT v.MaSV, HoTen, NgaySinh, DiemTB FROM VIEW_DTB AS v INNER JOIN SINHVIEN AS s ON v.MaSV = s.MaSV`



SELECT q.MaQuyien, ChucNang, DacTa FROM NGUOI_DUNG n JOIN ND_QUIYEN nq ON
 n.MaND = nq.MaND JOIN QUYEN q ON nq.MaQuyien = q.MaQuyien WHERE TenDangNhap =
 'login1' AND getdate() BETWEEN nq.NgayBatDau AND nq.NgayKetThuc

UNION

SELECT q.MaQuyien, Chucnang, DaTa FROM NGUOI-DUNG n JOIN ND_NHOM nn ON
 n.MaND = nn.MaND JOIN NHOM_ND nh ON nn.MaNhomND = nh.MaNhomND JOIN
 NhomND_QUIYEN nq ON nh.MaNhomND = nq.MaNhomND JOIN QUYEN q ON nq.MaQuyien =
 q.MaQuyien WHERE TenDangNhap = 'login1' AND getdate() BETWEEN nn.NgayBatDau AND
 nn.NgayKetThuc AND getdate() BETWEEN nq.NgayBatDau AND nq.NgayKetThuc

Khung nhìn – View (Virtual view)

```
CREATE VIEW VIEW_DTB AS SELECT d.MaSV,  
SUM(d.DiemTK * m.SoTC)/SUM(m.SoTC) AS DiemTB  
FROM MON_HOC AS m INNER JOIN LOP_TIN_CHI AS l  
ON m.MaMH = l.MaMH INNER JOIN BANG_DIEM AS d  
ON l.MaLopTC = d.MaLopTC GROUP BY MaSV
```

Khung nhìn – View (Virtual view)

- Sử dụng:
 - `SELECT * FROM VIEW_DTB;`
 - `SELECT MaSV, DiemTB FROM VIEW_DTB WHERE DiemTB >= 8.5;`
 - `SELECT TOP 2 FROM VIEW_DTB ORDER BY DiemTB DESC;`
 - `SELECT v.MaSV, HoTen, NgaySinh, DiemTB FROM VIEW_DTB AS v NATURAL JOIN SINHVIEN`
 - `SELECT v.MaSV, HoTen, NgaySinh, DiemTB FROM VIEW_DTB AS v INNER JOIN SINHVIEN AS s ON v.MaSV = s.MaSV`

INTERSECT – UNION - EXCEPT

- Tìm các SV đã học môn A và môn B

```
SELECT DISTINCT MaSV FROM BANG_DIEM WHERE  
MaMH = 'A'
```

INTERSECT

```
SELECT MaSV FROM BANG_DIEM WHERE MaMH = 'B'
```

INTERSECT – UNION - EXCEPT

- Tìm các SV đã học môn A hoặc môn B

```
SELECT DISTINCT MaSV FROM BANG_DIEM WHERE  
MaMH = 'A'
```

UNION

```
SELECT DISTINCT MaSV FROM BANG_DIEM WHERE  
MaMH = 'B'
```

Hoặc

```
SELECT MaSV FROM BANG_DIEM WHERE MaMH = 'A'  
OR MaMH = 'B'
```

INTERSECT – UNION - EXCEPT

- Tìm các SV đã học môn A nhưng chưa học môn B

```
SELECT DISTINCT MaSV FROM BANG_DIEM WHERE  
MaMH = 'A'
```

EXCEPT

```
SELECT DISTINCT MaSV FROM BANG_DIEM WHERE  
MaMH = 'B'
```

BT: Quản lý nhập, xuất hàng hoá

Có nhiều nhân viên, khách hàng, nhà cung cấp, kho bãi, hàng hoá, hoá đơn bán và hoá đơn nhập.

QLXH: Mỗi NV có thể lập nhiều HĐ, mỗi HĐ do một NV lập. Mỗi KH có thể có nhiều HĐ, mỗi HĐ thuộc về một KH. Mỗi HĐ được viết cho một NK, mỗi NK có thể có nhiều HĐ. Trong mỗi HĐ có thể có nhiều HH, mỗi HH có thể có mặt trong nhiều HĐ. Mỗi HH, nếu có mặt trong một HĐ nào đó, thì chỉ một lần duy nhất với số lượng và đơn giá cụ thể.

QLNH: Tương tự.

- 1) Vẽ sơ đồ TT-KH & sơ đồ CSDL QH
- 2) Tính số lượng mỗi HH đã nhập cho đến nnnn-tt-nn
- 3) Tính số lượng mỗi HH đã xuất cho đến nnnn-tt-nn
- 4) Tính tổng nhập, xuất, tồn cho mỗi HH cho đến nnnn-tt-nn
- 5) Tương tự 2-4,...theo NK...

BT: Quản lý nhập, xuất hàng hoá

Có nhiều nhân viên, khách hàng, nhà cung cấp, kho bãi, hàng hoá, hoá đơn bán và hoá đơn nhập....

Với cách quản lý trên, sau này có thể theo dõi quá trình thay đổi giá của một mặt hàng nào đó hay không? Cải tiến.

3.4 Ràng buộc toàn vẹn

- Định nghĩa. Ràng buộc tính nhất quán là một khẳng định nào đó mà nó có thể đúng hoặc sai phụ thuộc vào trạng thái của CSDL.
- Ví dụ về ràng buộc tính toàn vẹn có thể là những khẳng định sau đây:
 - Ví dụ 2. Độ tuổi của nhân viên phải nằm trong khoảng từ 18 đến 65 tuổi.
 - Ví dụ 3. Mỗi một sinh viên phải có một từ định danh ID không lặp lại.
 - Ví dụ 4. Mỗi giảng viên chỉ có thể là giảng viên cơ hữu của một trường và từ định danh của trường đó được ghi trong bản ghi về giảng viên.
 - Ví dụ 5. Tổng thành tiền của một hoá đơn phải bằng tổng các kết quả phép nhân số lượng cho đơn giá của tất cả các mặt hàng có trong hoá đơn.

3.4 Ràng buộc toàn vẹn

- Ví dụ sự phá huỷ tính toàn vẹn CSDL:
- Khi thêm một nhân viên mới cần phải đồng thời tăng giá trị của trường bp_soluong.
 - Bước 1. Thêm nhân viên vào bảng NHANVIEN: `insert into nhanvien values(6, 'Nguyen Van A', 1)`
 - Bước 2. Tăng giá trị trường bp_soluong: `update BOPHAN set bp_soluong = bp_soluong + 1 where bp_id = 1.`
- Nếu sau khi hoàn thành hành động đầu tiên và trước khi hoàn thành hành động thứ hai xảy ra sự cố trong hệ thống, thì thực tế chỉ hoàn thành hành động thứ nhất và CSDL sẽ có trạng thái không toàn vẹn.

3.4 Ràng buộc toàn vẹn

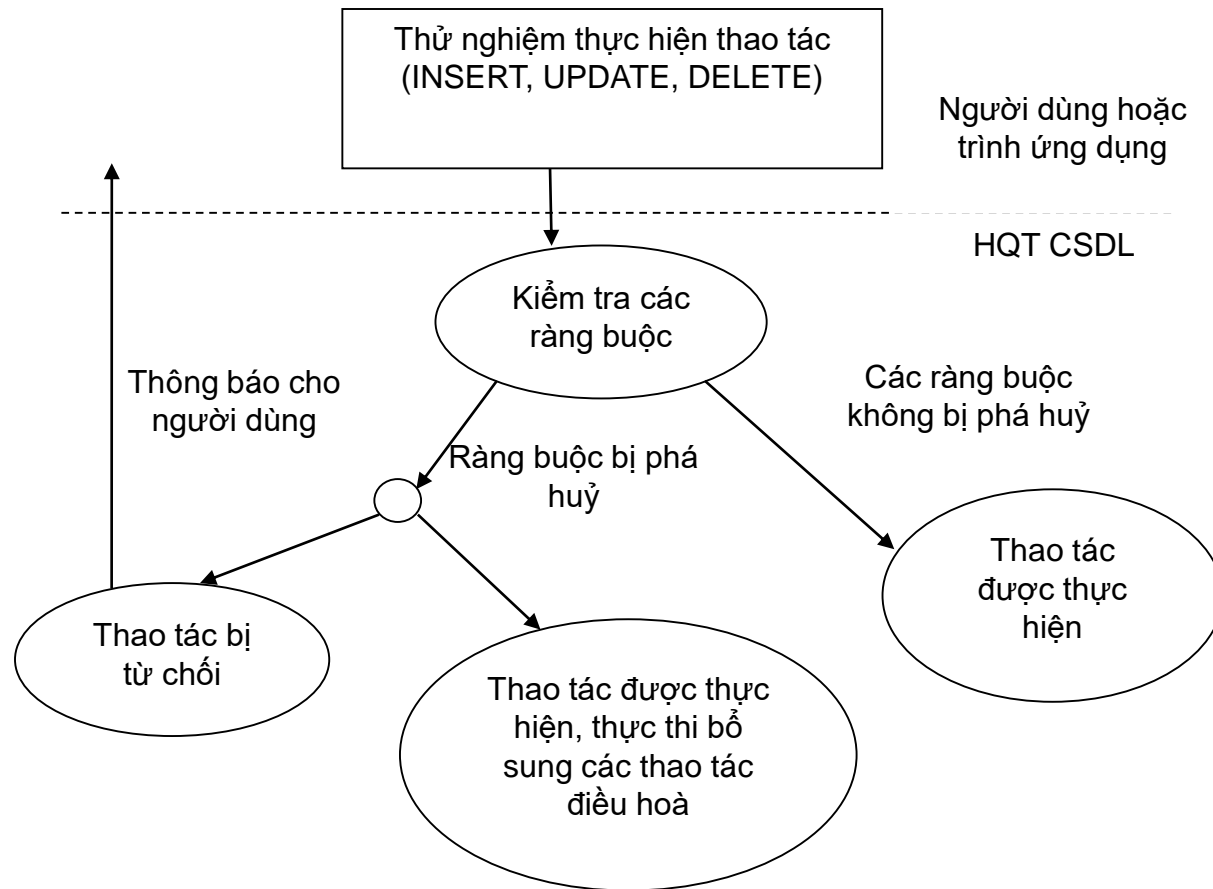
- Định nghĩa 3. CSDL nằm trong trạng thái toàn vẹn nếu thoả mãn tất cả các ràng buộc về tính toàn vẹn đã được định nghĩa cho CSDL

3.4 Ràng buộc toàn vẹn

- Phản ứng của hệ thống đối với thử nghiệm phá huỷ tính toàn vẹn: Có hai loại phản ứng với các thử nghiệm phá huỷ tính toàn vẹn:
 - Từ chối thực hiện thao tác “không hợp pháp”
 - Thực hiện một loạt các hành động điều chỉnh (điều hoà, bù trừ - compensating)

3.4 Ràng buộc toàn vẹn

- Công việc của hệ thống trong việc kiểm tra các ràng buộc:



