

## MỤC LỤC

<b>Chương I. TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU.....</b>	<b>9</b>
A. Mục tiêu.....	9
B. Phương pháp giảng dạy .....	9
C. Tài liệu tham khảo.....	9
D. Nội dung chi tiết.....	9
<b>I.1. Cơ sở dữ liệu .....</b>	<b>9</b>
I.1.1. Định nghĩa cơ sở dữ liệu.....	11
I.1.2. Các tính chất của một cơ sở dữ liệu .....	12
<b>I.2. Hệ quản trị cơ sở dữ liệu .....</b>	<b>14</b>
I.2.1. Định nghĩa hệ quản trị cơ sở dữ liệu .....	14
I.2.2. Các chức năng của một hệ HQTCSDDL .....	14
I.2.3. Các đặc trưng của giải pháp cơ sở dữ liệu.....	15
I.2.4. Ví dụ một cơ sở dữ liệu .....	15
<b>I.3. Mô hình cơ sở dữ liệu .....</b>	<b>16</b>
I.3.1. Các loại mô hình cơ sở dữ liệu .....	17
I.3.2. Lược đồ và trạng thái cơ sở dữ liệu.....	18
<b>I.4. Con người trong hệ quản trị cơ sở dữ liệu .....</b>	<b>19</b>
I.4.1. Người quản trị hệ cơ sở dữ liệu.....	19
I.4.2. Người thiết kế cơ sở dữ liệu.....	19
I.4.3. Những người sử dụng.....	19
I.4.4. Người phân tích hệ thống và lập trình ứng dụng .....	19
I.4.5. Người thiết kế và cài đặt hệ quản trị dữ liệu.....	19
I.4.6. Những người phát triển công cụ .....	19
I.4.7. Các thao tác viên và những người bảo trì.....	20
<b>I.5. Ngôn ngữ cơ sở dữ liệu và giao diện.....</b>	<b>20</b>
I.5.1. Các ngôn ngữ hệ quản trị cơ sở dữ liệu.....	20
I.5.2. Các loại giao diện của hệ quản trị cơ sở dữ liệu .....	20
<b>I.6. Tổng kết chương và câu hỏi ôn tập .....</b>	<b>21</b>

I.6.1. Tổng kết chương .....	21
I.6.2. Câu hỏi ôn tập .....	21
<b>Chương II. MÔ HÌNH THỰC THỂ - LIÊN KẾT .....</b>	<b>23</b>
A. Mục tiêu .....	23
B. Phương pháp giảng dạy .....	23
C. Tài liệu tham khảo .....	23
D. Nội dung chi tiết.....	23
<b>II.1. Sử dụng mô hình quan niệm bậc cao cho việc thiết kế cơ sở dữ liệu.....</b>	<b>24</b>
<b>II.2. Các thành phần cơ bản của mô hình thực thể-liên kết .....</b>	<b>25</b>
II.2.1. Thực thể.....	25
II.2.2. Tập thực thể.....	25
II.2.3. Thuộc tính.....	26
II.2.4. Khóa .....	27
II.2.5. Mối liên kết.....	28
II.2.5.1. Khái niệm mối liên kết .....	28
II.2.5.2. Bản số của mối liên kết.....	28
II.2.5.3. Các loại liên kết.....	29
II.2.5.4. Mối liên kết đệ qui .....	31
II.2.5.5. Mối liên kết đa ngôi.....	32
II.2.6. Thuộc tính của mối liên kết .....	32
II.2.7. Tập thực thể yếu.....	32
<b>II.3. Ví dụ về thiết kế mô hình thực thể-liên kết .....</b>	<b>33</b>
<b>II.4. Mô hình thực thể-liên kết mở rộng.....</b>	<b>35</b>
II.4.1. Lớp cha, lớp con và mối liên kết ISA .....	35
II.4.2. Tổng quát hóa, chuyên biệt hóa .....	36
II.4.3. Các ràng buộc trên mối liên kết ISA.....	37
<b>II.5. Tổng kết chương và câu hỏi ôn tập.....</b>	<b>39</b>
II.5.1. Tổng kết chương .....	39
II.5.2. Câu hỏi ôn tập.....	39
II.5.3. Bài tập .....	40
<b>Chương III. MÔ HÌNH QUAN HỆ.....</b>	<b>43</b>
A. Mục tiêu .....	43

B. Phương pháp giảng dạy .....	43
C. Tài liệu tham khảo .....	43
D. Nội dung chi tiết.....	43
<b>III.1. Các khái niệm trong mô hình quan hệ.....</b>	<b>44</b>
III.1.1. Các khái niệm .....	45
III.1.1.1. Thuộc tính.....	45
III.1.1.2. Quan hệ.....	45
III.1.1.3. Bộ giá trị.....	47
III.1.2. Các đặc trưng của quan hệ .....	47
<b>III.2. Các ràng buộc toàn vẹn trên quan hệ và lược đồ CSDL quan hệ .....</b>	<b>48</b>
III.2.1. Khóa - Siêu khóa .....	48
III.2.2. Khóa chính .....	49
III.2.3. Khóa ngoại .....	50
<b>III.3. Các phép toán trong mô hình quan hệ.....</b>	<b>52</b>
III.3.1. Đại số quan hệ .....	52
III.3.2. Các phép toán tập hợp.....	52
III.3.2.1. Phép hợp.....	53
III.3.2.2. Phép giao .....	53
III.3.2.3. Phép trừ .....	53
III.3.2.4. Phép tích Descartes .....	54
III.3.3. Các phép toán quan hệ .....	55
III.3.3.1. Phép chiếu .....	55
III.3.3.2. Phép chọn .....	56
III.3.3.3. Phép nối.....	57
III.3.3.4. Phép chia .....	60
III.3.4. Một số ví dụ về phép toán đại số quan hệ.....	62
<b>III.4. Chuyển mô hình thực thể-liên kết sang mô hình quan hệ.....</b>	<b>63</b>
III.4.1. Tập thực thể.....	63
III.4.2. Mối liên kết .....	64
III.4.3. Tập thực thể yếu .....	65
III.4.4. Thuộc tính đa trị .....	65
III.4.5. Mối liên kết đa ngôi .....	66
III.4.6. Mô hình hóa trường hợp mở rộng .....	66

<b>III.5. Tổng kết chương và câu hỏi ôn tập .....</b>	<b>69</b>
III.5.1. Tổng kết chương.....	69
III.5.2. Câu hỏi ôn tập.....	69
III.5.3. Bài tập .....	69
<b>Chương IV. TỔ CHỨC DỮ LIỆU VẬT LÝ .....</b>	<b>71</b>
A. Mục tiêu.....	71
B. Phương pháp giảng dạy .....	71
C. Tài liệu tham khảo .....	71
D. Nội dung chi tiết.....	71
<b>IV.1. Mô hình tổ chức bộ nhớ ngoài .....</b>	<b>71</b>
<b>IV.2. Tổ chức Tập băm và các thao tác trên tổ chức tập băm .....</b>	<b>72</b>
IV.2.1. Tổ chức tập băm .....	72
IV.2.2. Các thao tác trên tổ chức tập băm .....	74
IV.2.2.1. Tìm kiếm mẫu tin.....	74
IV.2.2.2. Thêm mới mẫu tin.....	74
IV.2.2.3. Xóa mẫu tin .....	74
IV.2.2.4. Sửa mẫu tin.....	74
<b>IV.3. Tổ chức đồng và thao tác trên tổ chức đồng .....</b>	<b>74</b>
IV.3.1. Tổ chức đồng.....	74
IV.3.2. Thao tác trên tổ chức đồng.....	75
IV.3.2.1. Tìm kiếm mẫu tin.....	75
IV.3.2.2. Thêm mẫu tin mới.....	75
IV.3.2.3. Xóa mẫu tin .....	75
IV.3.2.4. Sửa mẫu tin.....	75
<b>IV.4. Tổ chức tệp chỉ dẫn và thao tác trên tổ chức tệp chỉ dẫn .....</b>	<b>75</b>
IV.4.1. Tổ chức tệp chỉ dẫn .....	76
IV.4.2. Thao tác trên tổ chức tệp chỉ dẫn.....	76
IV.4.2.1. Tìm kiếm mẫu tin.....	76
IV.4.2.2. Thêm mẫu tin mới.....	77
IV.4.2.3. Xóa mẫu tin .....	77
IV.4.2.4. Sửa mẫu tin.....	77
<b>IV.5. Tổng kết chương và câu hỏi ôn tập.....</b>	<b>77</b>

IV.5.1. Tổng kết chương.....	77
IV.5.2. Câu hỏi ôn tập .....	77
<b>Chương V. NGÔN NGỮ TRUY VẤN DỮ LIỆU – SQL.....</b>	<b>79</b>
A. Mục tiêu.....	79
B. Phương pháp giảng dạy .....	79
C. Tài liệu tham khảo .....	79
D. Nội dung chi tiết.....	79
<b>V.1. Ngôn ngữ truy vấn dữ liệu - SQL.....</b>	<b>80</b>
V.1.1. Ngôn ngữ định nghĩa dữ liệu .....	80
V.1.2. Ngôn ngữ thao tác dữ liệu .....	80
V.1.3. Ngôn ngữ kiểm soát dữ liệu .....	80
<b>V.2. Biểu thức và hàm trong SQL .....</b>	<b>81</b>
V.2.1. Biểu thức .....	81
V.2.2. Hàm.....	81
V.2.2.1. Hàm xử lý chuỗi .....	81
V.2.2.2. Hàm thời gian .....	82
V.2.2.3. Các hàm khác.....	83
<b>V.3. Câu lệnh SQL .....</b>	<b>84</b>
V.3.1. Câu lệnh định nghĩa dữ liệu .....	85
V.3.1.1. Tạo mới một CSDL.....	85
V.3.1.2. Lệnh tạo mới một bảng.....	85
V.3.2. Câu lệnh thao tác dữ liệu.....	90
V.3.2.1. Câu lệnh bổ sung dữ liệu (INSERT) .....	90
V.3.2.2. Câu lệnh sửa dữ liệu (UPDATE) .....	91
V.3.2.3. Câu lệnh xóa dữ liệu (DELETE) .....	92
V.3.2.4. Câu lệnh truy xuất dữ liệu (SELECT).....	93
V.3.3. Câu lệnh kiểm soát dữ liệu.....	110
V.3.4. SQL và các toán tử đại số quan hệ.....	111
<b>V.4. Tổng kết chương và ôn tập.....</b>	<b>112</b>
V.4.1. Tổng kết chương.....	112
V.4.2. Câu hỏi ôn tập.....	112
V.4.3. Bài tập .....	113

<b>Chương VI. RÀNG BUỘC TOÀN VỆ DỮ LIỆU .....</b>	<b>116</b>
A. Mục tiêu .....	116
B. Phương pháp giảng dạy .....	116
C. Tài liệu tham khảo .....	117
D. Nội dung chi tiết.....	117
<b>VI.1. Ràng buộc toàn vẹn.....</b>	<b>117</b>
VI.1.1. Khái niệm ràng buộc toàn vẹn .....	117
VI.1.2. Các đặc trưng của RBTV .....	118
VI.1.2.1. Bối cảnh .....	118
VI.1.2.2. Nội dung.....	118
VI.1.2.3. Bảng tầm ảnh hưởng.....	118
VI.1.3. Giải pháp tránh vi phạm RBTV .....	119
<b>VI.2. Phân loại ràng buộc toàn vẹn .....</b>	<b>120</b>
VI.2.1. Ràng buộc toàn vẹn nội .....	120
VI.2.1.1. RBTV miền giá trị .....	120
VI.2.1.2. RBTV liên bộ .....	121
VI.2.1.3. RBTV liên thuộc tính.....	122
VI.2.2. Ràng buộc toàn vẹn ngoài.....	122
VI.2.2.1. RBTV khóa ngoại .....	122
VI.2.2.2. RBTV liên thuộc tính - liên quan hệ.....	123
VI.2.2.3. RBTV liên bộ - liên quan hệ .....	124
VI.2.2.4. RBTV chu trình .....	125
<b>VI.3. Tổng kết chương và câu hỏi ôn tập.....</b>	<b>126</b>
VI.3.1. Tổng kết chương.....	126
VI.3.2. Câu hỏi ôn tập .....	126
VI.3.3. Bài tập .....	127
<b>Chương VII. CHUẨN HÓA CƠ SỞ DỮ LIỆU .....</b>	<b>128</b>
A. Mục tiêu.....	128
B. Phương pháp giảng dạy .....	128
C. Tài liệu tham khảo .....	128
D. Nội dung chi tiết.....	128
<b>VII.1. Các nguyên tắc thiết kế lược đồ quan hệ .....</b>	<b>129</b>

VII.1.1. Nghĩa của các thuộc tính quan hệ.....	129
VII.1.2. Thông tin dư thừa trong các bộ và sự dị thường cập nhật.....	129
VII.1.3. Các giá trị không xác định trong các bộ.....	130
<b>VII.2. Phụ thuộc hàm.....</b>	<b>130</b>
VII.2.1. Định nghĩa phụ thuộc hàm.....	130
VII.2.2. Các quy tắc suy diễn đối với các phụ thuộc hàm.....	131
VII.2.2.1. Hệ tiên đề Amstrong.....	131
VII.2.2.2. Bao đóng của tập phụ thuộc hàm F.....	132
VII.2.2.3. Bao đóng của tập thuộc tính X.....	133
VII.2.3. Sự tương đương của các tập phụ thuộc hàm.....	134
VII.2.4. Tập phụ thuộc hàm tối thiểu (Phủ tối thiểu).....	135
VII.2.5. Khóa của lược đồ quan hệ.....	136
VII.2.5.1. Định nghĩa khóa của lược đồ quan hệ.....	136
VII.2.5.2. Thuật toán tìm một khóa.....	137
VII.2.5.3. Thuật toán tìm tất cả các khóa.....	138
<b>VII.3. Các dạng chuẩn.....</b>	<b>140</b>
VII.3.1. Dạng chuẩn 1 (1NF).....	141
VII.3.2. Dạng chuẩn 2 (2NF).....	142
VII.3.3. Dạng chuẩn 3 (3NF).....	143
VII.3.4. Dạng chuẩn Boyce-Codd (BCNF).....	143
<b>VII.4. Các thuật toán thiết kế cơ sở dữ liệu quan hệ và các dạng chuẩn cao hơn .</b>	<b>144</b>
VII.4.1. Các thuật toán thiết kế cơ sở dữ liệu quan hệ.....	<b>144</b>
VII.4.1.1. Phân rã có nối không mất thông tin (phân rã bảo toàn thông tin).....	144
VII.4.1.2. Phân rã bảo toàn phụ thuộc hàm.....	146
VII.4.1.3. Chuẩn hóa một lược đồ CSDL bằng cách tiếp cận phân rã.....	148
VII.4.2. Các phụ thuộc đa trị và dạng chuẩn 4.....	<b>151</b>
VII.4.2.1. Phụ thuộc đa trị.....	151
VII.4.2.2. Dạng chuẩn 4 (4NF).....	153
VII.4.3. Các phụ thuộc nối và dạng chuẩn 5.....	<b>154</b>
VII.4.3.1. Phụ thuộc nối.....	154
VII.4.3.2. Dạng chuẩn 5 (5NF).....	154
<b>VII.5. Tổng kết chương và câu hỏi ôn tập.....</b>	<b>155</b>

VII.5.1. Tổng kết chương .....	155
VII.5.2. Câu hỏi ôn tập .....	155
VII.5.3. Bài tập.....	155



## Chương I. TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

### A. Mục tiêu

#### ➤ Về kiến thức:

Sau khi xong chương này sinh viên nắm vững các kiến thức sau:

- Nguyên nhân ra đời của cơ sở dữ liệu
- Khái niệm về cơ sở dữ liệu, ưu điểm của cơ sở dữ liệu
- Khái niệm về hệ quản trị cơ sở dữ liệu, các đặc tính của hệ quản trị cơ sở dữ liệu.
- Khái niệm về mô hình dữ liệu, các loại mô hình dữ liệu. Ưu điểm của mô hình quan hệ so với các mô hình dữ liệu khác.

#### ➤ Về kỹ năng

Giúp sinh viên phát triển các kỹ năng:

- Mạnh dạn phát biểu ý kiến.
- Rèn luyện khả năng làm việc theo nhóm.

### B. Phương pháp giảng dạy

- Trình chiếu slides
- Đặt câu hỏi cho sinh viên trả lời
- Hướng sinh viên thảo luận theo nhóm

### C. Tài liệu tham khảo

1. Nguyễn An Tế, Đồng Thị Bích Thủy (1996). *Nhập Môn Cơ Sở Dữ Liệu*.
2. Nguyễn Đăng Ty, Đỗ Phúc (2003). *Giáo Trình Cơ Sở Dữ Liệu*.
3. Lê Tiến Vương (1994). *Nhập Môn Cơ Sở Dữ Liệu Quan Hệ*
4. Jeffrey D. Ullman (1998). *Mô Hình Dữ Liệu và Ngôn Ngữ Ván Tin*. Tập 1

### D. Nội dung chi tiết

#### I.1. Cơ sở dữ liệu

Trong những năm gần đây khi vấn đề tin học hóa ngày càng mở rộng, thuật ngữ *Cơ sở dữ liệu* (CSDL) đã trở nên khá quen thuộc không chỉ riêng với những người làm tin

học mà còn đối với những người làm trong các lĩnh vực khác như kinh tế, thương mại, khoa học, giáo dục, y tế... Có thể nói rằng Cơ sở dữ liệu là môn học về cách tổ chức lưu trữ và khai thác dữ liệu và là môn học bắt buộc đối với các sinh viên chuyên ngành tin học. Phải thừa nhận rằng không có một ứng dụng tin học nào mà không chứa đựng vấn đề về tổ chức lưu trữ và khai thác dữ liệu.

### **Tại sao cần phải có Cơ sở dữ liệu?**

Trong suốt một thời gian dài tại Việt Nam, vì nhiều lý do khác nhau, hầu hết các đơn vị kinh tế, hành chính sự nghiệp,... tin học hóa quản lý bằng cách sử dụng mô hình *hệ thống tập tin cổ điển*, chúng được tổ chức riêng rẽ, phục vụ cho một mục đích của một đơn vị con cụ thể. Lấy một ví dụ cụ thể như sau:

Tại công ty XYZ, người ta trang bị máy tính cho tất cả các phòng ban để phục vụ quản lý. Phòng Kế toán sử dụng máy tính để tính lương và in danh sách lương của từng bộ phận dựa trên danh sách nhân viên cùng với hệ số lương và phụ cấp do phòng tổ chức cung cấp, thông tin mà phòng Kế toán quản lý là: Họ và Tên, Hệ số lương, Hệ số phụ cấp và sử dụng công cụ văn phòng là Microsoft Excel. Phòng Tổ chức quản lý thông tin chi tiết về nhân viên gồm: Họ và Tên, Chức vụ, Giới tính, Ngày sinh, Hệ số lương, Hệ số phụ cấp, Hoàn cảnh gia đình, Quá trình công tác, phần mềm sử dụng để quản lý là Microsoft Word. Trong khi đó tại Tổng công ty sử dụng phần mềm Microsoft Excel quản lý tất cả các thông tin của cả 2 phòng Kế toán và Tổ chức.

Ta có nhận xét sau:

#### **❖ Ưu điểm của Hệ thống tập tin cổ điển:**

- Việc xây dựng hệ thống các tập tin riêng tại từng đơn vị quản lý ít tốn thời gian bởi khối lượng thông tin quản lý nhỏ, không cần phải đầu tư nhiều vật chất và chất xám, do đó triển khai ứng dụng nhanh.
- Thông tin triển khai chỉ phục vụ cho mục đích hẹp nên khả năng đáp ứng nhanh chóng, kịp thời.

#### **❖ Nhược điểm của Hệ thống tập tin cổ điển:**

- Do thông tin được tổ chức ở mỗi phòng ban cũng như phần mềm dùng để quản lý ở mỗi phòng ban khác nhau, nên sự phối hợp tổ chức và khai thác ở các phòng ban là khó khăn, thông tin phòng ban này không thể sử dụng được cho phòng ban khác, *thiếu sự chia sẻ thông tin giữa các nơi*. Chẳng hạn trong tháng 12 phòng Tổ chức tăng lương

cho nhân viên Nguyễn Văn Tuấn nhưng chưa kịp chuyển thông tin sang phòng Kế toán, cuối tháng 12 nhân viên Nguyễn Văn Tuấn vẫn nhận lương theo hệ sổ lương cũ. Điều này xảy ra do thiếu sự chia sẻ thông tin giữa 2 phòng ban.

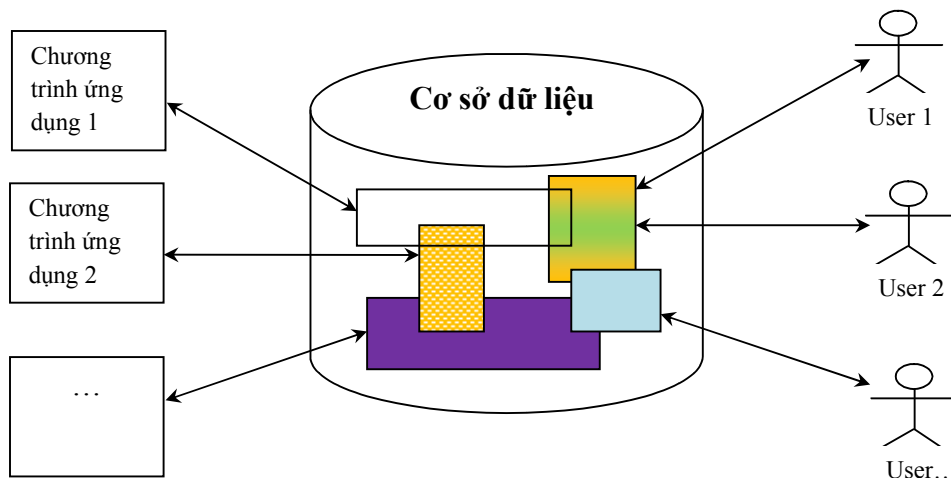
- Cùng một thông tin nhưng phải nhập vào máy tính tại nhiều bộ phận khác nhau gây nên sự lãng phí thời gian và công sức, tốn nhiều không gian để lưu trữ. Khi thông tin một nhân viên thay đổi phải thay đổi tại tất cả các phòng ban.

- Sự trùng lặp thông tin dễ dẫn đến *tình trạng không nhất quán dữ liệu*. Chẳng hạn, cùng nhân viên Nguyễn Văn Tuấn, phòng Tổ chức nhập đầy đủ học tên nhưng phòng Kế toán lại nhập tắt Nguyễn V. Tuấn.

Qua ví dụ trên, ta nhận thấy việc tổ chức dữ liệu theo hệ thống tập tin cổ điển có khá nhiều nhược điểm. Việc xây dựng một hệ thống đảm bảo được tính nhất quán dữ liệu, đáp ứng được nhu cầu khai thác thông tin đồng thời của tất cả các phòng ban trong công ty là thực sự cần thiết. Đây là nguyên nhân dẫn đến sự ra đời của phương pháp tổ chức dữ liệu mới: *Cơ sở dữ liệu*.

### 1.1.1. Định nghĩa cơ sở dữ liệu

*Cơ sở dữ liệu* (DataBase - DB) là hệ thống thông tin có cấu trúc được lưu trữ trên các thiết bị lưu trữ có thể thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng với nhiều mục đích khác nhau



Hình 1.1. Sơ đồ tổng quát về cơ sở dữ liệu

Chẳng hạn, ta cần xây dựng CSDL quản lý sinh viên trong một trường đại học, dữ liệu cần lưu trữ các thông tin về sinh viên, chuyên ngành, môn học, giảng viên giảng dạy môn học, điểm thi, ... Các thông tin này liên quan mật thiết với nhau và được lưu trữ

trong máy tính dưới dạng cấu trúc nào đó (chẳng hạn dạng bảng). Cùng một thời điểm, phòng Đào tạo có thể sử dụng CSDL để cập nhật thông tin của sinh viên, Khoa có thể xem thông tin giảng viên để sắp xếp thời khóa biểu, sinh viên có thể xem kết quả thi của mình.

### 1.1.2. Các tính chất của một cơ sở dữ liệu

- CSDL không chỉ đơn thuần là hợp nhất các hệ thống tập tin riêng lẻ mà là một tập hợp dữ liệu có cấu trúc liên kết với nhau một cách logic và mang một ý nghĩa cố hữu nào đó.

- Một cơ sở dữ liệu biểu thị một khía cạnh nào đó của thế giới thực. Những thay đổi của thế giới thực phải được phản ánh một cách trung thực vào trong cơ sở dữ liệu. Những thông tin được đưa vào trong cơ sở dữ liệu tạo thành một không gian cơ sở dữ liệu hoặc là một “thế giới thu nhỏ”.

#### ❖ Ưu điểm của CSDL

- Giảm sự trùng lặp thông tin đến mức thấp nhất và do đó đảm bảo được tính nhất quán và toàn vẹn dữ liệu.

- Có khả năng chia sẻ thông tin cho nhiều người sử dụng và nhiều ứng dụng khác nhau, tiết kiệm được tài nguyên và tăng hiệu quả sản xuất.

#### ❖ Yêu cầu đặt ra

Để đạt được các ưu điểm trên, CSDL đặt ra những vấn đề cần phải giải quyết:

- *Tính chủ quyền của dữ liệu*: Do khả năng chia sẻ thông tin của CSDL nên tính chủ quyền của dữ liệu có thể bị mờ nhạt. Tính chủ quyền được thể hiện trên vấn đề an toàn dữ liệu và tính chính xác của dữ liệu.

- *Tính bảo mật và quyền khai thác thông tin*: Do có nhiều người được phép khai thác CSDL nên cần phải có cơ chế bảo mật và phân quyền khai thác.

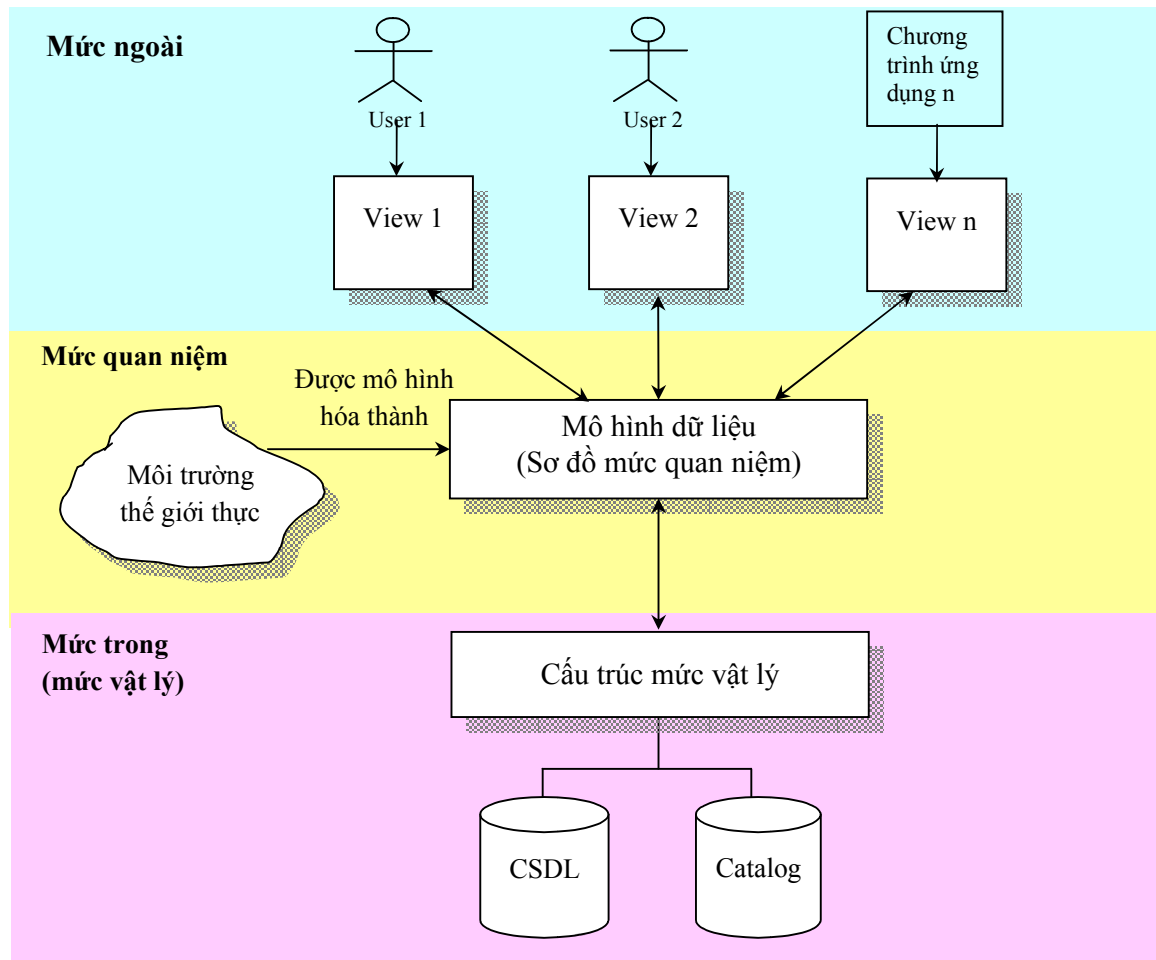
- *Giải quyết tranh chấp dữ liệu*: Nếu có nhiều người cùng muốn truy cập đến một CSDL, độ ưu tiên truy cập có thể được căn cứ trên quyền hạn khai thác của từng người sử dụng (ưu tiên cho người có quyền khai thác cao hơn), dựa trên đặc tính của sự truy cập (xem, thêm, xóa, sửa), hoặc dựa trên thời điểm truy cập dữ liệu, ...

- *Đảm bảo dữ liệu khi có sự cố*: Việc quản lý dữ liệu tập trung có thể làm tăng khả năng mất mát hoặc sai lệch thông tin khi có sự cố như mất điện đột xuất, một phần đĩa

lưu trữ CSDL bị hư, ... Để đảm bảo CSDL luôn ổn định cần thiết phải có một cơ chế phục hồi dữ liệu khi có sự cố xảy ra.

### ❖ Kiến trúc tổng quát của một cơ sở dữ liệu

Theo kiến trúc ANSI-PARC, một CSDL có 3 mức biểu diễn:



Hình 1.2. Kiến trúc tổng quát của một cơ sở dữ liệu

- Mức trong: Còn gọi là mức vật lý, đây là mức lưu trữ CSDL. Vấn đề cần giải quyết ở mức này là dữ liệu được lưu trữ như thế nào? ở đâu?. Những người hiểu và làm việc với CSDL tại mức này là người quản trị CSDL (Administrator)

- Mức quan niệm: Còn gọi là mức logic. Tại mức này sẽ giải quyết câu hỏi: cần phải lưu trữ những dữ liệu gì, mối quan hệ giữa các dữ liệu này như thế nào ?

Từ thế giới thực, các chuyên viên tin học (chuyên thiết kế CSDL) qua quá trình khảo sát nhu cầu và phân tích, sẽ xác định thông tin gì được cho là cần thiết phải đưa vào CSDL, đồng thời mô tả rõ mối quan hệ giữa các thông tin này. Kết quả của quá trình phân tích chính là mô hình dữ liệu. Có thể nói CSDL mức quan niệm là sự biểu

diễn trừu tượng CSDL mức vật lý, và ngược lại CSDL mức vật lý là sự cài đặt cụ thể CSDL mức quan niệm.

- Mức ngoài: Đây là mức của người sử dụng và các chương trình ứng dụng. Làm việc ở mức này chính là các kỹ sư lập trình và người sử dụng không chuyên (người dùng cuối).

Mỗi người sử dụng hay mỗi chương trình ứng dụng có thể "nhìn" (View) CSDL theo một góc độ khác nhau, có thể nhìn thấy toàn bộ hoặc cũng chỉ nhìn thấy một phần CSDL tùy thuộc vào qui định của người quản trị.

## **1.2. Hệ quản trị cơ sở dữ liệu**

### **1.2.1. Định nghĩa hệ quản trị cơ sở dữ liệu (DataBase Management System - DBMS)**

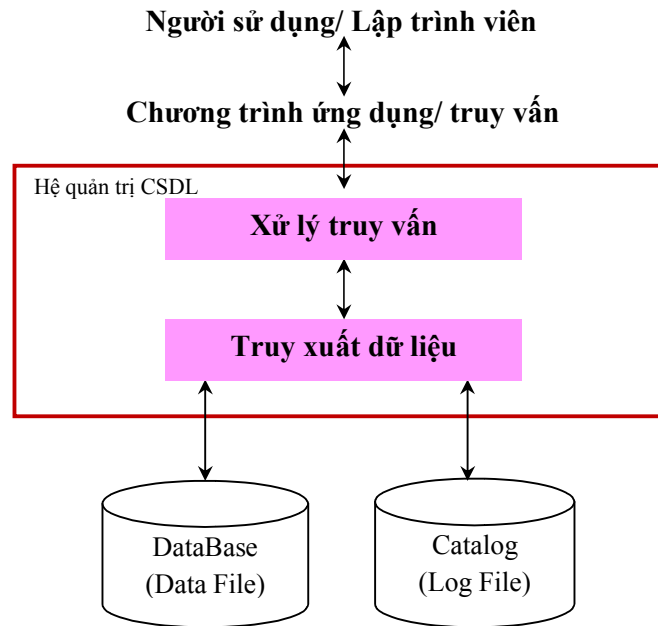
Để giải quyết tốt các vấn đề đặt ra cho CSDL như: tính chủ quyền, tính bảo mật hay phân quyền hạn khai thác, giải quyết tranh chấp trong quá trình truy cập dữ liệu, phục hồi dữ liệu khi có sự cố, ... cần phải có phần mềm chuyên dụng. Phần mềm chuyên dụng đó được gọi là *hệ quản trị cơ sở dữ liệu* (HQTCSDL). Các HQTCSDL là công cụ hỗ trợ tích cực cho các nhà thiết kế CSDL cũng như những người khai thác CSDL. Hiện nay có khá nhiều HQTCSDL trên thị trường như: FoxPro, DB2, MS Access, Sybase, Dbase, SQL Server, Oracle 9i/10g/11g, ... với chất lượng khác nhau.

Hệ quản trị cơ sở dữ liệu (HQTCSDL) là phần mềm chuyên dụng cho phép xây dựng và khai thác cơ sở dữ liệu. Mỗi HQTCSDL đều được cài đặt dựa trên một mô hình dữ liệu cụ thể, hầu hết (trên 90%) các HQTCSDL hiện nay đều dựa trên *mô hình quan hệ* (chi tiết về mô hình quan hệ trình bày trong chương III).

### **1.2.2. Các chức năng của một hệ HQTCSDL**

- Hệ quản trị cơ sở dữ liệu không chỉ chứa bản thân CSDL mà còn chứa các định nghĩa đầy đủ về CSDL (như mô tả về cấu trúc lưu trữ dữ liệu, các ràng buộc, quyền hạn sử dụng...), các định nghĩa này được lưu trữ trong từ điển dữ liệu (catalog). Dữ liệu lưu trong catalog còn được gọi là siêu dữ liệu (data of data).

- Có biện pháp bảo mật tốt khi có yêu cầu.
- Có cơ chế giải quyết tranh chấp dữ liệu.
- Có cơ chế sao lưu (backup) và phục hồi (restore) dữ liệu khi có sự cố xảy ra.
- Hỗ trợ việc truy cập dữ liệu thông qua một ngôn ngữ truy vấn.



Hình 1.3. Môi trường hệ quản trị cơ sở dữ liệu

### 1.2.3. Các đặc trưng của giải pháp cơ sở dữ liệu

- Bản chất tự mô tả của hệ cơ sở dữ liệu: một cơ sở dữ liệu không chỉ gồm có bản thân cơ sở dữ liệu mà còn có cả định nghĩa hoặc mô tả đầy đủ về cấu trúc cơ sở dữ liệu và các ràng buộc.

- Sự độc lập giữa chương trình và dữ liệu: vì định nghĩa về cấu trúc của một CSDL được lưu trữ trong catalog nên khi có thay đổi nhỏ về cấu trúc ta ít phải sửa lại chương trình truy cập đến CSDL đó.

- Tính trừu tượng của CSDL: hệ CSDL cho phép trình bày dữ liệu ở một mức trừu tượng cho phép, nhằm che bớt những chi tiết lưu trữ thật của dữ liệu.

- Hỗ trợ các khung nhìn dữ liệu: một cơ sở dữ liệu có nhiều người sử dụng, mỗi người có khả năng nhìn thấy một phần khác nhau của CSDL, khả năng nhìn thấy đó gọi là khung nhìn (View).

- Chia sẻ dữ liệu cho nhiều người sử dụng: phải có cơ chế để giải quyết tranh chấp dữ liệu.

### 1.2.4. Ví dụ một cơ sở dữ liệu

Chúng ta hãy xem xét một cơ sở dữ liệu quản lý điểm của sinh viên một trường học. Cơ sở dữ liệu này lưu giữ các thông tin liên quan đến sinh viên, các môn học, điểm... Cơ sở dữ liệu được tổ chức thành 4 bảng: bảng SINHVIEN lưu trữ dữ liệu về

các sinh viên, bảng MONHOC lưu trữ các dữ liệu về các môn học, bảng KHOA lưu trữ các dữ liệu các khoa trong trường, bảng KETQUA lưu trữ điểm từng môn học của các sinh viên.

MONHOC			KHOA			
MAMH	TENMH	SOTINCHI	MAKH	TENKH	NGAYTL	DIEN THOAI
DT001	Điện tử	2				
KT003	Nguyên lý kế toán	3	CNTT	Công nghệ thông tin		
TT001	Toán rời rạc	3	DTQT	Đào tạo Quốc tế		
TH001	Tin học ứng dụng	2	QTKD	Quản trị kinh doanh		
TH002	Cơ sở dữ liệu	3	XD	Xây dựng		

SINHVIEN							
MASV	HOTEN	NGAY SINH	GIOI TINH	DIACHI	CMND	DIEN THOAI	MAKHOA
SV0001	Nguyễn Thị Hải	23-02-1990	Nữ	Hà Nội	111222333		CNTT
SV0002	Trần Văn Chính	24-12-1992	Nam	Đà Nẵng	111555777		DTQT
SV0003	Lê Thu Bạch Yến	21-02-1990	Nữ	Đà Nẵng	333344445		QTKD
SV0004	Trần Anh Tuấn	20-12-1990	Nam		987654321		CNTT
SV0005	Trần Thanh Mai	12-08-1991	Nữ	Quảng Nam	555555555		DTQT
SV0006	Trần Thị Thu Thủy	02-01-1991	Nữ		123456789		CNTT

KETQUA			
MASV	MAMH	LANTHI	DIEM
SV0001	TH001	1	3
SV0001	TH001	2	6
SV0002	TH001	1	8.5
SV0002	TH002	1	8
SV0003	KT003	1	3
SV0003	KT003	2	9
SV0003	TT001	1	5
SV0003	TH001	1	10
SV0006	TH001	1	4.5
SV0006	TH002	1	5

Hệ quản trị cơ sở dữ liệu cho phép ta dễ dàng tạo ra và thao tác (thêm, xóa sửa, truy vấn) với cơ sở dữ liệu này. Ngoài ra HQTCSDL còn hỗ trợ sao lưu để phòng khi dữ liệu có sự cố, cho phép tạo tài khoản người dùng và cấp quyền sử dụng cho người dùng, ...

### 1.3. Mô hình cơ sở dữ liệu



### 1.3.1. Các loại mô hình cơ sở dữ liệu

*Mô hình dữ liệu* (Data Model) là sự trừu tượng hóa môi trường thế giới thực, nó là sự biểu diễn dữ liệu ở mức quan niệm. Mỗi loại mô hình dữ liệu đặc trưng cho một cách tiếp cận khác nhau của những nhà phân tích-thiết kế CSDL. Nói một cách hình thức, mô hình dữ liệu là một tập hợp các cấu trúc để mô tả dữ liệu và tập các phép toán dùng để thao tác trên dữ liệu này.

Có 5 mô hình dữ liệu thường được sử dụng nhất là:

- Mô hình mạng (Network Model)
- Mô hình phân cấp (Hierarchical Model)
- Mô hình thực thể-liên kết (Entity-Relationship Model)
- Mô hình quan hệ (Relational Model)
- Mô hình hướng đối tượng (Object Oriented Model)
- Mô hình bán cấu trúc (Semi-structured Model – XML)

Trước thập niên 1970, hầu hết các HQTCSDL đầu tiên được xây dựng dựa trên hai mô hình là *mô hình mạng* và *mô hình phân cấp*, một trong số đó là IMS – một hệ quản trị cơ sở dữ liệu của hãng IBM dựa trên mô hình phân cấp. Khi vừa mới ra đời *mô hình quan hệ* đã lật đổ hoàn toàn hai mô hình trước nó và nhanh chóng thống trị trong các HQTCSDL hàng đầu thế giới như DB2, Access, SQL Server, MySQL, Oracle... HQTCSDL đầu tiên dựa trên mô hình quan hệ là System-R của hãng IBM, ra đời vào năm 1976.

*Mô hình quan hệ* được E.F.Codd đề xuất vào năm 1970 dựa trên nền tảng vững chắc về lý thuyết tập hợp. Ưu điểm của mô hình này là đơn giản và truy vấn dữ liệu dễ dàng, một CSDL bao gồm các *quan hệ* (relation), mỗi quan hệ như là một *bảng* gồm nhiều cột và nhiều dòng, dữ liệu trình bày theo dạng bảng nên rất dễ hiểu và vì vậy khá dễ học đối với những người mới bắt đầu. Ví dụ cho trong mục I.2.4 là một cơ sở dữ liệu sử dụng mô hình quan hệ.

Ngày nay, *mô hình quan hệ* kết hợp cùng với *mô hình thực thể-liên kết* (được Perter Pin-Shan Chen đề xuất năm 1976) đang được sử dụng rộng rãi trong quá trình phân tích-thiết kế cơ sở dữ liệu (được trình bày chi tiết trong chương II và chương III).

Mặc dù *mô hình hướng đối tượng* (ra đời vào những năm cuối thập niên 1980 và những năm đầu thập niên 1990) là mô hình dữ liệu tiên tiến nhất hiện nay dựa trên

cách tiếp cận hướng đối tượng nhưng vẫn chưa được sử dụng rộng rãi một phần do chưa có nhiều HQTCSDL được cài đặt theo mô hình này. Các HQTCSDL đã sử dụng mô hình hướng đối tượng như Orion, Illustra,..., xét cho cùng hầu hết các hệ quản trị này đều vẫn là quan hệ-hướng đối tượng, nghĩa là xét về bản chất chúng vẫn dựa trên nền tảng của mô hình quan hệ. HQTCSDL hướng đối tượng thuần nhất có thể là ODMG ra đời vào năm 1996.

Vào những năm cuối thập niên 1990, khi internet bùng phát người ta bắt đầu quan tâm đến *mô hình bán cấu trúc*. Ngôn ngữ sử dụng mô hình bán cấu trúc là XML, với ngôn ngữ này thông tin mô tả về đối tượng thể hiện trong các tag. Cơ sở dữ liệu được xây dựng theo mô hình này có nhiều ưu điểm do lưu trữ được hầu hết các loại dữ liệu khác nhau. Hiện nay, cơ sở dữ liệu bán cấu trúc là hướng mới trong nghiên cứu và ứng dụng.

### 1.3.2. Lược đồ và trạng thái cơ sở dữ liệu

Trong một mô hình dữ liệu cần phải phân biệt rõ giữa cấu trúc mô tả của cơ sở dữ liệu và bản thân cơ sở dữ liệu. Cấu trúc mô tả của một cơ sở dữ liệu được gọi là *lược đồ cơ sở dữ liệu*, nó được xác định rõ trong quá trình thiết kế cơ sở dữ liệu và không bị thay đổi thường xuyên.

Các dữ liệu trong một cơ sở dữ liệu có thể thay đổi một cách thường xuyên. Các dữ liệu trong một cơ sở dữ liệu tại một thời điểm cụ thể được gọi là *một trạng thái cơ sở dữ liệu* hoặc thể hiện của lược đồ cơ sở dữ liệu. Một lược đồ cơ sở dữ liệu có thể có nhiều trạng thái cơ sở dữ liệu.

Khi chúng ta định nghĩa một cơ sở dữ liệu mới, ta chỉ đặc tả lược đồ cơ sở dữ liệu cho hệ quản trị cơ sở dữ liệu. Tại thời điểm này, trạng thái của cơ sở dữ liệu là một trạng thái rỗng, không có dữ liệu. Chúng ta nhận được trạng thái ban đầu của cơ sở dữ liệu khi ta nhập dữ liệu lần đầu tiên. Từ đó trở đi, mỗi khi một phép toán cập nhật được thực hiện đối với cơ sở dữ liệu, chúng ta nhận được một trạng thái cơ sở dữ liệu khác. Tại mọi thời điểm, cơ sở dữ liệu có một trạng thái hiện tại. Hệ quản trị cơ sở dữ liệu có trách nhiệm đảm bảo rằng mỗi trạng thái cơ sở dữ liệu là một trạng thái vững chắc, nghĩa là một trạng thái thoả mãn cấu trúc và các ràng buộc được đặc tả trong lược đồ. Vì vậy, việc đặc tả một lược đồ đúng đắn cho HQTCSDL là một việc làm rất quan trọng. Các định nghĩa này được trình bày chi tiết trong chương 3.

#### **I.4. Con người trong hệ quản trị cơ sở dữ liệu**

##### **I.4.1. Người quản trị hệ cơ sở dữ liệu (Database Administrator – DBA)**

Trong một tổ chức có nhiều người cùng sử dụng các tài nguyên, cần phải có một người giám sát và quản lý. Trong môi trường hệ cơ sở dữ liệu, các tài nguyên là cơ sở dữ liệu, người quản trị cơ sở dữ liệu là người chịu trách nhiệm quản lý các tài nguyên đó. Người này chịu trách nhiệm về việc cho phép truy cập cơ sở dữ liệu, tổ chức CSDL, cấp quyền khai thác CSDL và chịu trách nhiệm khi dữ liệu có sự cố.

##### **I.4.2. Người thiết kế cơ sở dữ liệu (Database Designer)**

Người này chịu trách nhiệm chọn lọc các dữ liệu cần được lưu trữ, chọn các cấu trúc thích hợp để biểu diễn và lưu trữ các dữ liệu đó trong cơ sở dữ liệu. Những nhiệm vụ này được thực hiện trước khi cơ sở dữ liệu được cài đặt và phổ biến. Người thiết kế phải tiếp xúc với những người sử dụng tương lai để hiểu được các đòi hỏi của họ và đưa ra một thiết kế thoả mãn các yêu cầu đó.

##### **I.4.3. Những người sử dụng (End User)**

Những người sử dụng là những người mà công việc của họ đòi hỏi truy cập đến cơ sở dữ liệu để truy vấn, cập nhật và sinh ra các thông tin. Thường thì những người sử dụng không có kiến thức về CSDL, chỉ khai thác CSDL thông qua công cụ là các chương trình ứng dụng.

##### **I.4.4. Người phân tích hệ thống và lập trình ứng dụng**

Những người này có kiến thức về CSDL, chuyên xây dựng các ứng dụng phục vụ cho các mục đích khác nhau trên CSDL.

##### **I.4.5. Người thiết kế và cài đặt hệ quản trị dữ liệu**

Đó là những người chuyên thiết kế và xây dựng ra các hệ quản trị cơ sở dữ liệu. Chẳng hạn, hãng Oracle đã thiết kế hệ quản trị Oracle, hãng Microsoft đã thiết kế hệ quản trị SQL server, Access, ...

##### **I.4.6. Những người phát triển công cụ**

Là những người thiết kế và cài đặt các công cụ (tool), đó là các phần mềm đóng gói hỗ trợ việc thiết kế và sử dụng cơ sở dữ liệu. Chẳng hạn, phần mềm hỗ trợ thiết kế CSDL như PowerDesigner, phần mềm sử dụng CSDL như bộ Visual.NET.