Hình II.3 Kiểm tra các ràng buộc

3.4.1. Phân loại ràng buộc toàn vẹn

- Bằng các cách triển khai (realization)
- Phân loại ràng buộc tính toàn vẹn theo thời điểm kiểm tra
- Phân loại ràng buộc tính toàn vẹn theo không gian ứng dụng

3.4.1.1 Bằng các cách triển khai

- Hỗ trợ ràng buộc toàn vẹn theo kiểu khai báo
 - Hỗ trợ ràng buộc toàn vẹn theo kiểu khai báo được bao hàm trong sự xác định các ràng buộc bằng các công cụ của ngôn ngữ định nghĩa dữ liệu (DDL - Data Definition Language)
 - VD: - nv_id sẽ là khoá ứng viên (chính) của quan hệ;
 - - nv_hoten không thể chứa các giá trị NULL
- Hỗ trợ ràng buộc toàn vẹn theo kiểu thủ tục
 - Hỗ trợ ràng buộc tính toán toàn vẹn theo thủ tục được bao hàm trong việc sử dụng các bẫy sự kiện (trigger) và các thủ tục (stored procedure)
 - VD: Để đề phòng các trường hợp sửa đổi thông tin về một bộ phận nhưng nhập vào trường bp_soluong một giá trị không đúng cần phải tạo ra một bẫy sự kiện tự động thực thi khi thêm vào hoặc sửa đổi trong bảng BOPHAN

3.4.1.2 Phân loại ràng buộc toàn vẹn theo thời điểm kiểm tra

- Ràng buộc được kiểm tra ngay lập tức là những ràng buộc được kiểm tra ngay lập tức vào thời điểm thực hiện các thao tác có thể phá huỷ ràng buộc đó
 - VD: Ví dụ kiểm tra tính độc nhất của khoá ứng viên (chính) được thực hiện vào thời điểm thêm một bản ghi mới vào bảng
- Ràng buộc với thời điểm kiểm tra trì hoãn là ràng buộc được kiểm tra vào thời điểm cố định giao tác bằng lệnh COMMIT WORK
 - VD: Thêm một bản ghi vào bảng NHANVIEN bao gồm thêm bản ghi về nhân viên mới vào bảng NHANVIEN và sửa đổi một bản ghi trong bảng BOPHAN

3.4.1.3 Phân loại ràng buộc tính toán vện theo không gian ứng dụng

- Ràng buộc tính toán vện miền giá trị là những ràng buộc dựa trên những giá trị được cho phép của miền giá trị
 - VD: “Độ tuổi của nhân viên không được nhỏ hơn 18 và không lớn hơn 65” là ràng buộc miền giá trị “Độ tuổi nhân viên”
- Ràng buộc tính toán vện của thuộc tính là những ràng buộc dựa trên những giá trị cho phép đối với thuộc tính, hệ quả của việc thuộc tính được khai báo trên một miền giá trị nào đó
 - VD: “Độ tuổi nhân viên” và phải là số nguyên

3.4.1.3 Phân loại ràng buộc tính toán vện theo không gian ứng dụng

- Ràng buộc tính toán vện bản ghi là những ràng buộc dựa trên những giá trị cho phép của từng bản ghi của quan hệ, và không phải là ràng buộc tính toán vện của thuộc tính
 - VD: Nếu trường chức_vụ có giá trị “giám đốc” thì trường mức_lương phải có giá trị lớn hơn 5.000.000. Ràng buộc này kết nối hai thuộc tính một bản ghi
 - VD: Mỗi quan hệ đó gắn chặt ba thuộc tính thành_tiền, số_lượng và đơn_giá với nhau

3.4.1.3 Phân loại ràng buộc tính toán vện theo không gian ứng dụng

Tên thuộc tính	Mô tả thuộc tính	Thuộc tính gốc?	Công thức để tính thuộc tính thứ sinh
Mặt_hàng	Tên hàng hoá	Đúng	
N	Số lượng	Đúng	
P1	Đơn giá	Đúng	
S1	Thành tiền theo đơn giá		$S1 = N * P1$
Percent	% phụ thu cho mỗi đơn vị hàng	Đúng	
P2	phụ thu cho mỗi đơn vị hàng		$P2 = P1 * Percent / 100$
S2	Tổng phụ thu cho mỗi mặt hàng		$S2 = N * P2$
P3	Đơn giá của đơn vị hàng có tính phụ thu		$P3 = P1 + P2$
S3	Thành tiền cho mặt hàng có tính phụ thu		$S3 = N * P3$
VAT	% VAT	Đúng	
P4	Tổng VAT cho mỗi đơn vị hàng		$P4 = P2 * NDS / 100$
S4	Tổng VAT cho mặt hàng		$S4 = N * P4$
P5	Thành tiền đơn vị hàng có tính phụ thu và VAT		$P5 = P3 + P4$
S5	Thành tiền mặt hàng có tính phụ thu và VAT		$S5 = N * P5$

3.4.1.3 Phân loại ràng buộc tính toàn vẹn theo không gian ứng dụng

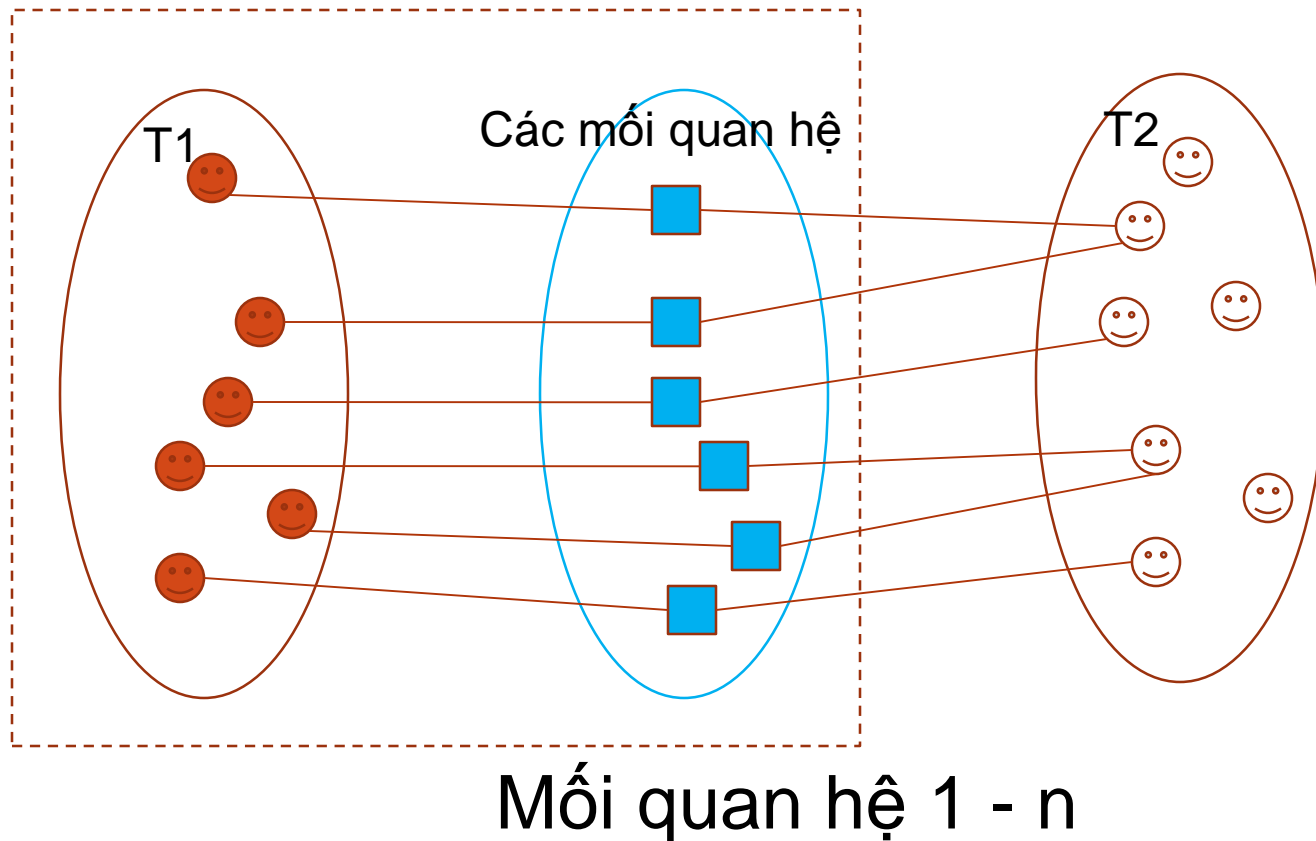
- Ràng buộc tính toàn vẹn của quan hệ là những ràng buộc dựa trên các giá trị cho phép của từng quan hệ, và không phải là ràng buộc toàn vẹn của bản ghi
- Ràng buộc tính toàn vẹn CSDL là những ràng buộc dựa trên những giá trị của hai hay nhiều quan hệ liên quan với nhau, kể cả trường hợp quan hệ tự liên quan đến bản thân

3.4.1.3 Phân loại ràng buộc tính toàn vẹn theo không gian ứng dụng

- Ràng buộc tính toàn vẹn bản ghi là những ràng buộc dựa trên những giá trị cho phép của từng bản ghi của quan hệ, và không phải là ràng buộc tính toàn vẹn của thuộc tính
 - VD: Nếu trường chức_vụ có giá trị “giám đốc” thì trường mức_lương phải có giá trị lớn hơn 5.000.000. Ràng buộc này kết nối hai thuộc tính một bản ghi
 - VD: Mỗi quan hệ đó gắn chặt ba thuộc tính thành_tiền, số_lượng và đơn_giá với nhau
- Ràng buộc tính toàn vẹn của quan hệ là những ràng buộc dựa trên các giá trị cho phép của từng quan hệ, và không phải là ràng buộc toàn vẹn của bản ghi
- Ràng buộc tính toàn vẹn CSDL

3.4.1.3 Phân loại ràng buộc tính toàn vẹn theo không gian ứng dụng

- Ràng buộc tính toàn vẹn CSDL là những ràng buộc dựa trên những giá trị của hai hay nhiều quan hệ liên quan với nhau, kể cả trường hợp quan hệ tự liên quan đến bản thân



3.4.2. Triển khai ràng buộc tính toàn vẹn dạng định nghĩa bằng công cụ SQL

- Các nguyên tắc chung
- Cú pháp ràng buộc theo chuẩn SQL

3.4.2.1 Các nguyên tắc chung

- Các ràng buộc kiểu khai báo theo các phương pháp:
 - Giống như các ràng buộc miền giá trị
 - Các ràng buộc trong khuôn khổ định nghĩa quan hệ
 - Các ràng buộc được lưu trữ trong CSDL dưới dạng các điều chỉnh (assertion) độc lập
- Ràng buộc kiểm tra ngay lập tức (NOT DEFERRED) và ràng buộc trì hoãn kiểm tra (DEFERRED):
 - Sau từng thao tác có thể làm thay đổi nội dung của bảng mà ràng buộc đó liên quan đến
 - Sau khi hoàn thành mỗi giao tác chứa các thao tác làm thay đổi nội dung các bảng mà ràng buộc đó liên quan đến
 - Ở mọi thời điểm trung gian nếu như người dùng khởi đầu sự kiểm tra