#### **I.4.7. Các thao tác viên và những người bảo trì**

Là những người chịu trách nhiệm về hoạt động của các phần cứng và phần mềm của hệ thống.

### **I.5. Ngôn ngữ cơ sở dữ liệu và giao diện**

#### **I.5.1. Các ngôn ngữ hệ quản trị cơ sở dữ liệu**

Trong các hệ quản trị cơ sở dữ liệu hiện nay thường có các ngôn ngữ giao tiếp giữa người sử dụng và CSDL:

- Ngôn ngữ mô tả dữ liệu (DDL – Data Definition Language): dùng để khai báo cấu trúc của CSDL.
- Ngôn ngữ thao tác dữ liệu (DML – Data Manipulation Language): dùng để thêm, xóa, sửa và truy vấn dữ liệu trong CSDL.
- Ngôn ngữ quản lý dữ liệu (DCL – Data Control Language): dùng để bảo mật thông tin, cấp quyền hạn khai thác CSDL cho người sử dụng.

Các ngôn ngữ trên được gọi chung là SQL. SQL là viết tắt của Structured Query Language - ngôn ngữ truy vấn có cấu trúc, cho phép người dùng sử dụng để tổ chức, quản lý và truy xuất dữ liệu từ CSDL, SQL là hệ thống các câu lệnh sử dụng tương tác với CSDL quan hệ. Tên gọi của SQL làm ta liên tưởng đến một ngôn ngữ dùng để truy vấn dữ liệu, thực sự mà nói khả năng của SQL vượt xa so với công cụ truy xuất dữ liệu, mặc dù đó là mục đích ban đầu để xây dựng nên ngôn ngữ SQL.

Trước đây SQL được gọi là SEQUEL (Structured English Query Language), là ngôn ngữ được IBM phát triển để sử dụng trong hệ thống CSDL thử nghiệm là SystemR. Cho tới nay, có thể nói rằng SQL là ngôn ngữ chuẩn trong CSDL, SQL được sử dụng trong rất nhiều các HQTCSDL thương mại hàng đầu thế giới như BD2, Access, SQL Server, Oracle, ...

#### **I.5.2. Các loại giao diện của hệ quản trị cơ sở dữ liệu**

Các hệ quản trị cơ sở dữ liệu cung cấp nhiều loại giao diện người dùng thân thiện. Các loại giao diện chính gồm có:

- Giao diện đồ họa: Giao diện này luôn hiển thị các mẫu biểu được thiết kế sẵn, rất thân thiện và hỗ trợ người dùng rất nhiều trong việc quản lý, cập nhật hay truy vấn CSDL. Người sử dụng không cần phải nhớ cú pháp của các ngôn ngữ sử dụng trong các hệ quản trị này.

- Giao diện lệnh: Việc tạo một cơ sở dữ liệu, cập nhật cơ sở dữ liệu, truy vấn thông tin từ cơ sở dữ liệu, hay tạo user và cấp quyền sử dụng, sao lưu, phục hồi... nếu sử dụng lệnh có thể thực hiện nhanh chóng và đơn giản hơn nhiều so với việc sử dụng giao diện đồ họa, tuy nhiên cần thiết phải ghi nhớ các cú pháp của các ngôn ngữ liên quan.

## **I.6. Tổng kết chương và câu hỏi ôn tập**

### **I.6.1. Tổng kết chương**

Chương này đã giới thiệu sơ lược về nguyên nhân ra đời của CSDL cũng như những ưu điểm của CSDL và những vấn đề mà CSDL đặt ra. Mỗi CSDL phải tuân theo một mô hình dữ liệu nhất định như mô hình quan hệ, mô hình mạng, mô hình phân cấp, mô hình hướng đối tượng,...

HQTCSDL là phần mềm chuyên dụng dùng để xây dựng và khai thác CSDL, trên 90% các HQTCSDL thương mại hiện nay đều sử dụng mô hình quan hệ như: DB2, Access, SQL server, MySQL, Oracle,... Hầu hết các HQTCSDL đều hỗ trợ 2 giao diện chính, đó là giao diện đồ họa rất thân thiện gần gũi với người dùng và giao diện lệnh giúp việc thực thi được thực hiện nhanh chóng và đơn giản.

### **I.6.2. Câu hỏi ôn tập**

1. Những hạn chế của hệ thống tập tin cổ điển?
2. Khái niệm CSDL? Ưu điểm của CSDL?
3. Muốn xây dựng một CSDL để quản lý bán hàng, ta cần lưu trữ những thông tin gì?
4. Khái niệm hệ quản trị cơ sở dữ liệu? Các chức năng của một HQTCSDL?
5. Kể tên các HQTCSDL được sử dụng rộng rãi trên thị trường hiện nay?
6. Một CSDL có bao nhiêu mức biểu diễn? Ta nói CSDL mức quan niệm là sự biểu diễn trừu tượng của mức vật lý, đúng hay sai?
7. Mô hình dữ liệu dùng để biểu diễn CSDL ở mức nào?

8. Mô hình dữ liệu nào thống trị trong các HQTCSDL hàng đầu thế giới hiện nay?  
Vì sao?
9. Những người nào trực tiếp sử dụng CSDL?
10. Hãy cho ví dụ để chứng minh sự quan trọng việc truy vấn CSDL bằng lệnh?

## Chương II. MÔ HÌNH THỰC THỂ - LIÊN KẾT

### A. Mục tiêu

#### ➤ Về kiến thức:

Sau khi xong chương này sinh viên nắm vững các kiến thức sau:

- Nắm vững các khái niệm trong mô hình thực thể-liên kết (ER): Thực thể và tập thực thể, thuộc tính và các loại thuộc tính, mối liên kết và các loại mối liên kết
- Nắm vững các khái niệm trong mô hình thực thể-liên kết mở rộng (EER): Mối liên kết ISA, tổng quát hóa và chuyên biệt hóa.
- Có thể thiết kế mô hình ER cho một số hệ thống quản lý đơn giản.

#### ➤ Về kỹ năng

Giúp sinh viên phát triển các kỹ năng:

- Năng động trong lớp học.
- Có khả năng trình bày trước đám đông
- Rèn luyện khả năng làm việc theo nhóm

### B. Phương pháp giảng dạy

- Trình chiếu slides
- Đặt câu hỏi cho sinh viên trả lời
- Yêu cầu sinh viên thảo luận theo nhóm
- Làm và giải bài tập tại lớp
- Yêu cầu sinh viên làm bài tập về nhà

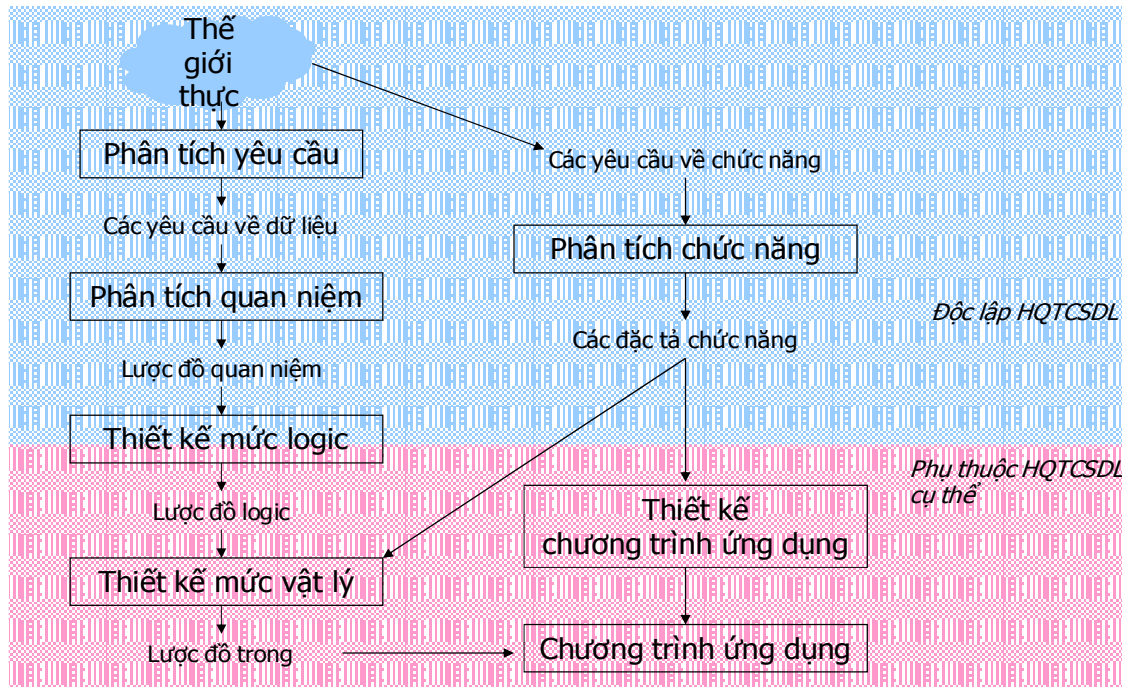
### C. Tài liệu tham khảo

1. Nguyễn An Tế, Đồng Thị Bích Thủy (1996). *Nhập Môn Cơ Sở Dữ Liệu*.
2. Nguyễn Đăng Ty, Đỗ Phúc (2003). *Giáo Trình Cơ Sở Dữ Liệu*.
3. Lê Tiến Vương (1994). *Nhập Môn Cơ Sở Dữ Liệu Quan Hệ*
4. Jeffrey D. Ullman (1998). *Mô Hình Dữ Liệu Và Ngôn Ngữ Ván Tin*. Tập 1
5. Hoàng Trung Sơn (2004). *Thiết Kế Cơ Sở Dữ Liệu*.

### D. Nội dung chi tiết

## II.1. Sử dụng mô hình quan niệm bậc cao cho việc thiết kế cơ sở dữ liệu

Quá trình thiết kế một cơ sở dữ liệu sử dụng mô hình quan niệm bậc cao được minh họa bằng hình 2.1.



Hình 2.1. Quá trình thiết kế một cơ sở dữ liệu

Bước đầu tiên là *phân tích yêu cầu*. Trong bước này, người thiết kế cơ sở dữ liệu lấy thông tin từ những người sử dụng cơ sở dữ liệu để đặc tả yêu cầu. Kết quả của bước này là một tập hợp ghi chép súc tích về các yêu cầu của những người sử dụng. Những yêu cầu sẽ được đặc tả càng đầy đủ và chi tiết càng tốt. Song song với việc đặc tả các yêu cầu dữ liệu, cần phải đặc tả các yêu cầu về chức năng của ứng dụng.

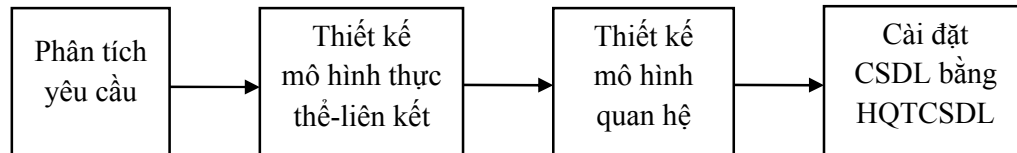
Khi tất cả các yêu cầu đã được thu thập và phân tích, bước tiếp theo là tạo ra lược đồ quan niệm bằng cách sử dụng các mô hình dữ liệu quan niệm mức cao (chẳng hạn mô hình thực thể-liên kết). Bước này gọi là *phân tích quan niệm*. Lược đồ quan niệm là một mô tả súc tích và ngắn gọn về các yêu cầu dữ liệu của những người sử dụng.

Bước tiếp theo trong việc thiết kế cơ sở dữ liệu chuyển lược đồ quan niệm thành lược đồ logic trên cơ sở một mô hình dữ liệu nào đó (chẳng hạn như mô hình quan hệ, mô hình phân cấp, mô hình mạng), hay nói cách khác bước này sẽ chuyển lược đồ quan niệm từ mô hình dữ liệu bậc cao thành mô hình dữ liệu có thể cài đặt trong một HQTCSDL. Bước này gọi là *thiết kế mức logic* hoặc là ánh xạ mô hình dữ liệu



Bước cuối cùng trong thiết kế cơ sở dữ liệu là *thiết kế vật lý*, trong bước này ta sẽ cài đặt CSDL vào một HQTCSDL có sẵn. Song song với bước này các chương trình ứng dụng sử dụng CSDL cũng được thiết kế và cài đặt.

Một cách đơn giản, các bước chính trong quá trình thiết kế một CSDL phổ biến từ trước tới nay có thể được minh họa bằng hình 2.2 bên dưới:



Hình 2.2. Các bước chính thiết kế một cơ sở dữ liệu

Trong chương này, chúng ta sẽ tìm hiểu chi tiết về mô hình thực thể-liên kết (ER). Mô hình quan hệ và các bước chuyển từ mô hình ER sang mô hình quan hệ sẽ được trình bày trong chương 3. Cách cài đặt CSDL bằng HQTCSDL sẽ được trình bày trong chương 5.

## II.2. Các thành phần cơ bản của mô hình thực thể-liên kết

*Mô hình thực thể-liên kết* (mô hình ER), còn được gọi là mô hình thực thể-kết hợp, được P.P.Chen đề xuất vào năm 1976. Mô hình này dùng để biểu diễn một cách trừu tượng cấu trúc của một CSDL, không chú ý đến việc thiết kế CSDL mức vật lý được mong đợi như ở các mô hình dữ liệu khác.

### II.2.1. Thực thể

*Thực thể* (entity) là một đối tượng trong thế giới thực, hay nói cách khác thực thể là một vật tồn tại và phân biệt được, có nghĩa là có thể phân biệt được thực thể này với thực thể khác. Khái niệm về “tính phân biệt được” rất gần gũi với “đặc tính nhận dạng đối tượng”.

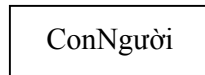
Chẳng hạn, mỗi con người là một thực thể, mỗi chiếc xe máy cũng là thực thể. Ta cũng có thể nói mỗi con kiến cũng là thực thể nếu ta biết cách phân biệt con này với con khác (như sơn 1 con số cực bé trên lưng của chúng)

### II.2.2. Tập thực thể

Một nhóm bao gồm tất cả các thực thể tương tự nhau tạo ra một tập thực thể (entity set). Ví dụ về các tập thực thể: Tất cả mọi người, tất cả xe gắn máy,... Một trong

những bước mấu chốt để xây dựng thành công mô hình ER cho một CSDL chính là việc chọn các tập thực thể.

Trong mô hình ER, tập thực thể được biểu diễn bằng hình chữ nhật.



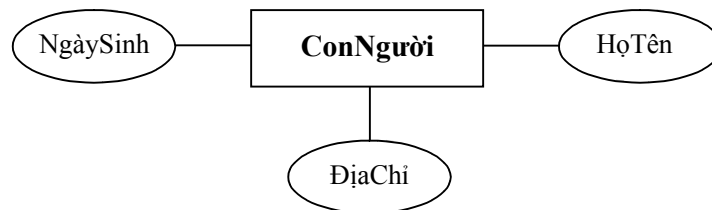
Hình 2.3. Biểu diễn tập thực thể trong mô hình ER

### II.2.3. Thuộc tính

Các *đặc tính* của tập thực thể gọi là các thuộc tính (attribute). Chúng liên kết với mỗi thực thể của tập thực thể với một giá trị từ một miền giá trị (domain) dành cho thuộc tính đó, miền giá trị của một thuộc tính có thể là số nguyên, số thực, chuỗi ký tự,...

*Ví dụ 2.1.* Một tập thực thể ConNgười được mô tả bằng các thuộc tính là HọTên, NgàySinh, ĐịaChỉ. Một thực thể cụ thể có HọTên: “Lê Anh Khoa”, NgàySinh: “1/5/1980”, ĐịaChỉ: “Đà Nẵng”.

Trong mô hình ER, các thuộc tính được biểu diễn bằng hình oval, chúng liên hệ với tập thực thể bằng các cạnh.

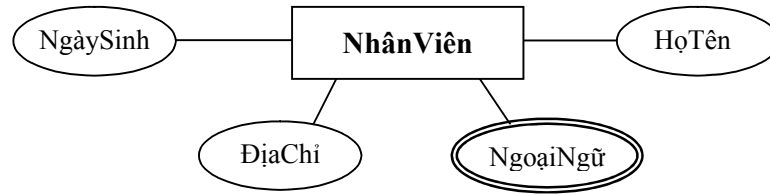


Hình 2.4. Biểu diễn thuộc tính trong mô hình ER

Trong mô hình ER có nhiều kiểu thuộc tính: thuộc tính đơn trị, thuộc tính đa trị, thuộc tính phức hợp:

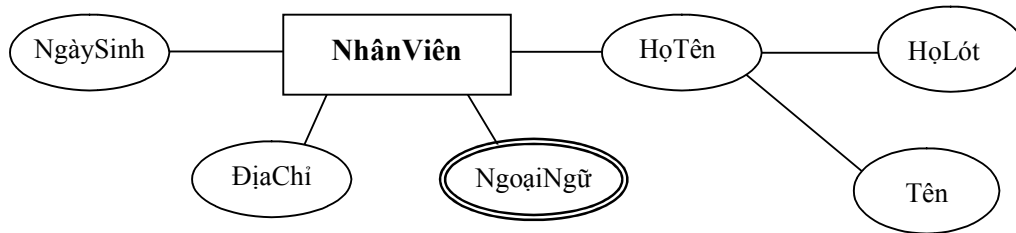
- *Thuộc tính đơn trị* là thuộc tính có giá trị duy nhất cho một thực thể cụ thể gọi là các thuộc tính đơn trị. Ví dụ, HọTên là một thuộc tính đơn trị của thực thể NhânViên, mỗi nhân viên có một họ tên duy nhất.
- *Thuộc tính đa trị* là thuộc tính có thể có nhiều giá trị cho cùng một thực thể, trong mô hình ER thuộc tính đa trị được biểu diễn bằng hình oval kép. Chẳng hạn, thuộc tính NgoạiNgữ của tập thực thể NhânViên, một người có thể không biết một ngoại ngữ nào nhưng cũng có thể biết nhiều ngoại ngữ. Như vậy, một

nhân viên có thể có một số giá trị khác nhau cho thuộc tính NgoạiNgữ. Thuộc tính NgoạiNgữ là một thuộc tính đa trị.



Hình 2.5. Biểu diễn thuộc tính đa trị trong mô hình ER

- *Thuộc tính phức hợp* là thuộc tính có thể phân chia được thành các thành phần nhỏ hơn, biểu diễn các thuộc tính cơ bản hơn với các ý nghĩa độc lập. Chẳng hạn, thuộc tính HọTên của thực thể NhânViên có thể phân chia thành các tính HọLót và Tên. Giá trị của một thuộc tính phức hợp là sự kết hợp các giá trị của các thuộc tính thành phần tạo nên nó. Việc phân chia một thuộc tính phức hợp thành các thuộc tính đơn tùy thuộc vào hoàn cảnh cụ thể.



Hình 2.6. Biểu diễn thuộc tính phức hợp trong mô hình ER

## II.2.4. Khóa

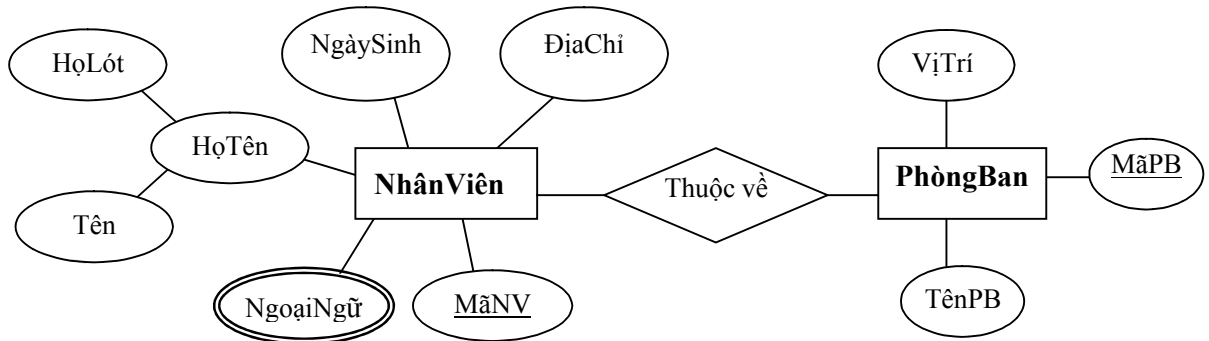
Một thuộc tính hoặc một tập các thuộc tính *tối thiểu* dùng để xác định tính duy nhất của mỗi thực thể trong tập thực thể gọi là *Khóa* (key) của tập thực thể đó. Về nguyên tắc, mỗi tập thực thể có một khóa, bởi vì chúng ta giả thiết rằng mỗi thực thể đều có thể phân biệt với tất cả các thực thể khác. Trong mô hình ER, các thuộc tính là thành phần của một khóa sẽ được gạch dưới.

*Ví dụ 2.2.* Một tập thực thể bao gồm các công dân Việt Nam, có thể dùng một thuộc tính SốCMND làm khóa vì ở nước ta mỗi công nhân đều có một SốCMND khác nhau. Tuy nhiên, nếu tập thực thể bao gồm các công dân của nhiều quốc gia thì không thể đảm bảo chắc chắn rằng hai quốc gia nào đó sẽ không dùng số chứng minh nhân dân giống nhau. Vì vậy khóa thích hợp phải gồm hai thuộc tính SốCMND và QuốcGia

## II.2.5. Môi liên kết

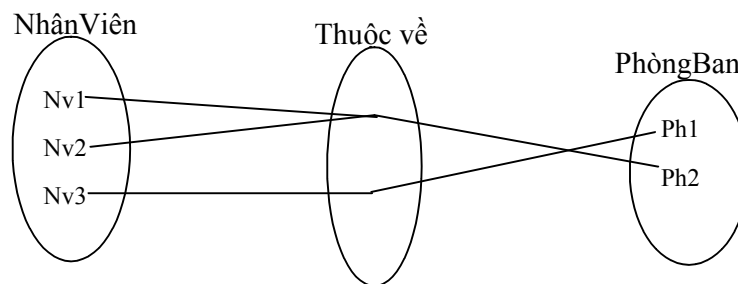
### II.2.5.1. Khái niệm môi liên kết

Môi liên kết là sự kết nối giữa hai hay nhiều thực thể phản ánh mối quan hệ nào đó trong thế giới thực. Trong mô hình ER, môi liên kết được biểu diễn bằng hình thoi và liên hệ với các tập thực thể bằng các cạnh vô hướng.



Hình 2.7. Biểu diễn môi liên kết trong mô hình ER

Một thể hiện của môi liên kết trong Hình 2.7:



Hình 2.8. Một thể hiện của môi liên kết

### II.2.5.2. Bản số của môi liên kết

Mỗi nhánh của liên kết được xác định kèm với một bản số. Một bản số là một bộ gồm 2 thành phần (min, max) là chỉ định số lượng (tối thiểu, tối đa) các thực thể của tập thực thể tham gia vào môi liên kết.

Chẳng hạn, các bản số trong hình 2.9 có nghĩa: Một phòng ban có *tối thiểu một* và *tối đa nhiều* nhân viên, một nhân viên chỉ thuộc duy nhất một phòng ban.



Hình 2.9. Bản số của môi liên kết

Có các loại bản số sau:

(0, 1): tối thiểu là 0, tối đa là 1

(1, 1): duy nhất 1

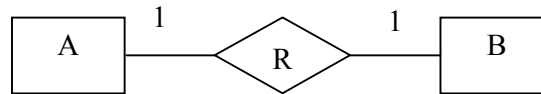
(0, n): tối thiểu 0, tối đa nhiều

(1, n): tối thiểu 1, tối đa nhiều

### II.2.5.3. Các loại liên kết

Để mô hình hóa đầy đủ cho thế giới thực, cần phải phân loại các liên kết theo số lượng các thực thể từ mỗi tập thực thể tham gia vào liên kết. Sau đây là các loại liên kết:

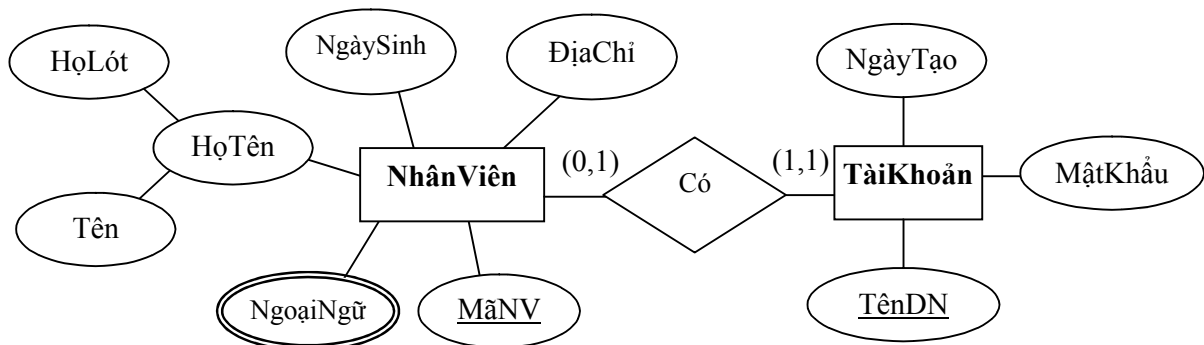
#### ▪ Mỗi liên kết Một-Một (1-1)



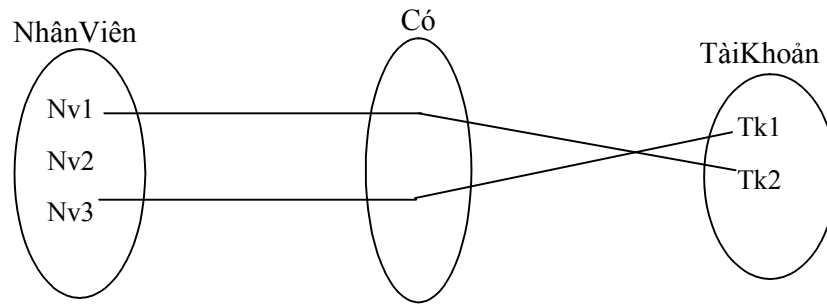
Hình 2.10. Liên kết Một-Một

Một thực thể trong tập thực thể A có *quan hệ tối đa* với *một* thực thể trong tập thực thể B, và ngược lại một thực thể trong B có *quan hệ tối đa* với *một* thực thể trong A.

*Ví dụ 2.3.* Mỗi quan hệ NhânViên <Có> TàiKhoản là liên kết Một-Một. Một tài khoản đăng nhập chỉ của duy nhất một nhân viên và một nhân viên có thể không có hoặc chỉ có một tài khoản đăng nhập.

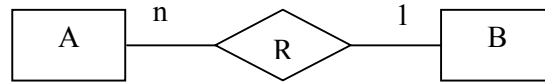


Một thể hiện của mối liên kết NhânViên <Có> TàiKhoản:



Hình 2.11. Ví dụ mối liên kết Một-Một

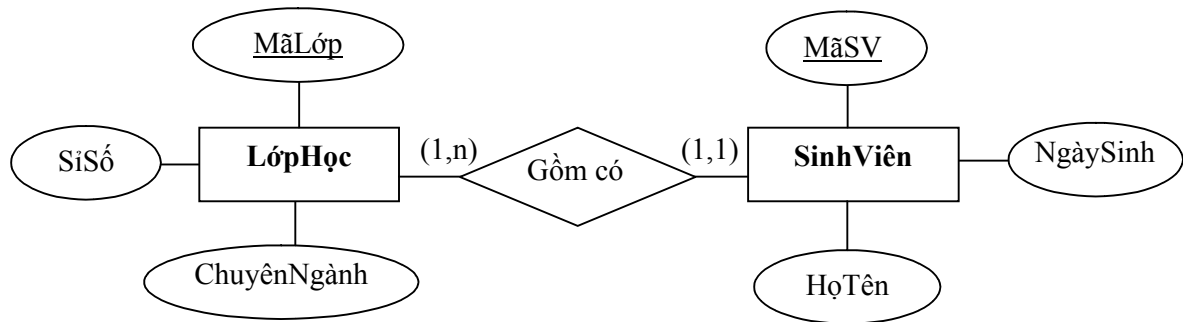
▪ **Mối liên kết Một-Nhiều (1-n)**



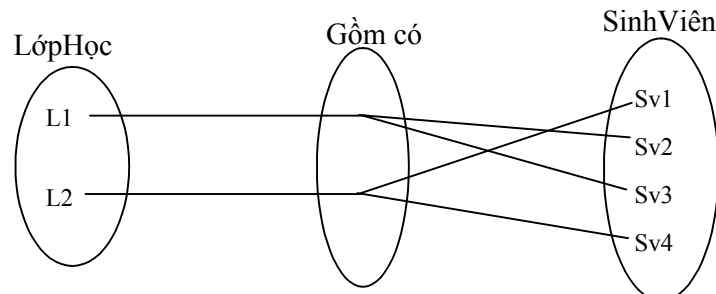
Hình 2.12. Mối liên kết Một-Nhiều

Một thực thể trong tập thực thể A có quan hệ với nhiều thực thể trong tập thực thể B, và ngược lại một thực thể trong B có quan hệ tối đa với một thực thể trong A.

*Ví dụ 2.4.* Mối quan hệ LớpHọc <gồm có> SinhViên là liên kết Một-Nhiều. Một lớp học có nhiều sinh viên và một sinh viên chỉ thuộc một lớp học duy nhất.

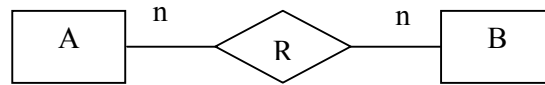


Một thể hiện của mối liên kết LớpHọc <Gồm có> SinhViên:



Hình 2.13. Ví dụ mối liên kết Một-Nhiều

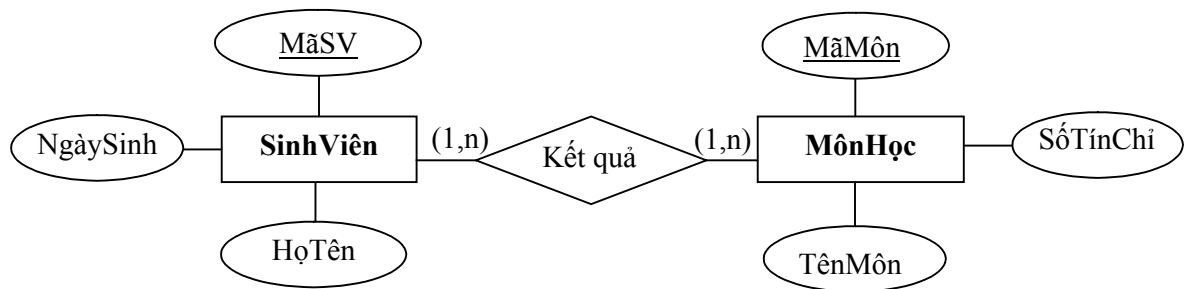
▪ **Mối liên kết Nhiều-Nhiều (n-n)**



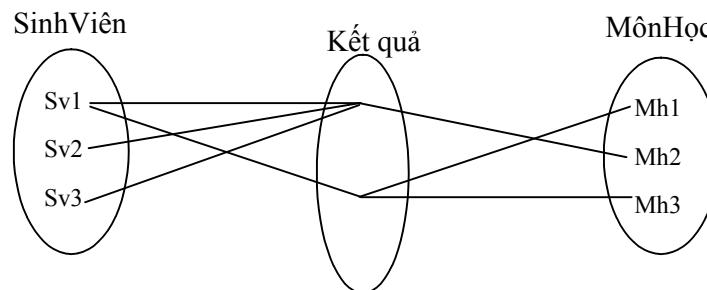
Hình 2.14. Mối liên kết Nhiều-Nhiều

Một thực thể trong tập thực thể A có quan hệ với nhiều thực thể trong tập thực thể B, và ngược lại một thực thể trong B có mối quan hệ với nhiều thực thể trong A.

*Ví dụ 2.5.* Mối quan hệ SinhViên <Kết quả> MônHọc là liên kết Nhiều-Nhiều. Một sinh viên có thể học nhiều môn học và một môn học có thể được nhiều sinh viên học.



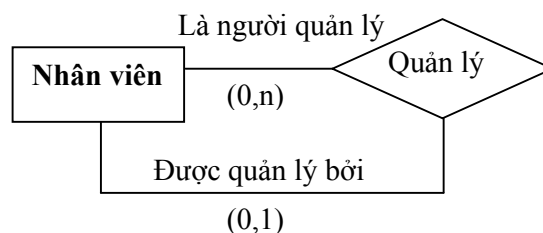
Một thể hiện của mối liên kết SinhViên <Kết quả> MônHọc:



Hình 2.15. Ví dụ mối liên kết Nhiều-Nhiều

#### II.2.5.4. Mối liên kết đệ qui

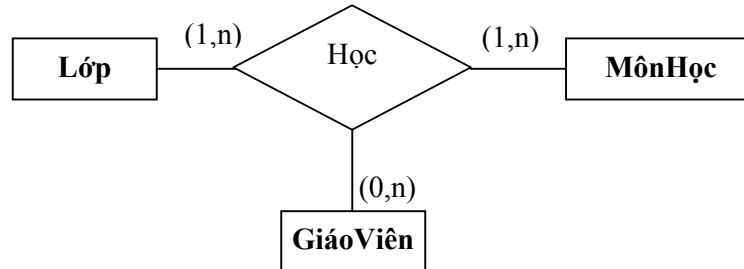
Một tập thực thể có thể tham gia nhiều lần vào một liên kết với nhiều vai trò khác nhau, mối liên kết như vậy gọi là mối liên kết đệ qui.



Hình 2.16. Mối liên kết đệ qui

### II.2.5.5. Mỗi liên kết đa ngôi

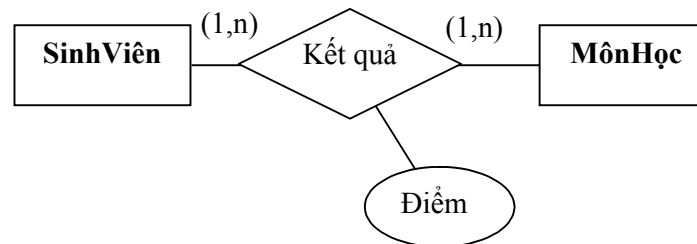
Bậc của mỗi liên kết là số các tập thực thể tham gia vào mỗi liên kết đó. Chẳng hạn, mỗi liên kết đệ qui có bậc là 1. Các mỗi liên kết 1-1, 1-n, n-n có bậc là 2. Liên kết với số tập thực thể tham gia lớn hơn 2 được gọi là *mỗi liên kết đa ngôi*.



Hình 2.17. Mỗi liên kết đa ngôi

### II.2.6. Thuộc tính của mỗi liên kết

Mỗi liên kết cũng có thể có các thuộc tính, tương tự các thuộc tính của tập thực thể. Thuộc tính trên mỗi liên kết dùng để mô tả tính chất riêng của mỗi liên kết đó. Chẳng hạn, mỗi liên kết <Kết quả> có thể có thuộc tính Điểm để ghi nhận kết quả thi của một sinh viên sau khi học xong một môn học.



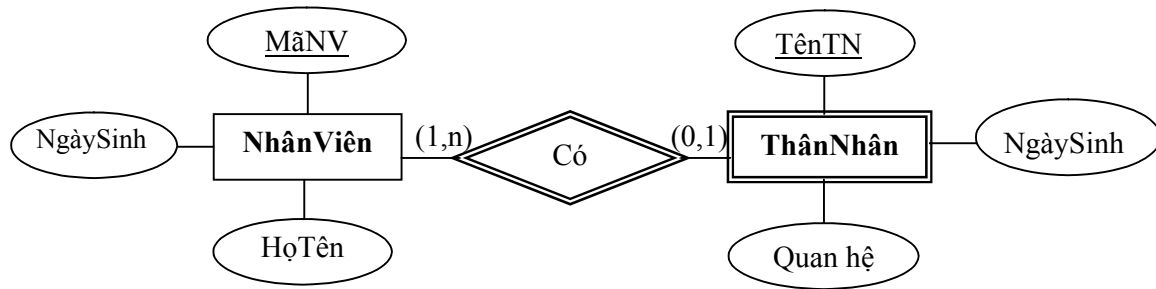
Hình 2.18. Thuộc tính của mỗi liên kết

### II.2.7. Tập thực thể yếu

Một tập thực thể được gọi là *tập thực thể yếu* khi sự tồn tại của nó phụ thuộc vào sự tồn tại của một tập thực thể khác (thực thể chủ). Tập thực thể yếu không có khóa mà chỉ có khóa bộ phận, nếu chỉ dựa vào khóa bộ phận không thể phân biệt được các thực thể trong tập thực thể yếu với nhau mà phải dựa vào khóa của thực thể chủ. Trong mô hình ER, tập thực thể yếu và mỗi liên kết được biểu diễn bằng hình chữ nhật nét đôi và hình thoi nét đôi.

Chẳng hạn, tập thực thể ThânNhân là một thực thể yếu, vì sự tồn tại của nó phụ thuộc vào sự tồn tại của tập thực thể NhânViên.





Hình 2.19. Tập thực thể yếu

### II.3. Ví dụ về thiết kế mô hình thực thể-liên kết

Như mô tả trong hình 2.2, bước đầu tiên trong việc xây dựng một cơ sở dữ liệu là phân tích yêu cầu thực tế. Kết quả của bước này là một tập hợp các ghi chép súc tích về các yêu cầu người sử dụng cũng như tình trạng của nơi ta cần xây dựng cơ sở dữ liệu.

Chúng ta cần xây dựng một CSDL cho hệ thống *quản lý dự án* của một công ty, dữ liệu cần quản lý liên quan đến thông tin nhân viên, phòng ban và các dự án. Giả sử rằng sau khi tập hợp các yêu cầu và phân tích hoạt động của công ty, ta có được các ghi chép lại như sau :

- Công ty có nhiều phòng ban, mỗi phòng ban có một mã phòng duy nhất, một tên phòng, địa điểm. Một nhân viên có thể quản lý (trưởng phòng) một phòng ban, việc nhân viên quản lý phòng ban được ghi lại bằng ngày nhân viên bắt đầu nhận chức quản lý.
- Mỗi phòng ban chủ trì một số dự án. Dự án có một mã duy nhất, một tên và được triển khai ở một địa điểm.
- Nhân viên có mã số, tên (họ lót và tên), địa chỉ, ngày sinh, phái và lương. Mỗi nhân viên làm việc ở một phòng ban nhưng có thể tham gia vào nhiều dự án với số giờ làm việc khác nhau.
- Một nhân viên có thể có nhiều thân nhân. Mỗi thân nhân lưu giữ tên, phái, ngày sinh và mối quan hệ với nhân viên đó. Khi thông tin của nhân viên bị xóa thì những thân nhân của nhân viên đó cũng bị xóa theo.

Theo ghi chép ở trên, chúng ta có thể xác định các tập thực thể và các mối liên kết như sau:

**▪ Xác định các thực thể:**

- Công ty là một đối tượng trong thế giới thực, tuy nhiên ở đây chúng ta không có tập thực thể Công ty vì trong quản lý này chỉ có duy nhất một công ty.

- PHONGBAN là một tập thực thể với các thuộc tính tên phòng (TENPHG), mã phòng (MAPHG), địa điểm (DIADIEM). Trong tập thực thể PHONGBAN, thuộc tính MAPHG là thuộc tính khóa vì mỗi phòng ban có một mã số duy nhất.

- Tập thực thể DUAN có các thuộc tính tên dự án (TENDA), mã dự án (MADA), địa điểm dự án (DDIEM\_DA). Thuộc tính MADA là thuộc tính khóa.

- Tập thực thể NHANVIEN với các thuộc tính mã nhân viên (MANV), họ tên (HOTEN), địa chỉ (DCHI), ngày sinh (NGSINH), phái (PHAI) và lương (LUONG). Thuộc tính khóa là MANV.

- Tập thực thể THANNHAN với các thuộc tính tên thân nhân (TENTN), phái (PHAI), ngày sinh (NGAYSINH), và quan hệ với nhân viên (QUANHE). Tập thực thể này có quan hệ phụ thuộc với tập thực thể NHANVIEN. Thuộc tính TENTN là khóa bộ phận của tập thực thể THANNHAN.

**▪ Xác định các mối liên kết:**

- Mối liên kết NHANVIEN <quản lý> PHONGBAN. Đây là mối liên kết 1-1, một nhân viên chỉ quản lý tối đa một phòng ban và một phòng ban chỉ có tối đa 1 nhân viên quản lý.

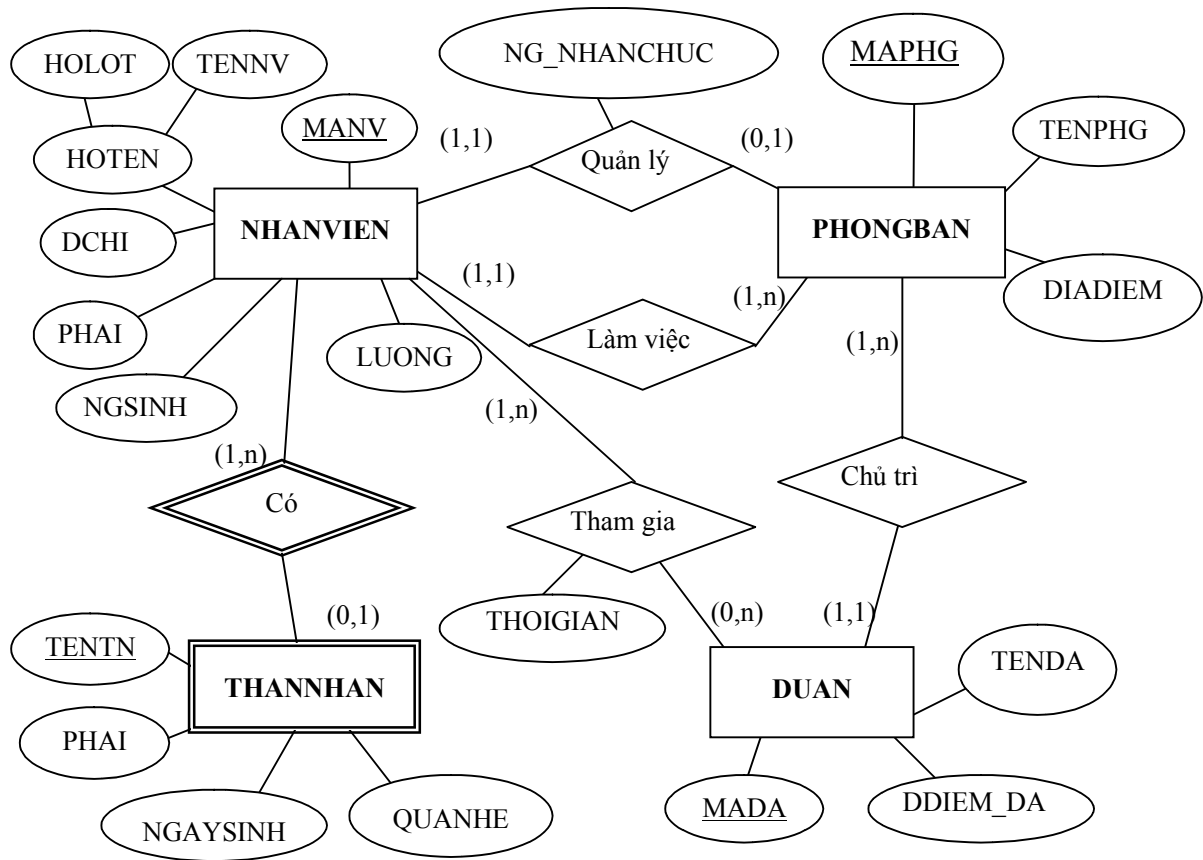
- Mối liên kết NHANVIEN <làm việc cho> PHONGBAN. Đây là mối liên kết 1-n, một nhân viên làm việc cho một phòng ban nhưng một phòng ban có thể có nhiều nhân viên.

- Mối liên kết PHONGBAN <chủ trì> DUAN. Đây là mối liên kết 1-n, một phòng ban có thể chủ trì nhiều dự án và một dự án được chủ trì bởi một phòng ban.

- Mối liên kết NHANVIEN <tham gia> DUAN. Đây là mối liên kết n-n, một nhân viên có thể tham gia làm việc cho nhiều dự án và một dự án có nhiều nhân viên tham gia.

- Mối liên kết NHANVIEN <có> THANNHAN. Đây là mối liên kết 1-n, một nhân viên có thể có nhiều thân nhân. Tập thực thể THANNHAN là tập thực thể yếu vì sự tồn tại của nó phụ thuộc vào sự tồn tại của tập thực thể NHANVIEN.

## ▪ Mô hình ER



Hình 2.20. Mô hình ER cho hệ thống quản lý dự án công ty

## II.4. Mô hình thực thể-liên kết mở rộng

Mô hình thực thể-liên kết mở rộng (EER – Enhanced Entity Relationship Model) bao gồm tất cả các khái niệm của mô hình ER, ngoài ra còn có các khái niệm như mối liên kết ISA, chuyên biệt hóa và tổng quát hóa.

### II.4.1. Lớp cha, lớp con và mối liên kết ISA

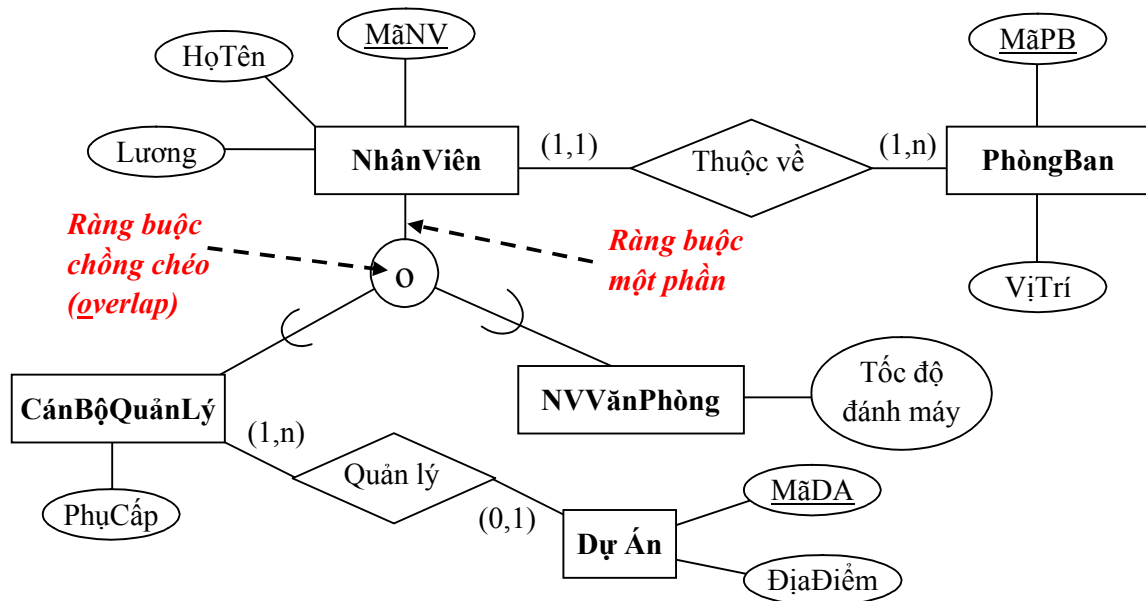
Kiểu liên kết cha-con giữa 2 tập thực thể còn được gọi là mối liên kết ISA. Ta nói rằng A isa B (A là một B) nếu tập thực thể A là một trường hợp đặc biệt của B, điều này có nghĩa A có thể kế thừa các thuộc tính và các mối liên kết của B ngoài ra A có thể có thêm những thuộc tính riêng và các mối liên kết riêng. Khi đó ta nói tập thực thể A kế thừa từ tập thực thể B, B là tập thực thể cha (lớp cha), A là tập thực thể con (lớp con).