3.4.2.1 Các nguyên tắc chung

- Ràng buộc CSDL - ràng buộc khoá ngoại: thao tác được thực thi theo các liên kết (referential triggered actions): enforce referential integrity
- Thao tác: ON UPDATE, ON DELETE
- Hành động (điều chỉnh):
 - CASCADE. Mọi thay đổi giá trị của khoá cha đều tự động đưa đến sự thay đổi tương tự giá trị khoá ngoài liên quan. Xóa bản ghi trong bảng cha dẫn đến việc xóa các bản ghi liên kết với nó trong bảng con
 - SET NULL. Tất cả các khoá ngoài liên kết đến khoá cha vừa được sửa đổi hoặc xóa đều nhận giá trị NULL
 - SET DEFAULT. Tất cả các khoá ngoài liên kết đến khoá cha vừa được sửa đổi hoặc xóa đều nhận giá trị mặc định cho các khoá đó
 - NO ACTION. Giá trị khoá ngoài không thay đổi. Nếu thao tác dẫn đến phá huỷ tính toàn vẹn liên kết (xuất hiện liên kết “treo”) thì nó sẽ không được thực thi

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- `Ràng_buộc_check::=`
`CHECK mệnh_đề_predicate`
- `Giới_hạn_bảng::=`
`[CONSTRAINT Tên_giới_hạn`
`{`
`{PRIMARY KEY (Tên_cột,...)}`
`| {UNIQUE (Tên_cột,...)}`
`| {FOREIGN KEY (Tên_cột,...) REFERENCES Tên_bảng [(Tên_cột,...)] [Đặc_tả_liên_kết]}`
`| {Giới_hạn_check }`
`}`
`[Các_cột_của_giới_hạn]`
- `Giới_hạn_cột::=`
`[CONSTRAINT Tên_giới_hạn`
`{`
`{NOT NULL}`
`| {PRIMARY KEY}`
`| {UNIQUE}`
`| {REFERENCES Tên_bảng [(Tên_cột)] [Đặc_tả_liên_kết]}`
`| {Giới_hạn_check }`
`}`
`[Các_cột_của_giới_hạn]`
- `Đặc_tả_liên_kết::=`
`[MATCH {FULL | PARTIAL}]`
`[ON UPDATE {CASCADE | SET NULL | SET DEFAULT | NO ACTION}]`
`[ON DELETE {CASCADE | SET NULL | SET DEFAULT | NO ACTION}]`
- `Các_cột_của_giới_hạn::=`
`{DEFERRABLE [INITIALLY DEFERRED | INITIALLY IMMEDIATE]}`
`| {NOT DEFERRABLE}`

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- Ràng buộc thể loại CHECK chứa các mệnh đề có thể nhận các giá trị TRUE, FALSE và UNKNOWN (NULL)
 - VD: CHECK (students.birthday IS NOT NULL) AND (students.surename IS NOT NULL)
 - VD: CHECK exists(select * from DANHSACH)

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- Ràng buộc bảng và ràng buộc cột là một phần khai báo bảng và các cột tương ứng
 - Ràng buộc bảng có thể liên quan đến một vài cột của bảng; Ràng buộc cột chỉ liên quan đến một cột của bảng
 - Bất cứ ràng buộc cột nào cũng có thể khai báo giống như một ràng buộc bảng, nhưng không tồn tại chiều ngược lại

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- Ràng buộc PRIMARY KEY cho một bảng hoặc là cột có nghĩa là một hoặc một nhóm các cột tạo thành khoá ứng viên (candidate key) của bảng
 - Điều đó cũng có nghĩa rằng tổ hợp các giá trị PRIMARY KEY phải là độc nhất cho mỗi bản ghi của bảng
 - Các giá trị lặp lại hoặc là NULL sẽ bị từ chối
 - Theo các thuật ngữ của chuẩn SQL gọi là khoá chính của bảng

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- Ràng buộc UNIQUE cho một bảng hoặc cột có nghĩa là một nhóm một hoặc nhiều cột tạo thành khoá ứng viên của bảng mà trong đó cho chấp nhận cả giá trị NULL
 - Nghĩa là hai bản ghi chứa cùng giá trị nhưng khác NULL thì sẽ bị cho là phá huỷ tính độc nhất và sẽ bị từ chối
 - Hai bản ghi chứa cùng giá trị NULL sẽ được cho là khác nhau và được chấp nhận
 - Với một bảng có thể định nghĩa nhiều ràng buộc UNIQUE

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- Ràng buộc FOREIGN KEY... REFERENCES... cho một bảng và ràng buộc REFERENCES cho một cột định nghĩa khoá ngoài của bảng.
 - Ràng buộc REFERENCES cho một cột định nghĩa khoá ngoài đơn giản - khoá được cấu thành từ một cột.
 - Ràng buộc FOREIGN KEY... REFERENCES... cho một bảng có thể định nghĩa khoá ngoài đơn giản hoặc là phức hợp - khoá được cấu thành từ nhiều cột của một bảng.
 - Cột hoặc nhóm cột của bảng mà khoá ngoài liên kết đến phải có ràng buộc PRIMARY KEY hoặc UNIQUE.
 - Các cột mà khoá ngoài liên kết đến phải có cùng kiểu dữ liệu giống như các cột thành viên của khoá ngoài.
 - Bảng có thể liên kết đến chính bản thân mình.
 - Ràng buộc khoá ngoài bị phá huỷ nếu các giá trị có trong khoá ngoài không trùng với các giá trị các khoá tương ứng (của bảng cha) nào của bất cứ một bản ghi nào của bảng cha.
 - Các thao tác dẫn đến sự phá huỷ ràng buộc khoá ngoài đều bị từ chối.
 - Các giá trị của khoá ngoài và các khoá bảng cha cần phải trùng khớp như thế nào, cũng như những hành động nào cần phải thực thi khi xảy ra thay đổi khoá bảng cha sẽ được miêu tả trong phần đặc tả liên kết

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- Ràng buộc NOT NULL của cột không chấp nhận sự xuất hiện trong cột các giá trị NULL
- Đặc tả liên kết định nghĩa các đặc tính của khoá ngoài của bảng
 - Mệnh đề MATCH {FULL|PARTIAL}

3.4.2.2 Cú pháp ràng buộc theo chuẩn SQL

- Các thuộc tính của ràng buộc định nghĩa thời điểm kiểm tra ràng buộc: NOT DEFERRABLE (không trì hoãn) hoặc DEFERRABLE (trì hoãn)
 - Mặc định: NOT DEFERRABLE.
 - NOT DEFERRABLE: được kiểm tra ngay lập tức sau khi thực thi mỗi thao tác có thể dẫn đến sự vi phạm ràng buộc như INSERT, UPDATE hoặc DELETE.
 - DEFERRABLE: có hai chế độ kiểm tra – không chậm trễ sau mỗi thao tác hoặc vào cuối giao tác
 - Chế độ kiểm tra có thể được thay đổi vào bất cứ lúc nào trong than giao tác bằng lệnh SET CONSTRAINTS
 - Trong khi định nghĩa ràng buộc có thể khai báo chế độ kiểm tra ban đầu INITIALLY DEFERRED (trì hoãn ban đầu) hoặc INITIALLY IMMEDIATE (ban đầu kiểm tra ngay lập tức)

3.4.2.2 Các lệnh sau đây có thể dùng các ràng buộc

- CREATE DOMAIN – tạo miền giá trị
- ALTER DOMAIN – thay đổi miền giá trị
- DROP DOMAIN – huỷ miền giá trị
- CREATE TABLE – tạo bảng mới
- ALTER TABLE – thay đổi bảng
- DROP TABLE – huỷ bảng
- CREATE ASSERTION – tạo khẳng định
- DROP ASSERTION – xoá khẳng định
- COMMIT WORK – cố định giao tác
- SET CONSTRAINTS – ấn định thời điểm kiểm tra các ràng buộc

Đại số quan hệ - Relational algebra

- Codd đã định nghĩa 8 toán tử, chia làm hai nhóm:
- Các toán tử truyền thống:
 - Kết hợp (Union)
 - Giao (Intersection)
 - Trừ (Difference)
 - Tích đề-cát (Cartesian product)
- Các toán tử quan hệ đặc biệt:
 - Select
 - Project
 - Join
 - Chia (Divide)

UNION – Phép hợp

Giả sử có hai quan hệ tương thích $r1$ và $r2$:

$$r1(c1_1, c1_2, \dots, c1_n) \qquad r1 = \{t1_i \mid i = 1..k\}$$

$$r2(c2_1, c2_2, \dots, c2_m) \qquad r2 = \{t2_j \mid j = 1..l\}$$

Hai quan hệ $r1$ và $r2$ được gọi là tương thích nếu:

$$n = m$$

$$\forall i = 1..n : \text{domain}(c1_i) = \text{domain}(c2_i)$$

Kết quả phép hợp của hai quan hệ $r1$ và $r2$ là quan hệ:

$$r(c_1, c_2, \dots, c_n)$$

$$r = r1 \cup r2 = \{t_i \mid t_i \in r1 \vee t_i \in r2\}$$

$$r = \{t1_1, t1_2, \dots, t1_k, t2_1, t2_2, \dots, t2_l\}$$

UNION – Phép hợp

$r1$

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2

$r2$

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S4	Sinh Viên 4	L2
S5	Sinh Viên 5	L2

$r1 \cup r2$

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2
S4	Sinh Viên 4	L2
S5	Sinh Viên 5	L2

INTERSECT – Phép giao

Kết quả phép giao của hai quan hệ tương thích $r1$ và $r2$ là quan hệ:

$$r(c_1, c_2, \dots, c_n)$$

$$r = r1 \cap r2 = \{t_i \mid t_i \in r1, t_i \in r2\}$$

INTERSECT – Phép giao

Kết quả phép giao của hai quan hệ tương thích $r1$ và $r2$ là quan hệ:

$$r(c_1, c_2, \dots, c_n)$$

$$r = r1 \cap r2 = \{t_i \mid t_i \in r1 \wedge t_i \in r2\}$$

$r1$

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2

$r2$

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S4	Sinh Viên 4	L2
S5	Sinh Viên 5	L2

$r1 \cap r2$

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1

DIFFERENCE (MINUS, EXCEPT) – Phép trừ

Kết quả phép hiệu của hai quan hệ tương thích $r1$ và $r2$ là quan hệ:

$$r12(c_1, c_2, \dots, c_n)$$

$$r21(c_1, c_2, \dots, c_n)$$

$$r12 = r1 - r2 = \{t_i \mid t_i \in r1 \wedge t_i \notin r2\}$$

$$r21 = r2 - r1 = \{t_i \mid t_i \in r2 \wedge t_i \notin r1\}$$

DIFFERENCE (MINUS, EXCEPT) – Phép trừ

r1

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2

r2

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S4	Sinh Viên 4	L2
S5	Sinh Viên 5	L2