Chẳng hạn, các thực thể của tập thực thể NhânViên có thể được chia thành 2 nhóm nhỏ: CánBộQuảnLý và NhânViênVănPhòng, các thực thể trong 2 nhóm này là một tập con của các thực thể trong tập thực thể NhânViên, nghĩa là mỗi thực thể là thành viên của một trong những nhóm này cũng là một nhân viên. Khi đó, ta nói tập thực thể CánBộQuảnLý và NhânViênVănPhòng kế thừa từ tập thực thể NhânViên, NhânViên là tập thực thể cha, CánBộQuảnLý và NhânViênVănPhòng là tập thực thể con.

#### II.4.2. Tổng quát hóa, chuyên biệt hóa

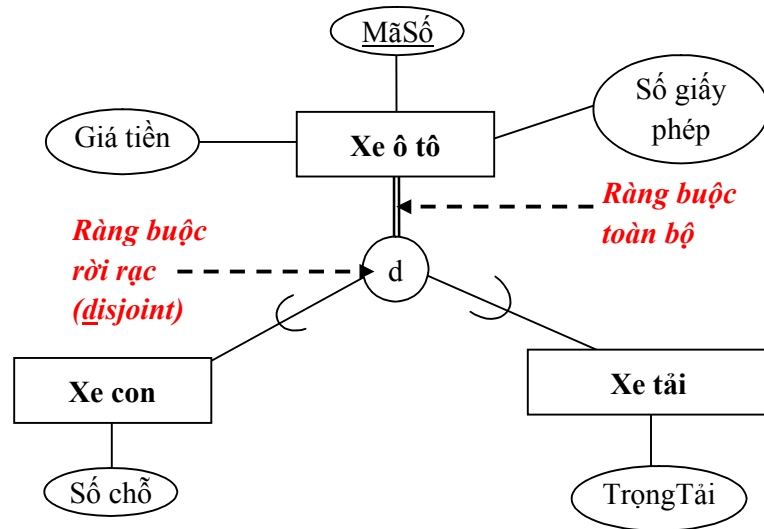
*Chuyên biệt hóa* là quá trình xác định các tập thực thể con (lớp con) của một tập thực thể cha (lớp cha), các lớp con này được tạo dựa trên một số đặc tính riêng biệt nào đó từ các thực thể trong lớp cha.

Chẳng hạn, 2 tập thực thể CánBộQuảnLý và NhânViênVănPhòng là lớp con của tập thực thể NhânViên dựa trên kiểu công việc của các thực thể trong lớp cha. Ngoài những thuộc tính và mối liên kết từ tập thực thể NhânViên, CánBộQuảnLý và NhânViênVănPhòng còn có những thuộc tính và mối liên kết riêng.



Hình 2.21. Chuyên biệt hóa của mô hình EER

*Tổng quát hóa* là quá trình đảo ngược của chuyên biệt hoá, trong đó ta bỏ qua sự khác nhau giữa các tập thực thể, xác định các đặc tính chung của chúng và tổng quát hoá chúng thành một lớp cha của các tập thực thể đó.



Hình 2.22. Tổng quát hóa của mô hình EER

### II.4.3. Các ràng buộc trên mối liên kết ISA

*Ràng buộc rời rạc* chỉ ra rằng các lớp con của một chuyên biệt phải rời rạc nhau, điều này có nghĩa là một thực thể là một thành viên của chỉ duy nhất một trong số các lớp con của chuyên biệt hoá. Nếu các lớp con không thỏa mãn ràng buộc rời rạc được gọi là *ràng buộc chồng chéo*, các thực thể của chúng có thể chồng chéo nhau, nghĩa là một thực thể có thể là một thành viên của nhiều lớp con trong chuyên biệt hoá.

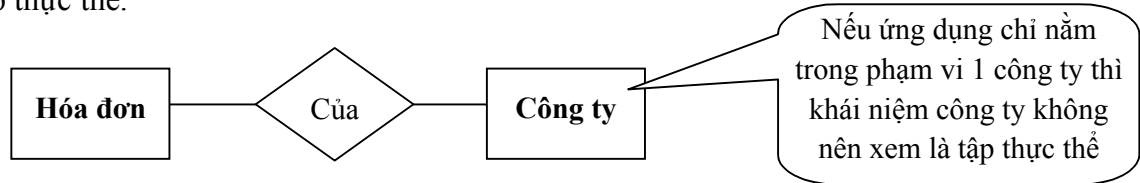
Trong sơ đồ của mô hình EER, nếu các lớp con là thỏa mãn ràng buộc rời rạc thì ở giữa hình tròn nối với các lớp con có ghi chữ d (disjoin), ngược lại nếu các lớp con là thỏa mãn ràng buộc chồng chéo thì ở giữa hình tròn nối các lớp con có ghi chữ o (overlap).

Ràng buộc thứ hai trong chuyên biệt hoá gọi là *ràng buộc đầy đủ*, nó có thể là *ràng buộc toàn bộ* hoặc *ràng buộc từng phần*. Một ràng buộc toàn bộ chỉ ra rằng mỗi thực thể trong lớp cha phải là một thành viên của một lớp con nào đó trong chuyên biệt. Một ràng buộc từng phần cho phép một thực thể của lớp cha có thể không thuộc về bất kỳ lớp con nào. Trong sơ đồ của mô hình EER, nếu các lớp con là thỏa mãn ràng buộc toàn bộ thì đường thẳng nối từ lớp cha đến hình tròn là đường nét đôi.

Xem ví dụ trong hình 2.21, một nhân viên cũng có thể vừa là nhân viên văn phòng, vừa là cán bộ quản lý (nên thỏa ràng buộc chồng chéo) và cũng có thể một nhân viên không thuộc 2 loại nhân viên trên, chẳng hạn là nhân viên kinh doanh (thỏa ràng buộc từng phần).

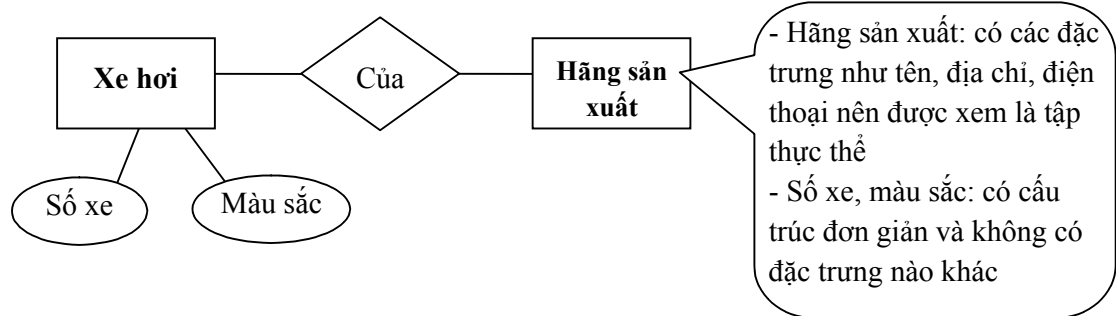
## ❖ Tiêu chí lựa chọn giữa các khái niệm

▪ *Tập thực thể hay không là tập thực thể?* Trong một số trường hợp thì khái niệm cần biểu diễn có thể là một đối tượng của thế giới thực nhưng trong phạm vi ứng dụng thì số thể hiện chỉ là một. Nếu không có nhu cầu mở rộng về sau thì không nên xem là tập thực thể.



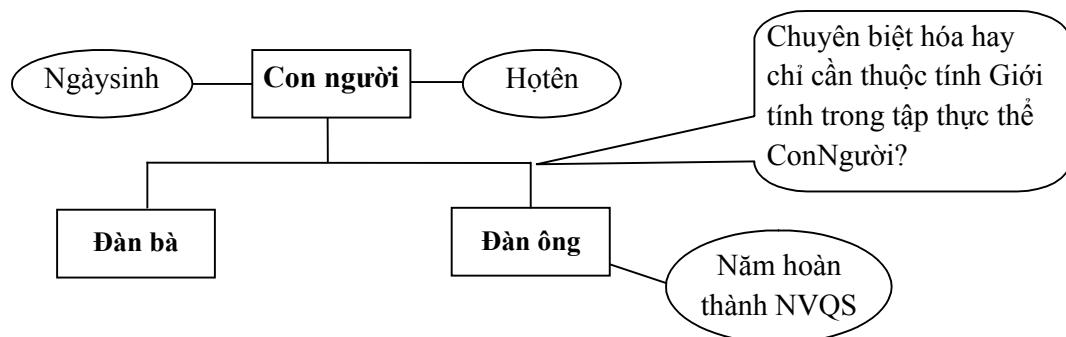
Hình 2.23. Tập thực thể hay không là tập thực thể

▪ *Tập thực thể hay thuộc tính?* Chọn là tập thực thể khi có thể xác định một số đặc trưng căn bản như thuộc tính, mối liên kết. Chọn là thuộc tính khi đối tượng có cấu trúc nguyên tố đơn giản và không có các đặc trưng khác.

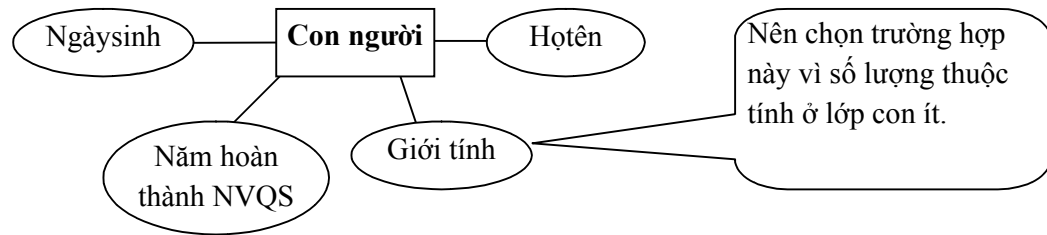


Hình 2.24. Tập thực thể hay thuộc tính

▪ *Chuyên biệt hóa hay thuộc tính?* Chuyên biệt hóa được chọn khi chúng ta cho rằng có một số đặc trưng ở cấp thấp hơn (như thuộc tính hay mối liên kết). Trường hợp ngược lại sẽ là thuộc tính.

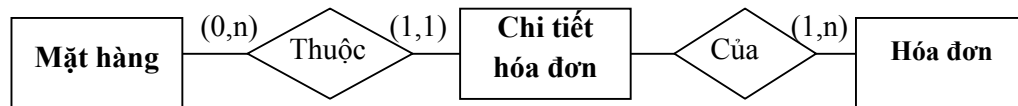


Hay

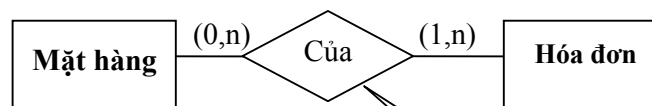


Hình 2.25. Chuyên biệt hóa hay thuộc tính

▪ *Mối liên kết hay tập thực thể?* Chọn là tập thực thể nếu khái niệm quan tâm có một số đặc trưng cần mô hình hóa, chẳng hạn như các mối liên kết đến các tập thực thể khác.



Hay



Hình 2.26. Mối liên kết hay tập thực thể

Là tập thực thể Chi tiết hóa đơn hay chỉ là mối liên kết? 2 trường hợp trong ví dụ này là như nhau.

## II.5. Tổng kết chương và câu hỏi ôn tập

### II.5.1. Tổng kết chương

Chương này đã giới thiệu các khái niệm trong mô hình ER: thực thể, tập thực thể, mối liên kết, thuộc tính,... và ứng dụng vào xây dựng một mô hình ER cụ thể cho hệ thống quản lý dự án của một công ty. Thêm vào đó đó, chương này cũng đã trình bày mô hình EER - mở rộng của mô hình ER, các khái niệm “mở rộng” trong mô hình ER như mối liên kết ISA, lớp cha/lớp con, chuyên biệt hoá, tổng quát hoá.

### II.5.2. Câu hỏi ôn tập

1. Giải thích sự khác nhau giữa thực thể và tập thực thể?
2. Thế nào là tập thực thể yếu?

3. Khóa dùng để làm gì? Khi nào thì một thuộc tính là khóa?
4. Mỗi liên kết là gì? Có những loại mỗi liên kết nào?
5. Mỗi liên kết có thuộc tính không?
6. Thế nào là mỗi liên kết đa ngôi?
7. Thế nào là mỗi liên kết ISA?
8. Tổng quát hóa và chuyên biệt hóa giống và khác nhau ở chỗ nào?
9. Có những ràng buộc nào trên mỗi liên kết ISA?

### II.5.3. Bài tập

**1. Hãy sửa lại mô hình ER *Quản lý dự án* công ty trong mục II.3 sao cho phù hợp với các yêu cầu được thêm vào dưới đây:**

- Mỗi nhân viên có thể được *quản lý trực tiếp* bởi một nhân viên khác, một nhân viên có thể quản lý trực tiếp nhiều nhân viên
- Mỗi phòng ban có thể thuộc nhiều địa điểm và một địa điểm có thể có nhiều phòng ban.

**2. Xây dựng mô hình ER cho hệ thống *Quản lý khách sạn* với các mô tả sau:**

- Một khách sạn cần xây dựng một hệ thống thông tin về việc thuê phòng và sử dụng các dịch vụ của khách trọ bao gồm các chức năng sau: quản lý phòng trọ, quản lý khách trọ, quản lý các dịch vụ, tính tiền và in hóa đơn cho khách trọ (tiền thuê phòng + tiền sử dụng dịch vụ).
- Thông tin về phòng gồm mã phòng, loại phòng .
- Thông tin về khách trọ gồm mã khách, họ tên, số CMND, địa chỉ, quốc tịch.
- Một khách trọ có thể đến thuê phòng tại khách sạn này nhiều lần, thông tin mỗi lần thuê của một khách gồm phòng thuê, ngày bắt đầu, ngày kết thúc.
- Giả sử tất cả các phòng đều là phòng đơn (phòng một người). Đơn giá thuê / một ngày của một phòng được ấn định trước tùy theo phòng thuộc loại nào. Trong mỗi lần thuê phòng, khách trọ có thể trả thêm các khoản tiền về dịch vụ (như điện thoại, ăn uống, karaoke,...)

**3. Xây dựng mô hình ER cho hệ thống *Quản lý chuyến bay* với các mô tả sau:**

- Một công ty hàng không cần xây dựng một hệ thống thông tin về các tuyến bay trong nước và nước ngoài do công ty quản lý bao gồm các chức năng sau: quản lý



tuyến bay/chuyến bay, quản lý đơn vị bay, quản lý phi công, quản lý thông tin khách hàng.

- Thông tin về tuyến bay gồm mã tuyến bay, tên thành phố khởi hành, tên thành phố đến, đơn giá vé, số giờ bay.
- Mỗi tuyến bay có nhiều chuyến bay; thông tin về chuyến bay gồm mã chuyến bay, loại máy bay, các phi công lái, các hành khách đi trên chuyến bay.
- Thông tin về khách hàng gồm mã số, họ tên, ngày sinh, phái; một hành khách có thể đi trên nhiều chuyến bay khác nhau.
- Công ty có nhiều đơn vị bay, thông tin về đơn vị bay gồm mã số, tên đơn vị, mỗi đơn vị bay gồm nhiều phi công
- Thông tin về phi công gồm mã số, họ tên, ngày sinh, phái; Mỗi phi công chỉ thuộc một đơn vị bay, và mỗi phi công có thể lái nhiều chuyến bay.

#### **4. Xây dựng Mô hình ER cho hệ thống *Quản lý sinh viên* với các mô tả sau:**

- Trường được chia thành các trường con: Trường ĐH Duy Tân, Trường ĐH Kinh tế, Trường ĐH Bách Khoa,... Mỗi trường có một mã, tên, năm thành lập và có một hiệu trưởng quản lý. Mỗi hiệu trưởng quản lý một trường.
- Mỗi trường có nhiều khoa. Chẳng hạn, trường ĐH Bách Khoa có các khoa Tin học, Toán, Hóa,... Mỗi khoa chỉ thuộc về một trường. Thông tin về Khoa gồm mã khoa, tên khoa, địa chỉ, số điện thoại.
- Mỗi khoa có nhiều lớp, thông tin của lớp gồm mã lớp, tên lớp, chế độ đào tạo (chính qui, liên thông...), mỗi lớp có 1 lớp trưởng và lớp trưởng là 1 sinh viên.
- Mỗi lớp có nhiều sinh viên, mỗi sinh viên chỉ thuộc về một lớp. Thông tin về mỗi sinh viên gồm: Mã sinh viên, Họ tên, địa chỉ, ngày sinh, giới tính, lớp
- Trong một học kỳ, mỗi lớp được học nhiều môn học và một môn học được nhiều lớp học. Với mỗi lớp, một môn học chỉ do 1 giáo viên giảng dạy. Mỗi môn học gồm có tên môn học, mã số, số tín chỉ.
- Mỗi khoa có nhiều giáo viên làm việc, nhưng mỗi giáo viên chỉ làm việc cho một khoa. Mỗi một khoa có một chủ nhiệm khoa, đó là một giáo viên.
- Mỗi lớp có một giáo viên chủ nhiệm.
- Sau mỗi học kỳ sẽ có một danh sách điểm để phân loại. Nó gồm các thông tin: mã sinh viên, mã môn học, điểm bằng chữ, điểm bằng số.

#### **5. Xây dựng Mô hình ER cho hệ thống *Quản lý thuê băng đĩa* với các mô tả sau:**

- Một cửa hàng cho thuê băng đĩa cần xây dựng một hệ thống thông tin về việc thuê băng đĩa của khách hàng bao gồm các chức năng sau: quản lý thông tin về các băng đĩa của cửa hàng, quản lý thông tin khách hàng, quản lý thuê/trả băng đĩa của khách hàng.
- Mỗi cuốn băng/đĩa có tên, và thời điểm phát hành và thuộc một hãng sản xuất băng đĩa (Sony, JVC,...)
- Mỗi phim có thể được chứa trong nhiều cuốn băng/đĩa, thông tin về một phim gồm tên phim, năm sản xuất, tên đạo diễn.
- Một phim có nhiều diễn viên đóng và chỉ do một hãng phim sản xuất. Thông tin về diễn viên gồm họ tên, ngày sinh, phái. Thông tin về hãng phim gồm tên hãng, năm thành lập.
- Thông tin của khách hàng gồm tin họ tên, ngày sinh, phái, địa chỉ, CMND. Mỗi khách hàng có thể đến cửa hàng này thuê băng đĩa nhiều lần, cửa hàng cần ghi lại thông tin về băng thuê, ngày giờ thuê, hạn trả, ngày giờ trả băng (sau khi khách hàng trả băng), và số tiền thuê.

**6. Xây dựng Mô hình ER cho hệ thống *Quản lý bán hàng* của Công ty Dược phẩm Thuận Thảo với hóa đơn bán hàng như sau:**

**HÓA ĐƠN** Mẫu số: 01/GTKT-3LL  
GIÁ TRỊ GIA TĂNG DL/2008B  
Liên 2: Giao khách hàng

Ngày: 23 tháng 07 năm 2008

Đơn vị bán hàng: Công ty TNHH DƯỢC PHẨM THUẬN THẢO  
Địa chỉ: Đc: 34 Ngõ Quyển TP Huế  
Số tài khoản: MS: 33.003.752.13  
Điện thoại: MS:

Họ tên người mua hàng: VANG PHONG OG CDE  
Tên đơn vị: 06 Ngõ Quyển Huế  
Địa chỉ: MS:  
Số tài khoản: MS:  
Hình thức thanh toán: TM MS:

STT	Tên hàng hoá, dịch vụ	Đơn vị tính	Số lượng	Đơn giá	Thành tiền
A	B	C	1	2	3=1x2
1	Prednisolon 5mg	viên	2.000	133	266.000
2	Cobum 480mg	viên	300	350	105.000
3	Cyproflaxacin 500mg	viên	1.000	381	381.000
4	Dop	lọ	200	1905	381.000
5	Torpan Cedem	viên	1.000	286	286.000
6	Cloramphenicol 0,1%	viên	1.000	762	762.000
7	Vitamin B6 (Pyridoxine)	viên	2.000	143	286.000
8	Methylnitramin	viên	2.000	190	380.000
9	Cyfloracin	viên	1.000	952	952.000
Tổng cộng hàng:					3.799.000
Thuế suất GTGT: 0% Tiền thuế GTGT:					189.950
Tổng cộng tiền thanh toán:					3.988.950
Số tiền viết bằng chữ: Ba triệu chín trăm tám mươi nghìn, chín trăm tám mươi đồng					
Người mua hàng (Ký, ghi rõ họ tên)		Người bán hàng (Ký, ghi rõ họ tên)		Thủ trưởng đơn vị (Ký, đóng dấu, ghi rõ họ tên)	
		Thủy Người Thu Hàng			

(Cần kiểm tra, đối chiếu khi lập, giao, nhận hóa đơn)

### Chương III. MÔ HÌNH QUAN HỆ

#### A. Mục tiêu

##### ➤ Về kiến thức:

Sau khi xong chương này sinh viên nắm vững các kiến thức sau:

- Nắm vững các khái niệm trong mô hình quan hệ: thuộc tính, quan hệ, bộ giá trị, khóa chính, khóa ngoại,...
- Nắm rõ cách chuyển đổi từ mô hình ER sang mô hình quan hệ
- Hiểu rõ các phép toán đại số quan hệ và ứng dụng giải một số bài tập liên quan đến vấn đề này.

##### ➤ Về kỹ năng

Giúp sinh viên phát triển các kỹ năng:

- Mạnh dạn phát biểu ý kiến.
- Rèn luyện khả năng làm việc theo nhóm.
- Rèn luyện khả năng tự học, tự nghiên cứu
- Chăm chỉ, chủ động, sáng tạo trong học tập

#### B. Phương pháp giảng dạy

- Trình chiếu slides
- Đặt câu hỏi cho sinh viên trả lời
- Cho sinh viên thảo luận theo nhóm
- Làm bài tập tại lớp và nộp vào cuối giờ.
- Yêu cầu sinh viên làm bài tập về nhà.

#### C. Tài liệu tham khảo

1. Nguyễn An Tế, Đồng Thị Bích Thủy (1996). *Nhập Môn Cơ Sở Dữ Liệu*.
2. Nguyễn Đăng Ty, Đỗ Phúc (2003). *Giáo Trình Cơ Sở Dữ Liệu*.
3. Lê Tiên Vương (1994). *Nhập Môn Cơ Sở Dữ Liệu Quan Hệ*
4. Jeffrey D. Ullman (1998). *Mô Hình Dữ Liệu Và Ngôn Ngữ Ván Tin*. Tập 1
5. Hoàng Trung Sơn (2004). *Thiết Kế Cơ Sở Dữ Liệu*.

#### D. Nội dung chi tiết

### III.1. Các khái niệm trong mô hình quan hệ

*Mô hình quan hệ* (Relational Model) được E.F.Codd đề xuất vào năm 1970 dựa trên nền tảng rất vững chắc về lý thuyết tập hợp. Trước mô hình quan hệ là hai là *mô hình mạng* (Network Model) và *mô hình phân cấp* (Hierarchical Model), nhưng khi vừa mới ra đời mô hình quan hệ đã lật đổ hoàn toàn hai mô hình trước nó và nhanh chóng thống trị trong các hệ quản trị cơ sở dữ liệu hàng đầu thế giới. Cho đến ngày nay mô hình quan hệ vẫn là số 1, người ta thống kê trên 90% các hệ quản trị cơ sở dữ liệu trên thế giới được cài đặt theo mô hình quan hệ.

Ưu điểm của mô hình quan hệ so với các mô hình dữ liệu trước nó ở chỗ nó đơn giản và truy vấn dữ liệu dễ dàng. Một CSDL bao gồm các *quan hệ* (relation), mỗi quan hệ như là một *bảng* gồm nhiều cột và nhiều dòng, dữ liệu trình bày theo dạng bảng nên rất dễ hiểu đối với người sử dụng kể cả những người mới bắt đầu. Mô hình quan hệ cho phép sử dụng ngôn ngữ cấp cao (high-level language) để truy vấn dữ liệu, những câu truy vấn phức tạp đều có thể thực hiện đơn giản và cho kết quả rất nhanh chóng.

Hãy xét một quan hệ NHANVIEN trong CSDL quản lý nhân viên như sau:

MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
01	Lê Tiến	Huy	1/1/1980	Nam	500
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
03	Lê Thị	Lan		Nữ	200
04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

Nhìn vào quan hệ NHANVIEN, ta dễ dàng thấy rằng hiện tại có tất cả 4 nhân viên, mỗi dòng là thông tin của một nhân viên. Vì chỉ có 4 nhân viên nên việc tìm kiếm thông tin của nhân viên quá đơn giản (chỉ cần liếc mắt là thấy ngay!), tuy nhiên nếu số lượng nhân viên trong quan hệ này lên tới con số hàng ngàn thì việc tìm kiếm bằng mắt thường không còn đơn giản. Lúc bấy giờ ta có thể sử dụng ngôn ngữ cấp cao chuyên dùng để truy xuất dữ liệu, đó là ngôn ngữ SQL. Chẳng hạn, ta muốn đếm xem “có bao nhiêu nhân viên sinh vào tháng 11”, câu truy vấn được viết đơn giản như sau:

```
Select count (*)
From NHANVIEN
Where month(NGSINH)=11
```

Ngôn ngữ truy vấn dữ liệu SQL sẽ được trình bày chi tiết trong chương 5. Trong chương 3 này chúng ta sẽ tìm hiểu về những khái niệm cơ bản nhất trong mô hình

quan hệ: thuộc tính, lược đồ quan hệ, thể hiện quan hệ, bộ giá trị, khóa, và các phép toán có thể thực hiện trên mô hình quan hệ. Đây là một chương rất quan trọng và mấu chốt về CSDL cần phải nắm vững.

### III.1.1. Các khái niệm

#### III.1.1.1. Thuộc tính

*Thuộc tính* (attribute) là các đặc tính riêng dùng để mô tả đối tượng. Mỗi thuộc tính có tên để phân biệt với thuộc tính khác, có kiểu dữ liệu (data type) và miền giá trị (domain). Thuộc tính còn được gọi là cột (column) hay trường (field).

Chẳng hạn, đối tượng NHANVIEN có các thuộc tính: MANV, TENNV, NGSINH, PHAI, LUONG,... Trong đó, thuộc tính PHAI có kiểu dữ liệu chuỗi và miền giá trị của thuộc tính phải là  $Dom(PHAI) = \{“Nam”, “Nữ”\}$ :

Miền giá trị của thuộc tính còn có thể chứa thêm một giá trị đặc biệt gọi là *giá trị rỗng* (null). Tùy theo ngữ cảnh, giá trị rỗng có thể đặc trưng cho một giá trị *không thể xác định* được hoặc đặc trưng cho một giá trị *chưa thể xác định* ở thời điểm đang xét.

Trong các ứng dụng thực tế, người phân tích-thiết kế thường đặt tên thuộc tính một cách gợi nhớ. Theo lý thuyết, người ta vẫn khuyên rằng không nên đặt tên thuộc tính quá dài bởi vì nó làm cho việc viết các câu lệnh truy vấn trở nên vất vả hơn, và cũng không nên đặt tên quá ngắn vì nó không cho thấy được ngữ nghĩa của chúng, đặc biệt là không nên đặt trùng tên hai thuộc tính mang ngữ nghĩa khác nhau thuộc hai đối tượng khác nhau. Chẳng hạn, nếu có hai đối tượng SINHVIEN và GIANGVIEN đều có thuộc tính TEN thì nên đặt thuộc tính rõ ràng là TENSX và TENGX.

#### III.1.1.2. Quan hệ

Cấu trúc chính để trình bày dữ liệu trong mô hình quan hệ chính là *quan hệ* (relation). Một quan hệ bao gồm *lược đồ quan hệ* (relation schema) và *thể hiện quan hệ* (relation instance).

#### Lược đồ quan hệ

*Lược đồ quan hệ* (relation schema) bao gồm tên quan hệ, tên các thuộc tính, kiểu dữ liệu và miền giá trị của các thuộc tính. Chẳng hạn ta có lược đồ quan hệ NHANVIEN như sau:


NHANVIEN(MaNV: *string*, HOLOT: *string*, TENNV: *string*, NGAYSINH: *datetime*, PHAI: *string*, LUONG: *int*)

### Thể hiện quan hệ

*Thể hiện quan hệ* (relation instance) là tập hợp các bộ giá trị (tuple), còn được gọi là mẫu tin (record). Số lượng các giá trị trong mỗi bộ bằng với số lượng các thuộc tính trong lược đồ quan hệ. Một thể hiện quan hệ có thể xem như là một bảng với các bộ giá trị là các dòng (row), tất cả các dòng này có cùng số lượng cột (column). Thông thường, *thể hiện quan hệ* được gọi ngắn gọn là *quan hệ*.

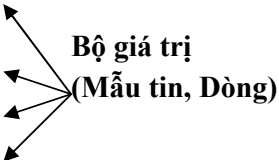
Ví dụ 3.1. Một thể hiện của lược đồ NHANVIEN như sau:

**Thuộc tính (Trường, Cột)**



MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
01	Lê Tiến	Huy	1/1/1980	Nam	500
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
03	Lê Thị	Lan		Nữ	200
04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

**Bộ giá trị  
(Mẫu tin, Dòng)**



Một cách hình thức, một lược đồ quan hệ R được định nghĩa:

$$R ( A_1 : D_1, A_2 : D_2, \dots, A_n : D_n )$$

Trong đó R là tên của lược đồ quan hệ,  $A_i$  ( $1 \leq i \leq n$ ) là tên các thuộc tính,  $D_i$  ( $1 \leq i \leq n$ ) là miền giá trị của  $A_i$  và được ký hiệu  $\text{Dom}(A_i)$ . Một thể hiện của lược đồ quan hệ R là tập hợp các bộ giá trị  $\{t_1, t_2, \dots, t_m\}$ . Mỗi bộ giá trị  $t_j$  ( $1 \leq j \leq m$ ) gồm n trường  $\{v_1, v_2, \dots, v_n\}$ , trong đó mỗi giá trị  $v_k$  ( $1 \leq k \leq n$ ) là một phần tử của miền giá trị  $\text{Dom}(A_k)$  hoặc giá trị rỗng.

### Lược đồ CSDL quan hệ và CSDL quan hệ:

*Lược đồ CSDL quan hệ* (Relational DataBase Schema) tập hợp tất cả các lược đồ quan hệ trong CSDL.

Ví dụ 3.2. Ta có lược đồ CSDL quan hệ trong CSDL quản lý dự án như sau:

**NHANVIEN** (MANV: *string*, HOLOT: *string*, TENNV: *string*, NGAYSINH: *datetime*, PHAI: *string*, LUONG: *int*, DCHI: *string*, MA\_NQL: *string*, PHG: *string*)

**PHONGBAN** (MAPHG:string, TENPHG:string, TRPHG:string, NG\_NHANCHUC:string)

**DIADIEM\_PHONG** (MAPHG:string, DIADIEM:string)

**DUAN** (MADA:string, TENDA:string, DDIEM\_DA:string, PHONG:string)

**PHANCONG** (MA\_NVIEN:string, SODA:string, THOIGIAN:float)

**THANNHAN** (MA\_NVIEN:string, TENTN:string, NGAYSINH:datetime, PHAI:string, QUANHE: string)

*CSDL quan hệ* (Relational DataBase) là tập hợp các thể hiện quan hệ trong CSDL. Trong CSDL quản lý dự án trên, CSDL quan hệ chính là tập các thể hiện của 5 lược đồ quan hệ trên, dĩ nhiên mỗi thể hiện quan hệ phải thỏa mãn những ràng buộc miền giá trị của các lược đồ quan hệ tương ứng.

### III.1.1.3. Bộ giá trị

Bộ giá trị là các thông tin của một đối tượng thuộc quan hệ, bộ giá trị còn được gọi là mẫu tin (record) hay dòng (row). Để lấy thành phần thứ  $A_i$  (tức là giá trị thuộc tính  $A_i$ ) của bộ giá trị  $t$  ta viết  $t[A_i]$  (hoặc  $t.A_i$ ), phép rút trích này gọi là phép chiếu một bộ lên thuộc tính  $A_i$ .

Chẳng hạn,  $t = \langle 01, \text{Lê Tiến, Huy}, 1/1/1980, \text{Nam}, 500 \rangle$  là một bộ của quan hệ NHANVIEN (MANV, HOLOT, TENNV, NGSINH, PHAI, LUONG). Khi đó,  $t[\text{TENNV}] = \text{“Huy”}$ .

### III.1.2. Các đặc trưng của quan hệ

- Thứ tự các bộ trong quan hệ là không quan trọng: Một quan hệ được định nghĩa như một tập hợp các bộ, các phần tử trong một tập hợp không có thứ tự, vì vậy các bộ trong một quan hệ không có một thứ tự cụ thể. Các quan hệ chứa cùng một số hàng như nhau nhưng các hàng được sắp xếp khác nhau được xem như đồng nhất với nhau.

- Thứ tự giữa các giá trị trong một bộ là quan trọng: Theo định nghĩa quan hệ là tập hợp  $n$ -bộ là một danh sách có thứ tự của  $n$  giá trị, vì thứ tự của các giá trị trong một bộ là quan trọng, từ đó suy ra thứ tự của các thuộc tính trong một lược đồ quan hệ cũng quan trọng. Tuy nhiên, theo định nghĩa trong mục III.1.1.2 về quan hệ, một bộ có thể xem như một tập hợp các cặp ( $\langle \text{thuộc tính} \rangle, \langle \text{giá trị} \rangle$ ), nếu tên thuộc tính luôn xuất hiện cùng với giá trị của nó thì thứ tự của các thuộc tính là không quan trọng.

- Mỗi giá trị trong một bộ là phải là một giá trị nguyên tử, điều đó có nghĩa là nó không phân chia được thành các thành phần nhỏ hơn. Như vậy, trong mô hình quan hệ không cho phép có các thuộc tính phức hợp và các thuộc tính đa trị. Các thuộc tính đa trị phải được biểu diễn bằng các quan hệ còn các thuộc tính phức hợp chỉ được biểu diễn bằng các thuộc tính thành phần đơn của nó.
- Các giá trị của một vài thuộc tính trong một bộ cụ thể có thể không biết được hoặc chưa biết được, khi đó giá trị của các thuộc tính này là giá trị null.
- Không có bộ nào trùng nhau trong mỗi thể hiện của lược đồ quan hệ.

### III.2. Các ràng buộc toàn vẹn trên quan hệ và lược đồ CSDL quan hệ

Ràng buộc toàn vẹn là những ràng buộc trên lược đồ CSDL mà tất cả các dữ liệu phải thỏa mãn để đảm bảo *tính nhất quán và toàn vẹn dữ liệu*. Phần này chúng ta chỉ tìm hiểu về các ràng buộc về khóa, khóa chính và khóa ngoại, các ràng buộc khác như ràng buộc miền giá trị, ràng buộc liên thuộc tính,... sẽ được trình bày chi tiết trong chương 6.

#### III.2.1. Khóa - Siêu khóa

**Định nghĩa:** Cho lược đồ quan hệ  $R$  được định nghĩa trên tập thuộc tính  $U = \{A_1, A_2, \dots, A_n\}$ . Một tập con  $K \subseteq U$  là **khóa** (Key) của  $R$  nếu thỏa đủ 2 điều kiện sau:

- (1)  $K$  là tập thuộc tính xác định duy nhất bộ giá trị trong  $R$
- (2) Mọi tập thuộc tính  $K'$  con của  $K$  ( $K' \neq K$ ) đều không có tính chất (1)

Điều này có nghĩa là không tồn tại hai bộ nào có giá trị bằng nhau trên mọi thuộc tính của  $K$  (điều kiện 1) và  $K$  là tập con nhỏ nhất thỏa mãn tính chất này (điều kiện 2). Ý nghĩa của khóa là dùng để nhận diện một bộ giá trị trong quan hệ, khi cần truy tìm bộ  $t$  nào đó ta chỉ cần biết giá trị của các thuộc tính khóa của  $t$  là đủ, không nhất thiết phải biết tất cả giá trị của các thuộc tính.

*Ví dụ 3.3.* Xét quan hệ:

NHANVIEN(MANV, HOLOT, TENNV, NGSINH, PHAI, LUONG, CMND)

$K1 = \{ \text{MANV} \}$  : là khóa

$K2 = \{ \text{MANV, TENNV} \}$  : không là khóa, thỏa (1) nhưng không thỏa (2).  $K2$  được gọi là **siêu khóa** (Super Key)



K3 = {CMND}: là khóa

K4 = { TENNV, NGSINH}: không là khóa, vì không thỏa (1) và (2).

### III.2.2. Khóa chính

Trong một quan hệ có thể có nhiều khóa, thông thường ta phải chỉ định một khóa làm cơ sở để nhận biết các bộ, khóa này được gọi là *khóa chính* (Primary Key). Qui ước rằng, trong một lược đồ quan hệ *khóa chính sẽ được gạch dưới*. Lưu ý, việc lựa chọn một khóa làm khóa chính là tùy ý, tuy nhiên tốt nhất là nên chọn khóa chính có số thuộc tính là ít nhất.

Chẳng hạn, trong ví dụ 3.3 ta nhận thấy quan hệ NHANVIEN có 2 khóa là MANV và CMND, ở đây ta chọn MANV làm khóa chính cho quan hệ NHANVIEN.

NHANVIEN(MANV, HOLOT, TENNV, NGSINH, PHAI, LUONG, CMND)

Ví dụ 3.4. Xác định khóa chính cho CSDL quản lý dự án.

NHANVIEN(MANV, HOLOT, TENNV, NGSINH, PHAI, LUONG, DCHI, MA\_NQL, PHG)

PHONGBAN(MAPHG, TENPHG, TRPHG, NG\_NHANCHUC)

DIADIEM\_PHONG(MAPHG, DIADIEM)

DUAN(MADA, TENDA, DDIEM\_DA, PHONG)

PHANCONG(MA\_NVIEN, SODA, THOIGIAN)

THANNHAN(MA\_NVIEN, TENTN, NGAYSINH, PHAI, QUANHE)

#### ▪ Lưu ý đối với khóa chính:

- Giá trị các thuộc tính của khoá chính trong quan hệ phải khác rỗng (null)

<u>MANV</u>	HOLOT	TENNV	NGSINH	PHAI	LUONG
01	Lê Tiến	Huy	1/1/1980	Nam	500
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
03	Lê Thị	Lan		Nữ	200
	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

- Giá trị các thuộc tính của khoá chính không được trùng nhau

<u>MANV</u>	HOLOT	TENNV	NGSINH	PHAI	LUONG
02	Lê Tiến	Huy	1/1/1980	Nam	500
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
03	Lê Thị	Lan		Nữ	200
04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

Trong hầu hết các HQTCSDL như MS Access, SQL Server, Oracle, ... có cài đặt cơ chế tự động kiểm tra hai yêu cầu trên của khóa chính. Tức là nếu thêm vào quan hệ

một bộ mới có giá trị khóa chính null hoặc giá trị khóa chính trùng với một bộ nào đó đã có trong quan hệ thì hệ thống sẽ báo lỗi và yêu cầu nhập lại giá trị khác.

### III.2.3. Khóa ngoại

Khóa ngoại (Foreign Key) dùng để biểu diễn mối quan hệ giữa hai quan hệ với nhau, diễn tả sự tương ứng giữa các bộ của hai quan hệ. Nói cách khác, khóa ngoại được hiểu là sự tồn tại của một giá trị thuộc một bộ trong một quan hệ có liên quan đến sự tồn tại của một của giá trị thuộc một bộ khác trong một quan hệ khác.

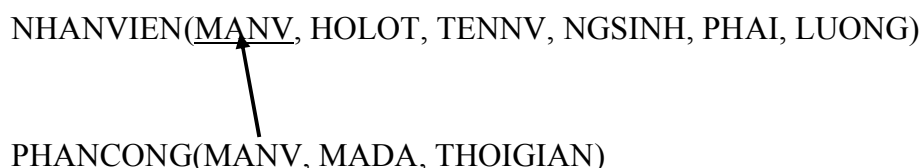
Một tập hợp các thuộc tính FK trong quan hệ  $R_1$  là một *khóa ngoại* của  $R$  tham chiếu đến quan hệ  $S$  nếu nó thỏa mãn hai quy tắc sau:

- (1) Các thuộc tính trong FK của quan hệ  $R$  có cùng miền giá trị như các thuộc tính của khóa chính PK của quan hệ  $S$ .
- (2) Một giá trị của các thuộc tính FK trong một bộ  $t_R$  của quan hệ  $R$  (chính là giá trị  $t_R[FK]$ ) hoặc là null, hoặc có mặt như một giá trị của khóa chính PK của một bộ  $t_S$  trong quan hệ  $S$  (chính là giá trị  $t_S[PK]$ ), có nghĩa là  $t_R[FK] = t_S[PK]$ .

Để dễ dàng xác định khóa ngoại của một quan hệ, xem xét khái niệm khác về khóa ngoại như sau: Một tập thuộc tính FK của quan hệ  $R$  được gọi là khóa ngoại của quan hệ  $R$  tham chiếu đến quan hệ  $S$  nếu nó là khóa chính của quan hệ  $S$ .

Để biểu diễn khóa ngoại một trong lược đồ quan hệ, ta vẽ một đoạn thẳng có hướng từ khóa ngoại đến khóa chính của quan hệ mà nó tham chiếu tới. Chú ý rằng một khóa ngoại của một quan hệ có thể tham chiếu đến khóa chính của chính quan hệ đó, trong trường hợp này khóa ngoại biểu thị một liên kết đệ quy.

Ví dụ 3.5. Xét 2 lược đồ quan hệ sau:



MaNV là khóa ngoại của quan hệ PHANCONG vì nó là khóa chính trong quan hệ NHANVIEN. Ta nói MaNV là khóa ngoại của quan hệ PHANCONG tham chiếu tới MANV của quan hệ NHANVIEN.

Để hiểu kỹ hơn về khóa ngoại, ta xét thể hiện của hai lược đồ quan hệ trên:

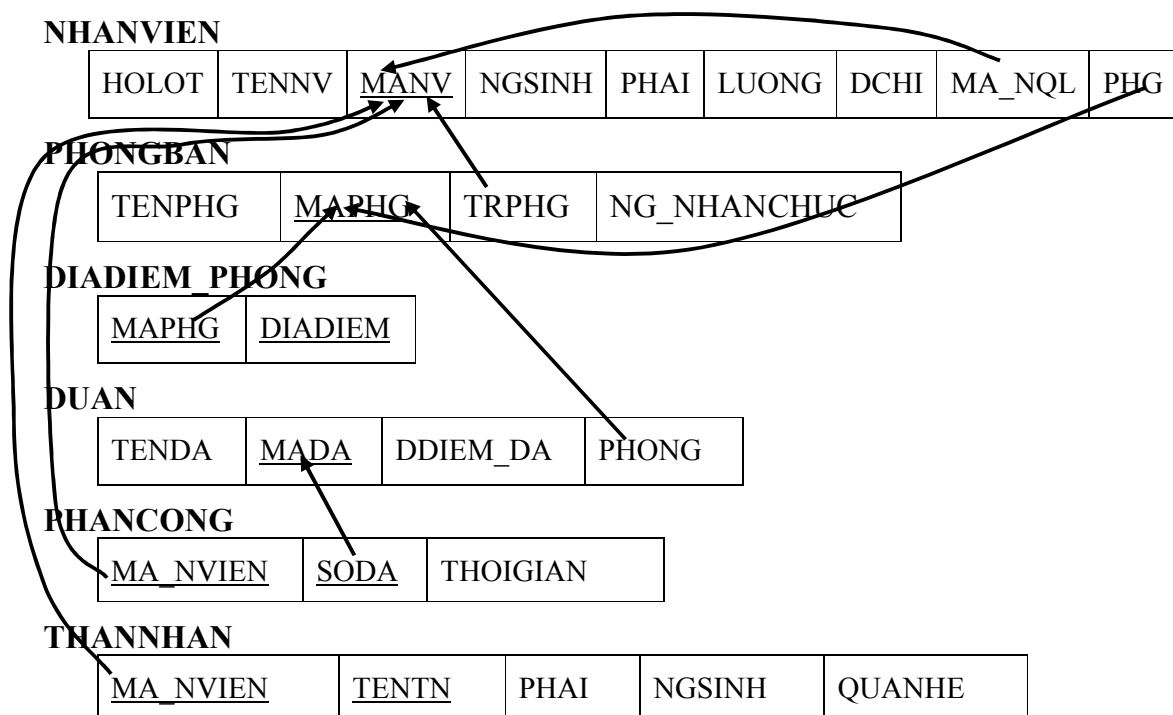
<u>MANV</u>	<u>MADA</u>	<u>THOIGIAN</u>
01	D001	5
01	D002	3
02	D001	4
05	D003	3

<u>MANV</u>	<u>HOLOT</u>	<u>TENNV</u>	<u>NGSINH</u>	<u>PHAI</u>	<u>LUONG</u>
01	Lê Tiến	Huy	1/1/1980	Nam	500
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
03	Lê Thị	Lan		Nữ	200
04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

Vì MANV trong quan hệ PHANCONG là khóa ngoại tham chiếu đến MANV trong quan hệ NHANVIEN, nên những giá trị của trường MANV trong PHANCONG phải tồn tại trong tập những giá trị của trường MANV trong NHANVIEN. Tuy nhiên, trong ví dụ trên tồn tại một nhân viên có mã 05 tham gia vào dự án D003 với thời gian 3 ngày mà nhân viên này lại không tồn tại trong quan hệ NHANVIEN. Vậy bộ giá trị <05, D003, 3> trong quan hệ PHANCONG vi phạm ràng buộc khóa ngoại. Về nguyên tắc, các HQTCSĐL sẽ không cho chèn bộ giá trị này vào CSĐL.

Một lưu ý nhỏ trong khóa ngoại, giá trị của thuộc tính làm khóa ngoại cho phép giá trị null và thuộc tính làm khóa ngoại và thuộc tính được tham chiếu tới không nhất thiết phải trùng tên với nhau, chỉ cần cùng miền giá trị.

**Ví dụ 3.6.** Xác định tất cả các khóa ngoại cho lược đồ quản lý dự án:



### III.3. Các phép toán trong mô hình quan hệ

#### III.3.1. Đại số quan hệ

Khi nói tới mô hình quan hệ người ta quan tâm tới 3 vấn đề chính:

- Cấu trúc để lưu trữ dữ liệu
- Các ràng buộc trên dữ liệu
- Các *phép toán để thao tác dữ liệu*

Như đã trình bày trong phần đầu của chương này về mô hình quan hệ, cấu trúc lưu trữ dữ liệu trong mô hình quan hệ chính là cấu trúc Bảng (Table). Các ràng buộc trên dữ liệu chúng ta cũng đã tìm hiểu như ràng buộc khóa chính, ràng buộc khóa ngoại và còn các loại ràng buộc khác sẽ được trình bày chi tiết trong chương 6.

*Tập hợp các phép toán để thao tác với dữ liệu* trong mô hình quan hệ được gọi là *đại số quan hệ*. Các phép toán này giúp cho người sử dụng xác định rõ các yêu cầu lấy tin cơ bản, kết quả của một phép lấy tin là một quan hệ mới, có thể được tạo ra từ một hoặc nhiều quan hệ. Một dãy các phép toán trên quan hệ tạo nên một biểu thức đại số quan hệ mà kết quả của nó cũng là một quan hệ.

Các phép toán đại số quan hệ được chia thành hai nhóm. Một nhóm bao gồm *các phép toán tập hợp* lấy từ lý thuyết tập hợp trong toán học, các phép toán đó là phép hợp, phép giao, phép trừ và phép tích Đề-các. Nhóm kia bao gồm các *phép toán quan hệ* được xây dựng đặc biệt cho cơ sở dữ liệu quan hệ, các phép toán đó là phép chọn, phép chiếu, phép nối và một số các phép toán khác.

#### III.3.2. Các phép toán tập hợp

Đó là các phép toán hợp, giao và trừ tập hợp và phép tích Descartes, đây là các phép toán hai ngôi, nghĩa là mỗi phép toán được áp dụng cho hai tập hợp. Khi áp dụng 3 phép toán hợp, giao và trừ cho cơ sở dữ liệu quan hệ, hai quan hệ tham gia vào một trong các phép toán trên phải có cùng một cấu trúc, điều kiện này được gọi là *khả hợp*. Hai quan hệ  $R(A_1, A_2, \dots, A_n)$  và  $S(B_1, B_2, \dots, B_m)$  được gọi là khả hợp nếu chúng có cùng số thuộc tính  $n = m$  và mỗi cặp thuộc tính tương ứng có cùng miền giá trị  $Dom(A_i) = Dom(B_i)$  với  $1 \leq i \leq n$ .