$r1 \setminus r2$

MaSV	HoTen	Lop
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2

$r2 \setminus r1$

MaSV	HoTen	Lop
S4	Sinh Viên 4	L2
S5	Sinh Viên 5	L2

CARTESIAN PRODUCT – Tích đề-cát

Giả sử có hai quan hệ $r1$ và $r2$:

$$r1(c1_1, c1_2, \dots, c1_n)$$

$$r1 = \{t1_i \mid i = 1..k\}$$

$$r2(c2_1, c2_2, \dots, c2_m)$$

$$r2 = \{t2_j \mid j = 1..l\}$$

Kết quả phép tích hai quan hệ $r1$ và $r2$ là quan hệ:

$$r = \{(c1_1, \dots, c1_n, c2_1, \dots, c2_m)\}$$

$$r = r1 \times r2 = \{(t1_i, t2_j) \mid i = 1..k, j = 1..l\}$$

CARTESIAN PRODUCT – Tích đề-cát

r1

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2

r2

MaMH	TenMH	SoTC
M1	Môn học 1	3
M2	Môn học 2	2

r1xr2 (các cặp sinh viên – môn học có khả năng)

MaSV	HoTen	Lop	MaMH	TenMH	SoTC
S1	Sinh Viên 1	L1	M1	Môn học 1	3
S1	Sinh Viên 1	L1	M2	Môn học 2	2
S2	Sinh Viên 1	L1	M1	Môn học 1	3
S2	Sinh Viên 1	L1	M2	Môn học 2	2
S3	Sinh Viên 3	L2	M1	Môn học 1	3
S3	Sinh Viên 3	L2	M2	Môn học 2	2

SELECTION – Phép chọn

$$r = \{t_i \mid i = 1..k\}$$

$$\delta_{\text{conditions}} < r > = \{t_j \mid j = 1..k, t_j(\text{conditions}) = \text{true}\}$$

SINH_VIEN

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2

$\delta_{(\text{Lop}='L1')}$ (SINH_VIEN)

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1

PROJECTION – Phép chiếu

$$r(c_1, c_2, \dots, c_n)$$

$$r = \{t_i \mid i = 1..k\}$$

$$r' = \pi_{(attribute_list)}(r)$$

$$attribute_list = \{c_j \mid j = 1..m, m \leq n\}$$

$$r' = \{t_i \mid \langle attribute_list \rangle \mid i = 1..k\}$$

PROJECTION – Phép chiếu

SINH_VIEN

MaSV	HoTen	Lop
S1	Sinh Viên 1	L1
S2	Sinh Viên 1	L1
S3	Sinh Viên 3	L2

$\pi_{(MaSV, HoTen)} (SINH_VIEN)$

$\pi_{(HoTen, Lop)} (SINH_VIEN)$

HoTen	Lop
Sinh Viên 1	L1
Sinh Viên 1	L1
Sinh Viên 3	L2

MaSV	HoTen
S1	Sinh Viên 1
S2	Sinh Viên 1
S3	Sinh Viên 3

JOIN – Phép nối

Giả sử có hai quan hệ $r1$ và $r2$:

$$r1(c1_1, c1_2, \dots, c1_n)$$

$$r1 = \{t1_i \mid i = 1..k\}$$

$$r2(c2_1, c2_2, \dots, c2_m)$$

$$r2 = \{t2_j \mid j = 1..l\}$$

Kết quả phép nối của hai quan hệ $r1$ và $r2$ với điều kiện conditions là quan hệ:

$$r(c1_1, \dots, c1_n, c2_1, \dots, c2_m)$$

$$r = r1 \overset{\text{condition}}{\bowtie} r2 = \{(t1_i, t2_j) \mid i = 1..k, j = 1..l, \\ (t1_i, t2_j)(\text{condition}) = \text{true}\}$$

JOIN – Phép nối

$r1$

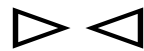
MaMH	TenMH	SoTC
M1	Môn học 1	2
M2	Môn học 2	3
M3	Môn học 3	3

$r2$

MaLopTC	MaMH	MaGV
LTC1	M1	GV1
LTC3	M2	GV2
LTC2	M2	GV2

$$r1.MaMH = r2.MaMH$$

$r = r1$



$r2$

$r1.MaLopTC$	MaMH	MaGV	$r2.MaMH$	TenMH	SoTC
LTC1	M1	GV1	M1	Môn học 1	2
LTC3	M2	GV2	M2	Môn học 2	3
LTC2	M2	GV2	M2	Môn học 2	3

JOIN – Phép nối

- CROSS JOIN (=decartesian product)
- INNER JOIN (=decartesian product + điều kiện nối)
- NATURAL JOIN (=inner join)
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN (=LEFT OUTER JOIN UNION RIGHT OUTER JOIN)

DEVIDE – Phép chia

Giả sử có hai quan hệ $r1$ và $r2$:

$$\begin{array}{ll} r1(c1_1, c1_2, \dots, c1_n, c2_1, c2_2, \dots, c2_m) & r1 = \{t1_i \mid i = 1..k\} \\ r2(c2_1, c2_2, \dots, c2_m) & r2 = \{t2_j \mid j = 1..l\} \end{array}$$

Kết quả phép chia của hai quan hệ $r1$ và $r2$ là quan hệ:

$$\begin{array}{l} r = r1 \div r2 = \{t_g \mid \forall j = 1..l : (t_g, t2_j) \in r1\} \\ r(c1_1, \dots, c1_n, c2_1, \dots, c2_m) \end{array}$$

DEVIDE – Phép chia

$r1$

Tên	Giới tính
A	M
B	F
A	F
C	F
D	M
C	M

$r2$

Giới tính
M
F

$r = r1 \div r2$

Tên
A
C

LÝ THUYẾT THIẾT KẾ CSDL

- Anomalies - Các bất thường
- Decomposition - Phân rã quan hệ
- Functional Dependencies - Phụ thuộc hàm
- Normal Forms - Các dạng chuẩn

Anomalies - Các bất thường

- Insertion anomalies - bất thường thêm mới
- Update anomalies - bất thường cập nhật
- Deletion anomalies - bất thường xóa
- VD: SV n-1 LopSH, SV n-n MH, 1 SV - 1 MH: 1 lần

MaSV	HoTen	NgaySinh	LopSH	Khoa	MaMH	Diem
S1	Sinh Viên 1	NS1	L1	Khoa 1	M1	6
S1	Sinh Viên 1	NS1	L1	Khoa 1	M2	7
S1	Sinh Viên 1	NS1	L1	Khoa 1	M3	9
S2	Sinh Viên 2	NS2	L2	Khoa 1	M1	8
S3	Sinh Viên 3	NS2	L3	Khoa 2	M1	5

Insertion anomalies – bất thường thêm mới

- Khi thêm mới thông tin về một sinh viên thi một môn nào đó đạt bao nhiêu điểm, ta phải thêm mới các thông tin bổ sung (các thông tin về họ tên, lớp, khoa...) ngoài các thông tin Sinh_viên - Môn_học - Điểm

MaSV	HoTen	NgaySinh	LopSH	Khoa	MaMH	Diem
S1	Sinh Viên 1	NS1	L1	Khoa 1	M1	6
S1	Sinh Viên 1	NS1	L1	Khoa 1	M2	7
S1	Sinh Viên 1	NS1	L1	Khoa 1	M3	9
S2	Sinh Viên 2	NS2	L2	Khoa 1	M1	8
S3	Sinh Viên 3	NS2	L3	Khoa 2	M1	5

Update anomalies – Bất thường cập nhật

- Khi cập nhật thông tin về một sinh viên, ta phải cập nhật ở tất cả các bản ghi liên quan đến sinh viên đó. Nếu không, có thể dẫn đến sự phá huỷ về tính nhất quán của dữ liệu.

MaSV	HoTen	NgaySinh	LopSH	Khoa	MaMH	Diem
S1	Sinh Viên 5	NS5	L1	Khoa 1	M1	6
S1	<i>Sinh Viên 1</i>	<i>NS1</i>	<i>L1</i>	<i>Khoa 1</i>	M2	7
S1	<i>Sinh Viên 1</i>	<i>NS1</i>	<i>L1</i>	<i>Khoa 1</i>	M3	9
S2	Sinh Viên 2	NS2	L2	Khoa 1	M1	8
S3	Sinh Viên 3	NS2	L3	Khoa 2	M1	5

Deletion anomalies – Bất thường xoá

- Khi xoá một bản ghi về điểm của sinh viên, ta có thể mất thông tin về sinh viên.

MaSV	HoTen	NgaySinh	LopSH	Khoa	MaMH	Diem
S1	Sinh Viên 1	NS1	L1	Khoa 1	M1	6
S1	Sinh Viên 1	NS1	L1	Khoa 1	M2	7
S1	Sinh Viên 1	NS1	L1	Khoa 1	M3	9
S2	Sinh Viên 2	NS2	L2	Khoa 1	M1	8
S3	Sinh Viên 3	NS2	L3	Khoa 2	M1	5

Functional Dependencies – Phụ thuộc hàm

- Quan hệ: $r = \{t_i \mid i = 1..n\}$
 - Lược đồ: $Q(c_1, c_2, \dots, c_m)$
 - Tập thuộc tính của quan hệ: $Q^+ = \{c_1, c_2, \dots, c_m\}$
- Phụ thuộc hàm:

$$A \subseteq Q^+, B \subseteq Q^+$$

$$\forall t_i \in r, t_j \in r : t_i.A = t_j.A \Rightarrow t_i.B = t_j.B$$

Ta nói rằng, B phụ thuộc hàm vào A, hoặc A xác định B, và ký hiệu:

$$f_i \in F : A \rightarrow B$$

F – tập phụ thuộc hàm

Functional dependencies – Ví dụ

MaSV	HoTen	NgaySinh	LopSH	Khoa	MaMH	Diem
S1	Sinh Viên 1	NS1	L1	Khoa 1	M1	6
S1	Sinh Viên 1	NS1	L1	Khoa 1	M2	6
S1	Sinh Viên 1	NS1	L1	Khoa 1	M3	9
S2	Sinh Viên 2	NS2	L2	Khoa 1	M1	8
S3	Sinh Viên 3	NS2	L3	Khoa 2	M1	5

$f_1 : \text{MaSV} \rightarrow \text{HoTen, NgaySinh, LopSH, Khoa}$

$f_2 : \text{LopSH} \rightarrow \text{Khoa}$

$f_3 : \text{MaSV, MaMH} \rightarrow \text{Diem}$

$F = \{f_1, f_2, f_3\}$

Các tính chất của phụ thuộc hàm

- Các khái niệm:
 - Bao đóng của tập PTH F - F^+ : F^+ là tập tất cả các phụ thuộc hàm suy ra được từ F dựa vào hệ luật dẫn Armstrong
 - Bao đóng của tập thuộc tính X - X_F^+ : X_F^+ là tập tất cả các thuộc tính xác định được từ X dựa trên tập F^+
- Hệ tiên đề Armstrong

Hệ tiên đề Armstrong

- Hệ tiên đề Armstrong: (3 luật dẫn)
 - Phản xạ - Reflexivity : $Y \subseteq X \Rightarrow X \rightarrow Y$
 - Tăng trưởng - Augmentation : $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$
 - Bắt cầu - Transitivity : $X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$
- 3 luật dẫn mở rộng nhằm đơn giản hoá quá trình tính bao đóng:
 - Luật hợp – Union : $X \rightarrow Y, X \rightarrow Z \Rightarrow X \rightarrow YZ$
 - Luật phân rã – Decomposition : $X \rightarrow YZ \Rightarrow X \rightarrow Y, X \rightarrow Z$
 - Luật bắt cầu giả - Pseudo - transitivity : $X \rightarrow Y, AY \rightarrow Z \Rightarrow AX \rightarrow Z$

Bao đóng – Closure

- Bao đóng của tập phụ thuộc hàm
- Bao đóng của tập thuộc tính

Bao đóng của tập phụ thuộc hàm

- Bao đóng của tập phụ thuộc hàm (ký hiệu F^+) của tập phụ thuộc hàm F là tập tất cả các phụ thuộc hàm có thể suy ra từ F dựa trên hệ tiên đề Armstrong.
- Các tính chất của F^+ :
 - Tính phản xạ: $F \subseteq F^+$
 - Tính đơn điệu: $F \subseteq G \Rightarrow F^+ \subseteq G^+$
 - Tính lũy đẳng: $F^+ = F^{++}$

Bao đóng của tập thuộc tính

- Bao đóng của tập thuộc tính X ($X \subseteq Q^+$ đối với F (ký hiệu X^+ hoặc X_F^+) là tập tất cả các thuộc tính có thể xác định được từ tập thuộc tính X dựa trên tập phụ thuộc hàm F và hệ tiên đề Armstrong.

$$X^+ = X_F^+ = \{A \mid A \in Q^+, X \rightarrow A \in F^+\}$$

- Các tính chất của bao đóng của tập thuộc tính

Các tính chất của bao đóng của tập thuộc tính

- Tính phản xạ $X \subseteq X^+$
- Tính đơn điệu $X \subseteq Y \Rightarrow X^+ \subseteq Y^+$
- Tính lũy đẳng $X^{++} = X^+$
- 4 $X^+Y^+ \subseteq (XY)^+$
- 5 $(X^+Y)^+ = (XY^+)^+ = (X^+Y^+)^+$
- 6 $f : X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X_F^+$
- 7 $X \rightarrow Y \Leftrightarrow Y^+ \subseteq X^+$

Thuật toán tính bao đóng của tập TT

- Dữ liệu vào: $Q, F, X \subseteq Q^+$
- Dữ liệu ra: X^+ (hay X_F^+)

B1: $X^+ = X$

B2: Temp = X^+

$\forall f : U \rightarrow V \in F :$

if ($U \subseteq X^+$) Begin

$$X^+ = X^+ \cup V$$

$$F = F \setminus \{f\}$$

End

B3: if ($X^+ = \text{Temp}$)

X^+ là bao đóng của X

else go_to_B2

Thuật toán tính bao đóng của tập TT - VD

- Dữ liệu vào: $Q, F, X \subseteq Q^+$

$$Q^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \\ \text{Khoa}, \text{MaMH}, \text{Diem}\}$$

$$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}$$

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$f_4 : \text{MaSV}, \text{LopSH} \rightarrow \text{HoTen}, \text{Khoa}$$

$$F = \{f_1, f_2, f_3, f_4\}$$

Thuật toán tính bao đóng của tập TT: VD1

Cho: $X = \{\text{MaSV}, \text{NgaySinh}\} \subseteq Q^+ . X^+ = ?$

$$\text{B1: } X^+ = \{\text{MaSV}, \text{NgaySinh}\}$$

$$\text{B2: Temp} = X^+ = \{\text{MaSV}, \text{NgaySinh}\}$$

$$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}$$

$$\Rightarrow X^+ = \{\text{MaSV}, \text{NgaySinh}\} \cup \{\text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$F = F \setminus \{f_1\} = \{f_2, f_3, f_4\}$$

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$\Rightarrow X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\} \cup \{\text{Khoa}\}$$

$$X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$F = F \setminus \{f_2\} = \{f_3, f_4\}$$

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem} \quad \text{- không thoả}$$

$$f_4 : \text{MaSV}, \text{LopSH} \rightarrow \text{Hoten}, \text{Khoa}$$

$$F = F \setminus \{f_4\} = \{f_3\}$$

$$\text{B3: Temp} \neq X^+$$

$$\text{B2: Temp} = X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

Tất cả $\{f_3\}$ không thoả

$$\text{B3: Temp} = X^+ \Rightarrow$$

$$\Rightarrow \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

là bao đóng của X

Thuật toán tính bao đóng của tập TT: VD2

Cho: $X = \{\text{MaSV}, \text{MaMH}\} \subseteq Q^+ . X^+ = ?$

B1: $X^+ = \{\text{MaSV}, \text{MaMH}\}$

B2: Temp = $X^+ = \{\text{MaSV}, \text{MaMH}\}$

$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}$

$\Rightarrow X^+ = \{\text{MaSV}, \text{MaMH}\} \cup \{\text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$

$X^+ = \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$

$F = F \setminus \{f_1\} = \{f_2, f_3, f_4\}$

$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$

$\Rightarrow X^+ = \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\} \cup \{\text{Diem}\}$

$X^+ = \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{Diem}\}$

$F = F \setminus \{f_3\} = \{f_2, f_4\}$

Thuật toán tính bao đóng của tập TT: VD2

$f_2 : \text{LopSH} \rightarrow \text{Khoa}$

Cho: $X = \{\text{MaSV}, \text{MaMH}\} \subseteq Q^+ . X^+ = ?$
 $\Rightarrow X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\} \cup \{\text{Khoa}\}$

$$X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$F = F \setminus \{f_2\} = \{f_4\}$$

$f_4 : \text{MaSV}, \text{LopSH} \rightarrow \text{HoTen}, \text{Khoa}$

$$\Rightarrow X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\} \cup \{\text{HoTen}, \text{Khoa}\}$$

$$X^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$F = F \setminus \{f_4\} = \{\}$$

B3: $\text{Temp} \neq X^+$

B2: $\text{Temp} = X^+ = \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{Diem}\}$

Tất cả $\{\}$ không thoả

B3: $\text{Temp} = X^+ \Rightarrow$

là bao đóng của X

$$\Rightarrow \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{Diem}\}$$

Bài toán thành viên

- Cho tập phụ thuộc hàm F và phụ thuộc hàm f .

$$f \in F^+ ?$$

- Vì F có thể nhỏ, nhưng F^+ có thể sẽ rất lớn, và trong trường hợp chung, có thể nói, tính F^+ là không thể! Trong khi đó, bao đóng của tập thuộc tính thường có số phần tử là nhỏ, có thể tính được! Ta có:

$$f : X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X_F^+$$

- Vậy, để xác định $f \in F^+$ hay không, ta chỉ cần kiểm tra $Y \subseteq X_F^+$ hay không!

Khoá và thuật toán tìm khoá

- Định nghĩa khoá
- Thuật toán tìm một khoá
- Thuật toán tìm tất cả các khoá

Key - Khoá

- Cho $Q(Q_1, Q_2, \dots, Q_n), Q^+ = \{Q_1, Q_2, \dots, Q_n\}$
- $K \subseteq Q^+$ được gọi là khoá của Q nếu thoả 2 điều kiện:

$$K^+ = Q^+$$

$$\neg \exists Z \subset K : Z^+ = Q^+$$

Cho: $Q^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH},$
 $\text{Khoa}, \text{MaMH}, \text{Diem}\}$

$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}$

$f_2 : \text{LopSH} \rightarrow \text{Khoa}$

$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$

$f_4 : \text{MaSV} \rightarrow \text{HoTen}, \text{Khoa}$

$F = \{f_1, f_2, f_3, f_4\}$

Key - Khoá – VD

$$K1 = \{\text{MaSV}\}$$

$$K1^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\} \neq Q^+$$

$\Rightarrow K1$ không là khoá

$$K2 = \{\text{MaSV}, \text{MaMH}\}$$

$$K2^+ = \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{Diem}\} = Q^+$$

$$K21 = \{\text{MaSV}\}, K22 = \{\text{MaMH}\}, K21 \subset K2, K22 \subset K2$$

$$K21^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\} \neq Q^+$$

$$K22^+ = \{\text{MaMH}\} \neq Q^+$$

$\Rightarrow K2$ là khoá

Key - Khoá – VD

$$K3 = \{\text{MaSV}, \text{MaMH}, \text{NgaySinh}\}$$

$$K3^+ = \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{Diem}\} = Q^+$$

$$K31 = \{\text{MaSV}, \text{MaMH}\}, K32 = \dots, K31 \subset K3, K32 \dots$$

$$K31^+ = \{\text{MaSV}, \text{MaMH}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{Diem}\} = Q^+$$

$\Rightarrow K3$ không là khoá

Thuật toán tìm một khoá bất kỳ

- Ý tưởng:
 - B1. $X = Q^+$
 - B2. Loại bỏ một thuộc tính A từ X. Nếu đã thử qua tất cả các thuộc tính, chuyển sang B4.
 - B3. Tính X^+ . Nếu $X^+ = Q^+ \rightarrow$ Có thể loại bỏ A. Ngược lại, $X^+ \neq Q^+$, không thể loại bỏ A (Vì X còn lại không chứa khoá). Chuyển sang B2.
 - B4. X là khoá của Q
-
- Chú ý: Với thứ tự loại A từ X khác nhau, ta có thể thu được một trong các khoá khác nhau nếu Q có nhiều khoá!!!!

Thuật toán tìm một khoá bất kỳ: VD

- $Q^+ = \{MaSV, HoTen, NgaySinh, CMND\}$
- $F = \{f1: M \rightarrow HNC; f2: M \rightarrow C; f3: C \rightarrow M; f4: MH \rightarrow C; f5: MN \rightarrow H\}$

Loại	X	X+	X+=Q+?	Loại?	X còn
M	HCN	MHCN	T	T	HCN
H	CN	MHCN	T	T	CN
C	N	N	F	F	CN
N	C	MHCN	T	T	C

Loại	X	X+	X+=Q+?	Loại?	X còn
N	MHC	MHCN	T	T	MHC
C	MH	MHCN	T	T	MH
H	M	MHNC	T	T	M

Thuật toán tìm tất cả các khoá (2)

- Ý tưởng:
- Tìm tất cả các tập con khác rỗng của Q^+ . Tính bao đóng của mỗi tập con đó. Nếu bao đóng bằng Q^+ , có khả năng là khoá. Ngược lại chắc chắn không, loại bỏ. Tiếp theo, xét mỗi tập thu được, nếu có tập con thực sự có bao đóng bằng Q^+ , loại bỏ. Các tập còn lại là các khoá.