**III.3.2.1. Phép hợp**

Cho 2 quan hệ R và S khả hợp:  $R(A_1, A_2, \dots, A_n)$ ,  $S(B_1, B_2, \dots, B_n)$ . Phép hợp của 2 quan hệ R và S sẽ tạo thành quan hệ mới  $Q(A_1, A_2, \dots, A_n)$  chứa tất cả các bộ có trong R hay trong S. Phép hợp sẽ loại bỏ những bộ trùng lặp. Ký hiệu:  $R \cup S$ .

Có nghĩa:  $Q = R \cup S = \{q / q \in R \vee q \in S\}$

Ví dụ 3.7. Cho 2 quan hệ R và S khả hợp với các thể hiện như sau:

R	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

S	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	07	Ngô Thanh	Ngọc	5/7/1968	Nam	400
	08	Hồ Anh	Khoa	21/12/1983	Nam	250

Q=R∪S	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350
	07	Ngô Thanh	Ngọc	5/7/1968	Nam	400
	08	Hồ Anh	Khoa	21/12/1983	Nam	250

**III.3.2.2. Phép giao**

Cho 2 quan hệ R và S khả hợp:  $R(A_1, A_2, \dots, A_n)$ ,  $S(B_1, B_2, \dots, B_n)$ . Phép giao của 2 quan hệ R và S sẽ tạo thành quan hệ mới  $Q(A_1, A_2, \dots, A_n)$  chứa các bộ có trong R và có trong S. Ký hiệu:  $R \cap S$ .

Có nghĩa:  $Q = R \cap S = \{q / q \in R \wedge q \in S\}$

Ví dụ 3.8. Cho 2 quan hệ R và S khả hợp với các thể hiện như trong ví dụ 3.7

Q=R∩S	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500

**III.3.2.3. Phép trừ**

Cho 2 quan hệ R và S khả hợp:  $R(A_1, A_2, \dots, A_n)$ ,  $S(B_1, B_2, \dots, B_n)$ . Phép trừ của quan hệ R cho S sẽ tạo thành quan hệ mới  $Q(A_1, A_2, \dots, A_n)$  chứa các bộ có trong R nhưng không có trong S. Ký hiệu:  $R - S$ .

Có nghĩa:  $Q = R - S = \{q / q \in R \wedge q \notin S\}$

Ví dụ 3.9. Cho 2 quan hệ R và S khả hợp với các thể hiện như trong ví dụ 3.7

Q=R-S	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

### III.3.2.4. Phép tích Descartes

Cho 2 quan hệ R và S:  $R(A_1, A_2, \dots, A_n)$ ,  $S(B_1, B_2, \dots, B_m)$ . Phép tích Descartes của quan hệ R và S sẽ tạo thành quan hệ mới Q gồm  $n+m$  thuộc tính,  $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ . Các bộ giá trị trong Q được tạo thành do sự kết hợp một bộ của R và một bộ của S. Ký hiệu:  $R \times S$ .

Có nghĩa:  $Q = R \times S = \{(q_1, q_2) / q_1 \in R \wedge q_2 \in S\}$

Ví dụ 3.10. Cho 2 quan hệ R (NHANVIEN) và S (PHANCONG) với các thể hiện:

R	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

S	MANV	MADA	THOIGIAN
	01	D001	5
	01	D002	3
	02	D001	4

Q= R × S	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG	MANV	MADA	THOIGIAN
	01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D001	5
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430	01	D001	5
	03	Lê Thị	Lan		Nữ	200	01	D001	5
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350	01	D001	5
	01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D002	3
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430	01	D002	3
	03	Lê Thị	Lan		Nữ	200	01	D002	3
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350	01	D002	3
	01	Lê Tiến	Huy	1/1/1980	Nam	500	02	D001	4
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430	02	D001	4
	03	Lê Thị	Lan		Nữ	200	02	D001	4
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350	02	D001	4

## ❖ Lưu ý đối với các phép toán trên tập hợp:

- Phép hợp và phép giao là các phép toán có tính chất giao hoán và kết hợp:

$$R \cup S = S \cup R \text{ và } R \cap S = S \cap R$$

$$R \cup (S \cap T) = (R \cup S) \cap T \text{ và } R \cap (S \cup T) = (R \cap S) \cup T$$

- Phép trừ tập hợp không có tính chất giao hoán:

$$R - S \neq S - R$$

## III.3.3. Các phép toán quan hệ

## III.3.3.1. Phép chiếu

Phép chiếu là phép toán chọn một số thuộc tính của quan hệ. Nếu chúng ta chỉ quan tâm đến một số thuộc tính, dùng phép chiếu để loại bỏ những thuộc tính không cần thiết.

Cho quan hệ R: R(A<sub>1</sub>, A<sub>2</sub>, ..., A<sub>n</sub>). Gọi X là tập con của tập thuộc tính của quan hệ R. Phép chiếu quan hệ R lên tập X được ký hiệu là:  $\pi_X(R)$ . Kết quả của phép chiếu này là quan hệ R sau khi loại bỏ đi các thuộc tính không có trong tập thuộc tính X. Nếu X chỉ bao gồm các thuộc tính không phải thuộc tính khoá của R thì quan hệ kết quả có thể có những bộ trùng nhau, phép chiếu sẽ loại bỏ đi các bộ lặp lại.

Ví dụ 3.11. Cho quan hệ R với thể hiện:

R	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350
	07	Ngô Thanh	Ngọc	5/7/1968	Nam	430
	08	Hồ Anh	Khoa	21/12/1983	Nam	200

$$\pi_{\text{MaNV, TenNV}}(R) =$$

MANV	TENNV
01	Huy
02	Tuấn
03	Lan
04	Hiền
07	Ngọc
08	Khoa

$$\pi_{\text{Luong}}(\mathbf{R}) =$$

LUONG
500
430
200
350

### III.3.3.2. Phép chọn

Phép chọn được sử dụng để chọn các bộ thoả mãn điều kiện chọn từ một quan hệ. Ta có thể xem phép chọn như một bộ lọc, nó chỉ giữ lại các bộ thoả mãn điều kiện đặt ra.

Cho quan hệ  $R: R(A_1, A_2, \dots, A_n)$ . Điều kiện  $P$  là tổ hợp các biểu thức logic cơ sở chứa phép so sánh:  $>, \geq, <, \leq, =, \neq$  được chỉ ra trên một số thuộc tính nào đó của  $R$ . Phép chọn quan hệ  $R$  dựa trên điều kiện  $P$  được ký hiệu là:  $\sigma_P(\mathbf{R})$ . Kết quả của phép chọn này là quan hệ  $R$  sau khi loại bỏ đi các bộ không thoả điều kiện  $P$ .

Ví dụ 3.12. Cho 2 quan hệ  $R$  với thể hiện:

R	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350
	07	Ngô Thanh	Ngọc	5/7/1968	Nam	430
	08	Hồ Anh	Khoa	21/12/1983	Nam	200

$$\sigma_{\text{Luong} \geq 300}(\mathbf{R}) =$$

MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
01	Lê Tiến	Huy	1/1/1980	Nam	500
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350
07	Ngô Thanh	Ngọc	5/7/1968	Nam	430

$$\sigma_{\text{Luong} \geq 300 \wedge \text{Phai} = \text{'Nam'}}(\mathbf{R}) =$$

MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
01	Lê Tiến	Huy	1/1/1980	Nam	500
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
07	Ngô Thanh	Ngọc	5/7/1968	Nam	430

#### ❖ Lưu ý đối với phép chọn:

- Phép chọn là một phép toán có tính chất giao hoán, nghĩa là:

$$\sigma_{P_1}(\sigma_{P_2}(\mathbf{R})) = \sigma_{P_2}(\sigma_{P_1}(\mathbf{R}))$$



- Có thể kết hợp nhiều các phép chọn thành một phép chọn bằng cách sử dụng phép toán AND:

$$\sigma_{P_1}(\sigma_{P_2}(R)) = \sigma_{P_1 \text{ and } P_2}(R)$$

### III.3.3.3. Phép nối

#### a) Phép nối nội (Inner join):

Phép nối được dùng để kết hợp các bộ có liên quan với nhau từ hai quan hệ. Phép toán này rất quan trọng đối với cơ sở dữ liệu có nhiều quan hệ bởi vì nó cho phép ta xử lý các mối liên kết giữa các quan hệ.

Cho 2 quan hệ R và S:  $R(A_1, A_2, \dots, A_n)$ ,  $S(B_1, B_2, \dots, B_m)$ . Phép nối giữa quan hệ R với quan hệ S với điều kiện nối P sẽ tạo thành quan hệ mới Q gồm  $n+m$  thuộc tính,  $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ .

Ký hiệu:  $R \bowtie_P S$

Mỗi bộ của Q là một sự nối giữa một bộ của R và một bộ của S khi chúng thỏa mãn điều kiện nối P. Điều kiện nối P có dạng:  $A_{i1} \theta B_{j1} \text{ and } A_{i2} \theta B_{j2} \text{ and } \dots$ , trong đó  $A_i$  là các thuộc tính của R,  $B_j$  là các thuộc tính của S,  $A_i$  và  $B_j$  có cùng miền và  $\theta$  là một trong các phép toán so sánh  $\{>, \geq, <, \leq, =, \neq\}$ . Một phép toán nối với điều kiện tổng quát như vậy gọi là một phép nối tên-tên.

Sự khác nhau giữa tích Descartes và phép nối là ở chỗ trong phép nối, chỉ có các bộ thỏa mãn điều kiện nối P mới xuất hiện trong kết quả, trong khi đó trong Descartes mọi tổ hợp của các bộ đều có trong kết quả.

Ví dụ 3.13. Cho 2 quan hệ R và S với các thể hiện:

R	A	B	C
	1	4	9
	6	2	8
	7	5	3
	11	10	12

S	D	E
	8	9
	7	1

$A \geq D$   
 $R \bowtie S =$

A	B	C	D	E
11	10	12	8	9
7	5	3	7	1
11	10	12	7	1

Ví dụ 3.14. Cho 2 quan hệ R và S với các thể hiện:

R	A	B	C
	1	4	9
		2	8
	7	5	3
	8	10	12

S	D	E
	8	9
	7	1

$A=D$   
 $R \bowtie S =$

A	B	C	D	E
7	5	3	7	1
8	10	12	8	9

- Nếu phép nối  $\theta$  trong điều kiện nối P là phép so sánh  $=$  thì ta gọi *phép kết bằng* (equi-join). Nhận thấy rằng trong kết quả của phép nối bằng luôn có một hoặc nhiều cặp thuộc tính có các giá trị như nhau trong mỗi bộ (xem ví dụ 3.14, các giá trị tại thuộc tính A và D luôn giống nhau). Việc có các cặp thuộc tính có giá trị như nhau là thừa, vì vậy người ta đề nghị một phép nối mới gọi là nối tự nhiên, ký hiệu là  $*$ . *Phép nối tự nhiên* nhằm loại bỏ thuộc tính thừa (thuộc tính lặp lại) trong điều kiện nối bằng, yêu cầu của phép nối tự nhiên là hai thuộc tính nối (hoặc mỗi cặp thuộc tính nối) phải có tên như nhau trong cả hai quan hệ.

Ví dụ 3.15. Thực hiện phép kết tự nhiên giữa hai quan hệ R (NHANVIEN) và S (PHANCONG) :

R	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

S	MANV	MADA	THOIGIAN
	01	D001	5
	01	D002	3
	02	D001	4

Kết quả của phép kết tự nhiên giữa NHANVIEN và PHANCONG là:

$Q = R * S$	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG	MADA	THOIGIAN
	01	Lê Tiến	Huy	1/1/1980	Nam	500	D001	5
	01	Lê Tiến	Huy	1/1/1980	Nam	500	D002	3
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430	D001	4

**b) Phép nối ngoài**

Các phép toán nối ngoài mô tả ở trên dùng để chọn ra những bộ thỏa mãn điều kiện nối, các bộ không có bộ liên kết sẽ bị loại khỏi kết quả nối. Một tập hợp các phép toán gọi là nối ngoài có thể được sử dụng khi chúng ta muốn giữ các bộ trong R hoặc S hoặc trong cả hai quan hệ trong kết quả của phép nối dù chúng có những bộ liên kết trong quan hệ kia hay không. Có ba phép nối ngoài gọi là nối ngoài trái (left outer join), nối ngoài phải (right outer join) và nối ngoài đầy đủ (full outer join)

- **Phép nối ngoài trái (left outer join)**

Phép nối ngoài trái giữ lại mọi bộ trong quan hệ bên trái R trong phép nối. Nếu không có bộ liên kết nào được tìm thấy trong S thì các thuộc tính của S trong kết quả phép nối được “làm đầy” bằng các giá trị null.

Ký hiệu:  $R \overset{P}{\bowtie} S$

Ví dụ 3.16. Thực hiện phép nối ngoài trái giữa hai quan hệ R (NHANVIEN) và S (PHANCONG) :

R	MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350

S	MANV	MADA	THOIGIAN
	01	D001	5
	01	D002	3
	02	D001	4
	05	D003	5

$\overset{MANV}{R} \overset{P}{\bowtie} S =$

MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG	MANV	MADA	THOI GIAN
01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D001	5
01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D002	3
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430	02	D001	4
03	Lê Thị	Lan		Nữ	200			
04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350			

### ▪ Phép nối ngoài phải (right outer join)

Phép nối ngoài phải giữ lại mọi bộ trong quan hệ bên phải S trong phép nối. Nếu không có bộ liên kết nào được tìm thấy trong R thì các thuộc tính của R trong kết quả phép nối được “làm đầy” bằng các giá trị null.

Ký hiệu:  $R \overset{P}{\bowtie} S$

Ví dụ 3.17. Thực hiện phép nối ngoài phải giữa hai quan hệ R (NHANVIEN) và S (PHANCONG) trong ví dụ 3.16 :

MaNV

R  $\overset{P}{\bowtie}$  S =

MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG	MANV	MADA	THOI GIAN
01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D001	5
01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D002	3
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430	02	D001	4
						05	D003	5

### ▪ Phép nối ngoài đầy đủ (full outer join)

Phép nối ngoài đầy đủ giữ lại mọi bộ trong quan hệ bên trái R và phải S trong phép nối. Kết quả của phép nối ngoài đầy đủ giữa R và S chính là phép hợp giữa phép nối ngoài trái và phép nối ngoài phải của R và S.

Ký hiệu:  $R \overset{P}{\boxplus} S$

R  $\overset{P}{\boxplus}$  S =

MANV	HOLOT	TENNV	NGSINH	PHAI	LUONG	MANV	MADA	THOI GIAN
01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D001	5
01	Lê Tiến	Huy	1/1/1980	Nam	500	01	D002	3
02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430	02	D001	4
03	Lê Thị	Lan		Nữ	200			
04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350			
						05	D003	5

### III.3.3.4. Phép chia

Cho 2 quan hệ R và S:  $R(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ ,  $S(A_1, A_2, \dots, A_n)$ . Phép chia R cho S sẽ tạo thành quan hệ mới Q gồm m thuộc tính:  $Q(B_1, B_2, \dots, B_m)$  và các bộ trong Q là tập các bộ q sao cho với mọi bộ  $u \in S$  thì bộ  $q \cup u \in R$

Ký hiệu:  $R \div S$

Ví dụ 3.18. Cho 2 quan hệ R và S với các thể hiện:

R	A	B	C
	1	2	5
	2	5	5
	3	4	6
	3	4	5
	1	2	6

S	A	B
	1	2
	3	4

$R \div S =$	C
	5
	6

Ví dụ 3.19. Cho 2 quan hệ R (PHANCONG) và S (DUAN):

R	MANV	MADA	THOIGIAN
	01	D001	5
	01	D002	3
	02	D001	4
	03	D002	5
	03	D001	2
	04	D002	6

S	MADA	TENDA
	D001	Dự án 1
	D002	Dự án 2

Phép chia PHANCONG(MANV, MADA) cho DUAN(MADA) sẽ cho ta danh sách các nhân viên tham gia tất cả các dự án:

$\pi_{\text{MANV, MADA}}(\text{PHANCONG}) \div \pi_{\text{MANV}}(\text{MADA}) =$	MANV
	01
	03

#### ❖ Lưu ý đối với các phép toán đại số quan hệ

Ta có thể áp dụng nhiều phép toán quan hệ liên tiếp nhau. Trong trường hợp đó hoặc xếp lồng các phép toán lại với nhau, hoặc có thể áp dụng mỗi phép toán tại một thời điểm và tạo ra các quan hệ kết quả trung gian. Trong trường hợp tạo các quan hệ trung gian, ta phải đặt tên cho quan hệ đó.

Ví dụ 3.20. Với quan hệ NHANVIEN, chọn ra họ tên của những nhân viên làm việc với mức lương trên 400.

NHANVIEN	MANV	HOLOT	TENN	NGSINH	PHAI	LUONG
	01	Lê Tiến	Huy	1/1/1980	Nam	500
	02	Huỳnh Anh	Tuấn	8/11/1967	Nam	430
	03	Lê Thị	Lan		Nữ	200
	04	Nguyễn Thị	Hiền	5/12/1977	Nữ	350
	07	Ngô Thanh	Ngọc	5/7/1968	Nam	430
	08	Hồ Anh	Khoa	21/12/1983	Nam	200

Ta có thể viết:

$$\pi_{\text{HoLot, TenNV}} (\sigma_{\text{Luong} > 400} (\text{NHANVIEN}))$$

Hoặc viết cách khác, sử dụng kết quả trung gian:

$$\text{TG} \leftarrow \sigma_{\text{Luong} > 400} (\text{NHANVIEN})$$

$$\text{KQ} \leftarrow \pi_{\text{HoLot, TenNV}} (\text{TG})$$

### III.3.4. Một số ví dụ về phép toán đại số quan hệ

Trong phần này, chúng ta xét một số ví dụ minh họa việc sử dụng các phép toán đại số quan hệ. Các ví dụ ở đây thực hiện trên cơ sở dữ liệu *quản lý dự án*. Nói chung, một truy vấn có thể được thực hiện bằng nhiều cách và sử dụng nhiều phép toán khác nhau.

NHANVIEN(MANV, HOLOT, TENNV, NGSINH, PHAI, LUONG, DCHI, MA\_NQL, PHG)

PHONGBAN(MAPHG, TENPHG, TRPHG, NG\_NHANCHUC)

DIADIEM\_PHONG(MAPHG, DIADIEM)

DUAN(MADA, TENDA, DDIEM\_DA, PHONG)

PHANCONG(MA\_NVIEN, SODA, THOIGIAN)

THANNHAN(MA\_NVIEN, TENTN, NGAYSINH, PHAI, QUANHE)

a. Cho biết tên và lương của tất cả các nhân viên làm việc cho phòng “Nghiên cứu”

- Cách 1:

$$\pi_{\text{TENNV, LUONG}} (\sigma_{\text{TENPHG} = \text{“Nghiên cứu”}} (\text{NHANVIEN} \bowtie_{\text{PHG} = \text{MAPHG}} \text{PHONGBAN}))$$

- Cách 2:

$$\text{TG}_1 \leftarrow \sigma_{\text{TENPHG} = \text{“Nghiên cứu”}} (\text{PHONGBAN})$$

$$\text{TG}_2 \leftarrow \text{TG}_1 \bowtie_{\text{PHG} = \text{MAPHG}} \text{NHANVIEN}$$

$$\text{KQ} \leftarrow \pi_{\text{HOLOT, TENNV, LUONG}} (\text{TG}_2)$$

b. Cho biết những dự án chưa có nhân viên nào tham gia, hiển thị: mã dự án, tên dự án và phòng chủ trì dự án

$$TG_1 \leftarrow \pi_{MADA} (DUAN)$$

$$TG_2 \leftarrow \pi_{SODA} (PHANCONG)$$

$$TG_3 \leftarrow TG_1 - TG_2$$

$$TG_4 \leftarrow TG_3 \bowtie_{MADA} DUAN$$

$$KQ \leftarrow \pi_{MADA, TENDA, PHONG} (TG_4)$$

c. Cho biết những nhân viên nào tham gia tất cả các dự án do phòng có mã P005 chủ trì, hiển thị: MANV, TENNV

$$TG_1 \leftarrow \pi_{MADA} (\sigma_{PHONG="P005"} (DUAN))$$

$$TG_2 \leftarrow \pi_{MANV, SODA \rightarrow MADA} (PHANCONG)$$

$$TG_3 \leftarrow TG_2 \div TG_1$$

$$TG_4 \leftarrow TG_3 * NHANVIEN$$

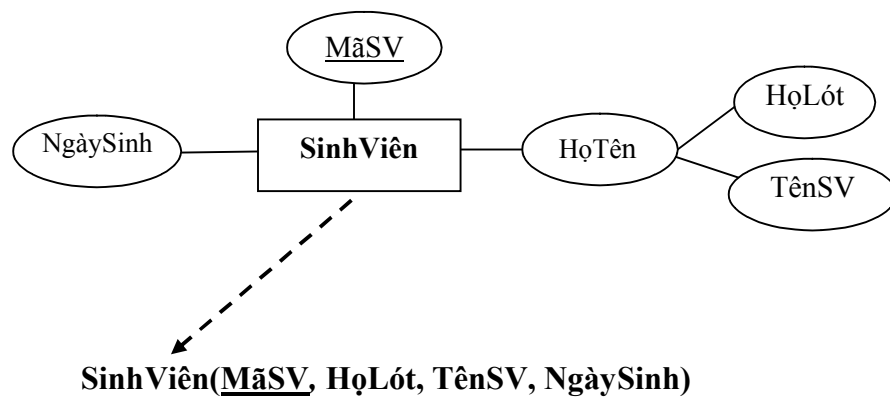
$$KQ \leftarrow \pi_{MANV, TENNV} (TG_4)$$

Đặt lại tên cho trường  
SODA thành MADA

### III.4. Chuyển mô hình thực thể-liên kết sang mô hình quan hệ

#### III.4.1. Tập thực thể

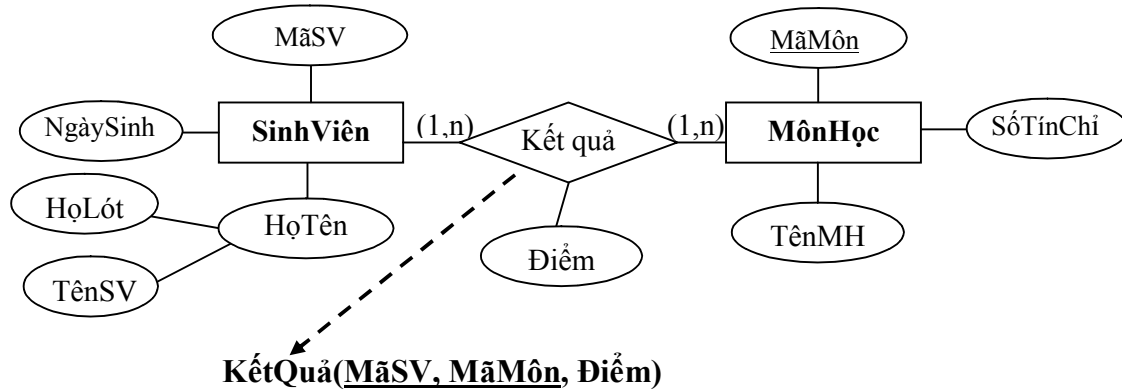
- Các tập thực thể (trừ tập thực thể yếu) chuyển thành các quan hệ có cùng tên và cùng thuộc tính.



Hình 3.1. Chuyển tập thực thể từ mô hình ER sang mô hình quan hệ

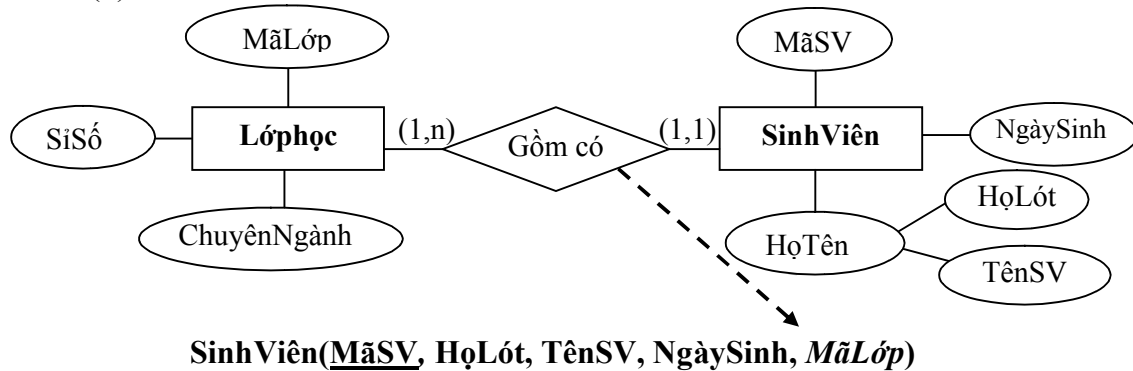
### III.4.2. Mỗi liên kết

- *Nhiều-Nhiều*: Tạo một quan hệ mới có tên là tên của mỗi liên kết và nhận thuộc tính khóa của hai quan hệ liên quan làm khóa.



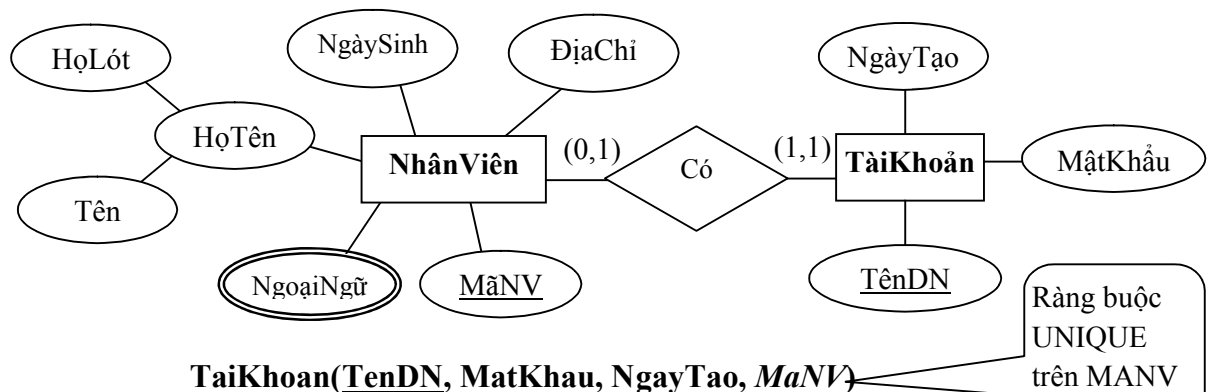
Hình 3.2. Chuyển mỗi liên kết n-n từ mô hình ER sang mô hình quan hệ

- *Một-Nhiều*: Thêm vào quan hệ bên một (1) thuộc tính khóa của quan hệ bên nhiều (n)



Hình 3.3. Chuyển mỗi liên kết 1-n từ mô hình ER sang mô hình quan hệ

- *Một-Một*: Thêm vào quan hệ này thuộc tính khóa của quan hệ kia hoặc thêm thuộc tính khóa vào cả 2 quan hệ và có ràng buộc duy nhất (UNIQUE) trên thuộc tính mới thêm vào này.

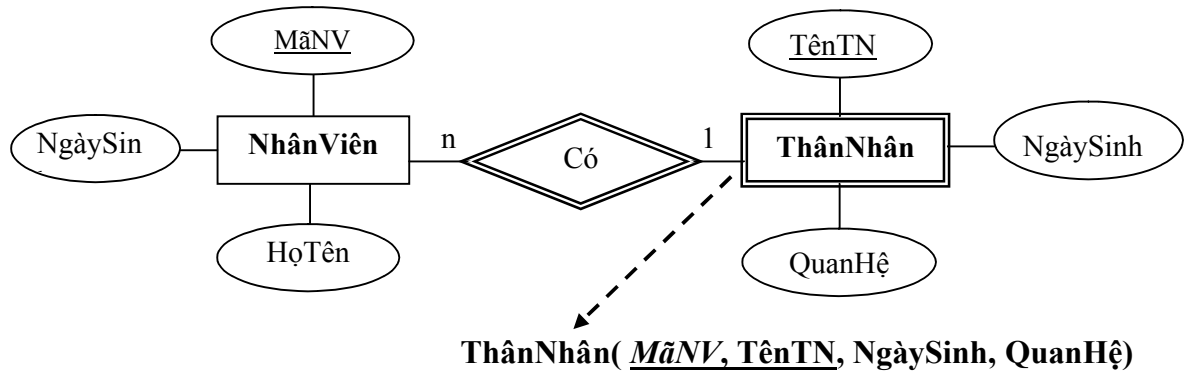


Hình 3.4. Chuyển mỗi liên kết 1-1 từ mô hình ER sang mô hình quan hệ



### III.4.3. Tập thực thể yếu

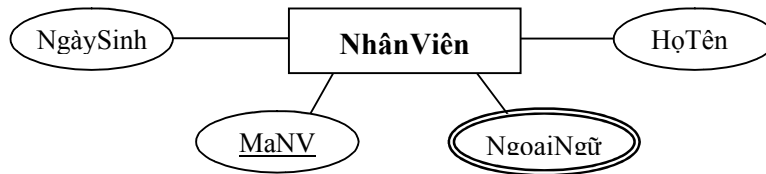
Chuyển thực thể yếu thành một quan hệ và thêm vào quan hệ này thuộc tính khóa của quan hệ liên quan.



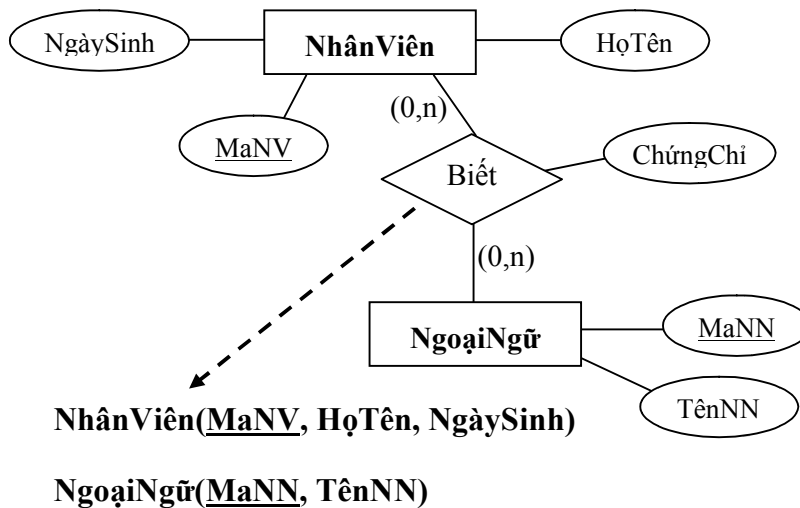
Hình 3.5. Chuyển tập thực thể yếu từ mô hình ER sang mô hình quan hệ

### III.4.4. Thuộc tính đa trị

Trước hết mô hình hóa thuộc tính đa trị bằng cách chuyển thuộc tính đa trị thành tập thực thể riêng và có quan hệ với tập thực thể ban đầu. Sau đó chuyển mô hình ER sau khi được mô hình hóa sang mô hình quan hệ.



Mô hình hóa thuộc tính đa trị trong hình trên như sau:

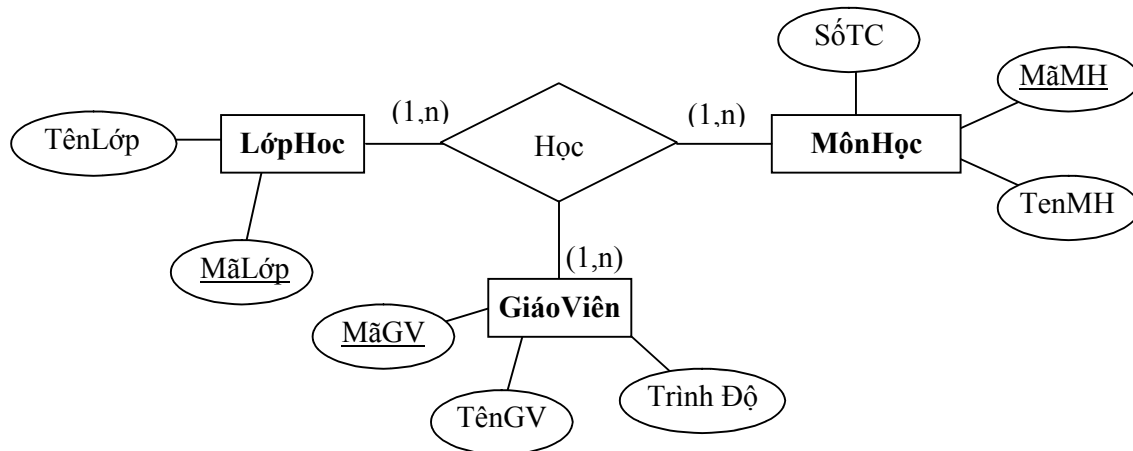


**TrìnhĐộNN(MãNV, MaNN, ChứngChỉ)**

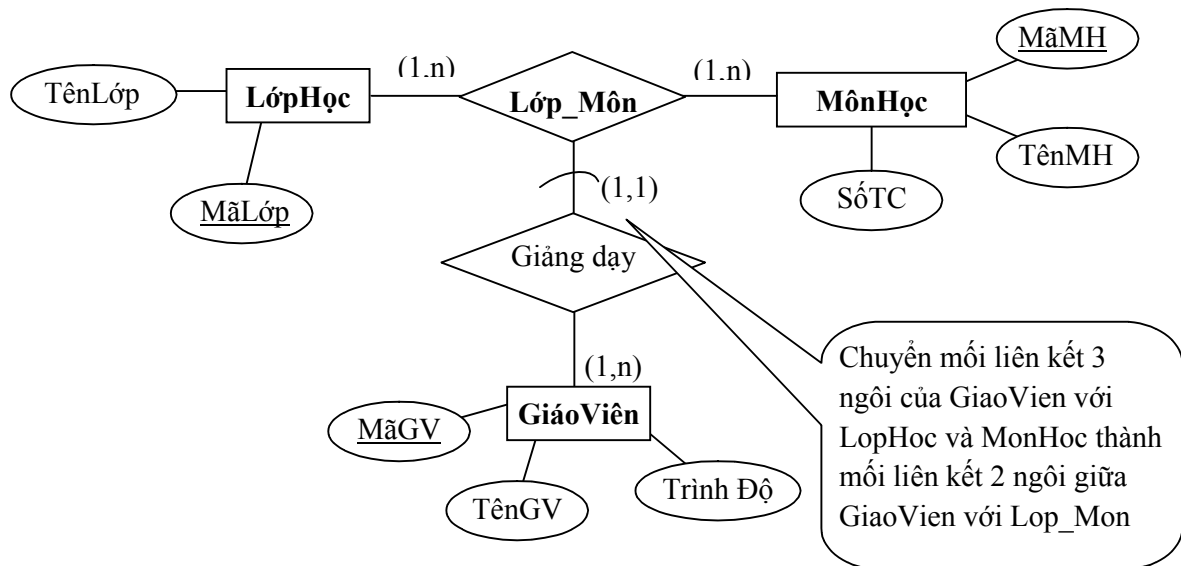
Hình 3.6. Chuyển thuộc tính đa trị từ mô hình ER sang mô hình quan hệ

### III.4.5. Môi liên kết đa ngôi

Mô hình hóa môi liên kết đa ngôi thành môi liên kết 2 ngôi trước khi chuyển sang mô hình quan hệ.



Mô hình hóa môi liên kết đa ngôi trong hình trên như sau:



**LớpHoc**(MaLớp, TênLớp)

**MônHoc**(MaMH, TênMH, SốTC)

**Lớp\_Môn**(MaLớp, MaMH, MaGV)

**GiáoViên**(MaGV, TênGV, TrìnhĐộ)

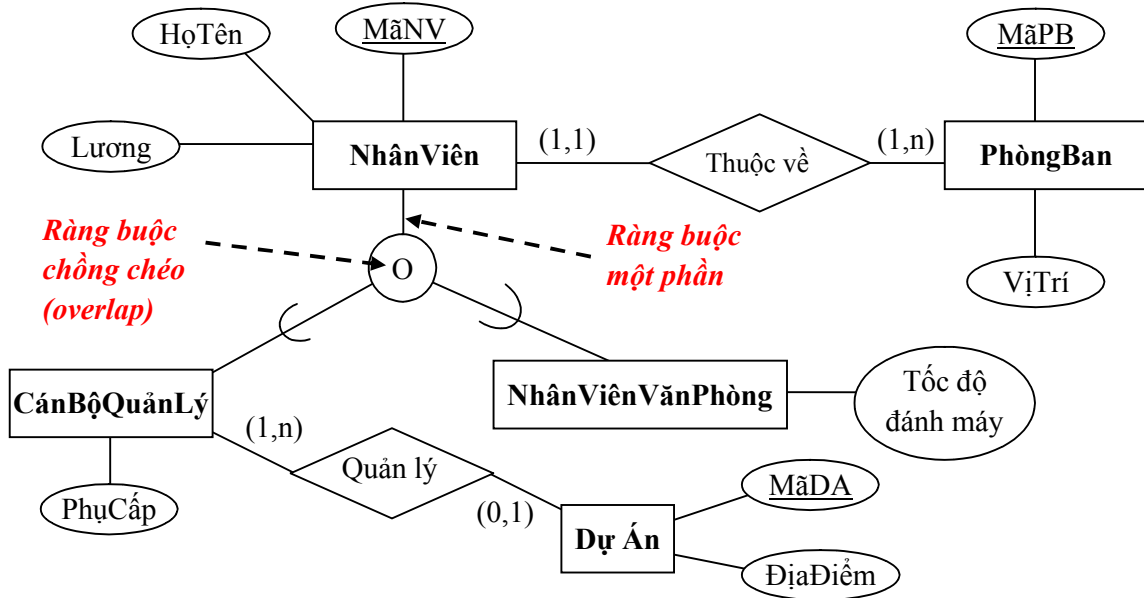
Hình 3.7. Chuyển môi liên kết đa ngôi từ mô hình ER sang mô hình quan hệ

### III.4.6. Mô hình hóa trường hợp mở rộng

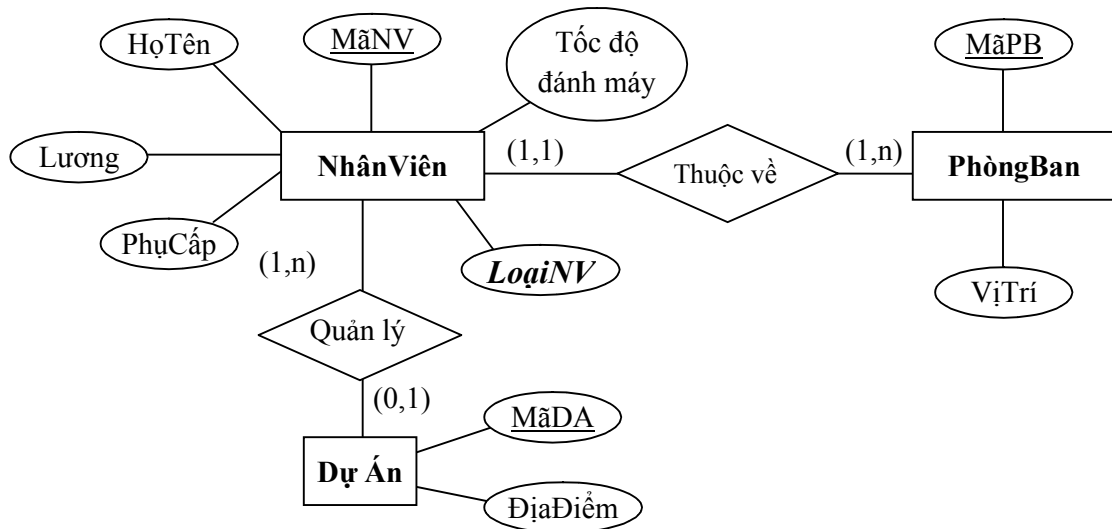
Trước khi chuyển sang mô hình quan hệ, ta loại bỏ tổng quát hóa/chuyên biệt hóa bằng cách tách hẳn thành các lớp con hay gộp chung vào lớp cha tùy từng trường hợp cụ thể. Chẳng hạn, đối với trường hợp thỏa mãn ràng buộc chồng chéo thì ta nên gộp

chung vào lớp cha; đối với trường hợp thỏa mãn ràng buộc rời rạc và nếu số lượng các thuộc tính tại các lớp con nhiều thì ta nên tách riêng các lớp con.

- Gộp các lớp con vào lớp cha:



Ta mô hình hóa trường hợp này bằng cách gộp:



**Nhân Viên**(MãNV, Họ Tên, Lương, Phụ Cấp, TD\_DanhMay, *LoạiNV*, MãPB)

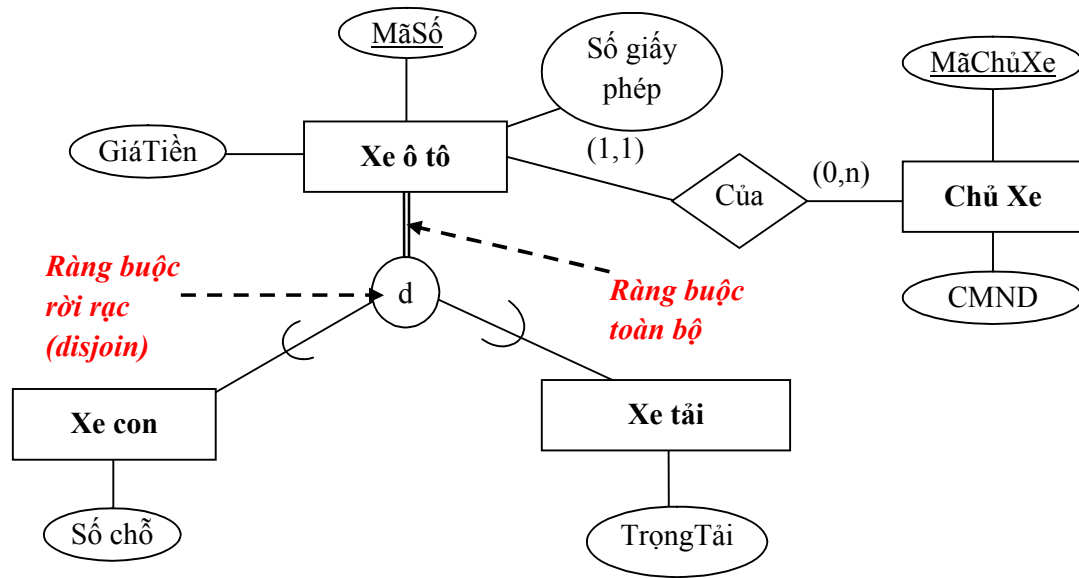
**Phòng Ban**(MãPB, Vị Trí)

**Dự Án**(MãDA, Địa Điểm, MãNV)

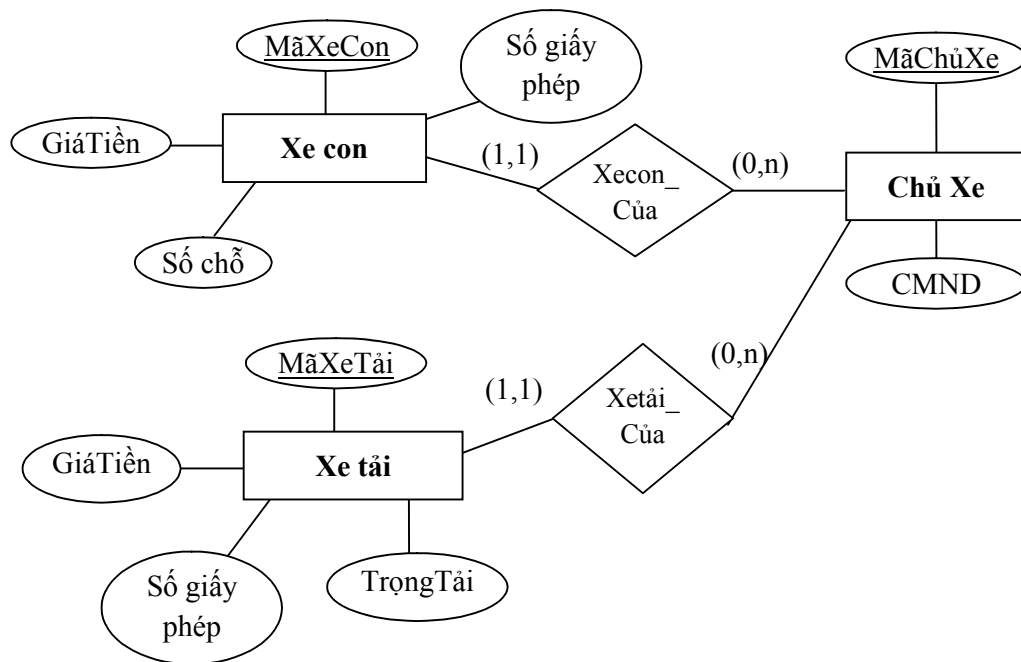
LoạiNV nhận 1 trong 3 giá trị: VanPhong, QuanLy và VP\_QL

Hình 3.8. Loại bỏ tổng quát hóa/chuyên biệt hóa bằng cách gộp vào lớp cha

- Tách các lớp con:



Ta có thể mô hình hóa trường hợp này bằng cách tách:



**ChủXe(MãChủXe, CMND)**

**XeCon(MãXeCon, SốGiấyPhép, GiáTiền, SốChỗ, MãChủXe)**

**XeTải (MãXeTải, SốGiấyPhép, GiáTiền, TrọngTải, MãChủXe)**

Hình 3.9. Loại bỏ tổng quát hóa/chuyên biệt hóa bằng cách tách riêng lớp con

### III.5. Tổng kết chương và câu hỏi ôn tập

#### III.5.1. Tổng kết chương

Chương này đã trình bày các định nghĩa trong mô hình quan hệ như lược đồ quan hệ, thể hiện quan hệ, thuộc tính, khóa, bộ giá trị và các phép toán được sử dụng trong mô hình quan hệ, các phép toán này được gọi chung là đại số quan hệ.

Bên cạnh đó, chương này cũng đã trình bày cách chuyển từ mô hình ER sang mô hình quan hệ: một tập thực thể ứng với một quan hệ, thuộc tính của tập thực thể tương ứng với thuộc tính của quan hệ, một mối liên kết được chuyển thành một quan hệ hay một thuộc tính tùy thuộc vào loại của mối liên kết, các trường hợp mở rộng trong mô hình ER cũng được mô hình hóa trước khi chuyển sang mô hình quan hệ.

#### III.5.2. Câu hỏi ôn tập

1. Mô hình quan hệ là gì?
2. Ưu điểm của mô hình quan hệ?
3. Định nghĩa quan hệ trong mô hình quan hệ?
4. Ý nghĩa của khóa chính? Yêu cầu của khóa chính?
5. Ý nghĩa của khóa ngoại? Yêu cầu của khóa ngoại?
6. Có mấy phép toán trên tập hợp? Đó là những phép toán nào?
7. Muốn thực hiện phép hợp, phép giao và phép trừ trên 2 quan hệ R và S thì 2 quan hệ này phải thỏa điều kiện gì?
8. Mục đích của phép chiếu? phép chọn? phép kết?
9. Có bao nhiêu loại phép kết? Đó là những phép kết nào? Hãy phân biệt chúng với nhau?
10. Cách chuyển từ mô hình ER sang mô hình quan hệ?