Thuật toán tìm tất cả các khoá (1)

- Ý tưởng:
- B1. Tìm tất cả các tập con khác rỗng của Q^+
 $K^1, K^2, \dots, K^i, \dots, K^j, \dots, K^{m=2^n-1}$
- B2. Tính bao đóng của tất cả các tập đó. Nếu $(K^j)^+ \neq Q^+$, loại K^j . Thu được $K^1, K^2, \dots, K^h, \dots, K^i, \dots, K^{k \leq m}$.
- B3. Xét mỗi K^h . Nếu $\exists K^i \subset K^h$, loại K^h .
- B4. Còn lại, $K^1, K^2, \dots, K^i, \dots, K^{l \leq k}$ là các khoá của Q.

Thuật toán tìm tất cả các khoá – cải tiến

- Ý tưởng:
- Các thuộc tính chỉ tham gia ở vế trái của các phụ thuộc hàm hoặc hoàn toàn không tham gia ở vế phải, chắc chắn là thành phần của khoá. Gọi tập các thuộc tính này là tập nguồn - TN.
- Các thuộc tính chỉ tham gia ở vế phải của các phụ thuộc hàm, chắc chắn không phải là thành phần khoá. Gọi tập các thuộc tính này là tập đích - TĐ.
- Các thuộc tính tham gia vừa ở vế phải, vừa ở vế trái của các phụ thuộc hàm, có thể là thành phần của khoá hoặc là không. Gọi tập các thuộc tính này là tập trung gian - TG (thường với số lượng thuộc tính nhỏ hơn X nhiều).

Thuật toán tìm tất cả các khoá – cải tiến

- Ý tưởng: Tìm các tập con của TG rồi kết hợp với TN :

- B1. Tìm tất cả các tập con (kể cả tập rỗng) của TG :
 $TG^1, TG^2, \dots, TG^i, \dots, TG^j, \dots, TG^{m=2^l}$

Kết hợp với TN : $K^i = TN \cup TG^i, i = 1 \dots m$

- B2. Tính bao đóng của tất cả các tập đó. Nếu $(K^j)^+ \neq Q^+$, loại K^j . Thu được $K^1, K^2, \dots, K^h, \dots, K^i, \dots, K^{k \leq m}$.
- B3. Xét mỗi K^h . Nếu $\exists K^i \subset K^h$, loại K^h .
- B4. Còn lại, $K^1, K^2, \dots, K^i, \dots, K^{l \leq k}$ là các khoá của Q .
- *Nhận xét*: Số lượng tập con phải xét để kiểm tra có phải là khoá hay không sẽ giảm với số lần:

$$s = \frac{2^{|Q^+|} - 1}{2^{|TG|}} \approx 2^{|Q^+| - |TG|}$$

Thuật toán tìm tất cả các khoá: VD

- $Q^+ = \{MaSV, HoTen, NgaySinh, CMND\}$
- $F = \{f1: M \rightarrow HNC; f2: M \rightarrow C; f3: C \rightarrow M; f4: MH \rightarrow C; f5: M \rightarrow H\}$

$TN = \{\}$ $TG = \{M, C, H\}$

STT	TG _i	(TN U TG _i)	(TN U TG _i) ⁺	SK: (TN U TG _i) ⁺ = Q ⁺	K?
1					
2	M	M	MHCN	+	+
3	C	C	MHCN	+	+
4	H	H	H		
5	MC	MC	MHCN	+	
6	MH	MH	MHCN	+	
7	HC	HC	MHCN	+	
8	MHC	MHC	MHCN	+	

Thuật toán tìm tất cả các khoá – cải tiến: VD

$$Q^+ = \{A, B, C, D, E, G, H\}$$

$$F = \{ABD \rightarrow CH, BC \rightarrow EG, CE \rightarrow BD\}$$

- Tìm tất cả các khoá

Gợi ý:

- 3 khoá: $\{A, B, C\}$, $\{A, C, E\}$, $\{A, B, D\}$
- Đạt dạng chuẩn 2, không đạt dạng chuẩn 3

Thuật toán tìm tất cả các khoá – cải tiến: VD

- $TN = \{A\}$ $TG = \{B, C, D, E\}$

STT	TG _i	(TN U TG _i)	(TN U TG _i) ⁺	SK: (TN U TG _i) ⁺ = Q ⁺	K?
1		A	A		
2	B	AB	AB		
3	C	AC	AC		
4	D	AD	AD		
5	E	AE	AE		
6	CD	ACD	ACD		
7	BC	ABC	ABCDEGH	+	+
8	BE	ABE	ABE		
9	CE	ACE	ABCDEGH	+	+
10	DE	ADE	ADE		
11	BD	ABD	ABCDEGH	+	+
12	BDE	ABDE	ABCDEGH	+	
13	CDE	ACDE	ABCDEGH	+	
14	BCE	ABCE	ABCDEGH	+	
15	BCD	ABCD	ABCDEGH	+	
16	BCDE	ABCDE	ABCDEGH	+	

Normal Forms - Các dạng chuẩn

- First Normal Form - 1NF - Dạng chuẩn 1
- Second Normal Form - 2NF - Dạng chuẩn 2
- Third Normal Form - 3NF - Dạng chuẩn 3
- Boyce-Codd Normal Form - BCNF - Dạng chuẩn Boyce-Codd
- Chú ý:
 - Khoá - là tập hợp tối thiểu các cột cho phép phân biệt hai bản ghi với nhau
 - Khoá chính - Mỗi quan hệ có thể có nhiều khoá. Khoá được chọn để sử dụng gọi là khoá chính

Minimal cover - Phủ tối thiểu

- Tập phụ thuộc hàm tương đương
- Phủ tối thiểu
- Thuật toán tìm phủ tối thiểu

Tập phụ thuộc hàm tương đương

- Tập phụ thuộc hàm tương đương: F và G được cho là tương đương, nếu và chỉ nếu:

$$\forall f \in F \Rightarrow f \in G^+$$

$$\forall g \in G \Rightarrow g \in F^+$$

Để xác định $f \in G^+$ hoặc $g \in F^+$ hay không, ta dùng tính chất:

$$f : X \rightarrow Y \in G^+ \Leftrightarrow Y \subseteq X_G^+$$

Tập phụ thuộc hàm tương đương - VD

- Tập phụ thuộc hàm tương đương: F và G được cho là tương đương, nếu và chỉ nếu:

$$\forall f \in F \Rightarrow f \in G^+$$

$$\forall g \in G \Rightarrow g \in F^+$$

Ví dụ, hai tập phụ thuộc hàm sau đây là tương đương:

$$Q^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{MaMH}, \text{Diem}\}$$

$$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}$$

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$f_4 : \text{MaSV}, \text{NgaySinh} \rightarrow \text{HoTen}, \text{Khoa}$$

$$F = \{f_1, f_2, f_3, f_4\}$$

$$g_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{LopSH}$$

$$g_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$g_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$g_4 : \text{MaSV} \rightarrow \text{NgaySinh}, \text{Khoa}$$

$$G = \{g_1, g_2, g_3, g_4\}$$

Khái niệm phủ tối thiểu

- F được gọi là phủ tối thiểu, nếu F thoả đồng thời 3 điều kiện:

- 1) $\forall f \in F : X \rightarrow A \Rightarrow A$ là một thuộc tính
- 2) $\neg \exists f \in F : X \rightarrow A \wedge Z \subset X \Rightarrow (F \setminus \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+ \cong F^+$
- 3) $\neg \exists f \in F \Rightarrow (F \setminus \{f\})^+ \cong F^+$

- 1) Cho phép loại bỏ các vế phải dư thừa
- 2) Cho phép loại bỏ các vế trái dư thừa
- 3) Cho phép loại bỏ các PTH dư thừa

- Chú ý: Phủ tối thiểu có ý nghĩa quan trọng trong việc thiết kế CSDL. Cụ thể, giúp thực hiện chuẩn hoá lược đồ CSDL.

Thuật toán tìm phủ tối thiểu – B1

- B1. Loại bỏ các vế phải dư thừa

$$G = \emptyset$$

$\forall f \in F: X \rightarrow Y$ (áp dụng luật phân rã)

Begin

$\forall A \subset Y:$ (A là một thuộc tính)

$$G = G \cup \{X \rightarrow A\}$$

$$F = F \setminus \{f\}$$

End

$$F = G$$

- B2. Loại bỏ các vế trái dư thừa

Thuật toán tìm phủ tối thiểu – B2

- B2. Loại bỏ các vế trái dư thừa

$$\forall f \in F: X \rightarrow A$$

Begin

$\forall B \subset X:$ (B là một thuộc tính)

Begin

$$G = F$$

$$\text{if } (F \setminus \{X \rightarrow A\} \cup \{(X \setminus \{B\}) \rightarrow A\})^+ \cong G^+$$

$$F = F \setminus \{f\} \cup \{(X \setminus \{B\}) \rightarrow A\}$$

End

End

- B3. Loại bỏ các PTH dư thừa

Thuật toán tìm phủ tối thiểu – B3

- B3. Loại bỏ các PTH dư thừa

$\forall f \in F$

Begin

$G = F$

if $(F \setminus \{f\})^+ \cong G^+$

$F = F \setminus \{f\}$

End

Thuật toán tìm phủ tối thiểu – VD

- Yêu cầu: Tìm phủ tối thiểu của F

$$Q^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}, \text{MaMH}, \text{Diem}\}$$

$$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}$$

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$f_4 : \text{MaSV}, \text{NgaySinh} \rightarrow \text{HoTen}, \text{Khoa}$$

$$f_5 : \text{MaSV} \rightarrow \text{HoTen}, \text{LopSH}$$

$$f_6 : \text{MaSV}, \text{LopSH} \rightarrow \text{Khoa}$$

$$f_7 : \text{MaSV}, \text{MaMH}, \text{LopSH} \rightarrow \text{Diem}$$

$$f_8 : \text{MaSV} \rightarrow \text{NgaySinh}, \text{Khoa}$$

$$F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$$

Thuật toán tìm phủ tối thiểu – VD: B1

- B1: Dùng luật phân rã để tách vế phải

$$f_{11} : \text{MaSV} \rightarrow \text{HoTen}$$

$$f_{12} : \text{MaSV} \rightarrow \text{NgaySinh}$$

$$f_{13} : \text{MaSV} \rightarrow \text{LopSH}$$

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$f_{41} : \text{MaSV}, \text{NgaySinh} \rightarrow \text{HoTen}$$

$$f_{42} : \text{MaSV}, \text{NgaySinh} \rightarrow \text{Khoa}$$

$$f_{51_x} : \text{MaSV} \rightarrow \text{HoTen}$$

$$f_{52_x} : \text{MaSV} \rightarrow \text{LopSH}$$

$$f_6 : \text{MaSV}, \text{LopSH} \rightarrow \text{Khoa}$$

$$f_7 : \text{MaSV}, \text{MaMH}, \text{LopSH} \rightarrow \text{Diem}$$

$$f_{81_x} : \text{MaSV} \rightarrow \text{NgaySinh}$$

$$f_{82} : \text{MaSV} \rightarrow \text{Khoa}$$

$$F = \{ f_{11}, f_{12}, f_{13}, f_2, f_3, f_{41}, f_{42}, f_6, f_7, f_{82} \}$$

Thuật toán tìm phủ tối thiểu – VD: B2

- B2: Loại bỏ vế trái dư thừa
- *Chỉ xét các PTH có vế trái nhiều hơn 1 thuộc tính

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$f_{41} : \text{MaSV}, \text{NgaySinh} \rightarrow \text{HoTen}$$

$$f_{42} : \text{MaSV}, \text{NgaySinh} \rightarrow \text{Khoa}$$

$$f_6 : \text{MaSV}, \text{LopSH} \rightarrow \text{Khoa}$$

$$f_7 : \text{MaSV}, \text{MaMH}, \text{LopSH} \rightarrow \text{Diem}$$

$$F = \{f_{11}, f_{12}, f_{13}, f_2, f_3, f_{41}, f_{42}, f_6, f_7, f_{82}\}$$

Thuật toán tìm phủ tối thiểu – VD: B2

- B2: Loại bỏ vế trái dư thừa

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$f_{3_g} : \text{MaMH} \rightarrow \text{Diem}$$

$$G = F \setminus \{f_3\} \cup \{f_{3_g}\}$$

$$X = \{\text{MaMH}\} \quad X_F^+ = \{\text{MaMH}\}$$

$$\{\text{Diem}\}_{\neg} \subseteq X_F^+ \Rightarrow f_{3_g} \notin F^+ \Rightarrow G_{\neg} \cong F$$

- \Rightarrow Không thể loại bỏ MaSV

$$f_{3_g} : \text{MaSV} \rightarrow \text{Diem}$$

$$G = F \setminus \{f_3\} \cup \{f_{3_g}\}$$

$$X = \{\text{MaSV}\} \quad X_F^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$\{\text{Diem}\}_{\neg} \subseteq X_F^+ \Rightarrow f_{3_g} \notin F^+ \Rightarrow G_{\neg} \cong F$$

- \Rightarrow Không thể loại bỏ MaMH

Thuật toán tìm phủ tối thiểu – VD: B2

- B2: Loại bỏ vế trái dư thừa

$$f_{41} : \text{MaSV}, \text{NgaySinh} \rightarrow \text{HoTen}$$

$$f_{41_g} : \text{MaSV} \rightarrow \text{HoTen}$$

$$G = F \setminus \{f_{41}\} \cup \{f_{41_g}\}$$

$$X = \{\text{MaSV}\} \quad X_F^+ = \{\text{MaSV}, \text{HoTen}, \text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$\{\text{HoTen}\} \subseteq X_F^+ \Rightarrow f_{41_g} \in F^+ : \text{MaSV} \rightarrow \text{HoTen} \Rightarrow F \cong G$$

\Rightarrow Loại bỏ NgaySinh

$$f_{41_g} : \text{NgaySinh} \rightarrow \text{HoTen}$$

$$G = F \setminus \{f_{41}\} \cup \{f_{41_g}\}$$

$$X = \{\text{NgaySinh}\} \quad X_F^+ = \{\text{NgaySinh}\}$$

$$\{\text{HoTen}\} \not\subseteq X_F^+ \Rightarrow f_{41_g} \notin F^+ \Rightarrow G \not\cong F$$

\Rightarrow Không thể loại bỏ MaSV

Thuật toán tìm phủ tối thiểu – VD: B2

- B2: Loại bỏ vế trái dư thừa
- Tương tự, cuối B2 sẽ thu được:

$$f_{11} : \text{MaSV} \rightarrow \text{HoTen}$$

$$f_{12} : \text{MaSV} \rightarrow \text{NgaySinh}$$

$$f_{13} : \text{MaSV} \rightarrow \text{LopSH}$$

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$f_{41_g} : \text{MaSV} \rightarrow \text{Khoa}$$

$$F = \{f_{11}, f_{12}, f_{13}, f_2, f_3, f_{41_g}\}$$

Thuật toán tìm phủ tối thiểu – VD: B3

- B3: Loại PTH dư thừa

$$f_{11} : \text{MaSV} \rightarrow \text{HoTen}$$

$$G = F \setminus \{f_{11}\}$$

$$X = \{\text{MaSV}\} X_G^+ = \{\text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$\{\text{HoTen}\} \not\subseteq X_G^+ \Rightarrow f_{11} \notin G^+$$

\Rightarrow Không thể loại bỏ

$$f_{41_g} : \text{MaSV} \rightarrow \text{Khoa}$$

$$G = F \setminus \{f_{41_g}\}$$

$$X = \{\text{MaSV}\} X_G^+ = \{\text{NgaySinh}, \text{LopSH}, \text{Khoa}\}$$

$$\{\text{Khoa}\} \subseteq X_G^+ \Rightarrow f_{41_g} \in G^+$$

$$F \cong G$$

\Rightarrow Loại bỏ f_{41_g}

Thuật toán tìm phủ tối thiểu – VD: B3

- B3: Loại PTH dư thừa
- Thu được phủ tối thiểu:

$$f_{11} : \text{MaSV} \rightarrow \text{HoTen}$$

$$f_{12} : \text{MaSV} \rightarrow \text{NgaySinh}$$

$$f_{13} : \text{MaSV} \rightarrow \text{LopSH}$$

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

$$f_3 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

$$F = \{f_{11}, f_{12}, f_{13}, f_2, f_3\}$$

First Normal Form – 1NF – Dạng chuẩn 1

Theo Date, một quan hệ được cho là thuộc 1NF nếu thoả các điều kiện:

1. Không có các bản ghi trùng nhau
 2. Mỗi trường (giao điểm giữa hàng và cột) chỉ chứa một giá trị từ miền giá trị hợp quy - chỉ chứa giá trị nguyên tố, không thể phân chia
 3. Không quan trọng thứ tự các cột
 4. Không quan trọng thứ tự các bản ghi
 5. Tất cả các cột là chuẩn, nghĩa là không có các cột ẩn như RowID, ObjectID...
- Tất cả các quan hệ đều thuộc 1NF (xem định nghĩa quan hệ)

First Normal Form – 1NF – Dạng chuẩn 1

Ngoài ra, còn có định nghĩa

- Một quan hệ được cho là thuộc 1NF nếu mỗi trường chỉ chứa một giá trị từ miền giá trị hợp quy - chỉ chứa giá trị nguyên tố, không thể phân chia
- Một quan hệ được cho là thuộc 1NF nếu không chứa các thuộc tính lặp

First Normal Form – 1NF – Dạng chuẩn 1

- Ví dụ, các “quan hệ” không đạt 1NF:

Số Hoá Đơn	Mã HH	Số Lượng	Đơn Giá
HD1	H1	10	5
	H2	4	7
HD2	H1	3	5

Mã SV	Họ và Tên		Lớp
	Họ	Tên	
SV1	Họ 1	Tên 1	Lop1
SV2	Họ 2	Tên 2	Lop 2

First Normal Form – 1NF – Dạng chuẩn 1

- Ví dụ, các “quan hệ” đạt 1NF:

Số Hoá Đơn	Mã HH	Số Lượng	Đơn Giá
HD1	H1	10	5
HD1	H2	4	7
HD2	H1	3	5

Mã SV	Họ	Tên	Lớp
SV1	Họ 1	Tên 1	Lop1
SV2	Họ 2	Tên 2	Lop 2

Mã SV	Họ và Tên	Lớp
SV1	Họ 1 Tên 1	Lop1
SV2	Họ 2 Tên 2	Lop 2

Second Normal Form – 2NF – Dạng chuẩn 2

- Một quan hệ được cho là đạt 2NF nếu đạt 1NF và không có cột không khoá phụ thuộc một phần vào khoá

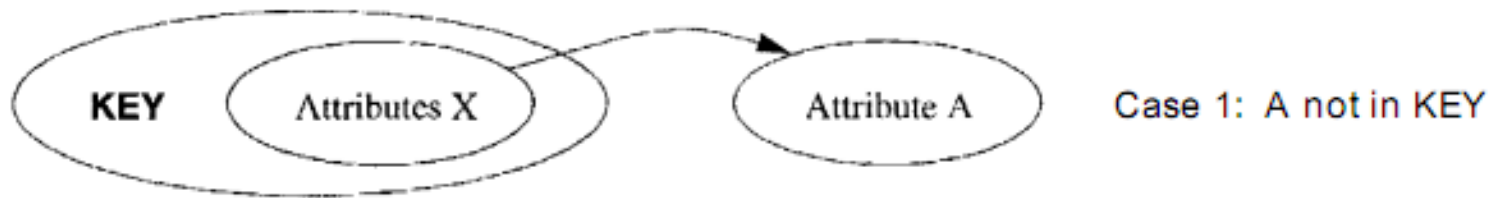


Figure 19.7 Partial Dependencies

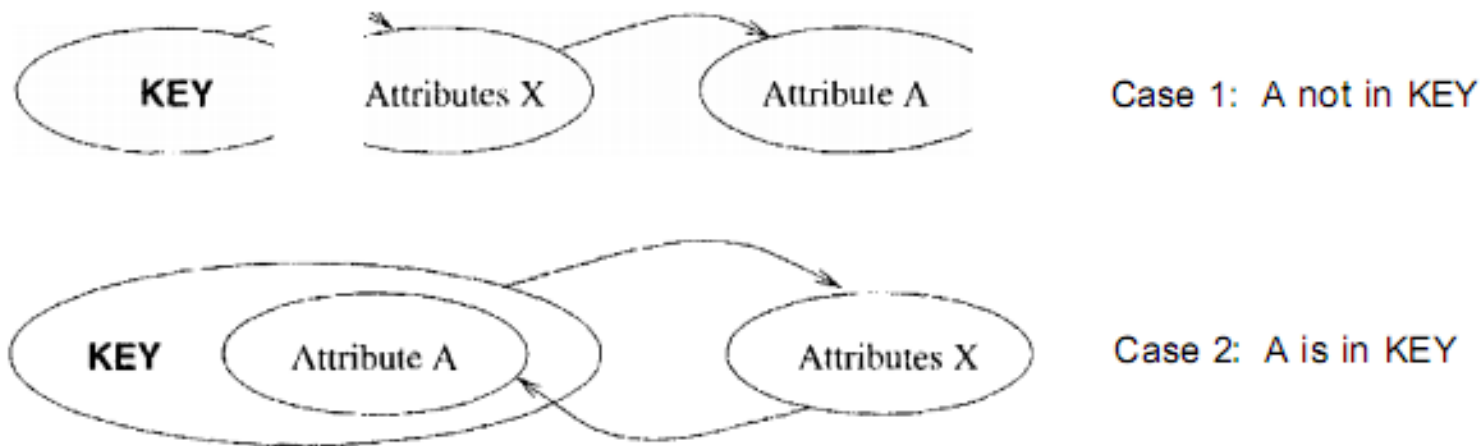


Figure 19.8 Transitive Dependencies

Second Normal Form – 2NF – Dạng chuẩn 2

- Ví dụ quan hệ không đạt 2NF

Quan hệ HOC (SV n-1 LopSH, SV n-n MH, 1 SV – 1 MH: 1 lần)