#### III.5.3. Bài tập

1. Chuyển các mô hình ER trong từ kết quả các bài tập của chương 2 sang mô hình quan hệ, bao gồm: quản lý khách sạn, quản lý chuyến bay, quản lý sinh viên, quản lý thuê băng đĩa, quản lý bán hàng
2. Với CSDL quản lý dự án, hãy trả lời các câu hỏi sau bằng các phép toán đại số quan hệ

NHANVIEN(MANV, HOLOT, TENNV, NGSINH, PHAI, LUONG, DCHI, MA\_NQL, PHG)  
PHONGBAN(MAPHG, TENPHG, TRPHG, NG\_NHANCHUC)  
DIADIEM\_PHONG(MAPHG, DIADIEM)  
DUAN(MADA, TENDA, DDIEM\_DA, PHONG)  
PHANCONG(MA\_NVIEN, SODA, THOIGIAN)  
THANNHAN(MA\_NVIEN, TENTN, NGAYSINH, PHAI, QUANHE)

**Câu hỏi:**

- a. Cho biết họ tên trưởng phòng của phòng ‘Quản lý’.
- b. Cho biết họ tên những nhân viên tham gia dự án có địa điểm ở ‘Hà Nội’
- c. Cho biết họ tên những nhân viên không có thân nhân.
- d. Cho biết họ tên những nhân viên chưa tham gia vào dự án nào.
- e. Cho biết họ tên những trưởng phòng chưa tham gia vào dự án nào
- f. Cho biết danh sách các nhân viên ‘nam’ thuộc phòng ‘Điều hành’.
- g. Cho biết các dự án (MADA, TENDA) có nhân viên họ ‘Nguyen’ tham gia trên 10 ngày.
- h. Danh sách các dự án (MADA) mà phòng ‘Nghiên cứu’ chủ trì
- i. Cho biết phòng ban (tên phòng, tên trưởng phòng) có địa điểm ‘Đà Nẵng’
- j. Danh sách những dự án có:
  - Người tham gia có họ “Đinh” hoặc
  - Người trưởng phòng chủ trì dự án có họ “Đinh”
- k. Cho biết nhân viên tham gia tất cả các dự án
- l. Cho phép nhân viên làm việc cho tất cả các dự án mà phòngng số 5 chủ trì
- m. Cho biết những nhân viên tham gia tất cả dự án mà nhân viên ‘33344555’ tham gia.

## Chương IV. TỔ CHỨC DỮ LIỆU VẬT LÝ

### A. Mục tiêu

#### ➤ Về kiến thức:

Sau khi xong chương này sinh viên nắm vững các kiến thức sau:

- Cách tổ chức lưu trữ CSDL trong máy
- Các phương pháp tổ chức lưu trữ dữ liệu, ưu điểm và hạn chế của từng phương pháp

#### ➤ Về kỹ năng

Giúp sinh viên phát triển các kỹ năng:

- Mạnh dạn phát biểu ý kiến.
- Rèn luyện khả năng đọc sách, tự nghiên cứu học hỏi.

### B. Phương pháp giảng dạy

- Trình chiếu slides
- Đặt câu hỏi cho sinh viên trả lời
- Sinh viên thảo luận theo nhóm

### C. Tài liệu tham khảo

1. Nguyễn An Tế, Đồng Thị Bích Thủy (1996). *Nhập Môn Cơ Sở Dữ Liệu*.
2. Lê Tiến Vương (1994). *Nhập Môn Cơ Sở Dữ Liệu Quan Hệ*
3. Jeffrey D. Ullman (1998). *Mô Hình Dữ Liệu Và Ngôn Ngữ Ván Tin*. Tập 2

### D. Nội dung chi tiết

#### IV.1. Mô hình tổ chức bộ nhớ ngoài

Cơ sở dữ liệu vật lý là một tập các *mẫu tin* (record), mỗi mẫu tin chứa một hoặc nhiều trường. Giá trị của các trường có thể là kiểu cơ bản như số nguyên, số thực, và chuỗi ký tự có chiều dài cố định và thay đổi, ở đây chúng ta cũng xem con trỏ như một kiểu dữ liệu cơ bản, đó là một tham chiếu đến một mẫu tin.

Giống như các mô hình dữ liệu cấp cao, các mẫu tin thường được xem là những thể hiện (instance) của một lược đồ, các mẫu tin này có cùng số lượng các trường, cùng

tên, cùng kiểu dữ liệu và cùng ý nghĩa. Chúng ta gọi danh sách các tên trường và kiểu dữ liệu là *khuôn dạng* (format) cho mẫu tin. *Tập dữ liệu* (file) là biểu thị một tập các mẫu tin có cùng khuôn dạng, tập dữ liệu là một cách biểu diễn vật lý thích hợp cho một quan hệ, một tập dữ liệu ứng với một quan hệ.

Thiết bị lưu trữ vật lý dùng để ghi các mẫu tin và các tập dữ liệu có thể xem như một mảng các byte được đánh số tuần tự. Thiết bị lưu trữ này có thể là có thể là vùng nhớ có các byte được đánh số từ 0 đến một con số rất lớn (có thể đến  $2^{30}$ ), vùng nhớ này có thể lưu trên bộ nhớ thứ cấp hay *bộ nhớ ngoài*, chẳng hạn như đĩa từ.

Khi số lượng dữ liệu trong CSDL khá lớn, ta không thể xem thiết bị trữ tin như bộ nhớ chính của hệ thống, vì ta phải xét đến việc dữ liệu cần thao tác với hệ thống phải được chuyển từ bộ nhớ ngoài vào bộ nhớ chính.

Đĩa từ được phân chia thành các *khối* vật lý (block) có kích cỡ giống nhau, mỗi khối chiếm khoảng 512 bytes đến 4096 bytes và được đánh địa chỉ khối, địa chỉ này được gọi là địa chỉ tuyệt đối trên đĩa. Mỗi tập dữ liệu lưu trên đĩa từ chiếm một hoặc nhiều khối, mỗi khối chứa một hoặc nhiều bản ghi. Việc thao tác với tập dữ liệu sẽ thông qua tên tập dữ liệu thực chất là thông qua địa chỉ tuyệt đối của các khối.

Các mẫu tin đều có địa chỉ và thường được xem là địa chỉ tuyệt đối của byte đầu tiên của mẫu tin hoặc là địa chỉ của khối chứa mẫu tin đó. Địa chỉ của các mẫu tin hoặc các khối thường được lưu ở một vị trí nào đó để khi cần, qua đó có thể truy cập tới dữ liệu cần thiết. Chỉ dẫn đó được gọi là *con trỏ* (pointer).

Các phép tính đặc trưng trên các tập dữ liệu là: thêm một mẫu tin, xóa một mẫu tin, sửa một mẫu tin, tìm một mẫu tin theo giá trị xác định nào đó. Lưu ý rằng trong các HQTCSDL việc xóa một mẫu tin là rất thận trọng, thông thường việc xóa mẫu tin được tiến hành bằng cách đánh dấu vào mẫu tin đó và con trỏ tới mẫu tin này không được dùng nữa, như vậy mẫu tin chỉ được xóa về mặt logic mà thôi. Nếu thực sự muốn xóa mẫu tin đó, chúng ta phải sử dụng thêm một thao tác xóa nữa.

## IV.2. Tổ chức Tập băm và các thao tác trên tổ chức tập băm

### IV.2.1. Tổ chức tập băm (Hashed File)



Trước hết ta tìm hiểu vài nét về hàm băm (H-function). Mỗi bản ghi đều có một khóa là giá trị số (ví dụ là giá trị  $v$ ), thì hàm băm  $h$  nhận giá trị  $v$  làm đối số tạo ra một số nguyên trong khoảng từ  $[0, k-1]$  (có nghĩa  $h(v) \in [0, k-1]$ ).

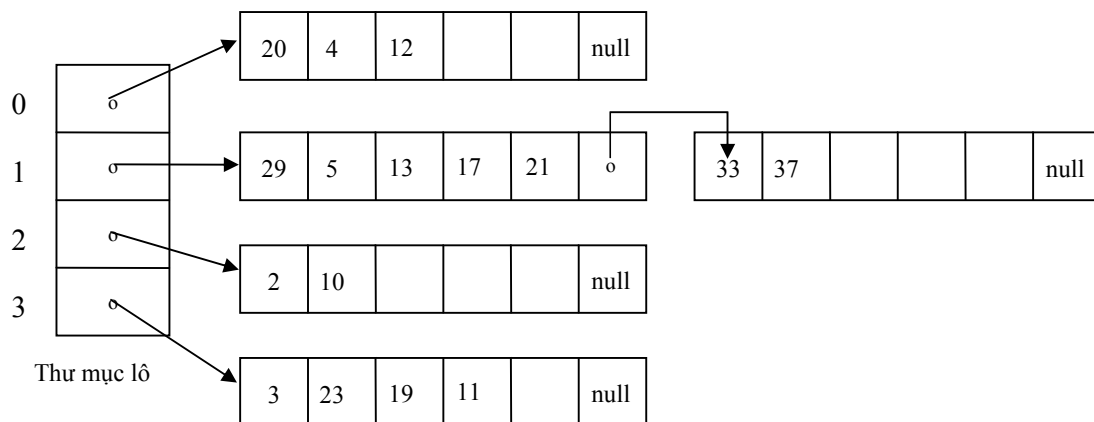
Có nhiều loại hàm được dùng làm hàm băm  $h$ , điều quan trọng là nó trả về giá trị trong khoảng từ 0 đến  $k-1$  và ta mong muốn rằng nó băm nhỏ được các khóa, có nghĩa là khi  $v$  chạy trên các giá trị khóa khả hữu,  $h(v)$  sẽ trả về các giá trị khả hữu với một xác suất như nhau.

Một loại hàm băm đơn giản cho phép biến đổi mỗi giá trị khóa thành một số nguyên rồi lấy phần dư khi chia số nguyên đó cho  $k$ . Nếu khóa là các số nguyên ta chỉ cần tính đơn giản  $h(v) = v \bmod k$ , nếu khóa là chuỗi kí tự ta có thể biến đổi chuỗi thành số nguyên bằng cách chia nhỏ chuỗi thành từng nhóm kí tự, xử lý các bit cho nhóm kí tự như một số nguyên rồi cộng các số nguyên này lại.

Ý tưởng của tổ chức tệp băm là chia tập các mẫu tin trong tệp dữ liệu thành các *lô* (bucket), mỗi lô gồm một hoặc nhiều khối, mỗi khối chứa một số lượng cố định các mẫu tin. Mỗi lô ứng với một giá trị băm, có nghĩa là có tất cả  $k$  lô. Ở đầu mỗi khối đều chứa con trỏ tới khối tiếp theo trong lô, khối cuối cùng chứa con trỏ rỗng.

Có một *thực mục lô* chứa  $k$  con trỏ, mỗi con trỏ ứng với mỗi cụm, đó là địa chỉ khối đầu tiên trong cụm.

*Ví dụ 4.1.* Trong hình 4.1 ta có một tệp dữ liệu được tổ chức như một tệp băm với 4 lô, nghĩa là  $k=4$ , mỗi lô có nhiều khối và mỗi khối chứa tối đa 5 mẫu tin. Ở đây ta sử dụng hàm băm  $h(v) = v \bmod 4$



Hình 4.1. Tổ chức tập tin băm

## IV.2.2. Các thao tác trên tổ chức tệp băm

### IV.2.2.1. Tìm kiếm mẫu tin

Để tìm kiếm mẫu tin có khóa  $x$ , chúng ta tính giá trị băm  $h(x)$ . Giả sử  $h(x)=i$  thì  $i$  sẽ là địa chỉ băm của lô  $i$ . Trong thư mục lô cho biết con trỏ tới khối đầu tiên, tìm trong khối này xem có mẫu tin có khóa  $x$  hay không, theo con trỏ ở đầu khối tìm tiếp tới các khối tiếp theo cho tới khi được mẫu tin mong muốn hoặc tới khối cuối cùng của lô  $i$  mà không có mẫu tin đó. Nếu không thấy mẫu tin cần tìm không cần phải kiểm tra các lô khác, bởi vì các mẫu tin có hàm băm  $i$  không thể nằm trong lô khác được.

### IV.2.2.2. Thêm mới mẫu tin

Giả sử cần thêm một mẫu tin có khóa là  $x$  vào tệp dữ liệu, thủ tục được thực hiện giống như tìm kiếm một mẫu tin. Nếu trong tệp dữ liệu đã có một mẫu tin trùng khóa  $x$  thì không thể chèn được, ngược lại mẫu tin sẽ được thêm vào khối cuối cùng trong lô nếu còn chỗ trống, nếu khối cuối cùng không còn chỗ trống ta phải tạo một khối mới và nối nó vào danh sách cuối cùng của lô  $h(x)$ , trong trường hợp này mẫu tin sẽ là mẫu tin đầu tiên của khối mới và khối mới sẽ là khối cuối cùng của lô  $h(x)$ .

### IV.2.2.3. Xóa mẫu tin

Để xóa một mẫu tin có khóa  $x$ , ta phải tìm mẫu tin cần xóa, nếu mẫu tin thuộc khối nào đó có nhiều mẫu tin thì loại bỏ mẫu tin có khóa  $x$  khỏi khối, nếu mẫu tin  $x$  là duy nhất trong khối thì việc xóa mẫu tin  $x$  đồng thời với việc giải phóng khối khỏi lô  $h(x)$ .

### IV.2.2.4. Sửa mẫu tin

Giả sử cần sửa mẫu tin có khóa  $x$ , nếu thông tin cần sửa liên quan đến khóa  $x$  thì việc sửa sẽ là loại bỏ mẫu tin này và thêm vào mẫu tin mới (vì khi sửa khóa, mẫu tin có thể thuộc vào một lô khác). Nếu trường cần sửa không thuộc khóa, tiến hành sửa sau khi tìm thấy mẫu tin trong lô  $h(x)$ .

## IV.3. Tổ chức đồng và thao tác trên tổ chức đồng

### IV.3.1. Tổ chức đồng (Heap)

Đây là một tổ chức tệp dữ liệu “tầm thường” nhất, trong đó các mẫu tin được “ném” vào trong khối không theo một thứ tự nào cả và các khối cũng không cần tổ

chức đặc biệt nào. Chúng ta chỉ giả sử đã sẵn có các con trỏ chỉ đến các khối và chúng được lưu trong bộ nhớ chính. Nếu cần quá nhiều khối để lưu các con trỏ này trong bộ nhớ chính thì bản thân chúng sẽ được lưu trong có khối ở bộ nhớ phụ và được truy xuất khi cần thiết.

Lưu ý rằng thời gian để thao tác với cá mẫu tin (thêm, xóa, sửa, tìm kiếm) được tính bằng *số các khối cần truy xuất được di chuyển giữa bộ nhớ chính và phụ*. Để cho thống nhất, ta giả sử rằng khởi đầu, toàn bộ tệp dữ liệu đều ở trong bộ nhớ phụ, và giả sử chúng ta có  $n$  mẫu tin và  $R$  là số lượng các mẫu tin có thể xếp vào trong 1 khối, do đó số lượng khối tối thiểu cần để lưu một tệp dữ liệu là  $n/R$ .

#### **IV.3.2. Thao tác trên tổ chức đồng**

##### **IV.3.2.1. Tìm kiếm mẫu tin**

Duyệt lần lượt các khối để tìm các mẫu tin có khóa  $x$ . Như vậy để tìm thấy mẫu tin trong tệp heap khi biết khóa của nó trung bình chúng ta phải truy xuất  $n/2R$  khối, nếu không tồn tại mẫu tin có khóa đó chúng ta phải truy xuất tất cả  $n/R$  khối.

##### **IV.3.2.2. Thêm mẫu tin mới**

Để chèn mẫu tin mới, chúng ta chỉ cần truy xuất khối cuối cùng của heap, là khối còn chỗ trống bên trong. Nếu khối cuối cùng không còn chỗ trống ta phải tạo khối mới. Ở mỗi trường hợp ta đều phải viết khối này vào bộ nhớ phụ sau khi chèn mẫu tin. Vậy thao tác chèn cần truy xuất 2 khối, một lần đọc khối và một lần ghi khối.

##### **IV.3.2.3. Xóa mẫu tin**

Thao tác xóa đòi hỏi phải tìm ta mẫu tin, nghĩa là thực hiện tìm kiếm rồi ghi lại khối chứa mẫu tin vào bộ nhớ phụ. Như vậy trung bình truy xuất  $n/2R+1$  khối nếu tìm thấy mẫu tin và  $n/R$  khối nếu không tìm thấy mẫu tin.

##### **IV.3.2.4. Sửa mẫu tin**

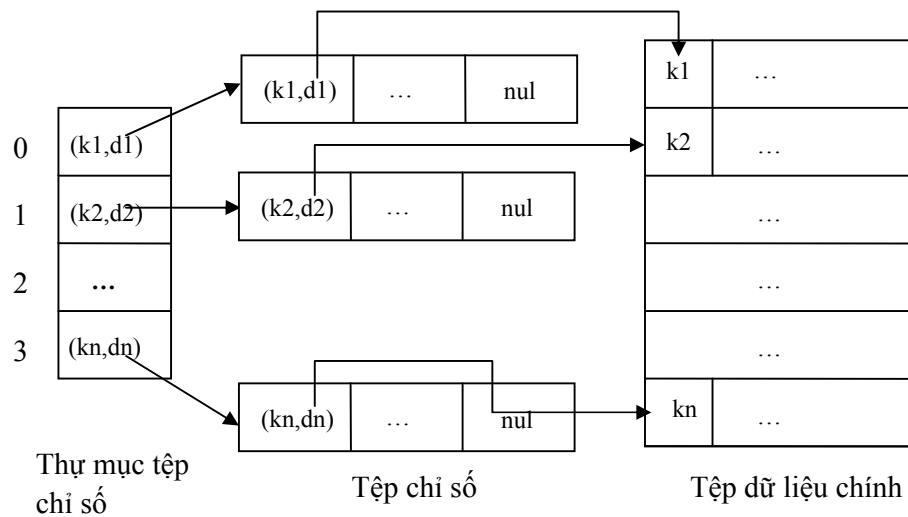
Thao tác chỉnh sửa cũng cần thời gian tương tự thao tác xóa. Chúng ta phải thực hiện  $n/2R+1$  lần truy xuất khối nếu thực hiện thành công và  $n/R$  khối nếu không tìm thấy mẫu tin cần sửa.

#### **IV.4. Tổ chức tệp chỉ dẫn và thao tác trên tổ chức tệp chỉ dẫn**

#### IV.4.1. Tổ chức tệp chỉ dẫn (Indexed Files)

Đây là một kiểu tổ chức tệp dữ liệu rất thường dùng trong CSDL. Để dễ dàng trình bày, giả sử rằng tệp dữ liệu luôn luôn được sắp xếp theo khóa (ví dụ tăng dần). Khóa luôn được quan niệm rằng, bao gồm một hoặc nhiều trường có thứ tự và có chiều dài cố định. Giá trị của khóa có thể là số, có thể là chuỗi kí tự. Nếu là chuỗi kí tự, việc sắp xếp được thực hiện theo thứ tự tự điển.

Để hỗ trợ cho tệp dữ liệu chính, cần tạo một *tệp chỉ số* theo khóa được chọn. Tệp chỉ số bao gồm các cặp  $(k, d)$  trong đó  $k$  là giá trị của khóa,  $d$  là địa chỉ của khối (hay con trỏ khối). Các cặp này được sắp xếp theo giá trị của khóa.



Hình 4.2. Tổ chức tệp chỉ dẫn

#### IV.4.2. Thao tác trên tổ chức tệp chỉ dẫn

##### IV.4.2.1. Tìm kiếm mẫu tin

Giả sử cần tìm mẫu tin có khóa  $x$ , có thể sử dụng nhiều cách duyệt trong tệp chỉ số.

**Tìm tuần tự:** Duyệt các tệp chỉ số  $(k, d)$  trong thư mục tệp chỉ số, so sánh giá trị  $x$  với  $k$ , gặp cặp đầu tiên  $(k', d')$  có  $k' > x$  thì dừng. Như vậy giá trị  $x$  thuộc khối đứng trước khối  $(k', d')$ , ví dụ là khối  $(k_l, d_l)$ . Trong khối  $(k_l, d_l)$  sẽ tìm tuần tự theo tệp chỉ số cho tới khi gặp giá trị  $k$  nào đó bằng  $x$  thì giá trị  $d$  tương ứng là địa chỉ của bộ cần tìm. Nếu không có giá trị  $k$  nào trong khối  $(k_l, d_l)$  bằng  $x$  thì xem như mẫu tin không tồn tại.

**Tìm nhị phân:** Phương pháp tìm kiếm tuần tự nói chung là chậm, thường việc tìm kiếm sử dụng phương pháp tìm nhị phân vì tệp chỉ số luôn được sắp xếp. Giả sử tệp chỉ số được lưu trên  $n$  khối  $(b_1, b_2, \dots, b_n)$ . Để tìm mẫu tin có khóa  $x$ , trước hết chọn

khối  $b_{[n/2]}$ , so sánh giá trị  $k$  thuộc khối này với  $x$ , nếu  $k > x$  thì bộ cần tìm nằm trong khối từ  $b_1, \dots, b_{[n/2]}$ , ngược lại trong các khối  $b_{[n/2+1]}, \dots, b_n$ . Quá trình lặp lại cho tới khi chỉ còn một khối chứa mẫu tin  $x$ , trong khối này tiếp tục tìm tuần tự trong các cặp  $(k, d)$  như phương pháp tìm tuần tự.

#### IV.4.2.2. Thêm mẫu tin mới

Muốn thêm mới mẫu tin có khóa  $x$ , sử dụng thủ tục tìm mẫu tin trên để xác định khối chứa mẫu tin mới này. Nếu trong khối còn chỗ thì thêm mẫu tin này vào khối đó đúng theo thứ tự sắp xếp của khóa. Việc đặt đúng vị trí này đòi hỏi phải chuyển chỗ các mẫu tin khác đứng sau mẫu tin có khóa  $x$ , nếu mẫu tin có khóa  $x$  là mẫu tin đầu tiên của khối thì phải sửa lại khóa  $k$  thành  $x$  trong thư mục tệp chỉ dẫn.

#### IV.4.2.3. Xóa mẫu tin

Quá trình thực hiện giống như thêm một mẫu tin. Khi xóa một mẫu tin có thể tạo nên một khối rỗng, khi đó có thể loại bỏ cả khối đó.

#### IV.4.2.4. Sửa mẫu tin

Sử dụng thủ tục tìm một mẫu tin để xác định mẫu tin cần sửa. Nếu các trường cần sửa không phải là khóa thì việc sửa đổi giá trị của mẫu tin thực hiện bình thường, giá trị của mẫu tin sau khi sửa vẫn được ghi lại ở vị trí cũ. Nếu các trường cần sửa là các trường khóa, quá trình sửa sẽ là quá trình xóa và thêm một mẫu tin.

### IV.5. Tổng kết chương và câu hỏi ôn tập

#### IV.5.1. Tổng kết chương

Chương này đã trình bày các phương pháp tổ chức lưu trữ CSDL mức vật lý thường được sử dụng trong các HQTCSDDL: tổ chức tệp băm, tổ chức đồng và tổ chức tệp chỉ dẫn. Mỗi phương pháp có một cách tổ chức lưu trữ khác nhau và ưu khuyết điểm khác nhau, điều đó được thể hiện qua các thao tác đặc trưng trên tệp dữ liệu: thêm, xóa và sửa và tìm kiếm một mẫu tin.

#### IV.5.2. Câu hỏi ôn tập

1. Dữ liệu được tổ chức trong tệp băm như thế nào? Các thủ tục thêm, xóa, sửa và tìm kiếm một mẫu tin được tiến hành như thế nào trên tệp băm?

2. Dữ liệu được tổ chức trong đồng như thế nào? Các thủ tục thêm, xóa, sửa và tìm kiếm một mẫu tin được tiến hành như thế nào trên đồng?
3. Dữ liệu được tổ chức trong tệp chỉ dẫn như thế nào? Các thủ tục thêm, xóa, sửa và tìm kiếm một mẫu tin được tiến hành như thế nào trên tệp chỉ dẫn?
4. Theo bạn, trong 3 phương pháp trên phương pháp nào hiệu quả nhất? Vì sao?

## Chương V. NGÔN NGỮ TRUY VẤN DỮ LIỆU – SQL

### A. Mục tiêu

#### ➤ Về kiến thức:

Sau khi xong chương này sinh viên nắm vững các kiến thức sau:

- Nắm vững cú pháp của các câu lệnh SQL và vận dụng nó để tạo mới CSDL, thêm, xóa, sửa và truy vấn dữ liệu từ CSDL.
- Nắm vững các câu SQL tương ứng với các phép toán đại số quan hệ.
- Có thể phân tích để tối ưu hóa câu truy vấn SQL.

#### ➤ Về kỹ năng

Giúp sinh viên phát triển các kỹ năng:

- Rèn luyện khả năng làm việc theo nhóm
- Chăm chỉ, chủ động, tích cực trong học tập
- Rèn luyện khả năng đọc sách, tự nghiên cứu học hỏi.

### B. Phương pháp giảng dạy

- Trình chiếu slides
- Làm nhiều bài tập và thảo luận tại lớp.
- Cho sinh viên làm một số bài kiểm tra nhanh
- Yêu cầu sinh viên làm bài tập về nhà
- Cho một số câu hỏi khó để kích thích suy nghĩ và suy luận của sinh viên

### C. Tài liệu tham khảo

1. Nguyễn An Tế, Đồng Thị Bích Thủy (1996). *Nhập Môn Cơ Sở Dữ Liệu*.
2. Nguyễn Đăng Ty, Đỗ Phúc (2003). *Giáo Trình Cơ Sở Dữ Liệu*.
3. Lê Tiến Vương (1994). *Nhập Môn Cơ Sở Dữ Liệu Quan Hệ*.
4. Jeffrey D. Ullman (1998). *Mô Hình Dữ Liệu Và Ngôn Ngữ Vấn Tin*. Tập 1
5. Hoàng Trung Sơn (2004). *Thiết Kế Cơ Sở Dữ Liệu*.
6. Phạm Thế Quế (2004). *Giáo Trình Cơ Sở Dữ Liệu*.

### D. Nội dung chi tiết

## V.1. Ngôn ngữ truy vấn dữ liệu - SQL

Như đã trình bày trong chương 1, một HQTCSDDL phải có ngôn ngữ giao tiếp giữa người sử dụng với CSDL. Ngôn ngữ giao tiếp gồm:

### V.1.1. Ngôn ngữ định nghĩa dữ liệu (DDL – Data Definition Language)

Cho phép khai báo cấu trúc các bảng của CSDL, khai báo các mối liên hệ của dữ liệu và các qui tắc, ràng buộc áp đặt lên dữ liệu đó.

### V.1.2. Ngôn ngữ thao tác dữ liệu (DML - Data Manipulation Language)

Cho phép người sử dụng có thể bổ sung dữ liệu như thêm (insert), xóa (delete), sửa (update) dữ liệu trong CSDL và truy xuất (select) các thông tin cần thiết từ CSDL.

### V.1.3. Ngôn ngữ kiểm soát dữ liệu (DCL - Data Control Language)

Cho phép người quản trị hệ thống khai báo và thay đổi việc bảo mật thông tin, cấp quyền khai thác dữ liệu cho người dùng.

Các ngôn ngữ trên được gọi chung là SQL. SQL là viết tắt của Structured Query Language - ngôn ngữ truy vấn có cấu trúc, cho phép người dùng sử dụng để tổ chức, quản lý và truy xuất dữ liệu từ CSDL. SQL là hệ thống các câu lệnh sử dụng tương tác với CSDL quan hệ, tên gọi của SQL làm ta liên tưởng đến một ngôn ngữ chỉ dùng để truy vấn dữ liệu, thực sự mà nói khả năng của SQL vượt xa so với công cụ truy xuất dữ liệu, mặc dù đó là mục đích ban đầu để xây dựng nên ngôn ngữ SQL.

Trước đây SQL được gọi là SEQUEL (Structured English Query Language), là ngôn ngữ được IBM phát triển để sử dụng trong hệ thống CSDL thử nghiệm là SystemR. Cho tới nay, có thể nói rằng SQL là ngôn ngữ chuẩn trong CSDL, SQL được sử dụng trong rất nhiều các HQTCSDDL thương mại hàng đầu thế giới như BD2, Access, SQL Server, Oracle, ...

SQL được thừa nhận là tiêu chuẩn của [ANSI](#) (American National Standards Institute) vào năm 1986 và [ISO](#) (International Organization for Standardization) năm 1987. Cho tới nay, ngôn ngữ SQL đã trải qua một số phiên bản sau:

- SQL-86: Được công bố đầu tiên bởi [ANSI](#) và được phê chuẩn bởi [ISO](#) năm [1987](#)
- SQL-89: có những thay đổi nhỏ
- SQL-92 (còn gọi là SQL2): đã có nhiều thay đổi lớn



- SQL-99 (còn gọi là SQL3)
- SQL-2003

Trong giáo trình này ta sử dụng phiên bản SQL-92 và sử dụng HQTCSDDL là SQL Server để cài đặt.

## V.2. Biểu thức và hàm trong SQL

### V.2.1. Biểu thức

Biểu thức (expression) là một dãy các toán hạng (operand) nối với nhau bởi các phép toán (operator). Các toán hạng có thể là: hằng, biến, thuộc tính, hàm. Các phép toán có thể là:

- Các phép toán số học: +, -, \*, /, ...
- Các phép toán so sánh: <, <=, >, >=, =, <>
- Các phép toán logic: not, and, or.
- Các phép toán phạm vi: in, like, between...and,...

### V.2.2. Hàm

Phần này sẽ trình bày một số các hàm thông dụng làm việc với các kiểu dữ liệu chuỗi, ngày tháng, số và giá trị NULL trong SQL Server 2005

#### V.2.2.1. Hàm xử lý chuỗi

Các hàm quan trọng bao gồm: LEFT, RIGHT, LEN, REPLACE, STUFF, SUBSTRING, LOWER, UPPER, LTRIM, and RTRIM

1. *LEFT*: trả về một chuỗi ký tự có chiều dài được chỉ định tính từ bên trái của chuỗi

*Ví dụ 5.1.* `select LEFT(N'Đà Nẵng', 2)`

Kết quả:

	(No column name)
1	Đà

2. *RIGHT*: tương tự hàm LEFT, trả về một chuỗi ký tự có chiều dài được chỉ định tính từ bên phải của chuỗi.
3. *SUBSTRING*: trả về một chuỗi con từ một chuỗi cho trước

*SUBSTRING* (chuỗi\_ban\_đầu, vị\_trí\_bắt\_đầu, chiều\_dài\_chuỗi\_con)

*Ví dụ 5.2.* `select SUBSTRING(N'Thành Phố Đà Nẵng', 7, 3)`

Kết quả:

	(No column name)
1	Phố

4. *LEN*: trả về chiều

dài một chuỗi

Ví dụ 5.3. `select LEFT('Đà Nẵng')`

Kết quả:

	(No column name)
1	7

5. *REPLACE*: thay thế một chuỗi bởi một chuỗi khác*REPLACE* (chuỗi\_ban\_đầu, chuỗi\_bị\_thay\_thế, chuỗi\_thay\_thế)Ví dụ 5.4. `select replace(N'Đà Nẵng', N'Năng', N'Thành')`

Kết quả:

	(No column name)
1	Đà Thành

6. *LOWER/UPPER*: chuyển chuỗi cho trước thành chuỗi thường/ chuỗi hoa7. *LTRIM/RTRIM*: cắt khoảng trắng bên trái/ bên phải của chuỗi**V.2.2.2. Hàm thời gian**1. *GETDATE*: trả về ngày giờ hiện tạiVí dụ 5.5. `select GETDATE()`

Kết quả:

	(No column name)
1	2011-12-14 14:41:55.173

2. *DAY/MONTH/YEAR*: trả về ngày/tháng/năm của một ngày cho trướcVí dụ 5.6. `select DAY(GETDATE()), MONTH(GETDATE()), YEAR(GETDATE())`

Kết quả:

	(No column name)	(No column name)	(No column name)
1	14	12	2011

3. *DATEPART*: trả về số nguyên chỉ định thời gian đại diện một ngày cho trước theo một đơn vị thời gian bất kỳ.

Ví dụ 5.7. Cho biết ngày hiện tại thuộc quý bao nhiêu trong năm nay, lưu ý ký tự “qq” đại diện đơn vị thời gian “quý”

`select DATEPART(qq, GETDATE())`

Kết

	(No column name)
1	4

quả:

Các ký tự đại diện

đơn vị thời gian gồm:

STT	Ký tự đại diện	Đơn vị thời gian
1	qq, q	Quý
2	ww, wk	Tuần
3	dy	Số ngày đã qua trong năm
4	dw	Số ngày đã qua trong năm
5	yyyy, yy	Năm
6	mm, m	Tháng
7	dd, d	Ngày
8	hh	Giờ
9	mi, n	Phút
10	ss, s	Giây

4. **DATENAME**: trả về tên đại diện ngày cho trước theo một đơn vị thời gian bất kỳ

*Ví dụ 5.8.* Cho biết ngày hiện tại thuộc tháng bao nhiêu trong năm nay

```
select DATENAME(mm, GETDATE())
```

Kết quả:

	(No column name)
1	December

5. **DATEDIFF**: trả về một số nguyên tính khoảng cách của hai ngày cho trước theo một đơn vị thời gian bất kỳ.

### V.2.2.3. Các hàm khác

1. **ROUND**: làm tròn số theo độ dài xác định

*Ví dụ 5.9.* `select ROUND(123.456, 1), ROUND(123.456, -1)`

Kết quả:

	(No column name)	(No column name)
1	123.500	120.000

2. **ISNUMERIC**: kiểm tra một giá trị có phải thuộc kiểu dữ liệu số hay không, trả về 1 nếu là số, ngược lại trả về 0.

3. **CAST**: chuyển đổi một giá trị (hay biểu thức) thuộc kiểu dữ liệu này sang một kiểu dữ liệu khác.

*CAST(Biểu\_thức/ Giá trị AS kiểu\_dữ\_liệu)*

4. **CONVERT**: chuyển đổi một giá trị (hay biểu thức) sang một kiểu dữ liệu bất kỳ nhưng có thể theo một định dạng nào đó (đặc biệt đối với kiểu dữ liệu ngày)

*CONVERT (Kiểu\_dữ\_liệu, Biểu\_thức/ giá trị[, định\_dạng])*

*Ví dụ 5.10.* `select CONVERT (varchar(11), GETDATE(), 106)`

Kết

	(No column name)
1	14 Dec 2011

quả:

STT	Định dạng ngày tháng	Hiển thị dữ liệu
1	101	mm/dd/yy
2	102	yy.mm.dd
3	103	dd/mm/yy
4	104	dd.mm.yy
5	105	dd-mm-yy
6	106	dd mon yy
7	107	mon dd, yy
8	108	hh:mm:ss
9	109	mon dd yyyy hh:mm:ss
10	110	mm-dd-yy
11	111	yy/mm/dd
12	112	yymmdd
13	113	dd mon yyyy hh:mm:ss
14	114	hh:mm:ss:mmm
15	21 hoặc 121	yyyy-mm-dd hh:mi:ss:mmm
16	20 hoặc 120	yyyy-mm-dd hh:mi:ss

### V.3. Câu lệnh SQL

Sau đây là một số qui ước để trình bày cú pháp câu lệnh SQL:

- **Từ khóa:** Là từ dành riêng có ý nghĩa nhất định trong ngôn ngữ, bắt buộc phải có trong câu lệnh. Không được dùng từ khóa cho mục đích khác, như đặt tên cho trường hay bảng

```
SELECT * FROM SinhVien
```

Trong đó, `SELECT` và `FROM` là các từ khóa

- **Thành phần bắt buộc:** Là biểu thức bắt buộc phải có trong câu lệnh. Thành phần bắt buộc được đặt trong dấu ngoặc nhọn `<...>`

```
CREATE TABLE <Tên_bảng>
```

- **Thành phần không bắt buộc:** Là biểu thức không bắt buộc phải có trong câu lệnh. Thành phần không bắt buộc được đặt trong dấu ngoặc vuông `[...]`

```
SELECT *
FROM <Tên_bảng>
[WHERE <Điều_kiện>]
```

- **Thành phần lựa chọn:** Việc chọn một trong các khả năng thể hiện bởi dấu `|`
- **Giá trị mặc định:** Được thể hiện bởi dấu gạch chân

```
SELECT *
FROM <Tên_bảng>
[WHERE <Điều kiện>]
[ORDER BY <tên cột> | <biểu thức> [ASC|DESC]]
```

❖ **Lưu ý:** Câu lệnh SQL có thể được viết trên nhiều dòng và không phân biệt hoa-thường. Thông thường tên CSDL, tên bảng, tên thuộc tính đều không chứa khoảng trắng, tuy nhiên SQL vẫn cho phép khoảng trắng với điều kiện khi sử dụng phải đi kèm với cặp dấu ngoặc vuông [].

Chẳng hạn, ta có câu lệnh select sau:

```
SELECT [Ho Ten], [Ngày Sinh]
FROM [Sinh Viên]
```

### V.3.1. Câu lệnh định nghĩa dữ liệu

#### V.3.1.1. Tạo mới một CSDL

##### 1. Lệnh tạo CSDL

- **Cú pháp:** `CREATE DATABASE <Tên_CSDL>`  
*Ví dụ 5.11.* `create database QLSinhVien`

##### 2. Lệnh sử dụng CSDL

- **Cú pháp:** `USE <Tên_CSDL>`  
*Ví dụ 5.12.* `use QLSinhVien`

##### 3. Lệnh xóa CSDL

- **Cú pháp:** `DROP DATABASE <Tên database>`  
*Ví dụ 5.13.* `drop database QLSinhVien`

#### V.3.1.2. Lệnh tạo mới một bảng

##### 1. Lệnh tạo mới cấu trúc bảng

- **Cú pháp:**

```
CREATE TABLE <Tên_bảng>
(
    <tên_cột_1> <kiểu dữ liệu> [not null|null] [ràng buộc 1]
    [, <tên_cột_2> <kiểu dữ liệu> [not null|null] [ràng buộc 2]]
    .....
    [, <tên_cột_n> <kiểu dữ liệu> [not null|null] [ràng buộc n]
    [, ràng buộc bảng]
)
```

▪ **Trong đó:**

- *Tên\_bảng*: Tên bảng (quan hệ) phải là duy nhất trong mỗi CSDL
- *Tên\_cột*: Tên cột (thuộc tính, trường) trong bảng phải duy nhất, mỗi bảng phải có ít nhất một cột
- *Kiểu dữ liệu*: Kiểu dữ liệu của cột

Kiểu dữ liệu	Bytes	Diễn giải
bit	1	0 hay 1 (0 false, 1 true)
tinyint	1	0 – 225
smallint	2	$-2^{15}$ (-32,768) $\rightarrow$ $2^{15} - 1$ (32,767)
int	4	$-2^{31}$ (-2,147,483,648) $\rightarrow$ $2^{31} - 1$ (2,147,483,647)
decimal		$-10^{38} + 1 \rightarrow 10^{38} - 1$
numeric		Tương tự decimal
money	4	-214,748.3648 $\rightarrow$ +214,748.3647
float		-1.79E + 308 $\rightarrow$ -2.23E - 308
datetime	8	January 1, 1753 $\rightarrow$ December 31, 9999, chính xác đến phần trăm giây
smalldatetime	4	January 1, 1900 $\rightarrow$ June 6, 2079, chính xác đến phút
char		Non-Unicode với chiều dài cố định, cho phép 8000 ký tự, mỗi ký tự 1 byte.
varchar		Non-Unicode với chiều dài biến đổi, cho phép 8000 ký tự, mỗi ký tự 1 byte.
nchar		Unicode với chiều dài cố định, cho phép 4000 ký tự, mỗi ký tự 2 byte.
nvarchar		Unicode với chiều dài biến đổi, cho phép 4000 ký tự, mỗi ký tự 2 byte.

- *Not null*: Không cho phép cột có nhận giá trị null, mặc định là null (null là giá trị đặc biệt, nghĩa là không xác định được giá trị, null không phải là giá trị khoảng trắng)
- *Ràng buộc*: Các ràng buộc trên bảng dữ liệu, chẳng hạn như các ràng buộc về khóa, các mặc định, các quy định về dạng dữ liệu...

Ví dụ 5.14. Tạo bảng SINHVIEN với các thuộc tính sau:

SINHVIEN(MASV,HOTEN,NGAYSINH,GIOITINH,DIACHI,CMND,DIENTHOAI,MAKHOA)

```
create table SINHVIEN
(
    MASV          char(10)          not null,
    HOTEN         nvarchar(60)      not null,
    NGAYSINH      datetime,
    GIOITINH      nchar(6),
    DIACHI        nvarchar(100),
```

```

CMND      char(10) ,
DIENTHOAI char(10) ,
MAKHOA    char(6)
)

```

▪ **Các ràng buộc trên bảng:**

- *Ràng buộc kiểm tra (Check)*: dùng để qui định các giá trị hay khuôn dạng có thể chấp nhận đối với một cột, trên một cột có thể có nhiều ràng buộc Check.

```
CONSTRAINT <Tên_ràng_buộc> CHECK (<biểu thức ràng buộc>)
```

- *Ràng buộc mặc định (Default)*: Là ràng buộc dùng để qui định giá trị mặc định cho một cột, giá trị này sẽ được gán cho cột nếu người dùng thêm một mẫu tin mới mà không chỉ định giá trị cho cột này.

```
CONSTRAINT <Tên_ràng_buộc> DEFAULT <giá trị mặc định>
```

- *Ràng buộc duy nhất (Unique)*: Sử dụng để bắt buộc giá trị trên cột không được trùng nhau.

```
CONSTRAINT <tên ràng buộc> UNIQUE (<Danh sách các cột>)
```

- *Ràng buộc khóa chính (Primary Key)*: Ràng buộc này dùng để định nghĩa khóa chính của bảng

```
CONSTRAINT <Tên_ràng_buộc> PRIMARY KEY(<Danh sách các cột>)
```

- *Ràng buộc khóa ngoại (Foreign Key)*: Ràng buộc này dùng để định nghĩa khóa ngoại của bảng

```

CONSTRAINT <Tên_ràng_buộc> FOREIGN KEY(<Danh sách các cột >)
REFERENCES <Bảng được tham chiếu>(<Danh sách các cột>)
[ON DELETE CASCADE | NO ACTION]
[ON UPDATE CASCADE | NO ACTION]

```

Trong đó, CASCADE là cách thức xử lý các mẫu tin trong *bảng tham chiếu* khi xóa hoặc sửa các mẫu tin trong *bảng được tham chiếu*, có nghĩa tự động xóa (ON DELETE) hay sửa (ON UPDATE) các mẫu tin tham chiếu nếu các mẫu tin được tham chiếu bị xóa hay sửa. Trường hợp ngược lại là NO ACTION (là trường hợp mặc định), có nghĩa không thể xóa (hay sửa các cột được tham chiếu) những mẫu tin trong bảng được tham chiếu.

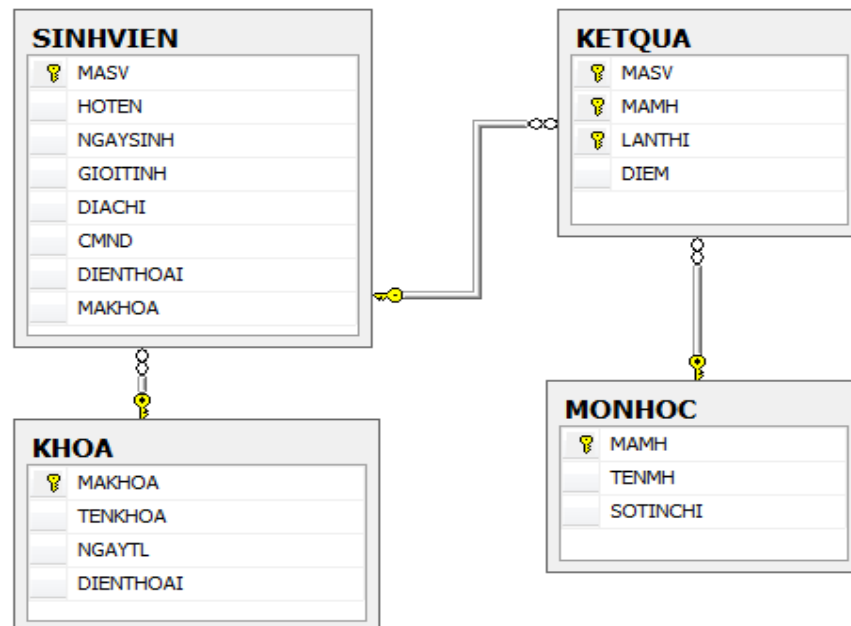
*Ví dụ 5.15.* Với bảng SINHVIEN, yêu cầu GIOITINH chỉ nhận một trong 2 giá trị ‘Nam’ hoặc ‘Nữ’ (Check) và mặc định là ‘Nam’ (Default), CMND phải là duy nhất (Unique), MASV là khóa chính và MAKHOA là khóa ngoại tham chiếu tới MAKHOA trong bảng KHOA. Khi sửa dữ liệu trong cột MAKHOA của bảng KHOA

thì dữ liệu trong cột MAKHOA của bảng SINHVIEN cũng thay đổi theo, không thể xóa bộ giá trị trong KHOA nếu bộ đó đã được tham chiếu trong bảng SINHVIEN.

```
create table SINHVIEN
(
    MASV      char(10)      not null,
    HOTEN     nvarchar(60) not null,
    NGAYSINH  datetime,
    GIOITINH  nchar(6) check(GIOITINH in ('Nam','Nữ')) default 'Nam',
    DIACHI    nvarchar(100),
    CMND      char(10) unique,
    DIENTHOAI char(10),
    MAKHOA    char(6),
    constraint PK_SV Primary Key(MASV),
    constraint FK_SV_KHOA Foreign Key(MAKHOA) references KHOA(MAKHOA)
    on update cascade
)
```

❖ **Lưu ý :** Đối với những ràng buộc không được đặt tên tường minh, HQTCSDDL sẽ tự động đặt tên cho ràng buộc này. Trong ví dụ 5.15, chỉ có khóa chính và khóa ngoại được đặt tên tường minh, còn các ràng buộc check, default, unique sẽ được hệ quản trị đặt tên. Dĩ nhiên, chúng ta hoàn toàn có thể đặt tên cho các ràng buộc check, default, unique nếu ta khai báo các ràng buộc này theo đúng cú pháp đã được định nghĩa.

Với CSDL quản lý sinh viên, sau khi tạo xong các bảng ta có sơ đồ diagram bên dưới, ta sẽ sử dụng CSDL này trong tất cả các ví dụ của chương 5.



Hình 5.1. Sơ đồ diagram



Dữ liệu của từng bảng cụ thể như sau:

MONHOC			KHOA			
MAMH	TENMH	SOTINCHI	MAKH	TENKH	NGAYTL	DIEN THOAI
DT001	Điện tử	2	CNTT	Công nghệ thông tin		
KT003	Nguyên lý kế toán	3	DTQT	Đào tạo Quốc tế		
TT001	Toán rời rạc	3	QTKD	Quản trị kinh doanh		
TH001	Tin học ứng dụng	2	XD	Xây dựng		
TH002	Cơ sở dữ liệu	3				

SINHVIEN							
MASV	HOTEN	NGAY SINH	GIOI TINH	DIACHI	CMND	DIEN THOAI	MAKHOA
SV0001	Nguyễn Thị Hải	23-02-1990	Nữ	Hà Nội	111222333		CNTT
SV0002	Trần Văn Chính	24-12-1992	Nam	Đà Nẵng	111555777		DTQT
SV0003	Lê Thu Bạch Yến	21-02-1990	Nữ	Đà Nẵng	333344445		QTKD
SV0004	Trần Anh Tuấn	20-12-1990	Nam		987654321		CNTT
SV0005	Trần Thanh Mai	12-08-1991	Nữ	Quảng Nam	555555555		DTQT
SV0006	Trần Thị Thu Thủy	02-01-1991	Nữ		123456789		CNTT

KETQUA			
MASV	MAMH	LANTHI	DIEM
SV0001	TH001	1	3
SV0001	TH001	2	6
SV0002	TH001	1	8.5
SV0002	TH002	1	8
SV0003	KT003	1	3
SV0003	KT003	2	9
SV0003	TT001	1	5
SV0003	TH001	1	10
SV0006	TH001	1	4.5
SV0006	TH002	1	5

## 2. Lệnh xóa bảng

### ▪ Cú pháp:

```
DROP TABLE <Tên_bảng>
```

❖ **Lưu ý:** không thể xóa bảng nếu bảng đang được khoá ngoại của bảng khác tham chiếu tới.