MaSV	HoTen	NgaySinh	MaMH	Diem
S1	Sinh Viên 1	NS1	M1	6
S1	Sinh Viên 1	NS1	M2	7
S1	Sinh Viên 1	NS1	M3	7
S2	Sinh Viên 2	NS2	M1	7
S3	Sinh Viên 3	NS2	M1	5

$$F = \{f_1, f_2\}$$

$$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}$$

$$f_2 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$$

- Khoá: $K = \{\text{MaSV}, \text{MaMH}\}$
- Cột khoá: MaSV, MaMH
- Cột không khoá: HoTen, NgaySinh, Diem

- Các cột không khoá HoTen, NgaySinh phụ thuộc vào MaSV – một phần của khoá
- Đạt 1NF nhưng vẫn gặp các bất thường

Second Normal Form – 2NF – Dạng chuẩn 2

- Dùng phương pháp tách không mất thông tin

$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}$

$\text{SINH_VIEN}(\text{MaSV}, \text{HoTen}, \text{NgaySinh})$

MaSV	HoTen	NgaySinh
S1	Sinh Viên 1	NS1
S2	Sinh Viên 2	NS2
S3	Sinh Viên 3	NS2

$f_2 : \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}$

$\text{DIEM}(\text{MaSV}, \text{MaMH}, \text{Diem})$

MaSV	MaMH	Diem
S1	M1	6
S1	M2	7
S1	M3	7
S2	M1	7
S3	M1	5

Third Normal Form – 3NF – Dạng chuẩn 3

- Phụ thuộc bắt cầu: A phụ thuộc bắt cầu vào X nếu:

$$X \rightarrow Y$$

$$Y \rightarrow A$$

$$Y \not\rightarrow X$$

$$A \notin XY$$

- Định nghĩa 1. Một quan hệ được cho là đạt 3NF, nếu đạt 2NF và không có các thuộc tính không khoá phụ thuộc bắt cầu vào khoá (không có các thuộc tính không khoá phụ thuộc lẫn nhau)
- Định nghĩa 2. Một quan hệ được cho là đạt 3NF, nếu mọi PTH $X \rightarrow A \in F^+$ với $A \notin X$ đều có:
 - Hoặc X là siêu khóa
 - Hoặc A là thuộc tính khoá

Third Normal Form – 3NF – Dạng chuẩn 3

- → - f_2 : Thuộc tính không khoá phụ thuộc lẫn nhau
- - Quan hệ thuộc 2NF, nhưng vẫn gặp các bất thường

MaSV	HoTen	NgaySinh	LopSH	Khoa
S1	Sinh Viên 1	NS1	L1	Khoa 1
S2	Sinh Viên 2	NS2	L1	Khoa 1
S3	Sinh Viên 3	NS2	L2	Khoa 2

$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}$

$f_2 : \text{LopSH} \rightarrow \text{Khoa}$

$F = \{f_1, f_2\}$

$K = \{\text{MaSV}\}$

Third Normal Form – 3NF – Dạng chuẩn 3

Dùng phương pháp tách không mất thông tin:

$$f_1 : \text{MaSV} \rightarrow \text{HoTen}, \text{NgaySinh}, \text{LopSH}$$

MaSV	HoTen	NgaySinh	LopSH
S1	Sinh Viên 1	NS1	L1
S2	Sinh Viên 2	NS2	L1
S3	Sinh Viên 3	NS2	L2

$$f_2 : \text{LopSH} \rightarrow \text{Khoa}$$

LopSH	Khoa
L1	Khoa 1
L2	Khoa 2

Boyce-Codd Normal Form – BCNF – Dạng chuẩn Boyce-Codd

- Một quan hệ được cho là đạt BCNF, nếu đạt 3NF và tất cả các định thức đều là khoá
- Định thức - vế trái của phụ thuộc hàm

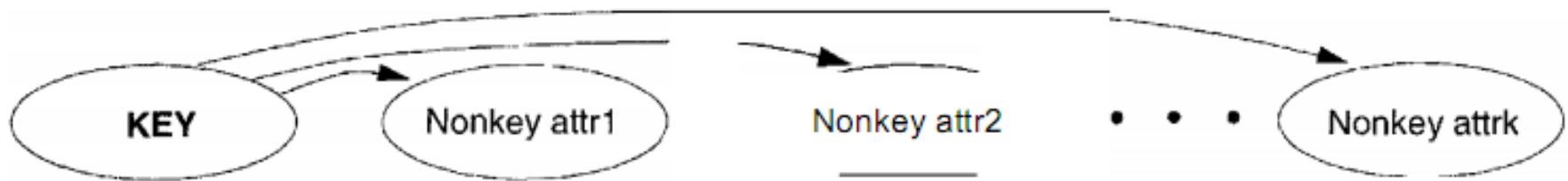


Figure 19.5 FDs in a BCNF Relation

Boyce-Codd Normal Form – BCNF – Dạng chuẩn Boyce-Codd

- Ví dụ quan hệ đạt 3NF nhưng không đạt BCNF

LAM_VIEC

(Phòng ban n – n Nhân viên, $f_1 : \text{MaPB} \rightarrow \text{TenPB}$

1 Phòng ban – 1 Nhân viên: một lần, tên phòng ban không trùng lại)

$f_2 : \text{TenPB} \rightarrow \text{MaPB}$

$f_3 : \text{MaPB}, \text{MaNV} \rightarrow \text{TenPB}, \text{CV}$

$f_4 : \text{TenPB}, \text{MaNV} \rightarrow \text{MaPB}, \text{CV}$

$F = \{f_1, f_2, f_3, f_4\}$

$K1 = \{\text{MaPB}, \text{MaNV}\}$

$K2 = \{\text{TenPB}, \text{MaNV}\}$

MaPB	TenPB	MaNV	CV
P1	Phòng 1	NV1	c1
P1	Phòng 1	NV2	c1
P2	Phòng 2	NV1	c2

Boyce-Codd Normal Form – BCNF – Dạng chuẩn Boyce-Codd

- Dùng phép tách không mất thông tin để đạt BCNF

MaPB	TenPB
P1	Phòng 1
P2	Phòng 2

$$f_1 : \text{MaPB} \rightarrow \text{TenPB}$$

$$f_2 : \text{TenPB} \rightarrow \text{MaPB}$$

$$K1 = \{\text{MaPB}\}$$

$$K2 = \{\text{TenPB}\}$$

MaPB	MaNV	CV
P1	NV1	c1
P1	NV2	c1
P2	NV1	c2

$$f_3 : \text{MaPB}, \text{MaNV} \rightarrow \text{CV}$$

$$K = \{\text{MaPB}, \text{MaNV}\}$$

Các dạng chuẩn: BT

$$Q^+ = \{A, B, C, D, E, G, H\}$$

$$F = \{ABD \rightarrow CH, BC \rightarrow EG, CE \rightarrow BD\}$$

- Tìm tất cả các khoá
- Dạng chuẩn cao nhất của Q là gì? Vì sao? Nếu chưa đạt dạng chuẩn BCNF, hãy chuẩn hoá để đạt BCNF.

Các dạng chuẩn: BT

$$Q^+ = \{A, B, C, D, E, G, H\}$$

$$F = \{ABD \rightarrow CH, BC \rightarrow EG, CE \rightarrow BD\}$$

Tập PTH tương đương - phủ tối thiểu:

$$Z = \{ABD \rightarrow C, ABD \rightarrow H, BC \rightarrow E, BC \rightarrow G, CE \rightarrow B, CE \rightarrow D\}$$

- Tìm tất cả các khoá
- Dạng chuẩn cao nhất của Q là gì? Vì sao? Nếu chưa đạt dạng chuẩn 3NF, hãy chuẩn hoá để đạt 3NF.

Gợi ý:

- 3 khoá: $\{A, B, C\}$, $\{A, C, E\}$, $\{A, B, D\}$
- Đạt dạng chuẩn 1, không đạt dạng chuẩn 3

Các dạng chuẩn: BT

- Việc quản lý khám bệnh của công dân có thể được quy định bởi những ràng buộc sau đây. Mỗi CÔNG DÂN có thể có nhiều HỒ SƠ, mỗi HỒ SƠ thuộc một CÔNG DÂN. Mỗi CÔNG DÂN có thể ĐĂNG KÝ khám bệnh nhiều lần, mỗi ĐĂNG KÝ khám bệnh thuộc về một CÔNG DÂN. Việc đăng ký khám bệnh của công dân được xác định thông qua hồ sơ của công dân đó. Mỗi ĐĂNG KÝ do một NHÂN VIÊN phụ trách tiếp nhận và mỗi NHÂN VIÊN có thể lập nhiều ĐĂNG KÝ. Với mỗi ĐĂNG KÝ, công dân có thể sử dụng nhiều DỊCH VỤ khám chữa bệnh khác nhau và mỗi DỊCH VỤ khám chữa bệnh có thể được thực hiện cho nhiều ĐĂNG KÝ khác nhau. Mỗi DỊCH VỤ khám chữa bệnh có thể có nhiều ĐƠN GIÁ theo ngày áp dụng và mỗi ĐƠN GIÁ thuộc về một DỊCH VỤ khám chữa bệnh. Vậy, có thể xác định mỗi ĐĂNG KÝ sử dụng những DỊCH VỤ nào với ĐƠN GIÁ bao nhiêu bằng ràng buộc giữa ĐĂNG KÝ và ĐƠN GIÁ thay vì giữa ĐĂNG KÝ và DỊCH VỤ. Tất nhiên, cho mỗi đăng ký, nếu công dân sử dụng một dịch vụ nào đó thì chỉ một lần, do một bác sĩ phụ trách và kết quả được lưu lại.
- Biết rằng, MãCD/MãHS/MãĐK/MãĐG xác định các thuộc tính còn lại của CÔNG DÂN/HỒ SƠ/ĐĂNG KÝ/ĐƠN GIÁ. Hãy cho biết lược đồ quan hệ Q(MãCD, MãHS, MãĐK, MãĐG, NgàyÁpDụng) đạt dạng chuẩn cao nhất là gì. Hãy chuẩn hoá để các lược đồ quan hệ đạt BCNF.

Các dạng chuẩn: BT

- Biết rằng, MãCD/MãHS/MãĐK/MãĐG xác định các thuộc tính còn lại của CÔNG DÂN/HỒ SƠ/ĐĂNG KÝ/ĐƠN GIÁ. Hãy cho biết lược đồ quan hệ Q(MãCD, MãHS, MãĐK, MãĐG, NgàyÁpDụng) đạt dạng chuẩn cao nhất là gì. Hãy chuẩn hoá để các lược đồ quan hệ đạt BCNF.
- Gợi ý:
 - Ta có phụ thuộc hàm: MãĐK \rightarrow MãHS, MãHS \rightarrow MãCD
 - Khoá của lược đồ là {MãĐK, MãĐG}
 - Đạt dạng chuẩn 1, không đạt dạng chuẩn 2. Tách để đạt dạng chuẩn 2, sau đó lại tách để đạt dạng chuẩn 3 và BC:
 - Q1(MãĐK, MãĐG, NgàyÁpDụng)
 - Q2 (MãCD, MãHS, MãĐK)
 - Q21(MãHS, MãĐK)
 - Q22(MãHS, MãCD)