KHOA(MAKHOA, TENKHOA, NGAYTL, DIEN THOAI)

SINHVIEN(MASV, HOTEN, NGAYSINH, GIOITINH, DIACHI, CMND, MAKHOA)

Bảng KHOA có khoá ngoại đang tham chiếu tới, do đó ta không thể xóa được bảng KHOA. Nếu muốn xóa KHOA thì phải xóa bảng SINHVIEN hoặc xóa ràng buộc khoá ngoại trước khi xóa bảng KHOA.

**3. Lệnh sửa bảng:** Sau khi đã tạo bảng, ta có thể sửa đổi cấu trúc hay thuộc tính bảng như: thêm cột, xóa khóa, thay đổi ràng buộc...

▪ **Cú pháp:**

```
ALTER TABLE <Tên_bảng>
  [ADD
    [<Tên_cột> <kiểu dữ liệu> [not null | null] [ràng buộc] ]
    | [, ràng buộc] ]
  | [DROP
    [COLUMN <Tên_cột>]
    | [[CONSTRAINT] <Tên_ràng_buộc>] ]
```

*Ví dụ 5.16.* Với bảng đã tạo như trong ví dụ 5.15, ta hủy cột CMND và ràng buộc khóa ngoại như sau:

```
alter table SINHVIEN
drop column DIENTHOAI, FK_SV_KHOA
```

Ta có thể tạo cột DIENTHOAI và khóa ngoại vừa được xóa bằng lệnh sửa bảng:

```
alter table SINHVIEN
add DIENTHOAI char(10) not null, constraint FK_SV_KHOA
Foreign Key(MAKHOA) references KHOA(MAKHOA)
```

### V.3.2. Câu lệnh thao tác dữ liệu

#### V.3.2.1. Câu lệnh bổ sung dữ liệu (INSERT)

Để bổ sung dữ liệu vào bảng ta có thể dùng 1 trong 2 cách sau:

- Thêm một dòng dữ liệu
- Thêm nhiều dòng dữ liệu bằng cách truy xuất từ các bảng dữ liệu khác

##### 1. Thêm 1 dòng dữ liệu

▪ **Cú pháp:**

```
INSERT INTO <tên bảng>([<Danh sách các cột>])
VALUES(<Danh sách các giá trị>)
```

*Ví dụ 5.17.* Thêm 1 dòng vào bảng KHOA, với bảng KHOA(MAKHOA, TENKHOA, NGAYTL, DIENTHOAI)

```
insert into KHOA (MAKHOA, NGAYTL, TENKHOA)
values ('DTQT', '19/01/2009', N'Đào tạo Quốc tế')
```

Khi đó, cột DIENTHOAI ở dòng dữ liệu này sẽ nhận giá trị mặc định (nếu có) hoặc nhận giá trị NULL. Nếu cột DIENTHOAI không có giá trị mặc định và không cho phép NULL thì câu lệnh sẽ bị lỗi.

Trong câu lệnh INSERT, *danh sách các cột* ngay sau tên bảng không nhất thiết phải khai báo nếu giá trị các cột được chỉ định đầy đủ trong *danh sách các giá trị*. Trong trường hợp này thứ tự các giá trị phải tuân theo đúng thứ tự của các cột trong bảng khi được định nghĩa. Câu lệnh insert trong ví dụ 5.17 có thể viết lại như sau:

```
insert into KHOA
values ('DTQT', N'Đào tạo Quốc tế', '19/01/2009', NULL)
```

Lưu ý đối với kiểu dữ liệu Unicode (nchar, nvarchar), cần phải thêm tiền tố N (viết hoa) vào trước chuỗi ký tự.

## 2. Thêm nhiều dòng dữ liệu từ câu SELECT

### ▪ Cú pháp:

```
INSERT INTO <tên bảng> ([<Danh sách các cột>])
<Câu lệnh SELECT>
```

*Ví dụ 5.18.* Giả sử ta có bảng SINHVIEN\_DANANG gồm các thông tin: MASV, HOTEN, NGAYSINH. Câu lệnh dưới đây sẽ thêm vào bảng này những sinh viên có địa chỉ Đà Nẵng từ bảng SINHVIEN

```
insert into SINHVIEN_DANANG
select MASV, HOTEN, NGAYSINH
from SINHVIEN
where DIACHI like N'%Đà Nẵng%'
```

❖ **Lưu ý:** Kết quả của câu lệnh SELECT phải có các cột tương thích với các cột được chỉ định trong bảng đích về số lượng và kiểu dữ liệu.

### V.3.2.2. Câu lệnh sửa dữ liệu (UPDATE)

### ▪ Cú pháp:

```
UPDATE <Tên_bảng>
SET <Tên_cột 1> = <giá trị 1>,
    <Tên_cột 2> = <giá trị 2>,
    ...
```

```
<Tên_cột n> = <giá trị n>
[FROM <Danh sách các bảng>]
[WHERE <Các điều kiện>]
```

Một câu lệnh UPDATE có thể cập nhập dữ liệu cho nhiều cột. Mệnh đề WHERE dùng để chỉ định các dòng dữ liệu chịu tác động của câu lệnh, nếu không có mệnh đề WHERE câu lệnh sẽ có tác động tới tất cả các dòng dữ liệu trong bảng được chỉ định.

*Ví dụ 5.19.* Cập nhật lại thông tin NGÀYTL là 12/2/2009, DIENTHOAI là 0510123456 cho khoa Đào tạo quốc tế

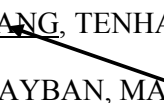
```
update KHOA
set NGÀYTL = '12/2/2009', DIENTHOAI='0510123456'
where MAKHOA = 'DTQT'
```

### Điều kiện cập nhật liên quan tới nhiều bảng

Mệnh đề FROM trong câu lệnh UPDATE được sử dụng khi cần chỉ định các giá trị và điều kiện cập nhật liên quan đến bảng khác, trong trường hợp này điều kiện WHERE thường có thêm điều kiện nối giữa các bảng.

*Ví dụ 5.20.* Giả sử ta có 2 bảng dữ liệu sau:

```
MATHANG(MAHANG, TENHANG, DONGIA)
NHATKY_BH(NGAYBAN, MAHANG, SOLUONG, THANHTIEN)
```



Câu lệnh sau cập nhật cột THANHTIEN = SOLUONG\*DONGIA

```
update NHATKY_BH
set THANHTIEN = SOLUONG*DONGIA
from MATHANG
where NHATKY_BH.MAHANG = MATHANG.MAHANG
```

### V.3.2.3. Câu lệnh xóa dữ liệu (DELETE)

#### ▪ Cú pháp:

```
DELETE [FROM] <Tên_bảng>
[FROM <Danh sách các bảng>]
[WHERE <Các điều kiện>]
```

Mệnh đề WHERE dùng để chỉ định các dòng dữ liệu chịu tác động của câu lệnh DELETE, nếu không có mệnh đề WHERE câu lệnh sẽ có tác động tới tất cả các dòng dữ liệu trong bảng được chỉ định.

*Ví dụ 5.21.* Xóa tất cả dữ liệu trong bảng SINHVIEN

```
delete from SINHVIEN
```

### Điều kiện xóa liên quan tới nhiều bảng

Mệnh đề FROM trong câu lệnh DELETE được sử dụng khi cần chỉ định các giá trị và điều kiện cập nhật liên quan đến bảng khác, trong trường hợp này điều kiện WHERE thường có thêm điều kiện nối giữa các bảng.

*Ví dụ 5.22.* Xóa tất cả những sinh viên nữ thuộc khoa Đào tạo Quốc tế

```
delete SINHVIEN
from KHOA
where KHOA.MAKHOA = SINHVIEN.MAKHOA
and TENKHOA = N'Đào tạo Quốc tế'
and GIOITINH = N'Nữ'
```

### V.3.2.4. Câu lệnh truy xuất dữ liệu (SELECT)

#### ▪ Cú pháp:

```
SELECT  [DISTINCT] [TOP n] <Danh sách các cột>
        [INTO <tên bảng mới> ]
FROM    <Danh sách các bảng>
[WHERE  <Các điều kiện>]
[GROUP BY <Danh sách các cột>]
[HAVING <Các điều kiện cho nhóm>]
[ORDER BY <Danh sách các cột>]
```

Câu lệnh SELECT được dùng để truy xuất dữ liệu từ các bảng và kết quả trả về cũng là dạng bảng. Lưu ý là các thành phần trong câu lệnh SELECT nếu được sử dụng phải tuân theo đúng cú pháp, nếu không câu lệnh sẽ không thực hiện được.

#### 1. Mệnh đề FROM

Chỉ định các bảng cần truy xuất dữ liệu, tên của các bảng được phân cách nhau bởi dấu phẩy.

*Ví dụ 5.23.* Cho biết họ tên, ngày sinh, CMND, địa chỉ của tất cả các sinh viên

```
select HOTEN, NGAYSINH, CMND, DIACHI
from SINHVIEN
```

Kết quả truy  
vấn:

	HOTEN	NGAYSINH	CMND	DIACHI
1	Nguyễn Thị Hải	1990-02-23 00:00:00.000	111222333	Hà Nội
2	Trần Văn Chính	1992-12-24 00:00:00.000	111555777	Đà Nẵng
3	Lê Thu Bạch Yến	1990-02-21 00:00:00.000	333344445	Đà Nẵng
4	Trần Anh Tuấn	1990-12-20 00:00:00.000	987654321	NULL
5	Trần Thanh Mai	1991-08-12 00:00:00.000	555555555	Quảng Nam
6	Trần Thị Thu Thủy	1991-01-02 00:00:00.000	123456789	NULL

## 2. Mệnh đề SELECT

Mệnh đề SELECT để chỉ định các trường, các biểu thức cần hiển thị trong câu lệnh truy vấn. Liên hệ với phép toán đại số quan hệ, mệnh đề SELECT chính là phép chiếu. Mệnh đề SELECT bao gồm các trường hợp sau:

### a. Chọn tất cả các cột

Khi cần hiển thị tất cả các trường trong bảng, ta sử dụng ký tự \* trong mệnh đề SELECT thay vì liệt kê tất cả các trường. Trong trường hợp này, các trường trong kết quả truy vấn hiển thị theo đúng thứ tự khi định nghĩa bảng.

*Ví dụ 5.24.* Cho biết thông tin của tất cả các sinh viên

```
select * from SINHVIEN
```

### b. Thay đổi tiêu đề cho cột

Mặc định, tiêu đề của cột trong kết quả truy vấn chính là tên các trường tương ứng trong bảng. Tuy nhiên, ta có thể đặt lại tiêu đề để kết quả hiển thị thân thiện hơn. Ta có các cách đặt tiêu đề cho cột như sau:

- Tiêu\_đề\_cột = Tên\_cột
- Tên\_cột as Tiêu\_đề\_cột
- Tên\_cột Tiêu\_đề\_cột

Trong trường hợp tiêu đề cột có khoảng trắng, ta phải đặt chúng trong cặp ngoặc vuông [ ] hoặc cặp dấu nháy đơn ‘’.

*Ví dụ 5.25.* Cho biết họ tên, ngày sinh, CMND, địa chỉ của tất cả các sinh viên

```
select 'Họ tên' = HOTEN, NGAYSINH as [Ngàysinh], CMND,
DIACHI 'Địa Chỉ'
```

from SINHVIEN

Kết quả truy vấn:

	Họ tên	Ngày sinh	CMND	Địa Chỉ
1	Nguyễn Thị Hải	1990-02-23 00:00:00.000	111222333	Hà Nội
2	Trần Văn Chính	1992-12-24 00:00:00.000	111555777	Đà Nẵng
3	Lê Thu Bạch Yến	1990-02-21 00:00:00.000	333344445	Đà Nẵng
4	Trần Anh Tuấn	1990-12-20 00:00:00.000	987654321	NULL
5	Trần Thanh Mai	1991-08-12 00:00:00.000	555555555	Quảng Nam
6	Trần Thị Thu Thủy	1991-01-02 00:00:00.000	123456789	NULL

### c. Sử dụng cấu trúc CASE trong câu lệnh SELECT

Ví dụ 5.26. Với bảng KETQUA(MASV, MAMH, LANTHI, DIEM), cho biết kết quả học tập của tất cả các sinh viên.

```
select MASV 'Mã SV', MAMH 'Mã môn', LANTHI 'Lần thi', Diem
      'Điểm', 'Kết quả' = CASE
                        WHEN DIEM >= 4 THEN N'Đậu'
                        ELSE N'Rớt'
                        END
```

from KETQUA

Kết quả:

	Mã SV	Mã môn	Lần thi	Điểm	Kết quả
1	SV0001	TH001	1	3.0	Rớt
2	SV0001	TH001	2	6.0	Đậu
3	SV0002	TH001	1	8.5	Đậu
4	SV0002	TH002	1	8.0	Đậu
5	SV0003	KT003	1	3.0	Rớt
6	SV0003	KT003	2	9.0	Đậu
7	SV0003	TT001	1	5.0	Đậu
8	SV0003	TH001	1	10.0	Đậu
9	SV0006	TH001	1	4.5	Đậu
10	SV0006	TH002	1	5.0	Đậu

### d. Biểu thức trong mệnh đề SELECT

Ví dụ 5.27. Với bảng MONHOC(MAMH, TENMH, SOTINCHI), cho biết thông tin của các môn học và số tiết tương ứng, biết 1 tín chỉ = 15 tiết.

```
select *, SOTIET = SOTINCHI*15
from MONHOC
```

Kết

	MAMH	TENMH	SOTINCHI	SOTIET
1	DT001	Điện tử	2	30
2	KT003	Nguyên lý kế toán	3	45
3	TT001	Toán rời rạc	3	45
4	TH001	Tin học ứng dụng	2	30
5	TH002	Cơ sở dữ liệu	3	45

quả:

### e. Dùng từ khóa DISTINCT

Từ khóa DISTINCT sử dụng để loại bỏ những bộ trùng nhau trong bảng kết quả. Trong mệnh đề SELECT, từ khóa DISTINCT được đặt trước danh sách các cột.

Ví dụ 5.28. Cho biết những năm sinh của tất cả các sinh viên:

```
select year(NGAYSINH) 'Năm sinh'
from SINHVIEN
```

Kết quả:

	Năm sinh
1	1990
2	1992
3	1990
4	1990
5	1991
6	1991

Sử dụng từ khóa DISTINCT:

```
select DISTINCT year(NGAYSINH) 'Năm sinh'
from SINHVIEN
```

Kết

	Năm sinh
1	1990
2	1991
3	1992

quả:

#### f. Dùng từ khóa TOP

Từ khóa TOP dùng để hạn chế số dòng hiển thị trong kết quả truy vấn.

Ví dụ 5.29. Cho biết 3 sinh viên đầu tiên trong danh sách các sinh viên:

```
select TOP 3 *
from SINHVIEN
```

Kết quả:

	MASV	HOTEN	NGAYSINH	GIOITINH	DIACHI	CMND	DIENHAI	MAKHOA
1	SV0001	Nguyễn Thị Hải	1990-02-23 00:00:00.000	Nữ	Hà Nội	111222333	NULL	CNTT
2	SV0002	Trần Văn Chính	1992-12-24 00:00:00.000	Nam	Đà Nẵng	111555777	NULL	DTQT
3	SV0003	Lê Thu Bạch Yến	1990-02-21 00:00:00.000	Nữ	Đà Nẵng	333344445	NULL	QTKD

### 3. Mệnh đề WHERE

Mệnh đề WHERE dùng để xác định điều kiện truy vấn dữ liệu, chỉ những dòng dữ liệu nào thỏa mãn biểu thức sau WHERE mới được hiển thị trong kết quả truy vấn. Liên hệ với phép toán đại số quan hệ, mệnh đề WHERE chính là phép chọn. Trong mệnh đề WHERE thường sử dụng:

- Các toán tử so sánh: =, >, <, >=, <=, ...



- Giới hạn: BETWEEN, NOT BETWEEN
- Danh sách: IN, NOT IN
- Khuôn dạng: LIKE, NOT LIKE
- Giá trị chưa biết: IS NULL, IS NOT NULL
- Toán tử logic: AND, OR, NOT
- Điều kiện tồn tại: EXISTS, NOT EXISTS

**a. Toán tử so sánh:**

*Ví dụ 5.30.* Cho biết những sinh viên sinh nam trên 20 tuổi

```
select *  
from SINHVIEN  
where GIOITINH = 'Nam'  
and year(getdate()) - year(NGAYSINH) > 20
```

**b. Giới hạn (BETWEEN, NOT BETWEEN):**

*Ví dụ 5.31.* Cho biết những sinh viên sinh nam từ 20 tới 22 tuổi

```
select *  
from SINHVIEN  
where GIOITINH = 'Nam'  
and year(getdate()) - year(NGAYSINH) between 20 and 22
```

**c. Danh sách (IN, NOT IN):**

*Ví dụ 5.32.* Ví dụ 5.31 có thể được viết lại như sau:

```
select *  
from SINHVIEN  
where GIOITINH = 'Nam'  
and year(getdate()) - year(NGAYSINH) in (20, 21, 22)
```

**d. Khuôn dạng (LIKE, NOT LIKE):**

Dùng để tìm kiếm dữ liệu theo khuôn dạng nào đó. Chúng thường được sử dụng kết hợp với các ký tự đại diện sau:

- % : đại diện cho 1 chuỗi ký tự bất kỳ gồm 0 hoặc nhiều ký tự
- \_ : đại diện cho 1 ký tự đơn bất kỳ
- [] : đại diện cho 1 ký tự đơn bất kỳ trong giới hạn được chỉ định

*Ví dụ 5.33.* Cho biết những sinh viên có họ là ‘Trần’

```
select MASV, HOTEN, GIOITINH  
from SINHVIEN
```

```
where HOTEN like N'Trần%
```

Kết quả:

	MASV	HOTEN	GIOITINH
1	SV0002	Trần Văn Chính	Nam
2	SV0004	Trần Anh Tuấn	Nam
3	SV0005	Trần Thanh Mai	Nữ
4	SV0006	Trần Thị Thu Thủy	Nữ

Ví dụ 5.34. Cho biết những sinh viên có họ tên có lý tự thứ 2 là 'g'

```
select MASV,HOTEN,GIOITINH
from SINHVIEN
where HOTEN like '_g%'
```

Ví dụ 5.35. Cho biết những sinh viên có họ tên có chứa các ký tự trong khoảng a-c

```
select MASV,HOTEN,GIOITINH
from SINHVIEN
where HOTEN like '%[a-c]%'
```

#### e. Giá trị null (IS NULL/ IS NOT NULL):

Ví dụ 5.36. Cho biết những sinh viên chưa cung cấp địa chỉ liên lạc

```
select MASV,HOTEN,DIACHI
from SINHVIEN
where DIACHI is null
```

Kết quả:

	MASV	HOTEN	DIACHI
1	SV0004	Trần Anh Tuấn	NULL
2	SV0006	Trần Thị Thu Thủy	NULL

#### f. Toán tử logic (AND, OR, NOT)

Ví dụ 5.37. Cho biết những sinh viên khoa 'DTQT' hoặc 'CNTT' có tuổi <=20

```
select HOTEN,MAKHOA,'TUỔI'= year(getdate())-year(NGAYSINH)
from SINHVIEN
where MAKHOA in ('DTQT','CNTT')
and year(getdate())-year(NGAYSINH)<=20
```

Kết

	HOTEN	MAKHOA	TUỔI
1	Trần Văn Chính	DTQT	19
2	Trần Thanh Mai	DTQT	20
3	Trần Thị Thu Thủy	CNTT	20

quả:

#### 4. Mệnh đề ORDER BY

Dùng để sắp xếp kết quả truy vấn theo một hay nhiều cột, theo thứ tự tăng (ASC) hoặc giảm (DESC), mặc định là tăng dần.

Ví dụ 5.38. Cho biết thông tin tất cả các môn học, sắp xếp các môn học giảm dần số tín chỉ.

```
select *
from MONHOC
order by SOTINCHI DESC
```

Kết

	MAMH	TENMH	SOTINCHI
1	KT003	Nguyên lý kế toán	3
2	TT001	Toán rời rạc	3
3	TH002	Cơ sở dữ liệu	3
4	DT001	Điện tử	2
5	TH001	Tin học ứng dụng	2

quả:

Câu lệnh sau sẽ sắp xếp các môn học giảm dần số tín chỉ, nếu số tín chỉ giống nhau thì sắp tăng dần theo tên môn học.

```
select *
from MONHOC
order by SOTINCHI DESC, TENMH
```

Kết

	MAMH	TENMH	SOTINCHI
1	TH002	Cơ sở dữ liệu	3
2	KT003	Nguyên lý kế toán	3
3	TT001	Toán rời rạc	3
4	DT001	Điện tử	2
5	TH001	Tin học ứng dụng	2

quả:

## 5. Liên kết nhiều bảng

### a. Phép nối nội

Khi cần thực hiện yêu cầu truy vấn dữ liệu từ trên 2 bảng trở lên, ta phải sử dụng đến phép nối giữa các bảng. Liên hệ với phép toán đại số quan hệ, liên kết nhiều bảng chính là *phép nối nội*.

Ví dụ 5.39. Cho biết thông tin tất cả các sinh viên, hiển thị: MASV, HOTEN, TENKHOA.

Bảng SINHVIEN

	MASV	HOTEN	NGAYSINH	GIOITINH	DIACHI	CMND	DIENHOA	MAKHOA
1	SV0001	Nguyễn Thị Hải	1990-02-23 00:00:00.000	Nữ	Hà Nội	111222333	NULL	CNTT
2	SV0002	Trần Văn Chính	1992-12-24 00:00:00.000	Nam	Đà Nẵng	111555777	NULL	DTQT
3	SV0003	Lê Thu Bạch Yến	1990-02-21 00:00:00.000	Nữ	Đà Nẵng	333344445	NULL	QTKD
4	SV0004	Trần Anh Tuấn	1990-12-20 00:00:00.000	Nam	NULL	987654321	NULL	CNTT
5	SV0005	Trần Thanh Mai	1991-08-12 00:00:00.000	Nữ	Quảng Nam	555555555	NULL	DTQT
6	SV0006	Trần Thị Thu Thủy	1991-01-02 00:00:00.000	Nữ	NULL	123456789	NULL	CNTT

Bảng KHOA

	MAKHOA	TENKHOA	NGAYTL	DIENTHOAI
1	CNTT	Công nghệ thông tin	NULL	NULL
2	DTQT	Đào tạo Quốc tế	NULL	NULL
3	QTKD	Quản trị kinh doanh	NULL	NULL
4	XD	Xây dựng	NULL	NULL

Nếu ta chỉ chọn 1 bảng SINHVIEN để truy vấn thì không thể biết được TENKHOA. Như vậy, để thực hiện câu truy vấn theo đúng yêu cầu hiển thị như trên ta phải nối từ 2 bảng SINHVIEN và KHOA (với điều kiện nối là MAKHOA của bảng SINHVIEN bằng với MAKHOA của bảng KHOA). Câu lệnh trong ví dụ này được viết như sau:

```
select MASV, HOTEN, TENKHOA
from SINHVIEN, KHOA
where SINHVIEN.MAKHOA = KHOA.MAKHOA
```

Kết quả:

	MASV	HOTEN	TENKHOA
1	SV0001	Nguyễn Thị Hải	Công nghệ thông tin
2	SV0002	Trần Văn Chính	Đào tạo Quốc tế
3	SV0003	Lê Thu Bạch Yến	Quản trị kinh doanh
4	SV0004	Trần Anh Tuấn	Công nghệ thông tin
5	SV0005	Trần Thanh Mai	Đào tạo Quốc tế
6	SV0006	Trần Thị Thu Thủy	Công nghệ thông tin

Trong phép nối, việc xác định điều kiện để thực hiện phép nối giữa các bảng đóng vai trò quan trọng nhất. Trong đa số trường hợp, điều kiện phép nối được xác định dựa vào mối quan hệ giữa các bảng, *thông thường đó là điều kiện bằng nhau giữa khóa chính và khóa ngoại của hai bảng có quan hệ với nhau.*

Nếu một cột xuất hiện trong câu SELECT nằm trong nhiều bảng thuộc mệnh đề FROM, cột đó phải được chỉ định cụ thể là thuộc bảng nào bằng cú pháp: Tên\_bảng.Tên\_cột. Câu lệnh dưới sẽ báo lỗi “*Ambiguous column name 'MAKHOA'.*”

```
select MASV, HOTEN, TENKHOA, MAKHOA
from SINHVIEN, KHOA
where MAKHOA = MAKHOA
```

Câu lệnh bị này bị lỗi  
vì trường MAKHOA  
không rõ ràng

Câu lệnh viết đúng phải là:

```
select MASV, HOTEN, TENKHOA, KHOA.MAKHOA
from SINHVIEN, KHOA
where SINHVIEN.MAKHOA = KHOA.MAKHOA
```

Ta cũng có thể sử dụng bí danh cho bảng để câu truy vấn đơn giản hơn, cách đặt bí danh cho bảng:

- Tên\_bảng as Tên\_bí\_danh
- Tên\_bảng Tên\_bí\_danh

```
select MASV, HOTEN, TENKHOA, k.MAKHOA
from SINHVIEN s, KHOA as k
where s.MAKHOA = k.MAKHOA
```

Nếu không có điều kiện nối, kết quả chính là phép tích Descartes giữa các bảng. Hãy xem kết quả của câu truy vấn sau:

```
select MASV, HOTEN, s.MAKHOA, k.MAKHOA, k.TENKHOA
from SINHVIEN s, KHOA k
```

Kết quả:

	MASV	HOTEN	MAKHOA	MAKHOA	TENKHOA
1	SV0001	Nguyễn Thị Hải	CNTT	CNTT	Công nghệ thông tin
2	SV0002	Trần Văn Chính	DTQT	CNTT	Công nghệ thông tin
3	SV0003	Lê Thu Bạch Yến	QTKD	CNTT	Công nghệ thông tin
4	SV0004	Trần Anh Tuấn	CNTT	CNTT	Công nghệ thông tin
5	SV0005	Trần Thanh Mai	DTQT	CNTT	Công nghệ thông tin
6	SV0006	Trần Thị Thu Thủy	CNTT	CNTT	Công nghệ thông tin
7	SV0001	Nguyễn Thị Hải	CNTT	DTQT	Đào tạo Quốc tế
8	SV0002	Trần Văn Chính	DTQT	DTQT	Đào tạo Quốc tế
9	SV0003	Lê Thu Bạch Yến	QTKD	DTQT	Đào tạo Quốc tế
10	SV0004	Trần Anh Tuấn	CNTT	DTQT	Đào tạo Quốc tế
11	SV0005	Trần Thanh Mai	DTQT	DTQT	Đào tạo Quốc tế
12	SV0006	Trần Thị Thu Thủy	CNTT	DTQT	Đào tạo Quốc tế
13	SV0001	Nguyễn Thị Hải	CNTT	QTKD	Quản trị kinh doanh
14	SV0002	Trần Văn Chính	DTQT	QTKD	Quản trị kinh doanh
15	SV0003	Lê Thu Bạch Yến	QTKD	QTKD	Quản trị kinh doanh
16	SV0004	Trần Anh Tuấn	CNTT	QTKD	Quản trị kinh doanh
17	SV0005	Trần Thanh Mai	DTQT	QTKD	Quản trị kinh doanh
18	SV0006	Trần Thị Thu Thủy	CNTT	QTKD	Quản trị kinh doanh
19	SV0001	Nguyễn Thị Hải	CNTT	XD	Xây dựng
20	SV0002	Trần Văn Chính	DTQT	XD	Xây dựng
21	SV0003	Lê Thu Bạch Yến	QTKD	XD	Xây dựng
22	SV0004	Trần Anh Tuấn	CNTT	XD	Xây dựng
23	SV0005	Trần Thanh Mai	DTQT	XD	Xây dựng
24	SV0006	Trần Thị Thu Thủy	CNTT	XD	Xây dựng

## b. Sử dụng phép nối trong SQL2

Chuẩn SQL2 (SQL-92) đưa ra một cách khác để biểu diễn phép nối, ở cách biểu diễn này điều kiện của phép nối không những được chỉ định ở mệnh đề WHERE mà ở

còn có thể ở ngay mệnh đề FROM. Cách sử dụng này biểu diễn phép nối cũng như điều kiện nối rõ ràng hơn.

*Ví dụ 5.40.* Cho biết thông tin tất cả các sinh viên khoa ‘Đào tạo Quốc tế’, hiển thị: MASV, HOTEN, TENKHOA. Ta có 2 cách viết như sau:

Cách 1.

```
select MASV, HOTEN, TENKHOA
from SINHVIEN, KHOA
where SINHVIEN.MAKHOA = KHOA.MAKHOA
and TENKHOA = N'Đào tạo Quốc tế'
```

Như đã trình bày ở trên, câu truy vấn này thực chất là thực hiện tích Descartes giữa SINHVIEN và KHOA sau đó chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE. Ta có thể viết lại câu truy vấn theo chuẩn SQL2 như sau:

Cách 2. Chuẩn SQL2

```
select MASV, HOTEN, TENKHOA
from SINHVIEN inner join KHOA
on SINHVIEN.MAKHOA = KHOA.MAKHOA
where TENKHOA = N'Đào tạo Quốc tế'
```

Vì mệnh đề FROM được thực hiện trước mệnh đề WHERE nên so với cách 1, cách 2 sau khi thực hiện xong mệnh đề FROM sẽ có số bộ giá trị ít hơn, nên mệnh đề WHERE thực hiện nhanh hơn và do đó, việc thực thu có lẽ cải tiến hơn.

Phép tích Descartes cũng có thể được thực hiện trong SQL2 bằng phép nối CROSS JOIN:

```
select *
from SINHVIEN cross join KHOA
```

### c. Phép nối ngoài

SQL2 cung cấp các phép nối ngoài như sau:

- Phép nối ngoài trái (LEFT OUTER JOIN)
- Phép nối ngoài phải (RIGHT OUTER JOIN)
- Phép nối ngoài đầy đủ (FULL OUTER JOIN)

*Ví dụ 5.41.* Đếm số sinh viên của từng khoa, hiển thị MAKHOA, TENKHOA, SỐSV. Để hiểu được ví dụ này, nên xem ý nghĩa của mệnh đề GROUP BY được trình bày bên dưới.

Nếu chúng ta sử dụng phép nối nội để thực hiện truy vấn:

```
select k.MAKHOA, TENKHOA, count(MASV) 'SỐSV'
from KHOA k inner join SINHVIEN sv
      on sv.MAKHOA = k.MAKHOA
group by k.MAKHOA, TENKHOA
```

Kết

	MAKHOA	TENKHOA	SỐSV
1	CNTT	Công nghệ thông tin	3
2	DTQT	Đào tạo Quốc tế	2
3	QTKD	Quản trị kinh doanh	1

quả:

Vì khoa ‘Xây dựng’ không có sinh viên nên không được hiển thị trong bảng kết quả. Ở đây ta mong muốn nếu khoa không có sinh viên thì số sinh viên tương ứng với khoa sẽ là 0. Để thực hiện được điều này ta phải sử dụng phép nối ngoài, cụ thể là phép nối ngoài trái.

```
select k.MAKHOA, TENKHOA, count(MASV) 'SỐSV'
from KHOA k left outer join SINHVIEN sv
      on sv.MAKHOA = k.MAKHOA
group by k.MAKHOA, TENKHOA
```

Kết

	MAKHOA	TENKHOA	SỐSV
1	CNTT	Công nghệ thông tin	3
2	DTQT	Đào tạo Quốc tế	2
3	QTKD	Quản trị kinh doanh	1
4	XD	Xây dựng	0

quả:

Vì sao như vậy? để hiểu hơn về ví dụ 5.41 ta hãy xem kết quả của phép nối trong và nối ngoài giữa 2 bảng KHOA và SINHVIEN:

Phép nối trong giữa KHOA và SINHVIEN:

```
select k.MAKHOA, TENKHOA, MASV
from KHOA k inner join SINHVIEN sv
      on sv.MAKHOA = k.MAKHOA
```

Kết quả:

	MAKHOA	TENKHOA	MASV
1	CNTT	Công nghệ thông tin	SV0001
2	DTQT	Đào tạo Quốc tế	SV0002
3	QTKD	Quản trị kinh doanh	SV0003
4	CNTT	Công nghệ thông tin	SV0004
5	DTQT	Đào tạo Quốc tế	SV0005
6	CNTT	Công nghệ thông tin	SV0006

Phép nối ngoài trái giữa KHOA và SINHVIEN:

```
select k.MAKHOA, TENKHOA, MASV
```

```
from KHOA k left outer join SINHVIEN sv
on sv.MAKHOA = k.MAKHOA
```

Kết

	MAKHOA	TENKHOA	MASV
1	CNTT	Công nghệ thông tin	SV0001
2	CNTT	Công nghệ thông tin	SV0004
3	CNTT	Công nghệ thông tin	SV0006
4	DTQT	Đào tạo Quốc tế	SV0002
5	DTQT	Đào tạo Quốc tế	SV0005
6	QTKD	Quản trị kinh doanh	SV0003
7	XD	Xây dựng	NULL

quả:

Như vậy, khác với *phép nối trong*, *phép nối ngoài trái* hiển thị trong bảng kết quả luôn cả những dòng dữ liệu của bảng bên trái dù không thỏa điều kiện nối.

Tương tự phép nối ngoài trái, *phép nối ngoài phải* hiển thị trong bảng kết quả luôn cả những dòng dữ liệu của bảng bên phải dù không thỏa điều kiện nối. Trong ví dụ trên, nếu bảng SINHVIEN đứng trước bảng KHOA và phép kết nối trái được thay bằng phép kết nối phải thì kết quả hai trường hợp là như nhau.

*Phép nối ngoài đầy đủ* hiển thị trong bảng kết quả những dòng dữ liệu của bảng bên trái và những dòng dữ liệu bên phải dù những dòng này không thỏa điều kiện nối.

## 6. Thống kê dữ liệu với mệnh đề GROUP BY và HAVING

Ta dùng mệnh đề GROUP BY và HAVING để thống kê dữ liệu theo nhóm, GROUP BY tổ chức (phân hoạch) dữ liệu thành các nhóm rồi trên mỗi nhóm thực hiện tính toán như tính tổng, tính trung bình,... Mệnh đề HAVING thiết lập điều kiện cho các kết quả tính toán. Các hàm thống kê được dùng với GROUP BY là:

- SUM(<Biểu\_thức>): Tính tổng các giá trị  
SUM(DISTINCT <Biểu\_thức>): Tính tổng các giá trị khác nhau
- AVG(<Biểu\_thức>): Tính trung bình các giá trị  
AVG(DISTINCT <Biểu\_thức>): Tính trung bình các giá trị khác nhau
- COUNT(\*): Đếm số lượng các dòng  
COUNT(<Biểu\_thức>): Đếm số lượng giá trị (khác NULL)  
COUNT(DISTINCT <Biểu\_thức>): Đếm số lượng giá trị (khác NULL) khác nhau
- MAX(<Biểu\_thức>): Tính giá trị lớn nhất
- MIN(<Biểu\_thức>): Tính giá trị nhỏ nhất



**a. Thống kê trên toàn bộ dữ liệu**

Khi cần thống kê trên toàn bộ dữ liệu, ta sử dụng hàm thống kê trong mệnh đề SELECT của câu lệnh truy vấn. Ngoài các hàm cần thống kê, ta không được sử dụng thêm bất kỳ cột hay biểu thức nào.

Ví dụ 5.42. Cho biết điểm thi thấp nhất và cao nhất trong lần 1 của **tất cả** các sinh viên.

Bảng KETQUA có dữ liệu:

	MASV	MAMH	LANTHI	DIEM
1	SV0001	TH001	1	3.0
2	SV0001	TH001	2	6.0
3	SV0002	TH001	1	8.5
4	SV0002	TH002	1	8.0
5	SV0003	KT003	1	3.0
6	SV0003	KT003	2	9.0
7	SV0003	TT001	1	5.0
8	SV0003	TH001	1	10.0
9	SV0006	TH001	1	4.5
10	SV0006	TH002	1	5.0

Câu truy vấn được viết như sau:

```
select min(DIEM) 'Min lần 1', max(DIEM) 'Max lần 1'
from KETQUA
where LANTHI = 1
```

Kết quả:

	Min lần 1	Max lần 1
1	3.0	10.0

**b. Thống kê trên các nhóm dữ liệu**

Nếu cần thống kê trên các nhóm dữ liệu, ta cần sử dụng GROUP BY để phân hoạch dữ liệu thành các nhóm, các hàm thống kê sẽ được thực hiện trên từng nhóm dữ liệu.

Ví dụ 5.43. Cho biết điểm thi thấp nhất và cao nhất trong lần 1 của **từng** các sinh viên.

```
select MASV, min(DIEM) 'Min lần 1', max(DIEM) 'Max lần 1'
from KETQUA
where LANTHI = 1
group by MASV
```

Kết

	MASV	Min lần 1	Max lần 1
1	SV0001	3.0	3.0
2	SV0002	8.0	8.5
3	SV0003	3.0	10.0
4	SV0006	4.5	5.0

quả:

Ví dụ 5.44. Cho biết điểm trung bình lần 1 của **từng** các sinh viên, hiển thị: MASV, HOTEN và điểm trung bình lần 1.

```
select k.MASV, HOTEN, avg(DIEM) 'Trung bình lần 1'
from KETQUA k inner join SINHVIEN s on k.MASV=s.MASV
where LANTHI = 1
group by k.MASV, HOTEN
```

Kết

	MASV	HOTEN	Trung bình lần 1
1	SV0001	Nguyễn Thị Hải	3.000000
2	SV0002	Trần Văn Chính	8.250000
3	SV0003	Lê Thu Bạch Yến	6.000000
4	SV0006	Trần Thị Thu Thủy	4.750000

quả:

❖ **Lưu ý:** Danh sách các tên cột và biểu thức được chọn để hiển thị trong câu lệnh truy vấn (thuộc mệnh đề SELECT) phải có mặt đầy đủ trong mệnh đề GROUP BY, nếu không câu lệnh sẽ không hợp lệ.

Nếu ví dụ 5.44 được viết lại như bên dưới sẽ báo lỗi “*Column 'SINHVIEN.HOTEN' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause*” vì thiếu cột HOTEN trong mệnh đề group by.

```
select k.MASV, HOTEN, avg(DIEM) 'Trung bình lần 1'
from KETQUA k inner join SINHVIEN s on k.MASV = s.MASV
where LANTHI = 1
group by k.MASV
```

Câu lệnh bị lỗi vì thiếu trường HOTEN trong mệnh đề GROUP BY

Câu lệnh viết đúng phải là:

```
select k.MASV, HOTEN, avg(DIEM) 'Trung bình lần 1'
from KETQUA k inner join SINHVIEN s on k.MASV = s.MASV
where LANTHI = 1
group by k.MASV, HOTEN
```

### c. Chỉ định điều kiện lên hàm thống kê

Mệnh đề HAVING có chức năng lọc dữ liệu tương tự mệnh đề WHERE nhưng được dùng để chỉ định điều kiện đối với các giá trị thống kê được sinh ra từ các hàm thống kê. Mệnh đề HAVING phải được dùng kèm với mệnh đề GROUP BY.

Ví dụ 5.45. Cho biết điểm trung bình lần 1 trên 5.0 của **từng** các sinh viên, hiển thị: MASV, HOTEN và điểm trung bình lần 1.

```
select k.MASV, HOTEN, avg(DIEM) 'Trung bình lần 1'
```

```

from KETQUA k inner join SINHVIEN s on k.MASV=s.MASV
where LANTHI = 1
group by k.MASV, HOTEN
having avg(DIEM)>5

```

Kết

	MASV	HOTEN	Trung bình lần 1
1	SV0002	Trần Văn Chính	8.250000
2	SV0003	Lê Thu Bạch Yến	6.000000

quả:

Lưu ý mệnh đề WHERE không chấp nhận các hàm thống kê. Mệnh đề WHERE dùng để chỉ định điều kiện lên các dòng của bảng dữ liệu còn mệnh đề HAVING chỉ định điều kiện lên kết quả tính toán của các hàm thống kê.

## 7. Truy vấn con

Truy vấn con là câu lệnh truy vấn thường được lồng vào bên trong mệnh đề FROM, mệnh đề WHERE hay mệnh đề HAVING của câu truy vấn khác. Loại truy vấn này được sử dụng để biểu diễn cho những câu truy vấn cần sử dụng đến kết quả của câu truy vấn khác.

### a. Truy vấn con trong mệnh đề FROM

*Ví dụ 5.46.* Cho biết các điểm thi lần 1 của sinh viên có mã 'SV0001', hiển thị: MASV, HOTEN, MAMH, LANTHI, DIEM

Cách 1.

```

select k.MASV, HOTEN, MAMH, LANTHI, DIEM
from KETQUA k, SINHVIEN s
where k.MASV = s.MASV
and LANTHI = 1
and k.MASV = 'SV0001'

```

Cách 2.

```

select k.MASV, HOTEN, MAMH, LANTHI, DIEM
from (select *
      from KETQUA
      where LANTHI = 1
        and MASV = 'SV0001') k, SINHVIEN s
where k.MASV = s.MASV

```

Cả hai cách đều cho kết quả như nhau:

	MASV	HOTEN	MAMH	LANTHI	DIEM
1	SV0001	Nguyễn Thị Hải	TH001	1	3.0

**b. Truy vấn con trong mệnh đề WHERE**

Các toán tử thường sử dụng kết hợp với câu truy vấn con là: IN, EXISTS, ALL, ANY.

*Ví dụ 5.47.* Cho biết những khoa nào không có nhân viên, hiển thị: MAKHOA, TENKHOA

Cách 1.

```
select MAKHOA, TENKHOA
from KHOA
where MAKHOA not in (select MAKHOA
                      from SINHVIEN)
```

Cách 2.

```
select MAKHOA, TENKHOA
from KHOA
where not exists(select *
                 from SINHVIEN
                 where KHOA.MAKHOA=SINHVIEN.MAKHOA)
```

Kết quả:

	MAKHOA	TENKHOA
1	XD	Xây dựng

*Ví dụ 5.48.* Cho biết những sinh viên nào có điểm thi cao nhất, hiển thị MASV, HOTEN, DIEM

Cách 1.

```
select s.MASV, HOTEN, DIEM
from SINHVIEN s, KETQUA k
where s.MASV = k.MASV
and DIEM = (select max(DIEM)
            from KETQUA)
```

Cách 2.

```
select s.MASV, HOTEN, DIEM
from SINHVIEN s, KETQUA k
where s.MASV = k.MASV
and DIEM >= All (select DIEM
                 from KETQUA)
```

*Ví dụ 5.49.* Cho biết những sinh viên khoa CNTT sinh cùng năm với những sinh viên khoa DTQT

```
select *
from SINHVIEN
where MAKHOA = 'CNTT'
```

```
and year(NGAYSINH) = ANY(select year(NGAYSINH)
                           from SINHVIEN
                           where MAKHOA = 'DTQT')
```

### c. Truy vấn con trong mệnh đề HAVING

Các toán tử thường sử dụng kết hợp với câu truy vấn con là: ALL, ANY.

*Ví dụ 5.50.* Cho biết những khoa nào có đông sinh viên nhất

```
select k.MAKHOA, TENKHOA, count(*) 'SốSV'
from KHOA k, SINHVIEN sv
where k.MAKHOA = sv.MAKHOA
group by k.MAKHOA, TENKHOA
having count(*) >= All(select count(*) 'SốSV'
                     from SINHVIEN
                     group by MAKHOA)
```

Kết quả:

	MAKHOA	TENKHOA	SốSV
1	CNTT	Công nghệ thông tin	3

*Ví dụ 5.51.* Cho biết những sinh viên nào có điểm trung bình thấp nhất

```
select sv.MASV, HOTEN, avg(DIEM) 'SốSV'
from SINHVIEN sv, KETQUA kq
where kq.MASV = sv.MASV
group by sv.MASV, HOTEN
having avg(DIEM) <= All(select avg(DIEM)
                      from KETQUA
                      group by MASV)
```

Kết quả:

	MASV	HOTEN	SốSV
1	SV0001	Nguyễn Thị Hải	4.500000

### ❖ Lưu ý:

- Câu lệnh SELECT không cho phép hai hàm thống kê lồng nhau.

*Ví dụ 5.52.* Viết như bên dưới sẽ nhận được thông báo lỗi “Cannot perform an aggregate function on an expression containing an aggregate or a subquery”

```
select sv.MASV, HOTEN, avg(DIEM) 'SốSV'
from SINHVIEN sv, KETQUA kq
where kq.MASV = sv.MASV
group by sv.MASV, HOTEN
having avg(DIEM) = (select min(avg(DIEM)))
```

```
from KETQUA
group by MASV)
```

- Thứ tự thực hiện trong câu lệnh SQL như sau:

**FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY**

Ví dụ 5.53. So sánh 2 câu truy vấn sau:

```
(1) select HOTEN, year(getdate()) - year(NGAYSINH) as Tuoi
    from SINHVIEN
    where Tuoi > 20
```

Câu lệnh lỗi vì  
không hiểu  
trường Tuoi

```
(2) select HOTEN, year(getdate()) - year(NGAYSINH) as Tuoi
    from SINHVIEN
    order by Tuoi
```

Câu truy vấn 1 có thông báo lỗi là “**Invalid column name 'Tuoi'**” nhưng ở câu truy vấn thứ 2 thì không, lý do mệnh đề WHERE được thực hiện trước mệnh đề SELECT nên trong câu truy vấn 1 ở mệnh đề WHERE không hiểu được trường Tuoi, còn mệnh đề ORDER BY thực hiện sau mệnh đề SELECT nên không xuất hiện lỗi này.

### V.3.3. Câu lệnh kiểm soát dữ liệu

Bảo mật dữ liệu là một trong những yếu tố quan trọng đối với sự sống còn của CSDL. Hầu hết các HQTCSDL hiện nay đều cung cấp khả năng bảo mật CSDL với các chức năng: cấp quyền truy cập CSDL cho người dùng, thu hồi quyền đã cấp phát,... SQL Server cung cấp 2 câu lệnh để phục vụ cho chức năng trên là lệnh: GRANT và REVOKE

Ví dụ 5.54. Gán quyền thực thi các câu lệnh SELECT, UPDATE, DELETE trên bảng SINHVIEN cho người dùng có tên UserA

```
grant SELECT, UPDATE, DELETE
on SINHVIEN
to UserA
```

Với quyền trên, người dùng UserA có thể thực hiện câu lệnh truy vấn, xóa và sửa trên bảng SINHVIEN như:

```
select *
from SINHVIEN
```

Nhưng lại không được thao tác trên các bảng khác, câu lệnh sau người dùng UserA lại không thực hiện được vì không có quyền:

```
select *
from KETQUA
```

Người dùng UserA không thực hiện được câu lệnh này vì UserA không có quyền thao tác trên bảng KETQUA

Ví dụ 5.55. Thu hồi quyền thực thi các câu lệnh SELECT trên bảng SINHVIEN từ người dùng có tên UserA

```
grant SELECT
on SINHVIEN
from UserA
```

### V.3.4. SQL và các toán tử đại số quan hệ

STT	ĐSQH	SQL
1	Phép hợp: $R \cup S$	<pre>select * from R Union select * from S</pre>
2	Phép giao: $R \cap S$ với: $R(A_1, A_2, \dots, A_n)$ $S(A_1, A_2, \dots, A_n)$	<pre>select * from R where exists (select *               from S               where R.A<sub>1</sub> = S.A<sub>1</sub>                  and R.A<sub>2</sub> = S.A<sub>2</sub>                  and ...                  and R.A<sub>n</sub> = S.A<sub>n</sub>)</pre>
3	Phép trừ: $R - S$ với: $R(A_1, A_2, \dots, A_n)$ $S(A_1, A_2, \dots, A_n)$	<pre>select * from R where not exists (select *                   from S                   where R.A<sub>1</sub> = S.A<sub>1</sub>                      and R.A<sub>2</sub> = S.A<sub>2</sub>                      and ...                      and R.A<sub>n</sub> = S.A<sub>n</sub>)</pre>
4	Tích Descartes: $R \times S$ với: $R(A_1, A_2, \dots, A_n)$ $S(B_1, B_2, \dots, B_m)$	<pre>Select * From A,B       Hoặc Select * From A cross join B</pre>
5	Phép chiếu: $\pi_{A_i, A_j}(R)$ với: $R(A_1, A_2, \dots, A_n)$	<pre>Select A<sub>i</sub>, A<sub>j</sub> From R</pre>
6	Phép chọn: $\delta_{A_i=x}(R)$ với:	<pre>Select * From R</pre>

	$R(A_1, A_2, \dots, A_n)$	Where $A_i = x$
7	Phép nối: $R \bowtie_{A_i=B_j} S$ với: $R(A_1, A_2, \dots, A_n)$ $S(B_1, B_2, \dots, B_m)$	Select * From R inner join S on $A_i = B_j$ <b>Hoặc</b> Select * From R, S Where $A_i = B_j$
8	Phép nối ngoài trái: $R \ltimes_{A_i=B_j} S$	Select * From R left outer join S on $A_i = B_j$
9	Phép nối ngoài phải: $R \rtimes_{A_i=B_j} S$	Select * From R right outer join S on $A_i = B_j$
10	Phép chia : $R \div S$ với: $R(A, B, C)$ $S(A)$	select $R_1.B, R_1.C$ from R as $R_1$ where not exists ( select * from S where not exists(select * from R as $R_2$ where $R_2.A = S.A$ and $R_2.B = R_1.B$ and $R_2.C = R_1.C$

#### V.4. Tổng kết chương và ôn tập

##### V.4.1. Tổng kết chương

Chương này đã trình bày 3 ngôn ngữ giao tiếp giữa người và máy trong các HQTCSDL là ngôn ngữ định nghĩa dữ liệu, ngôn ngữ thao tác dữ liệu và ngôn ngữ kiểm soát dữ liệu. Các ngôn ngữ này được gọi chung là ngôn ngữ SQL.

Cấu trúc các câu lệnh đã được trình bày rất chi tiết, đặc biệt là câu lệnh SELECT dùng để truy vấn dữ liệu. Một câu SELECT bao gồm các mệnh đề phải theo đúng thứ tự: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY. Tùy vào yêu cầu cụ thể mà câu truy vấn có thể bỏ đi một vài mệnh đề, 2 mệnh đề SELECT và FROM bắt buộc phải có trong mọi câu select.

##### V.4.2. Câu hỏi ôn tập

1. Các ngôn ngữ giao tiếp giữa người và máy trong các HQTCSDL? Ý nghĩa của từng ngôn ngữ.
2. Câu truy vấn dữ liệu (câu select) bao gồm các mệnh đề nào? Ý nghĩa của từng mệnh đề?



3. Có mấy cách để đặt tiêu đề cho một cột trong bảng kết quả của câu truy vấn?  
Cho ví dụ?
4. Từ khóa DISTINCT dùng để làm gì? Có cách nào khác để loại bỏ các dòng giống nhau trong bảng kết quả truy vấn mà không dùng DISTINCT không?
5. Giá trị null có phải là giá trị khoảng trắng không?
6. Muốn truy vấn dữ liệu từ nhiều bảng ta phải nối những bảng cần thiết lại với nhau, cho biết điều kiện nối nằm ở mệnh đề FROM hay mệnh đề WHERE? Cho ví dụ.
7. Khi nào sử dụng mệnh đề GROUP BY? Mệnh đề GROUP BY có phải luôn được đi kèm với mệnh đề HAVING không? Cho ví dụ?
8. Câu truy vấn con là gì? Câu truy vấn con thường xuất hiện ở mệnh đề nào trong câu truy vấn?
9. Bí danh cho bảng có quan trọng không? Cho ví dụ khi nào nó thực sự quan trọng, khi nào thì không?
10. Cho ví dụ hai hàm thống kê không được lồng nhau trong câu truy vấn, và sử dụng phép toán khác để thay thế 2 hàm thống kê trong ví dụ này?

#### V.4.3. Bài tập

Với CSDL quản lý dự án cho bên dưới, hãy thực hiện các câu hỏi bằng SQL:

NHANVIEN(MANV, HOLOT, TENNV, NGSINH, PHAI, LUONG, DCHI, MA\_NQL, PHG)

PHONGBAN(MAPHG, TENPHG, TRPHG, NG\_NHANCHUC)

DIADIEM\_PHONG(MAPHG, DIADIEM)

DUAN(MADA, TENDA, DDIEM\_DA, PHONG)

PHANCONG(MA\_NVIEN, SODA, THOIGIAN)

THANNHAN(MA\_NVIEN, TENTN, PHAI, NGAYSINH, QUANHE)

PHONGBAN			
<u>MAPH</u>	TENPHG	TRPHG	NG_NHANCHUC
5	Nghiên cứu	333445555	22/05/1978
4	Điều hành	987987987	01/01/1985
1	Quản lý	888665555	19/06/1971

DIADIEM_PHONG	
<u>MAPHG</u>	<u>DIADIEM</u>
1	Đà Nẵng
4	Tp HCM
5	Quảng Nam
5	Đà Nẵng
5	Hà Nội

NHANVIEN
----------

<u>MANV</u>	<u>HOLOT</u>	<u>TENNV</u>	<u>NGSINH</u>	<u>PHAI</u>	<u>LUONG</u>	<u>DCHI</u>	<u>MA_NQL</u>	<u>PHG</u>
123456789	Đinh Bá	Tiến	09/01/1965	Nam	300	Đà Nẵng	333445555	5
333445555	Nguyễn Thanh	Tùng	08/12/1975	Nam	350	Đà Nẵng	888665555	5
999887777	Bùi Thúy	Vũ	19/07/1978	Nam	250	Hà Nội	987654321	4
987654321	Lê Thị	Nhàn	20/06/1963	Nu	430	Quảng Nam	888665555	4
666884444	Nguyễn Mạnh	Hùng	15/09/1972	Nam	280	Quảng Nam	333445555	5
453453453	Trần Thanh	Tâm	31/07/1982	Nam	250	Bình Định	333445555	5
987987987	Trần Hồng	Quân	29/03/1989	Nam	200	Đà Nẵng	987654321	4
888665555	Vương Ngọc	Quyên	10/10/1977	Nu	650	Quảng Nam		1

THANNHAN				
<u>MA NVIEN</u>	<u>TENTN</u>	<u>PHAI</u>	<u>NGAYSINH</u>	<u>QUANHE</u>
333445555	Như	Nu	05/04/2005	Con gái
333445555	Khang	Nam	25/10/2010	Con trai
333445555	Hiền	Nu	03/05/1980	Vo chong
987654321	Đăng	Nam	29/02/1970	Vo chong
123456789	Duy	Nam	01/01/1990	Con trai
123456789	Như	Nu	31/12/1994	Con gái
123456789	Hiền	Nu	05/05/1967	Vo chong

PHANCONG		
<u>MA NVIEN</u>	<u>SODA</u>	<u>THOIGIAN</u>
123456789	1	32.5
123456789	2	7.5
666884444	3	40
453453453	1	20
453453453	2	20
333445555	3	10
333445555	10	10
333445555	20	10
999887777	30	30
999887777	10	10
987987987	10	35
987987987	30	5
987654321	30	20
987654321	20	15
888665555	20	

DUAN			
<u>MADA</u>	<u>TENDA</u>	<u>DDIEM_DA</u>	<u>PHONG</u>
1	Sản phẩm X	Đà Nẵng	5
2	Sản phẩm Y	Quảng Nam	5
3	Sản phẩm Z	Hà Nội	5
10	Tin học hóa	Đà Nẵng	4
20	Quảng cáo	TpHCM	1
30	Đào tạo	Quảng Nam	4

**Câu hỏi:**

1. Cho biết danh sách các nhân viên thuộc phòng ‘Điều hành’.
2. Cho biết họ tên trưởng phòng ‘Quản lý’.

3. Cho biết những trưởng phòng tham gia dự án ở “Hà Nội”
4. Cho biết họ tên nhân viên có thân nhân.
5. Cho biết họ tên nhân viên được phân công tham gia dự án.
6. Cho biết danh sách các nhân viên nam thuộc phòng ‘Điều hành’.
7. Cho biết mã nhân viên có người thân tham gia dự án với mã ‘10’.
8. Danh sách các nhân viên (HOLOT, TENNV) được ‘Nguyễn Thanh Tùng’ phụ trách trực tiếp.
9. Danh sách các dự án (MADA) có nhân viên họ ‘Nguyễn’ tham gia.
10. Danh sách các dự án (MADA) mà phòng có họ trưởng phòng là ‘Nguyễn’ chủ trì.
11. Danh sách những dự án có:
  - Người tham gia có họ “Đinh”
  - Người trưởng phòng chủ trì dự án có họ “Đinh”
12. Viết lại tất cả các câu trên thành các câu truy vấn lồng.
13. Cho biết những nhân viên người thân trùng tên nhau
14. Cho biết những nhân viên không có người thân nào
15. Cho biết danh sách những nhân viên có 2 thân nhân trở lên
16. Cho biết những trưởng phòng có tối thiểu 1 thân nhân
17. Cho biết những trưởng phòng có mức lương ít hơn nhân viên của mình
18. Cho biết tên phòng, số lượng nhân viên và tổng lương của từng phòng
19. Cho biết tên phòng, số lượng nhân viên và mức lương trung bình của từng phòng.
20. Cho biết tên phòng, mức lương trung bình của phòng đó >40000.
21. Cho biết nhân viên nào có nhiều thân nhân nhất.
22. Với mỗi nhân viên, cho biết họ tên nhân viên và số thân nhân của nhân viên
23. Cho biết lương trung bình của tất cả các nhân viên nữ
24. Cho biết tên các nhân viên và tên các phòng ban mà họ làm trưởng phòng(nếu có)
25. Cho biết họ tên nhân viên và tên các dự án mà nhân viên đó tham gia
26. Cho biết tổng số thời gian thực hiện của từng dự án
27. Cho biết những dự án nào có tổng thời gian thực hiện nhiều nhất
28. Cho biết họ tên nhân viên có tổng thời gian tham gia dự án cao nhất

29. Cho biết họ tên trưởng phòng của phòng có đông nhân viên nhất
30. Ứng với mỗi phòng cho biết họ, tên nhân viên có mức lương cao nhất
31. Cho biết nhân viên tham gia tất cả các dự án
32. Cho phép nhân viên làm việc cho tất cả các dự án mà phòng số 5 chủ trì
33. Cho biết những nhân viên tham gia tất cả dự án mà có nhân viên '987654321' tham gia
34. Thêm dòng < '943775543' , 'Tran Minh', 'Tuấn' , '6/2/1980', 'Quảng Bình', 'Nam', 280, '888665555', 1> vào NHANVIEN
35. Thêm dòng <4, 'San pham A', 'Huế', 2> vào DEAN
36. Xóa trong PHANCONG các bộ thỏa mãn MA\_NVIEN = '333445555'
37. Xóa trong NHANVIEN các bộ thỏa mãn MANV = '987654321'
38. Đổi giá trị tại thuộc tính TRPHG, NG\_NHANCHUC đối với các bộ trong PHONGBAN thỏa mãn MAPHG = 5 thành '123456789' và '01/10/88'.

## Chương VI. RÀNG BƯỚC TOÀN VỆN DỮ LIỆU

### A. Mục tiêu

#### ➤ Về kiến thức:

Sau khi xong chương này sinh viên nắm vững các kiến thức sau:

- Hiểu được ý nghĩa của ràng buộc toàn vẹn
- Có thể phát biểu các ràng buộc toàn vẹn cho một lược đồ quan hệ bất kỳ và cài đặt chúng vào hệ quản trị CSDL.

#### ➤ Về kỹ năng

Giúp sinh viên phát triển các kỹ năng:

- Rèn luyện khả năng làm việc theo nhóm
- Chủ động, tích cực trong học tập
- Rèn luyện khả năng đọc sách, tự nghiên cứu học hỏi.

### B. Phương pháp giảng dạy

- Trình chiếu slides
- Làm bài tập và thảo luận tại lớp.

- Ra bài tập về nhà cho sinh viên

### C. Tài liệu tham khảo

1. Nguyễn An Tế, Đồng Thị Bích Thủy (1996). *Nhập Môn Cơ Sở Dữ Liệu*.
2. Nguyễn Đăng Ty, Đỗ Phúc (2003). *Giáo Trình Cơ Sở Dữ Liệu*.
3. Lê Tiến Vương (1994). *Nhập Môn Cơ Sở Dữ Liệu Quan Hệ*.
4. Jeffrey D. Ullman (1998). *Mô Hình Dữ Liệu Và Ngôn Ngữ Vấn Tin*. Tập 1

### D. Nội dung chi tiết

#### VI.1. Ràng buộc toàn vẹn

##### VI.1.1. Khái niệm ràng buộc toàn vẹn

Trong mỗi CSDL luôn tồn tại nhiều mối liên hệ giữa các thuộc tính, giữa các bộ; sự liên hệ này có thể xảy ra trong cùng một quan hệ hoặc trong các quan hệ của một lược đồ CSDL. Các mối liên hệ này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thoả mãn ở mọi thời điểm. Những điều kiện bất biến đó được gọi là *ràng buộc toàn vẹn*. Chẳng hạn, điểm đánh giá kết quả học tập của một sinh viên khi học một môn học không thể là một số âm, mỗi sinh viên phải có một mã số duy nhất để phân biệt với các sinh viên khác, mỗi sinh viên dự thi một môn học không quá 4 lần, ...

Nhiệm vụ của người phân tích thiết kế là phải phát hiện càng đầy đủ các ràng buộc toàn vẹn càng tốt và mô tả chúng một cách chính xác trong hồ sơ phân tích thiết kế, đó là một việc làm rất quan trọng. Ràng buộc toàn vẹn được xem như là một công cụ để diễn đạt ngữ nghĩa của CSDL. Một CSDL được thiết kế cồng kềnh nhưng nó thể hiện được đầy đủ ngữ nghĩa của thực tế vẫn có giá trị cao hơn rất nhiều so với một cách thiết kế gọn nhẹ nhưng nghèo nàn về ngữ nghĩa vì thiếu các ràng buộc toàn vẹn của CSDL.

Công việc kiểm tra ràng buộc toàn vẹn thường được tiến hành vào thời điểm cập nhật dữ liệu (thêm, sửa, xoá). Những ràng buộc toàn vẹn phát sinh phải cần được ghi nhận và xử lý một cách tường minh (thường là bởi một hàm chuẩn hoặc một đoạn chương trình).

Ràng buộc toàn vẹn và kiểm tra sự vi phạm ràng buộc toàn vẹn là hai trong số những vấn đề quan trọng trong quá trình phân tích thiết kế cơ sở dữ liệu, nếu không quan tâm đúng mức đến những vấn đề trên, thì có thể dẫn đến những hậu quả nghiêm trọng về tính an toàn và toàn vẹn dữ liệu, đặc biệt là đối với những cơ sở dữ liệu lớn.

### VI.1.2. Các đặc trưng của RBTV

Đặc trưng của một RBTV được thể hiện bằng: bối cảnh, nội dung và bảng tầm ảnh hưởng

#### VI.1.2.1. Bối cảnh

Bối cảnh là những quan hệ có khả năng dẫn tới vi phạm RBTV khi thực hiện các thao tác cập nhật (thêm, xóa, sửa).

*Ví dụ 6.1.* Với CSDL quản lý sinh viên trong chương 5, ta có ràng buộc: mỗi sinh viên dự thi một môn học không quá 4 lần.

- Các thao tác cập nhật có nguy cơ dẫn tới vi phạm RBTV: sửa lần thi, thêm một kết quả học tập mới.
- Bối cảnh: quan hệ KETQUA

#### VI.1.2.2. Nội dung

Nội dung của một RBTV thường được biểu diễn bằng ngôn ngữ tự nhiên hoặc ngôn ngữ hình thức. Dùng ngôn ngữ tự nhiên để biểu diễn RBTV thì khá dễ hiểu nhưng thiếu tính chặt chẽ, ngược lại ngôn ngữ hình thức thì cô đọng, chặt chẽ nhưng lại khó hiểu. Các ngôn ngữ hình thức thường dùng là: mã giả, đại số quan hệ, phép tính quan hệ, ...

*Ví dụ 6.2.* Xét ràng buộc toàn vẹn sau:

- Ngôn ngữ tự nhiên: mỗi sinh viên phải có một mã số duy nhất để phân biệt với các sinh viên khác
- Ngôn ngữ hình thức:

$$\forall s_1, s_2 \in \text{SINHVIEN}, \text{ nếu } (s_1 \neq s_2) \text{ thì:} \\ s_1.\text{MASV} \neq s_2.\text{MASV}$$

#### VI.1.2.3. Bảng tầm ảnh hưởng

Trong quá trình phân tích thiết kế một CSDL, người phân tích cần lập *bảng tầm ảnh hưởng* cho một ràng buộc toàn vẹn nhằm xác định xác định thao tác cập nhật nào cần phải kiểm tra RBTV khi được thực hiện trên quan hệ bối cảnh.

Bảng tầm ảnh hưởng có thể được lập cho từng RBTV, cũng có thể được lập chung cho tất cả các RBTV trong một CSDL.

Bảng tầm ảnh hưởng của một ràng buộc toàn vẹn có dạng sau:

Tên_RBTV	Thêm	Xóa	Sửa
Quan hệ 1	+	–	+
Quan hệ 2	–	+	–
...			
Quan hệ n	–	+	+

Trong đó: + : vi phạm RBTV

– : không vi phạm RBTV

Chẳng hạn + tại ô (dòng *Quan hệ 1*, cột *Thêm*): có nghĩa là khi *Thêm* một bộ vào *Quan hệ 1* thì có nguy cơ dẫn tới vi phạm ràng buộc toàn vẹn có tên là *Tên\_RBTV*.

Bảng tầm ảnh hưởng tổng hợp của nhiều ràng buộc toàn vẹn trong CSDL có dạng sau:

	Tên RBTV1			Tên RBTV2			...			Tên RBTVn		
	Thêm	Xóa	Sửa	Thêm	Xóa	Sửa	...	...	...	Thêm	Xóa	Sửa
Quan hệ 1	+	–	+							+	+	+
Quan hệ 2	–	+	–	+	–	–						
Quan hệ 3				–	+	+				–	+	–
...												
Quan hệ n	–	+	+							+	–	+

### VI.1.3. Giải pháp tránh vi phạm RBTV

Để tránh vi phạm ràng buộc toàn vẹn cần phải có những hành động thích hợp. Thông thường hệ thống từ chối thao tác cập nhật và đưa ra thông báo và yêu cầu sửa chữa dữ liệu của các thuộc tính cho phù hợp với quy tắc đảm bảo tính nhất quán dữ liệu. Thông báo phải đầy đủ và phải thân thiện với người sử dụng.

Hầu hết các HQTCSDDL trên thị trường hiện nay nói chung và SQL Server nói riêng đều hỗ trợ các phương pháp để kiểm tra RBTV, đó là các ràng buộc về khóa chính, ràng buộc khóa ngoại, ràng buộc kiểm tra (CHECK), ràng buộc duy nhất

(UNIQUE), ràng buộc không được phép rỗng (NOT NULL), các ràng buộc này chúng ta đã tìm hiểu trong mục V.3.1.2 chương 5. Các loại RBTV này có thể tạo cùng lúc với tạo cấu trúc bảng cũng có thể tạo khi bảng đã tồn tại. Ngoài ra, SQL server còn hỗ trợ các TRIGGER, PROCEDURE để kiểm tra RBTV rất hiệu quả.

## VI.2. Phân loại ràng buộc toàn vẹn

### VI.2.1. Ràng buộc toàn vẹn nội

Là các ràng buộc toàn vẹn liên quan đến một quan hệ, gồm 3 loại ràng buộc: RBTV miền giá trị, RBTV liên bộ, RBTV liên thuộc tính.

#### VI.2.1.1. RBTV miền giá trị

RBTV miền giá trị là RBTV có liên quan đến miền giá trị của thuộc tính nào đó trong một quan hệ. Nói cách khác, RBTV miền giá trị qui định các giá trị cho một thuộc tính trong một quan hệ.

*Ví dụ 6.3.* Xét RBTV1: Điểm đánh giá kết quả học tập của sinh viên phải thuộc khoảng [0,10]

- Bối cảnh: KETQUA
- Nội dung:

$\forall kq \in \text{KETQUA}$ , nếu  $(kq.DIEM \neq \text{null})$  thì:  
 $kq.DIEM \geq 0 \wedge kq.DIEM \leq 10$

- Bảng tầm ảnh hưởng:

RBTV1	Thêm	Xóa	Sửa
KETQUA	+	–	+

*Ví dụ 6.4.* Xét RBTV2: Giới tính của sinh viên là ‘Nam’ hoặc ‘Nữ’

- Bối cảnh: SINHVIEN
- Nội dung:

$\forall sv \in \text{SINHVIEN}$ , nếu  $(sv.GIOITINH \neq \text{null})$  thì :  
 $sv.GIOITINH \in \{\text{‘Nam’}, \text{‘Nữ’}\}$

- Bảng tầm ảnh hưởng:

RBTV2	Thêm	Xóa	Sửa
SINHVIEN	+	–	+



### ❖ Cài đặt RBTV miền giá trị trong SQL Server

RBTV miền giá trị chính là các ràng buộc kiểm tra (CHECK), xem lại cú pháp ràng buộc này tại mục V.3.1.2 chương 5.

*Ví dụ 6.5.* Giả sử bảng KETQUA đã được tạo, ta có thể tạo RBTV1 trong ví dụ 6.3 như sau:

```
alter table KETQUA
add constraint CK_DIEM check(DIEM >=0 and DIEM <=10)
```

#### VI.2.1.2. RBTV liên bộ

RBTV liên bộ biểu diễn sự tồn tại của bộ này phụ thuộc vào sự tồn tại của một hay nhiều bộ khác trong cùng quan hệ. Phổ biến trong RBTV liên bộ là RBTV về khóa chính.

*Ví dụ 6.6.* Xét RBTV3: Mỗi sinh viên có một mã số duy nhất để phân biệt với các sinh viên khác

- Bối cảnh: SINHVIEN
- Nội dung:

$$\forall s_1, s_2 \in \text{SINHVIEN}, \text{nếu } (s_1 \neq s_2) \text{ thì:} \\ s_1.\text{MASV} \neq s_2.\text{MASV}$$

- Bảng tầm ảnh hưởng:

RBTV3	Thêm	Xóa	Sửa
SINHVIEN	+	–	+

*Ví dụ 6.7.* Xét RBTV4: Mỗi sinh viên có một số CMND duy nhất

- Bối cảnh: SINHVIEN
- Nội dung:

$$\forall s_1, s_2 \in \text{SINHVIEN}, \text{nếu } (s_1 \neq s_2 \wedge s_1.\text{CMND} \neq \text{null} \wedge s_1.\text{CMND} \neq \text{null}) \text{ thì:} \\ s_1.\text{CMND} \neq s_2.\text{CMND}$$

- Bảng tầm ảnh hưởng:

RBTV4	Thêm	Xóa	Sửa
SINHVIEN	+	–	+

### ❖ Cài đặt RBTV liên bộ trong SQL Server

RBTV liên bộ được thường được cài đặt bằng ràng buộc PRIMARY KEY, UNIQUE. Ví dụ 6.6 cài đặt bằng ràng buộc PRIMARY KEY, ví dụ 6.7 cài đặt bằng ràng buộc UNIQUE.

### VI.2.1.3. RBTV liên thuộc tính

RBTV liên thuộc tính là ràng buộc giữa các thuộc tính trong cùng quan hệ.

*Ví dụ 6.8.* Cho quan hệ THUE\_PHONG(MAPHIEUTHUE, PHONG, KHACHHANG, NGAYTHUE, NGAYTRA)

Xét RBTV5: Mỗi phiếu thuê phòng phải có ngày trả sau ngày thuê

- Bối cảnh: THUE\_PHONG
- Nội dung:

$\forall t \in \text{THUE\_PHONG}$ , nếu  $(t.\text{NGAYTHUE} \neq \text{null} \wedge t.\text{NGAYTRA} \neq \text{null})$  thì:  
 $t.\text{NGAYTRA} \geq t.\text{NGAYTHUE}$

- Bảng tầm ảnh hưởng:

RBTV5	Thêm	Xóa	Sửa
THUE_PHONG	+	–	+

### ❖ Cài đặt RBTV liên thuộc tính trong SQL Server

Các RBTV liên thuộc tính được cài đặt bằng ràng buộc CHECK.

*Ví dụ 6.9.* Giả sử bảng THUE\_PHONG đã được tạo, RBTV5 trong ví dụ 6.8 có thể được cài đặt như sau:

```
alter table THUE_PHONG
add constraint CK_NGAYTRA check(NGAYTRA >= NGAYTHUE)
```

### VI.2.2. Ràng buộc toàn vẹn ngoài

Là các ràng buộc toàn vẹn liên quan đến nhiều quan hệ, gồm 3 loại ràng buộc: RBTV khóa ngoại, RBTV liên bộ - liên quan hệ, RBTV liên thuộc tính - liên quan hệ.

#### VI.2.2.1. RBTV khóa ngoại

Ràng buộc toàn vẹn khoá ngoại còn được gọi là ràng buộc toàn vẹn phụ thuộc tồn tại. Bối cảnh của RBTV này thường có 2 quan hệ.

Ví dụ 6.10. Xét RBTv6: Mỗi sinh viên phải thuộc một khoa nhất định do trường quản lý.

- Bối cảnh: SINHVIEN, KHOA
- Nội dung:

$\forall sv \in \text{SINHVIEN}$ , nếu ( s.MAKHOA  $\neq$  null) thì:  
 $\exists kh \in \text{KHOA}$  sao cho:  
 sv.MAKHOA = kh.MAKHOA

Hoặc:

$\text{SINHVIEN}[\text{MAKHOA}] \subseteq \text{KHOA}[\text{MAKHOA}]$

- Bảng tầm ảnh hưởng:

RBTv6	Thêm	Xóa	Sửa
SINHVIEN	+	–	+
KHOA	–	+	+

#### ❖ Cài đặt RBTv khóa ngoại trong SQL Server

Các RBTv khóa ngoại được cài đặt bằng ràng buộc FOREIGN KEY.

#### VI.2.2.2. RBTv liên thuộc tính - liên quan hệ

Là các ràng buộc liên quan đến nhiều thuộc tính trên nhiều quan hệ.

Ví dụ 6.11. Cho 2 quan hệ:

PHIEUDATHANG(MaDH, MaKH, NgayDat)

PHIEUGIAOHANG(MaGH, MaDH, NgayGiao)

Xét RBTv7: Với mỗi phiếu đặt hàng, ngày giao hàng phải sau ngày đặt.

- Bối cảnh: PHIEUDATHANG, PHIEUGIAOHANG
- Nội dung:

$\forall g \in \text{PHIEUGIAOHANG}$ , nếu (g.NgayGiao  $\neq$  null) thì:  
 $\forall d \in \text{PHIEUDATHANG}$ , nếu (g.MaDH=d.MaDH  $\wedge$  d.NgayDat  $\neq$  null) thì:  
 g.NgayGiao  $\geq$  d.NgayDat

- Bảng tầm ảnh hưởng:

RBTv7	Thêm	Xóa	Sửa
PHIEUGIAOHANG	+	–	+
PHIEUDATHANG	–	–	+

❖ **Cài đặt RBTV liên thuộc tính - liên quan hệ trong SQL Server**

Các RBTV liên thuộc tính - liên quan hệ thường được cài đặt bằng các TRIGGER.

*Ví dụ 6.12.* BTV7 trong ví dụ 6.11 có thể được cài đặt bằng TRIGGER như sau:

```
create trigger KiemTra_NgayDat_NgayGiao
on PHIEUGIAOHANG after INSERT, UPDATE
as
begin
    declare @madh char(10), @ngaygiao datetime, @ngaydat
    datetime
    if((select count(*) from inserted)>1)
        select @madh = MADH, @ngaygiao = NGAYGIAO
        from inserted
        select @ngaydat = NGAYDAT
        from PHIEUDATHANG
        where MADH = @madh
        if(@ngaydat > @ngaygiao)
            begin
                Raiserror('ngay giao phai sau ngay dat',16,1)
                rollback
                return
            end
    else
        select @madh = MADH, @ngaygiao = NGAYGIAO
        from updated
        select @ngaydat = NGAYDAT
        from PHIEUDATHANG
        where MADH = @madh
        if(@ngaydat > @ngaygiao)
            begin
                raiserror('ngay giao phai sau ngay dat',16,1)
                rollback
                return
            end
    end
end
```

**VI.2.2.3. RBTV liên bộ - liên quan hệ**

Là các ràng buộc liên quan đến bộ giá trị trên nhiều quan hệ.

*Ví dụ 6.13.* Cho 2 quan hệ:

LOP(MaLop, TenLop, MaKhoa, SiSo)

HOCVIEN(MaHV, HoTen, NgaySinh, Phai, DiaChi, *MaLop*)

Xét RBTV8: Sĩ số của một lớp bằng tổng số lượng học viên trong lớp đó.

- Bối cảnh: LOP, HOCVIEN
- Nội dung:

$\forall lp \in LOP$ , nếu ( $lp.SiSo \neq null$ ) thì:

$$lp.SiSo = \text{count} \{ h \in HOCVIEN / h.MaLop = p.MaLop \}$$

- Bảng tầm ảnh hưởng:

RBTV8	Thêm	Xóa	Sửa
LOP	+	-	+
HOCVIEN	+	+	+

#### ❖ Cài đặt RBTV liên bộ - liên quan hệ trong SQL Server

Các RBTV liên bộ - liên quan hệ thường được cài đặt bằng các TRIGGER.

#### VI.2.2.4. RBTV chu trình

Ví dụ 6.14. Cho một CSDL gồm các quan hệ sau:

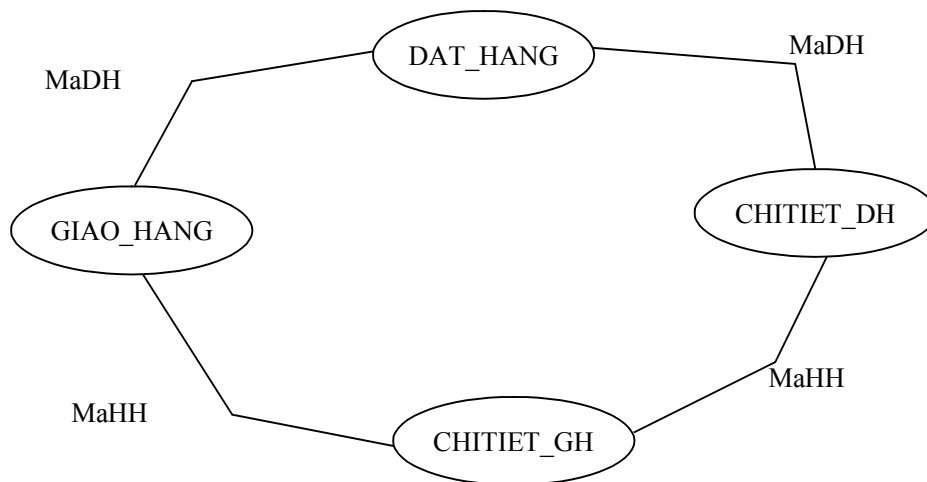
DAT\_HANG(MaDH, NgayDH, KhachHang)

CHITIET\_DH(MaDH, MaHH, SoLuongDat)

GIAO\_HANG(MaGH, MaDH, NgayGH,)

CHITIET\_GH(MaGH, MaHH, SoLuongGiao)

Biểu diễn CSDL trên bằng đồ thị như sau:



Trong hình vẽ ta nhận thấy đồ thị biểu diễn có chứa chu trình gồm 4 quan hệ. Ta xét ràng buộc toàn vẹn sau:

RBTV9: Một phiếu giao hàng cho một phiếu đặt hàng chỉ giao những mặt hàng mà khách hàng yêu cầu.

- Bối cảnh: DAT\_HANG, CHITIET\_DH, GIAO\_HANG, CHITIET\_GH
- Nội dung:

$$DH\_GH \leftarrow GIAO\_HANG \bowtie DAT\_HANG \bowtie CHITIET\_DH$$

$$\forall t \in CHITIET\_GH,$$

$$\exists s \in DH\_GH \text{ sao cho:}$$

$$t.MaHH = s.MaHH \wedge t.MaGH = s.MaGH \}$$

- Bảng tầm ảnh hưởng:

RBTV9	Thêm	Xóa	Sửa
DAT_HANG	–	–	+
CHITIET_DH	–	–	+
GIAO_HANG	–	–	+
CHITIET_GH	+	–	+

### ❖ Cài đặt RBTV chu trình trong SQL Server

Các RBTV chu trình thường được cài đặt bằng các TRIGGER.

## VI.3. Tổng kết chương và câu hỏi ôn tập

### VI.3.1. Tổng kết chương

Nói đến mô hình dữ liệu là nói đến 3 vấn đề chính : cấu trúc lưu trữ dữ liệu, các phép toán thao tác dữ liệu và các ràng buộc toàn vẹn trong mô hình mà dữ liệu cần phải thỏa mãn để đảm bảo tính nhất quán và toàn vẹn dữ liệu. Trong mô hình quan hệ, cấu trúc lưu trữ dữ liệu có dạng bảng, các phép toán thao tác với dữ liệu chính là đại số quan hệ (hai vấn đề này đã được trình bày trong chương 3).

Chương này đã trình bày khái niệm về RBTV, đặc trưng của RBTV cùng ý nghĩa của nó. RBTV được chia làm 2 nhóm chính: RBTV nội và RBTV ngoại, mỗi nhóm gồm nhiều loại RBTV, các loại này được trình bày khá chi tiết về đặc trưng cũng như cách cài đặt chúng trong mô hình quan hệ.

### VI.3.2. Câu hỏi ôn tập

1. Ràng buộc toàn vẹn là gì ? RBTV có quan trọng đối với mô hình dữ liệu không? Vì sao?
2. Các đặc trưng của một ràng buộc toàn vẹn?
3. Có thể biểu diễn RBTV bằng những ngôn ngữ nào? Ưu điểm của từng loại ngôn ngữ?
4. RBTV nội là gì? Có bao nhiêu loại RBTV nội? Ứng với mỗi loại cho một ví dụ?

5. RBTV ngoại là gì? Có bao nhiêu loại RBTV nội? Ứng với mỗi loại cho một ví dụ?

### VI.3.3. Bài tập

1. Hãy phát biểu tất cả các RBTV có thể có trong các mô hình quan hệ từ kết quả các bài tập 2, 3, 4, 5 và 6 trong chương 2: quản lý khách sạn, quản lý chuyến bay, quản lý sinh viên, quản lý thuê băng đĩa, quản lý bán hàng.
2. Với CSDL *quản lý dự án*, hãy phát biểu các RBTV sau bằng: bối cảnh, nội dung và bảng tầm ảnh hưởng:
  - a. Mỗi nhân viên phải có một mã số duy nhất để phân biệt với những nhân viên khác
  - b. Mỗi nhân viên thuộc một phòng nhất định
  - c. Trưởng phòng cũng là một nhân viên
  - d. Giới tính của nhân viên chỉ nhận một trong hai giá trị: Nam hoặc nữ
  - e. Lương của mỗi nhân viên phải lớn hơn 0
  - f. Ngày sinh của trưởng phòng phải nhỏ hơn ngày nhận chức
  - g. Lương của nhân viên không lớn hơn người quản lý trực tiếp nhân viên đó
  - h. Mỗi dự án chỉ có tối đa 5 nhân viên tham gia
  - i. Nhân viên và người thân không được trùng tên nhau
  - j. Thời gian nhân viên tham gia dự án phải lớn hơn 0.
  - k. Mỗi phòng chỉ có tối đa 3 địa điểm
  - l. Tên mỗi phòng ban phải là duy nhất.
  - m. Một nhân viên không được quản lý trực tiếp chính mình
  - n. Nhân viên chỉ được phân công vào những dự án do phòng mình chủ trì.

## Chương VII. CHUẨN HÓA CƠ SỞ DỮ LIỆU

### A. Mục tiêu

#### ➤ Về kiến thức:

Sau khi xong chương này sinh viên nắm vững các kiến thức sau:

- Hiểu được phụ thuộc hàm và sử dụng các tiên đề Amstrong để chứng minh một phụ thuộc hàm từ tập phụ thuộc hàm F cho trước.
- Định nghĩa khóa của một lược đồ quan hệ, áp dụng thuật toán để tìm một khóa và tất cả các khóa của một lược đồ quan hệ.
- Xác định được dạng chuẩn của một lược đồ quan hệ.
- Chuẩn hóa một lược đồ về dạng chuẩn cao hơn dựa vào phương pháp tách bảo toàn thông tin và bảo toàn phụ thuộc hàm.

#### ➤ Về kỹ năng

Giúp sinh viên phát triển các kỹ năng:

- Rèn luyện khả năng làm việc theo nhóm
- Chủ động, tích cực trong học tập
- Rèn luyện khả năng đọc sách, tự nghiên cứu học hỏi.

### B. Phương pháp giảng dạy

- Trình chiếu slides
- Làm nhiều bài tập và thảo luận tại lớp.
- Ra bài tập về nhà cho sinh viên

### C. Tài liệu tham khảo

1. Nguyễn An Tế, Đồng Thị Bích Thủy (1996). *Nhập Môn Cơ Sở Dữ Liệu*.
2. Nguyễn Xuân Huy, Lê Hoài Bắc (2008). *Bài Tập Cơ Sở Dữ Liệu*
3. Nguyễn Đăng Ty, Đỗ Phúc (2003). *Giáo Trình Cơ Sở Dữ Liệu*.
4. Lê Tiến Vương (1994). *Nhập Môn Cơ Sở Dữ Liệu Quan Hệ*
5. Jeffrey D. Ullman (1998). *Mô Hình Dữ Liệu Và Ngôn Ngữ Ván Tin*. Tập 1
6. Hoàng Trung Sơn (2004). *Thiết Kế Cơ Sở Dữ Liệu*.

### D. Nội dung chi tiết



## VII.1. Các nguyên tắc thiết kế lược đồ quan hệ

### VII.1.1. Ngữ nghĩa của các thuộc tính quan hệ

Khi chúng ta nhóm các thuộc tính để tạo nên một lược đồ quan hệ, ta giả thiết rằng có một ý nghĩa nào đó gắn với từng thuộc tính và các thuộc tính quan hệ với nhau như thế nào, ý nghĩa này còn gọi là *ngữ nghĩa*.

Nguyên tắc sau sẽ hỗ trợ cho việc thiết kế một lược đồ quan hệ:

Nguyên tắc 1: Các thuộc tính trong lược đồ quan hệ phải có ý nghĩa rõ ràng, tránh mập mờ. Một lược đồ quan hệ nên tương ứng với một tập thực thể hoặc một mối kết hợp.

### VII.1.2. Thông tin dư thừa trong các bộ và sự dị thường cập nhật

Một trong những mục tiêu của việc thiết kế lược đồ CSDL là làm tối thiểu không gian lưu trữ. Nếu cùng một thông tin được lưu giữ nhiều lần trong cơ sở dữ liệu thì ta gọi đó là dư thừa thông tin.

KETQUA(MASV, HOTEN, MONHOC, DIEMTHI)

Một thể hiện của lược đồ quan hệ KETQUA:

MASV	HOTEN	MONHOC	DIEMTHI
301.345	Nguyễn Văn Thành	Tin học đại cương	7
301.109	Trần Thu Hà	Toán cao cấp	6
301.345	Nguyễn Văn Thành	Tin học đại cương	8
301.345	Nguyễn Văn Thành	Lập trình cơ sở	7

Ta nhận thấy có ở đây có sự dư thừa thông tin. Nếu một sinh viên học nhiều môn học thì thông tin về sinh viên đó (MASV, HOTEN) được lưu giữ nhiều lần trong quan hệ. Ngoài việc dư thừa dẫn tới lãng phí không gian lưu trữ nó còn dẫn đến những dị thường là: dị thường khi sửa, dị thường khi xoá, dị thường khi sửa đổi.

Dị thường khi sửa: khi có nhu cầu cập nhật họ tên của sinh viên ‘301.345’ bắt buộc ta phải cập nhật tất cả các bộ giá trị có mã sinh viên này, sự cập nhật này dễ dẫn tới tình trạng thiếu nhất quán dữ liệu vì có thể ta bỏ sót hoặc có sự cố xảy ra trong quá trình cập nhật.

Dị thường khi chèn: chúng ta không thể biết họ tên của một sinh viên nếu hiện tại sinh viên đó không dự thi môn nào.

Dị thường khi xoá: chúng ta có thể xoá tất cả các môn thi của một sinh viên, vô ý làm mất dấu vết để tìm ra họ tên của sinh viên này.

Dựa vào các dị thường trên, ta có thể phát biểu nguyên tắc thiết kế như sau:

Nguyên tắc 2: Thiết kế các lược đồ quan hệ sao cho không sinh ra những dị thường trong các thể hiện quan hệ.

### VII.1.3. Các giá trị không xác định trong các bộ

Nếu một thuộc tính nào đó không thích hợp cho mọi bộ trong một quan hệ, chúng ta sẽ nhận giá trị null trong các bộ trong quan hệ đó. Điều này có thể làm tăng không gian ở mức lưu trữ gây nên lãng phí bộ nhớ.

Nguyên tắc 3: Tránh những thuộc tính mà trong một quan hệ thường xuyên nhận giá trị null.

## VII.2. Phụ thuộc hàm

### VII.2.1. Định nghĩa phụ thuộc hàm

Cho  $Q(U)$  là một lược đồ quan hệ với  $U = \{A_1, \dots, A_n\}$  là tập thuộc tính,  $X$  và  $Y$  là tập con khác rỗng của  $U$ . Nói rằng  $X \rightarrow Y$  là một phụ thuộc hàm được định nghĩa trên  $Q$  nếu với mọi thể hiện quan hệ  $R$  của lược đồ  $Q$  ta có:

$$\forall t_1, t_2 \in R: \text{ nếu } t_1[X] = t_2[X] \text{ thì } t_1[Y] = t_2[Y]$$

Nghĩa là không thể tồn tại 2 bộ trong  $R$  giống nhau ở các thuộc tính trong tập  $X$  mà lại khác nhau ở thuộc tính nào đó trong tập thuộc tính  $Y$ . Khi đó, phụ thuộc hàm  $X \rightarrow Y$  cũng có thể nói  $X$  xác định hàm  $Y$  hay  $Y$  phụ thuộc hàm vào  $X$ .

Phụ thuộc hàm  $X \rightarrow X$  được gọi là phụ thuộc hàm hiển nhiên. Người ta thường dùng  $F$  để chỉ tập các phụ thuộc hàm định nghĩa trên  $Q$ . Vì tập các thuộc tính  $U$  hữu hạn nên  $F$  cũng hữu hạn, các phần tử của  $F$  thường được đánh số  $f_1, f_2, \dots$

$F$  là tập các phụ thuộc hàm không hiển nhiên được định nghĩa trên  $Q$ , các phụ thuộc hàm hiển nhiên được ngầm hiểu là đã có trong  $F$ .

*Ví dụ 7.1.* Cho lược đồ quan hệ:

KETQUA\_HOCTAP(MASV, HOTEN, CMND, MAMH, TENMH, DIEMTHI)

Ta có các phụ thuộc hàm sau :

$MASV \rightarrow HOTEN, CMND$

CMND  $\rightarrow$  MASV, HOTEN

MAMH  $\rightarrow$  TENMH

MASV, MAMH  $\rightarrow$  DIEMTHI

Ví dụ 7.2. Cho lược đồ Q(ABCDE) với một thể hiện lược đồ R được xác định trên Q như sau:

A	B	C	D	E
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>	e <sub>1</sub>
a <sub>2</sub>	b <sub>1</sub>	c <sub>3</sub>	d <sub>3</sub>	e <sub>1</sub>
a <sub>2</sub>	b <sub>1</sub>	c <sub>4</sub>	d <sub>3</sub>	e <sub>1</sub>
a <sub>3</sub>	b <sub>2</sub>	c <sub>3</sub>	d <sub>1</sub>	e <sub>1</sub>

- Những phụ thuộc hàm thỏa R :  $AB \rightarrow D, A \rightarrow E, B \rightarrow E, C \rightarrow E, D \rightarrow E, \dots$
- Những phụ thuộc hàm không thỏa R:  $A \rightarrow D, E \rightarrow A, \dots$

## VII.2.2. Các quy tắc suy diễn đối với các phụ thuộc hàm

### VII.2.2.1. Hệ tiên đề Amstrong

Để có thể suy diễn (xác định) được các phụ thuộc hàm chưa có từ tập phụ thuộc hàm đã có ta dùng hệ tiên đề Armstrong, bao gồm các luật sau:

#### 1. Luật phản xạ

$$Y \subseteq X \Rightarrow X \rightarrow Y$$

#### 2. Luật tăng trưởng

$$X \rightarrow Y \text{ và } Z \subseteq W \Rightarrow XW \rightarrow YZ$$

#### 3. Luật bắc cầu

$$X \rightarrow Y \text{ và } Y \rightarrow Z \Rightarrow X \rightarrow Z$$

#### 4. Luật bắc cầu giả

$$X \rightarrow Y \text{ và } YW \rightarrow Z \Rightarrow XW \rightarrow Z$$

#### 5. Luật phân rã

$$X \rightarrow Y \text{ và } Z \subseteq Y \Rightarrow X \rightarrow Z$$

#### 6. Luật hội

$$X \rightarrow Y \text{ và } X \rightarrow Z \Rightarrow X \rightarrow YZ$$

**VII.2.2.2. Bao đóng của tập phụ thuộc hàm F**

Bao đóng của F, ký hiệu  $F^+$  là tập hợp tất cả các phụ thuộc hàm có thể suy diễn từ F dựa vào hệ tiên đề Amstrong. Tập F chính là tập con của  $F^+$ .

*Ví dụ 7.3.* Cho lược đồ quan hệ Q(ABCDEFGH) và tập phụ thuộc hàm F được cho như sau:

$$F = \{ B \rightarrow A; DA \rightarrow CE; D \rightarrow H; GH \rightarrow C; AC \rightarrow D \}$$

Khi đó:  $F^+ = \{ B \rightarrow A; DA \rightarrow CE; D \rightarrow H; GH \rightarrow C; AC \rightarrow D; BC \rightarrow D; AC \rightarrow H; GD \rightarrow C; AC \rightarrow E \dots \}$

Trên lý thuyết chúng ta hoàn toàn có thể tìm  $F^+$  vì tập thuộc tính U là hữu hạn. Nhưng trong thực tế bài toán xác định  $F^+$  từ tập F là không khả thi vì trong thực tế số lượng thuộc tính cũng như số lượng phụ thuộc hàm có thể rất lớn và sẽ dẫn đến sự bùng nổ về tổ hợp. Thay vào đó, chúng ta sẽ xét một bài toán khác: “*Kiểm tra một phụ thuộc hàm f có thuộc bao đóng  $F^+$  hay không?*”. Bài toán này được gọi là *bài toán thành viên*.

*Ví dụ 7.4.* Cho lược đồ quan hệ Q(ABCDEFG) với tập phụ thuộc hàm F:

$$F = \{ \begin{array}{l} f_1: AE \rightarrow C; \\ f_2: CG \rightarrow A; \\ f_3: BD \rightarrow G; \\ f_4: GA \rightarrow E \end{array} \}$$

$$\text{CMR: } BDC \rightarrow ABCDEG \text{ (hay } BDC \rightarrow ABCDEG \in F^+)$$

Ta có:

$$\begin{array}{ll} BD \rightarrow G & \text{(theo } f_3) \text{ (1)} \\ CG \rightarrow A & \text{(theo } f_2) \text{ (2)} \\ \text{(1) \& (2) suy ra : } BDC \rightarrow A & \text{(luật giả bắc cầu) (5)} \\ BDC \rightarrow BD & \text{(luật phản xạ) (3)} \\ BD \rightarrow G & \text{(theo } f_3) \text{ (4)} \\ \text{(3) \& (4) suy ra : } BDC \rightarrow G & \text{(luật bắc cầu) (6)} \\ \text{(5) \& (6) suy ra: } BDC \rightarrow GA & \text{(luật hội) (7)} \\ GA \rightarrow E & \text{(theo } f_4) \text{ (8)} \\ \text{(7) \& (8) suy ra: } BDC \rightarrow E & \text{(luật bắc cầu) (9)} \\ BDC \rightarrow BDC & \text{(luật phản xạ) (10)} \end{array}$$

(7) & (9) & (10) suy ra:  $BDC \rightarrow BDCEGA$  (luật hội)

### VII.2.2.3. Bao đóng của tập thuộc tính X

Cho lược đồ quan hệ  $Q(U)$  và tập các phụ thuộc hàm  $F = \{f_1, f_2, \dots, f_m\}$ , giả sử  $X$  là tập con của tập thuộc tính  $U$ . Bao đóng của tập thuộc tính  $X$  đối với  $F$  ký hiệu là  $X^+$  (hoặc  $X_F^+$ ) là tập tất cả các thuộc tính  $Y \subseteq U$  được suy ra từ tập thuộc tính  $X$  dựa vào các phụ thuộc hàm trong  $F$  và hệ tiên đề Armstrong:

$$X^+ = \{ Y \in U / X \rightarrow Y \in F^+ \}$$

*Ví dụ 7.5.* Cho lược đồ quan hệ  $Q(ABCDEFGH)$  với tập phụ thuộc hàm  $F$ :

$$F = \{ \begin{array}{l} f_1: B \rightarrow A \\ f_2: DA \rightarrow CE \\ f_3: D \rightarrow H \\ f_4: GH \rightarrow C \\ f_5: AC \rightarrow D \end{array} \}$$

Tính bao đóng của  $X = \{AC\}$

- $X^+ = AC$
- Do  $f_5$ :  $X^+ = AC\underline{D}$
- Do  $f_2$ :  $X^+ = ACDE$
- Do  $f_3$ :  $X^+ = ACDEH$

Vậy  $X^+ = (AC)^+ = ACDEH$

❖ **Định lý:**  $X \rightarrow Y \in F^+$  khi và chỉ khi  $Y \in X^+$

Dựa vào định lý trên chúng ta có thể đưa bài toán thành viên về một bài toán đơn giản hơn, đó là bài toán *tìm bao đóng của tập thuộc tính X*.

❖ **Thuật toán 7.1: Thuật toán tìm bao đóng  $X^+$**

Thuật toán tìm bao đóng với độ phức tạp  $O(n^2)$ , với  $n$  là số lượng thuộc tính của lược đồ quan hệ  $Q$ .

*Dữ liệu vào:*  $Q(U)$ , tập phụ thuộc hàm  $F$ , tập thuộc tính  $X \subseteq U$

*Dữ liệu ra:*  $X^+$

Thuật toán:

Bước 1.  $X^+ = X$

Bước 2.  $Old\_X^+ = X^+$

Với mỗi phụ thuộc hàm  $f: Y \rightarrow Z$  trong  $F$ :

If (  $Y \subseteq X^+$  )

$$X^+ = X^+ \cup Z$$

$$F = F - f$$

Bước 3. If (  $X^+ = Old\_X^+$  )

$X^+$  chính là kết quả cần tìm

Dừng thuật toán

Else

Quay lại Bước 2

Ví dụ 7.6. Cho lược đồ quan hệ  $Q(ABCDEFGH)$  với tập phụ thuộc hàm  $F$ :

$$F = \{ \begin{array}{l} f_1: B \rightarrow A \\ f_2: DA \rightarrow CE \\ f_3: D \rightarrow H \\ f_4: GH \rightarrow C \\ f_5: AC \rightarrow D \end{array} \}$$

Tính bao đóng của  $X = \{AC\}$

- $X^+ = AC$
- $f_5$  có vế trái chứa trong  $X^+ \rightarrow X^+ = ACD$
- $f_2$  có vế trái chứa trong  $X^+ \rightarrow X^+ = ACDE$
- $f_3$  có vế trái chứa trong  $X^+ \rightarrow X^+ = ACDEH$
- Thuật toán dừng và ta được  $(AC)^+ = ACDEH$

### VII.2.3. Sự tương đương của các tập phụ thuộc hàm

Một tập hợp các phụ thuộc hàm  $E$  được phủ bởi một tập các phụ thuộc hàm  $F$  nếu mỗi một phụ thuộc hàm trong  $E$  đều ở trong  $F^+$ , điều đó có nghĩa là mỗi phụ thuộc hàm trong  $E$  có thể suy diễn được từ  $F$ .

Hai tập phụ thuộc hàm  $E$  và  $F$  là *tương đương* nếu  $E^+ = F^+$ , hay nói cách khác là  $E$  phủ  $F$  và  $F$  phủ  $E$ . Có nghĩa, mỗi phụ thuộc hàm trong  $E$  có thể suy diễn được từ  $F$  và mỗi phụ thuộc hàm trong  $F$  có thể suy diễn được từ  $E$ .

*Ví dụ 7.7.* Chứng minh 2 tập phụ thuộc hàm bên dưới tương đương nhau:

$$F = \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H\}$$

$$F' = \{A \rightarrow CD, E \rightarrow AH\}$$

*Chứng minh  $F$  phủ  $F'$ :* ta cần chứng minh các phụ thuộc hàm trong  $F'$  suy ra từ  $F$

$$\left. \begin{array}{l} A_F^+ = ACD \Rightarrow A \rightarrow CD \\ E_F^+ = EADH \Rightarrow E \rightarrow AH \end{array} \right\} \Rightarrow \text{Các phụ thuộc hàm trong } F' \text{ suy ra từ } F$$

*Chứng minh  $F$  phủ  $F'$ :* ta cần chứng minh các phụ thuộc hàm trong  $F$  suy ra từ  $F'$

$$\left. \begin{array}{l} A_{F'}^+ = ACD \Rightarrow A \rightarrow C \\ AC_{F'}^+ = ACD \Rightarrow AC \rightarrow D \\ E_{F'}^+ = EAHCD \Rightarrow E \rightarrow AD, E \rightarrow H \end{array} \right\} \Rightarrow \text{Các phụ thuộc hàm trong } F \text{ có thể suy ra từ } F'$$

Vậy  $F$  và  $F'$  tương đương nhau.

#### VII.2.4. Tập phụ thuộc hàm tối thiểu (Phủ tối thiểu)

Một tập phụ thuộc hàm là tối thiểu nếu nó thỏa mãn các điều kiện sau đây:

- (1) Vế phải của các phụ thuộc hàm trong  $F$  chỉ có một thuộc tính.
- (2) Chúng ta không thể thay thế bất kỳ một phụ thuộc hàm  $X \rightarrow A$  trong  $F$  bằng phụ thuộc hàm  $Y \rightarrow A$ , với  $Y$  là tập con của  $X$  mà vẫn có một tập phụ thuộc hàm tương đương với  $F$ .
- (3) Chúng ta không thể bỏ đi bất kỳ phụ thuộc hàm nào ra khỏi  $F$  mà vẫn có một tập phụ thuộc hàm tương đương với  $F$ .

Chúng ta có thể nghĩ về tập tối thiểu các phụ thuộc hàm như là một tập hợp các phụ thuộc hàm không có sự dư thừa. Điều kiện 1 đảm bảo rằng mỗi phụ thuộc hàm là ở dạng chính tắc với chỉ một thuộc tính ở vế phải nên không có thuộc tính nào ở vế phải là dư thừa, điều kiện 2 đảm bảo không có thuộc tính nào ở vế trái dư thừa và điều kiện 3 đảm bảo không có phụ thuộc hàm nào dư thừa. Tập tối thiểu các phụ thuộc hàm còn được gọi là *phủ tối thiểu*.

Thường có rất nhiều phủ tối thiểu cho một tập các phụ thuộc hàm. Chúng ta luôn luôn có thể tìm được ít nhất là một phủ tối thiểu cho một tập các phụ thuộc hàm  $F$  bất kỳ theo thuật toán sau đây:

❖ **Thuật toán 7.2: Thuật toán tìm phủ tối thiểu của  $F$**

*Dữ liệu vào:*  $Q(U)$ , tập phụ thuộc hàm  $F$

*Dữ liệu ra:* phủ tối thiểu của  $F$

Thuật toán:

*Bước 1.* Chuyển các phụ thuộc hàm trong  $F$  thành các phụ thuộc hàm vế phải có 1 thuộc tính.

*Bước 2.* Loại bỏ các thuộc tính dư thừa bên vế trái bằng cách: với mỗi phụ thuộc hàm  $f: X \rightarrow Y \in F$ , duyệt lần lượt các thuộc tính  $A$  bên vế trái của  $f$ , nếu  $f \in (F - \{f\} \cup \{X - A \rightarrow Y\})^+$  thì loại  $A$  ra khỏi  $f$ .

*Bước 3.* Loại bỏ các phụ thuộc hàm dư thừa bằng cách duyệt lần lượt phụ thuộc hàm  $f$  trong  $F$ , nếu  $f \in (F - \{f\})^+$  thì loại bỏ  $f$  ra khỏi  $F$ .

*Ví dụ 7.8.* Cho lược đồ quan hệ  $Q(ABCD)$ . Tìm phủ tối thiểu tập phụ thuộc hàm:

$$F = \{ AB \rightarrow CD; B \rightarrow C; C \rightarrow D \}$$

Kết quả của bước 1 là:

$$F = \{ AB \rightarrow C; AB \rightarrow D; B \rightarrow C; C \rightarrow D \}$$

Kết quả của bước 2 là:

$$F = \{ B \rightarrow C; B \rightarrow D; B \rightarrow C; C \rightarrow D \}$$

Kết quả của bước 3 cho phủ tối thiểu:

$$F = \{ B \rightarrow C; C \rightarrow D \}$$

## VII.2.5. Khóa của lược đồ quan hệ

### VII.2.5.1. Định nghĩa khóa của lược đồ quan hệ

Cho lược đồ quan hệ  $Q(U)$  và tập các phụ thuộc hàm  $F$ . Khóa của lược đồ quan hệ là tập thuộc tính con  $K$  của  $U$  thỏa 2 điều kiện:

$$(1) \quad K^+ = U$$

$$(2) \quad \forall K' \subset K, (K')^+ \neq U$$

Một lược đồ quan hệ có thể có nhiều khóa.



Ví dụ 7.9. Tìm 1 khóa của lược đồ  $Q(ABCDEH)$  với tập phụ thuộc hàm  $F$ :

$$F = \{ BC \rightarrow E; D \rightarrow A; C \rightarrow A; AE \rightarrow D; BE \rightarrow CH \}$$

$$\text{Ta có: } (BC)^+ = BCEADH = U$$

$$B^+ = B \neq U$$

$$C^+ = CA \neq U$$

$$\Rightarrow BC \text{ là khóa của } Q$$

#### VII.2.5.2. Thuật toán tìm một khóa

##### ❖ Thuật toán 7.3:

Ý tưởng: Xuất phát từ tập thuộc tính  $U$  của lược đồ (gán  $K=U$ ), duyệt lần lược các thuộc tính  $A$  trong  $K$ , nếu  $(K - \{A\})^+ = U$  thì loại  $A$  ra khỏi  $K$ .  $K$  còn lại chính là một khóa cần tìm.

*Dữ liệu vào*:  $Q(U)$ , tập phụ thuộc hàm  $F$

*Dữ liệu ra*: Một khóa của  $Q$

Thuật toán:

*Bước 1.*  $K = U$

*Bước 2.* Với mỗi thuộc tính  $A \in K$

$$\text{if } (K - A)^+ = U)$$

$$K = K - A$$

*Bước 3.* return  $K$

Nếu muốn tìm các khóa khác (nếu có) của lược đồ quan hệ, ta có thể thay đổi thứ tự loại bỏ các thuộc tính trong  $K$ .

Ví dụ 7.10. Cho lược đồ quan hệ  $Q(ABC)$  và tập phụ thuộc hàm  $F$ . Hãy tìm một khóa của  $Q$ .

$$F = \{ A \rightarrow B; A \rightarrow C; B \rightarrow A \}$$

Bước 1.  $K = ABC$

Bước 2. - Loại thuộc tính  $A$ , do  $(K - A)^+ = U$  nên  $K = BC$

- Thuộc tính  $B$  không loại được do  $(K - B)^+ \neq U$  nên  $K = BC$

- Loại thuộc tính  $C$ , do  $(K - C)^+ = U$  nên  $K = B$

Bước 3. B là một khóa của Q

### VII.2.5.3. Thuật toán tìm tất cả các khóa

❖ **Thuật toán 7.4** (cơ bản):

*Dữ liệu vào:*  $Q(U)$ , tập phụ thuộc hàm  $F$

*Dữ liệu ra:* tất cả các khóa của  $Q$

Thuật toán:

*Bước 1:* Xác định tất cả các tập con của  $Q$ . Để xác định tất cả các tập con của một lược đồ quan hệ  $Q(U)$  với  $U = \{A_1, A_2, \dots, A_n\}$  ta lần lượt duyệt tất cả  $2^n - 1$  tập hợp con khác rỗng của  $U$  ( $n$  là số thuộc tính của  $Q$ ). Kết quả tìm được giả sử là các tập thuộc tính:  $S = \{X_1, X_2, \dots, X_{2^n-1}\}$

*Bước 2:* Tính  $X_i^+$ . Nếu  $X_i^+ = U$  thì  $X_i$  là siêu khoá. Giả sử sau bước này có  $m$  siêu khoá:  $S = \{S_1, S_2, \dots, S_m\}$

*Bước 3.* Tìm các khóa từ tập siêu khóa  $S$ . Xét mọi  $S_i$  và  $S_j$  con của  $S$  ( $i \neq j$ ), nếu  $S_i \subset S_j$  thì ta loại  $S_j$  ra khỏi  $S$ . Kết quả  $S$  còn lại chính là tập tất cả các khóa cần tìm.

*Ví dụ 7.11.* Tìm tất cả các khóa của lược đồ quan hệ  $Q(ABC)$  với tập phụ thuộc hàm được cho như sau:

$$F = \{A \rightarrow B; A \rightarrow C; B \rightarrow A\}$$

$X_i$	$X_i^+$	Siêu khóa	Khóa
A	ABC	A	A
B	ABC	B	B
C	C		
AB	ABC	AB	
AC	ABC	AC	
BC	ABC	BC	
ABC	ABC	ABC	

Vậy lược đồ quan hệ  $Q$  có hai khóa là:  $\{A\}$  và  $\{B\}$

Chú ý rằng thuật toán này tìm được tất cả các siêu khóa, tất cả các khóa. Thuật toán trên thì dễ hiểu dễ cài đặt, tuy nhiên nếu lược đồ với tập thuộc tính  $U$  lớn thì phép duyệt để tìm ra tập tất cả các tập con của tập  $U$  là điều không hiệu quả, do vậy cần thu

hợp không gian duyệt. Chúng ta sẽ nghiên cứu thuật toán cải tiến theo hướng giảm số thuộc tính của tập cần duyệt.

❖ **Thuật toán 7.5** (cải tiến):

Trước khi đi vào thuật toán cải tiến, ta cần đưa thêm một số khái niệm sau:

- Tập nguồn(TN): chứa tất cả các thuộc tính có xuất hiện ở vế trái và không xuất hiện ở vế phải của tập phụ thuộc hàm. Những thuộc tính không tham gia vào bất kỳ một phụ thuộc hàm nào thì cũng đưa vào tập nguồn.

- Tập đích (TD): chứa tất cả các thuộc tính có xuất hiện ở vế phải và không xuất hiện ở vế trái của tập phụ thuộc hàm.

- Tập trung gian(TG): chứa tất cả các thuộc tính vừa tham gia vào vế trái vừa tham gia vào vế phải của tập phụ thuộc hàm.

Ta có nhận xét: *Nếu K là khóa của lược đồ Q thì K chứa tất cả các thuộc tính nguồn và không chứa bất kỳ thuộc tính đích nào.*

*Dữ liệu vào:* Q(U), tập phụ thuộc hàm F

*Dữ liệu ra:* tất cả các khóa của Q

Thuật toán:

*Bước 1.* Tìm tập thuộc tính nguồn(TN), tập thuộc tính trung gian(TG)

*Bước 2.* if (TG =  $\phi$ )

Tập Khoá = Tập nguồn

Kết thúc

else

Tìm tất cả các tập con của tập trung gian

Qua bước 3

*Bước 3.* Với mỗi  $X_i$  thuộc tập con của tập trung gian

if  $((TN \cup X_i)^+ = U)$

$S = S \cup \{ TN \cup X_i \}$  ( S là tập các siêu khóa)

*Bước 4.* Xây dựng tập chứa tất cả các khóa của Q từ tập S. Xét mọi  $S_i, S_j$  con của S ( $i \neq j$ ), nếu  $S_i \subset S_j$  thì ta loại  $S_j$  ra khỏi S, kết quả còn lại chính là tập tất cả các khóa cần tìm.

*Ví dụ 7.12.* Áp dụng thuật toán cải tiến cho lược đồ quan hệ trong ví dụ 7.11:

$Q(ABC)$ , với  $F = \{ A \rightarrow B; A \rightarrow C; B \rightarrow A \}$

Ta có lời giải như sau:

$TN = \{ \phi \} ; TG = \{ A, B \}$

Gọi  $X_i$  là các tập con của tập TG

$X_i$	$TN \cup X_i$	$(TN \cup X_i)^+$	Siêu khóa	Khóa
$\phi$	$\phi$	$\phi$		
A	A	ABC	A	A
B	B	BAC	B	B
AB	AB	ABC	ABC	

Vậy lược đồ quan hệ Q có hai khóa là:  $\{A\}$  và  $\{B\}$

Chú ý rằng thuật toán này tìm được tất cả các khóa, nhưng không chắc tìm được tất cả các siêu khóa.

### VII.3. Các dạng chuẩn

Việc xác định chuẩn cho một lược đồ quan hệ có liên quan mật thiết với khóa, nên trước khi định nghĩa các dạng chuẩn chúng ta xem lại định nghĩa liên quan đến khóa của một lược đồ quan hệ:

- *Siêu khóa (superkey)*: Một siêu khóa của một lược đồ quan hệ  $Q(U)$  với tập thuộc tính  $U = A_1, A_2, \dots, A_n$  là một tập con S các thuộc tính của Q ( $S \subseteq U$ ), có tính chất là không có hai bộ khác nhau  $t_1, t_2$  trong một thể hiện nào của lược đồ Q thỏa  $t_1[S] = t_2[S]$ .

- *Khóa (key)*: Một khóa K của lược đồ quan hệ Q là một siêu khóa của Q và có tính chất là nếu bỏ đi bất kỳ thuộc tính nào ra khỏi K thì K không còn là siêu khóa nữa. Điều đó có nghĩa là khóa là một siêu khóa tối thiểu.

- *Khóa chính (primary key)*: Nếu một lược đồ quan hệ có nhiều hơn một khóa thì các khóa đó được gọi là các *khóa dự tuyển*. Một trong những khóa dự tuyển sẽ được chỉ định làm *khóa chính* và các khóa còn lại được gọi là các *khóa phụ* (secondary key). Một lược đồ quan hệ có duy nhất một khóa chính.

Một số khái niệm liên quan đến dạng chuẩn:

- *Thuộc tính khóa/ thuộc tính không khóa*: A là thuộc tính khóa nếu A tham gia vào bất kỳ một khóa nào của quan hệ, ngược lại A gọi là thuộc tính không khóa.

• *Phụ thuộc đầy đủ* : Thuộc tính A được gọi là phụ thuộc đầy đủ vào tập thuộc tính X nếu  $X \rightarrow A$  và không tồn tại  $X' \subset X$  sao cho  $X' \rightarrow A \in F^+$ . Khi đó  $X \rightarrow A$  được gọi là phụ thuộc hàm đầy đủ.

*Ví dụ 7.13.* Cho lược đồ Q(ABC) với tập phụ thuộc hàm:

$$F = \{A \rightarrow B; A \rightarrow C; AB \rightarrow C\}$$

Khi đó:  $A \rightarrow B$ ,  $A \rightarrow C$  là các phụ thuộc hàm đầy đủ, phụ thuộc hàm  $AB \rightarrow C$  không là phụ thuộc hàm đầy đủ vì có  $A \rightarrow C$ .

Chú ý rằng, một phụ thuộc hàm mà vế trái chỉ có một thuộc tính là phụ thuộc hàm đầy đủ.

### VII.3.1. Dạng chuẩn 1 (1NF)

Một lược đồ quan hệ Q được gọi là ở dạng chuẩn 1 nếu mọi thuộc tính trong Q đều là các thuộc tính đơn. Tức là những giá trị trong các thuộc tính đều không thể phân chia được nữa.

*Ví dụ 7.14.* Giả sử có lược đồ quan hệ:

CHUYENMON(MAGV, MONHOC)

Xét một thể hiện quan hệ được định nghĩa trên lược đồ CHUYENMON sau:

MAGV	MONHOC
GV01	CSDL, LTCS, THDC
GV02	TCC

Trong đó thuộc tính MonHoc là thuộc tính kép. Để có được dạng chuẩn 1 cho quan hệ trên, ta có thể tách thuộc tính kép thành thuộc tính đơn như sau:

CHUYENMON (MAGV, MONHOC)

MAGV	MONHOC
GV01	CSDL
GV01	LTCS
GV01	THDC
GV02	TCC

Chú ý rằng nếu ta không nói gì thêm, thì lược đồ quan hệ đang xét ít nhất là đạt dạng chuẩn 1.

**VII.3.2. Dạng chuẩn 2 (2NF)**

Một lược đồ quan hệ Q đạt dạng chuẩn 2 nếu Q đạt dạng chuẩn 1 và tất cả các thuộc tính không khoá của Q đều phụ thuộc đầy đủ vào khoá.

*Ví dụ 7.15.* Cho lược đồ quan hệ:

DON\_HANG(SODH, NGAYDH, KHACHHANG, MAHH, SOLUONG)

$F = \{SODH \rightarrow NGAYDH, KHACHHANG;$

$SODH, MAHH \rightarrow SOLUONG \}$

Lược đồ có 1 khóa là  $K = \{SODH, MAHH\}$ .

Lược đồ DON\_HANG không ở dạng chuẩn 2 vì thuộc tính không khóa NGAYDH, KHACHHANG không phụ thuộc đầy đủ vào khóa vì có phụ thuộc hàm  $SODH \rightarrow NGAYDH, KHACHHANG$  (phụ thuộc hàm này gọi là phụ thuộc hàm bộ phận, và thuộc tính KHACHHANG được gọi là phụ thuộc bộ phận vào khóa). Vậy lược đồ DON\_HANG đạt dạng chuẩn 1.

*Ví dụ 7.16.* Cho lược đồ quan hệ Q(ABCD) và tập phụ thuộc hàm:

$F = \{AB \rightarrow CD; B \rightarrow D; C \rightarrow A\}$

Q có 2 khóa là  $\{AB\}$  và  $\{BC\}$ . Do D là thuộc tính không khóa,  $AB \rightarrow D$  không là phụ thuộc hàm đầy đủ vì có  $B \rightarrow D$  nên Q không đạt 2NF. Vậy lược đồ đã cho đạt 1NF.

*Ví dụ 7.17.* Cho lược đồ quan hệ Q(GMVNHP) và tập phụ thuộc hàm:

$F = \{G \rightarrow N; G \rightarrow H; G \rightarrow P; M \rightarrow V; NHP \rightarrow M\}$

Dễ thấy Q có 1 khóa là  $\{G\}$ . Thuộc tính không khóa là M, V, N, H, P.

Do các phụ thuộc hàm  $G \rightarrow M; G \rightarrow V; G \rightarrow N; G \rightarrow H; G \rightarrow P$  là các phụ thuộc hàm đầy đủ, nên lược đồ quan hệ Q đạt 2NF

**❖ Hệ quả:**

- Q đạt 2NF nếu Q đạt 1NF và tập thuộc tính không khóa của Q bằng rỗng.
- Nếu khóa của quan hệ có một thuộc tính thì quan hệ đó ít nhất đạt chuẩn 2.

**VII.3.3. Dạng chuẩn 3 (3NF)**

Định nghĩa 1. Một lược đồ quan hệ Q được gọi là ở dạng chuẩn 3 nếu Q ở dạng chuẩn 2 và mọi thuộc tính không khóa của Q đều không phụ thuộc bắc cầu vào bất kỳ khóa nào.

Định nghĩa 2. Một lược đồ quan hệ Q đạt 3NF nếu mọi phụ thuộc hàm không hiển nhiên  $X \rightarrow A$  ( $A$  là 1 thuộc tính) thỏa:

- Hoặc X là một siêu khoá của Q
- Hoặc A là một thuộc tính khóa

*Ví dụ 7.18.* Cho lược đồ quan hệ Q(ABCD), với

$$F = \{ AB \rightarrow C ; D \rightarrow B ; C \rightarrow ABD \}$$

Q có 3 khóa là  $\{AB\}$ ,  $\{AD\}$  và  $\{C\}$ .

Tất cả các thuộc tính trong Q đều là thuộc tính khóa nên theo định nghĩa 1, Q đạt 3NF. Theo định nghĩa 2, tất cả các phụ thuộc hàm được cho trong F đều thỏa 1 trong 2 điều kiện. Vậy Q đạt 3NF.

❖ **Hệ quả:** Q đạt 3NF nếu tập thuộc tính không khóa của Q bằng rỗng.

*Ví dụ 7.19.* Lược đồ quan hệ sau có đạt 3NF không?

$$Q(NGPM) \text{ với tập phụ thuộc hàm } F = \{ NGP \rightarrow M ; M \rightarrow P \}$$

Dễ thấy các khóa của Q là  $\{NGP\}$ ,  $\{NGM\}$ . Theo định nghĩa 2, ta có:

$NGP \rightarrow M$  có vế trái là siêu khóa

$M \rightarrow P$  có vế phải là thuộc tính khóa.

Nên Q đạt chuẩn 3.

**VII.3.4. Dạng chuẩn Boyce-Codd (BCNF)**

Một lược đồ quan hệ Q được gọi là ở dạng chuẩn BC nếu mọi phụ thuộc hàm không hiển nhiên của F đều có vế trái chứa khóa.

*Ví dụ 7.20.* Cho lược đồ Q(ACDEIB) và tập phụ thuộc hàm :

$$F = \{ ACD \rightarrow EBI ; CE \rightarrow AD \}$$

Dễ thấy Q có hai khóa là: ACD và CE. Các phụ thuộc hàm của F đều có vế trái là siêu khóa, nên Q đạt dạng chuẩn BC.

❖ **Định lý:** Các lớp dạng chuẩn của một lược đồ quan hệ có quan hệ lồng nhau: nghĩa là lớp sau nằm trọn trong lớp trước.  $BCNF \subseteq 3NF \subseteq 2NF \subseteq 1NF$

*Ví dụ 7.21.* Chẳng hạn cho lược đồ quan hệ  $Q(ABCD)$ :

Nếu  $F = \{AB \rightarrow C; D \rightarrow B; C \rightarrow ABD\}$  thì  $Q$  đạt chuẩn 3NF nhưng không là BCNF

Nếu  $F = \{B \rightarrow D; A \rightarrow C; C \rightarrow ABD\}$  thì  $Q$  đạt dạng chuẩn 2NF nhưng không là 3NF

❖ **Lưu ý:** Dạng chuẩn của một lược đồ cơ sở dữ liệu là dạng chuẩn thấp nhất của các lược đồ quan hệ con.

#### VII.4. Các thuật toán thiết kế cơ sở dữ liệu quan hệ và các dạng chuẩn cao hơn

##### VII.4.1. Các thuật toán thiết kế cơ sở dữ liệu quan hệ

###### VII.4.1.1. Phân rã có nối không mất thông tin (phân rã bảo toàn thông tin)

Phân rã lược đồ  $Q(A_1, A_2, \dots, A_n)$  là thay nó bằng một tập các lược đồ  $p = \{Q_1, Q_2, \dots, Q_k\}$  sao cho  $Q = Q_1 \cup Q_2 \cup \dots \cup Q_k$ . Các tập thuộc tính  $Q_i$  không nhất thiết phải rời nhau. Một trong những lý do cần phân rã đó là nó có thể loại bỏ được một số vấn đề như đã được nói đến trong mục VII.1.

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ ,  $p$  là một phép phân rã  $Q$  thành các lược đồ con  $Q_1, Q_2, \dots, Q_k$  và  $F$  là tập phụ thuộc hàm, ta nói  $p$  là phép phân rã bảo toàn thông tin đối với  $F$  nếu với mỗi quan hệ  $R$  trên  $Q$  thỏa  $F$ :

$$R = \pi_{Q_1}(R) \bowtie \pi_{Q_2}(R) \bowtie \dots \bowtie \pi_{Q_k}(R)$$

Tức là  $R$  được tạo nên từ phép kết tự nhiên của các hình chiếu của nó trên các  $Q_i$

❖ **Thuật toán 7.6: Thuật toán kiểm tra tính kết nối không mất**



Dữ liệu vào:  $Q(A_1, A_2, \dots, A_n)$ ,  $F$ , phép phân rã  $\rho = \{Q_1, Q_2, \dots, Q_k\}$ .

Dữ liệu ra: Phép phân rã  $\rho$  có bảo toàn thông tin hay không?

### Thuật toán:

**Bước 1.** Thiết lập bảng với  $k$  dòng,  $n$  cột. Cột  $j$  ứng với thuộc tính  $A_j$  ( $j=1..n$ ), hàng  $i$  ứng với lược đồ quan hệ  $Q_i$  ( $i=1..k$ ). Tại vị trí hàng  $i$ , cột  $j$  ta điền ký hiệu  $a_j$  nếu  $A_j \in Q_i$ . Các ô còn lại được điền các ký hiệu  $b_t$  với  $t$  bắt đầu từ 1,  $t$  tăng dần một đơn vị theo chiều từ trái sang phải và từ trên xuống dưới.

**Bước 2.** Xét lần lượt các phụ thuộc hàm trong  $F$ , áp dụng cho bảng vừa mới thành lập ở trên. Chẳng hạn xét  $X \rightarrow Y \in F$ , chúng ta tìm những hàng giống nhau ở tất cả các thuộc tính của  $X$ , nếu có những hàng thỏa tính chất này ta sẽ làm cho các ký hiệu của hai hàng này bằng nhau ở tất cả các thuộc tính của  $Y$ . Khi làm cho hai ký hiệu của hai hàng này bằng nhau, ta gặp 2 trường hợp sau đây:

(1) Nếu một trong hai ký hiệu là  $a_j$  thì cho ký hiệu kia trở thành  $a_j$

(2) Nếu hai ký hiệu cùng là  $a$  hoặc cùng là  $b$  ( $b_k, b_l$ ) thì giữ nguyên

Chú ý rằng bước này có thể được lặp lại (cho các phụ thuộc hàm) cho đến khi không còn áp dụng được nữa (nghĩa là cho đến khi nào ở một lần duyệt qua tất cả các phụ thuộc hàm trong  $F$  mà bảng không có sự thay đổi nào).

**Bước 3.** Xét bảng kết quả, nếu thấy trong bảng này có một hàng chứa toàn  $a_j$  ( $j=1..n$ ) thì kết luận phép phân rã là bảo toàn thông tin, ngược lại thì phép phân rã không bảo toàn thông tin.

**Ví dụ 7.22.** Cho lược đồ quan hệ  $Q(\text{SAIP})$ , phép phân rã  $\rho = \{Q_1(\text{SA}); Q_2(\text{SIP})\}$  và tập phụ thuộc hàm  $F = \{S \rightarrow A; SI \rightarrow P\}$

Thực hiện bước 1 ta được:

	S	A	I	P
$Q_1(\text{SA})$	$a_1 \uparrow$	$a_2 \uparrow$	$b_1$	$b_2$
$Q_2(\text{SIP})$	$a_1 \downarrow$	$b_3 \downarrow$	$a_3$	$a_4$

Xét phụ thuộc hàm  $S \rightarrow A$  ta biến đổi bảng trên thành:

	S	A	I	P
$Q_1(\text{SA})$	$a_1$	$a_2$	$b_1$	$b_2$

$Q_2(\text{SIP})$	a1	a2	a3	a4
-------------------	----	----	----	----

Dòng thứ 2 của bảng chứa toàn giá trị  $a_j$  ( $j=1..4$ ) nên phép phân rã trên là bảo toàn thông tin. Thuật toán 7.6 có thể áp dụng cho phân rã với số lượng lược đồ bất kỳ. Tuy nhiên, đối với các phân rã thành hai lược đồ, chúng ta có một phép kiểm tra đơn giản hơn, đó là nội dung của định lý sau đây:

#### ❖ Định lý

Nếu  $p = \{Q_1, Q_2\}$  là một phân rã của  $Q$ , và  $F$  là một tập phụ thuộc hàm, thì  $p$  có nối không mất ứng với  $F$  nếu và chỉ nếu  $(Q_1 \cap Q_2) \rightarrow (Q_1 - Q_2)$  hoặc  $(Q_1 \cap Q_2) \rightarrow (Q_2 - Q_1)$ . Chú ý rằng những phụ thuộc hàm này không nhất thiết thuộc  $F_i$ , chỉ cần thuộc  $F^+$

*Ví dụ 7.23.* Giả sử  $Q(ABC)$  và  $F = \{A \rightarrow B\}$ . Thế thì phân rã của  $Q$  thành  $Q_1(AB)$  và  $Q_2(AC)$  có nối không mất bởi vì  $AB \cap AC = A$ ;  $AB - AC = B$  và  $A \rightarrow B$  đúng. Tuy nhiên nếu chúng ta phân rã  $Q$  thành  $Q_1(AB)$  và  $Q_2(BC)$ , chúng ta có  $Q_1 \cap Q_2 = B$ , nhưng cả  $Q_1 - Q_2$  lẫn  $Q_2 - Q_1$  đều không phụ thuộc vào  $B$ . Vì vậy phân rã này không có tính chất nối không mất ứng với  $F = \{A \rightarrow B\}$

#### VII.4.1.2. Phân rã bảo toàn phụ thuộc hàm

Cho phân rã  $p = \{Q_1, Q_2, \dots, Q_k\}$  của một lược đồ quan hệ  $Q$  và tập phụ thuộc hàm  $F$ . Hình chiếu của tập  $F$  lên tập thuộc tính  $Z$  ký hiệu  $\pi_Z(F)$  là tập các phụ thuộc hàm  $X \rightarrow Y \in F^+$  sao cho  $XY \subseteq Z$ . Ta nói phép phân rã  $p$  bảo toàn phụ thuộc hàm  $F$  nếu hợp của tất cả các phụ thuộc hàm trong  $\pi_{Q_i}(F)$  ( $i=1..k$ ) suy ra được tất cả các phụ thuộc hàm trong  $F$ .

Lý do  $p$  cần bảo toàn tập phụ thuộc hàm vì các phụ thuộc hàm trong  $F$  có thể được xem là các ràng buộc toàn vẹn cho quan hệ  $Q$ . Nếu các phụ thuộc hình chiếu không suy ra được  $F$  thì khi biểu diễn  $Q$  qua  $p$ , chúng ta có thể thấy rằng giá trị hiện hành của các  $Q_i$  biểu diễn một quan hệ không thoả  $F$ , ngay cả nếu  $p$  là phép phân rã không mất thông tin ứng với  $F$ . Khi đó mỗi thao tác cập nhật trên mỗi  $Q_i$  sẽ cần phải thực hiện một phép nối để kiểm tra lại rằng các ràng buộc không bị vi phạm. Do vậy vấn đề bảo toàn phụ thuộc hàm khi thực hiện phép phân rã là quan trọng.

*Ví dụ 7.24.* Cho lược đồ quan hệ  $Q(\text{CSZ})$  và tập phụ thuộc hàm:

$$F = \{CS \rightarrow Z; Z \rightarrow C\}$$

Phân rã lược đồ quan hệ  $Q$  thành  $Q_1(SZ)$  và  $Q_2(CZ)$ , phép phân rã này có nổi không mất vì  $SZ \cap CZ = Z$  và  $CZ - SZ = C$  và  $Z \rightarrow C \in F$ . Tuy nhiên hình chiếu của  $F$  lên  $Q_1(SZ)$  chỉ cho các phụ thuộc hàm hiển nhiên, còn chiếu trên  $Q_2(CZ)$  cho ra  $Z \rightarrow C$  và những phụ thuộc hàm hiển nhiên khác. Chúng ta có thể thấy rằng  $Z \rightarrow C$  và các phụ thuộc hàm hiển nhiên không suy ra  $CS \rightarrow Z$  được, vì thế phân rã này không bảo toàn phụ thuộc hàm.

Lưu ý rằng một phân rã có thể bảo toàn thông tin nhưng không chắc bảo toàn tập phụ thuộc hàm  $F$  (chẳng hạn như ví dụ 7.23), và cũng có những phân rã bảo toàn phụ thuộc hàm nhưng không bảo toàn thông tin, chẳng hạn như phân rã sau đây:

$Q(ABCD)$  với  $F = \{A \rightarrow B; C \rightarrow D\}$  và phân rã  $p = \{Q_1(AB), Q_2(CD)\}$ .

#### ❖ Thuật toán 7.7: Thuật toán kiểm tra bảo toàn phụ thuộc hàm

Về nguyên tắc chúng ta có thể dễ dàng kiểm tra xem một phân rã  $p = \{Q_1, Q_2, \dots, Q_k\}$  có bảo toàn tập phụ thuộc hàm hay không bằng cách tính  $F^+$  rồi chiếu nó trên tất cả các thành phần  $Q_i$ . Sau đó lấy hợp của các tập phụ thuộc kết quả rồi kiểm tra xem tập này có tương đương với tập  $F$  hay không ?

Tuy nhiên trong thực tế, tính  $F^+$  là một công việc hết sức khó khăn vì số lượng các phụ thuộc chứa trong nó thường là hàm mũ theo kích thước của  $F$ . Nhưng có một cách để kiểm tra tính bảo toàn này mà không cần phải tính  $F^+$ , phương pháp này có chi phí thời gian tỷ lệ với hàm đa thức theo kích thước của  $F$ . Sau đây là một thuật toán theo ý tưởng này.

*Dữ liệu vào:*  $Q(U), F$ , phân rã  $p = \{Q_1, Q_2, \dots, Q_k\}$

*Dữ liệu ra:* Phép phân rã  $p$  có bảo toàn phụ thuộc hàm hay không ?

Thuật Toán:

Gọi  $G$  là hợp của các hình chiếu của  $F$  lên các  $Q_i$ , tức là  $\pi_{Q_i}(F)$ . Để tính xem phép phân rã  $p$  có bảo toàn phụ thuộc hàm hay không chúng ta xét mỗi phụ thuộc hàm  $X \rightarrow Y \in F$  và xác định xem  $X_G^+$  có chứa  $Y$  hay không ( $X_G^+$  là  $X^+$  được tính ứng với tập phụ thuộc hàm  $G$ ), nếu có thì phụ thuộc hàm này  $X \rightarrow Y$  thuộc  $G^+$ . Nếu tất cả các phụ thuộc hàm trong  $F$  đều thuộc  $G^+$  thì phép phân rã  $p$  bảo toàn phụ thuộc hàm.

Ví dụ 7.25. Xét lược đồ quan hệ  $Q(ABCD)$  với phân rã  $p=(AB, BC, CD)$  Và tập phụ thuộc hàm  $F=\{A \rightarrow B; B \rightarrow C; C \rightarrow D\}$ . Theo thuật toán trên ta kết luận phép phân rã này bảo toàn phụ thuộc hàm.

### VII.4.1.3. Chuẩn hóa một lược đồ CSDL bằng cách tiếp cận phân rã

Quá trình chuẩn hoá trên một lược đồ quan hệ là thực hiện một loạt các kiểm tra lược đồ để xác nhận nó có thoả mãn dạng chuẩn nào và chuẩn hóa về dạng chuẩn mong muốn bằng phương pháp phân rã hoặc tổng hợp, giáo trình này chỉ đề cập đến phương pháp phân rã.

Theo quan điểm của cách tiếp cận phân rã, các quan hệ con của lược đồ cơ sở dữ liệu ban đầu sẽ lần lượt được phân rã thành những quan hệ con với số thuộc tính ít hơn sao cho cấu trúc kết quả đạt các tiêu chuẩn đề ra ở mức cao nhất. Quá trình phân rã là một quá trình được lặp lại đối với các quan hệ con nào được đánh giá là còn có thể phân rã được nữa.

## 1. Phân rã thành BCNF và bảo toàn thông tin

### ❖ Thuật toán 7.8

*Dữ liệu vào:* Lược đồ quan hệ  $Q(U)$  và tập phụ thuộc hàm  $F$

*Dữ liệu ra:* Phân rã  $p=\{Q_1, Q_2, \dots, Q_k\}$  đạt BCNF và bảo toàn thông tin

Thuật toán

*Bước 1.* Tìm tất cả các khoá của  $Q$ .

*Bước 2.* Tìm phụ thuộc hàm  $X \rightarrow Y \in F$  có  $X$  không là siêu khoá.

Nếu tìm thấy thì tách  $Q$  thành  $Q_1$  và  $Q_2$  theo quy tắc sau:

- $Q_1=Q[XY]$ , với  $F_1=\pi_{Q_1}(F)$  tìm bao đóng tất cả các tập con của  $XY$  để suy ra  $\pi_{Q_1}(F)$ .
- $Q_2=Q[Q^+-Y]$ , với  $F_2=\pi_{Q_2}(F)$  tìm bao đóng tất cả các tập con của  $Q^+-Y$  để suy ra  $\pi_{Q_2}(F)$ .

Thực hiện thuật toán phân rã  $(Q_1, F_1)$

Thực hiện thuật toán phân rã  $(Q_2, F_2)$

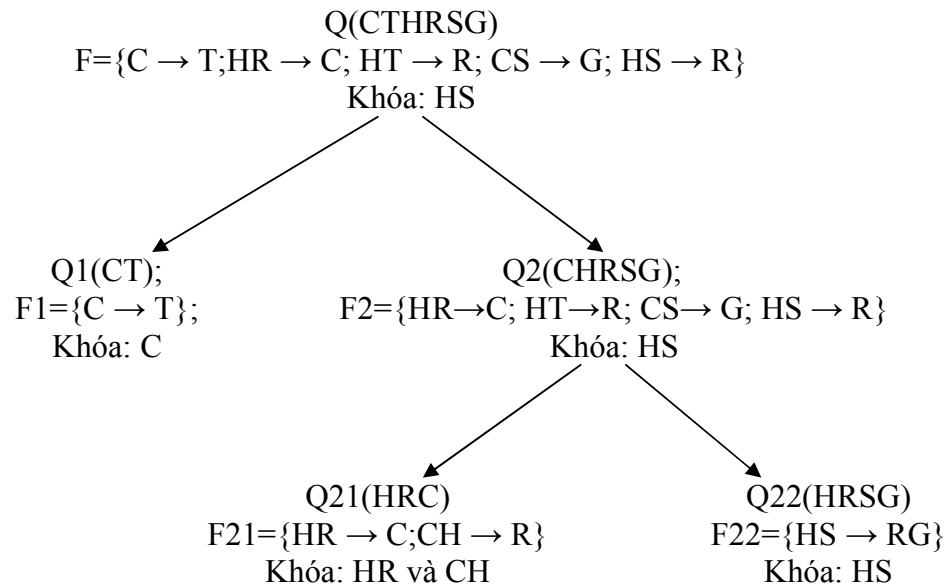
Ngược lại, thì không tìm thấy và kết thúc thuật toán.

Ví dụ 7.26. Cho lược đồ quan hệ  $Q(CTHRSG)$  và tập phụ thuộc hàm  $F$ . Hãy phân rã  $Q$  thành các lược đồ con đạt dạng chuẩn BC bảo toàn thông tin?

$$F = \{C \rightarrow T; HR \rightarrow C; HT \rightarrow R; CS \rightarrow G; HS \rightarrow R\}$$

Bước 1. Khoá của  $Q$ :  $HS$

Bước 2. Phụ thuộc hàm  $C \rightarrow T$  có vẻ trái không là siêu khoá



Sau khi phân rã  $Q(CTHRSG)$ , ta được lược đồ CSDL mới như sau:

$Q_1(CT)$ , với  $F_1 = \{C \rightarrow T\}$

$Q_{21}(HRC)$ , với  $F_{21} = \{HR \rightarrow C; CH \rightarrow R\}$

$Q_{22}(HRSG)$ , với  $F_{22} = \{HS \rightarrow RG\}$

Các lược đồ quan hệ trên đều đạt dạng chuẩn BC.

❖ Nhận xét: Theo thuật toán trên, khi phân rã  $Q$  thành  $Q_1$  và  $Q_2$  thì tập khoá của  $Q_2$  luôn bằng tập khoá của  $Q$ .

## 2. Phân rã thành 3NF và bảo toàn phụ thuộc hàm

Không phải lúc nào cũng có thể phân rã một lược đồ quan hệ thành dạng BC mà vẫn bảo toàn được các phụ thuộc hàm, tuy nhiên chúng ta luôn có thể tìm được một phân rã vừa bảo toàn phụ thuộc hàm vừa đạt 3NF.

❖ Thuật toán 7.9:

*Dữ liệu vào:* Lược đồ quan hệ  $Q(U)$  và tập phụ thuộc hàm  $F$

*Dữ liệu ra:* Phân rã  $p=\{Q_1, Q_2, \dots, Q_k\}$  đạt 3NF bảo toàn phụ thuộc hàm.

Thuật toán:

*Bước 1.* Tìm phủ tối thiểu của  $F$

*Bước 2.* Nếu có một phụ thuộc hàm nào của  $F$  mà liên quan đến tất cả các thuộc tính của  $Q$  thì kết quả ra chính là  $Q$  (  $Q$  không thể phân rã)

*Bước 3.* Cứ mỗi phụ thuộc hàm  $X \rightarrow A \in F$  thì  $XA$  là một lược đồ cần tìm

*Ví dụ 7.27.* Cho lược đồ quan hệ  $Q(ABCDEFGH)$  với tập phụ thuộc hàm:

$$F=\{AB \rightarrow D; EH \rightarrow G; G \rightarrow C; D \rightarrow C\}$$

Theo thuật toán 7.9, kết quả phân rã thành lược đồ CSDL sau:

$$Q_1(ABD), F1 = \{AB \rightarrow D\}$$

$$Q_2(EHG), F2 = \{EH \rightarrow G\}$$

$$Q_3(GC), F3 = \{G \rightarrow C\}$$

$$Q_4(DC), F4 = \{D \rightarrow C\}$$

Kết quả phân rã này cho một lược đồ CSDL đạt 3NF và bảo toàn phụ thuộc hàm, tuy nhiên phép phân rã này không bảo toàn thông tin (sử dụng thuật toán 7.6 để kiểm tra)

### 3. Phân rã thành 3NF vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin

Chúng ta hoàn toàn có thể phân rã một lược đồ thành các lược đồ con đạt dạng chuẩn BC có nối không mất, và chúng ta cũng có thể phân rã một lược đồ thành 3NF bảo toàn phụ thuộc hàm. Liệu chúng ta có thể tìm được một phân rã thành 3NF mà có cả hai đặc tính là bảo toàn tập phụ thuộc và có tính kết nối không mất thông tin hay không? Chúng ta có thể làm được điều đó thông qua phương pháp rất đơn giản (mà hiệu quả ) sau đây:

Tìm một phân rã  $p$  của  $Q$  có dạng 3NF như vừa mới phân tích ở trên, và tìm một khoá  $X$  của  $Q$ , thì  $X \cup p$  là một phân rã của  $Q$  mà tất cả các lược đồ quan hệ đều có tính kết nối không mất và bảo toàn phụ thuộc, phân rã cuối cùng là  $X \cup p$ , nghĩa là  $X$  chỉ thêm vào  $p$  nếu  $X$  chưa có trong  $p$ .

#### ❖ Thuật toán 7.10:

*Dữ liệu vào:* Lược đồ quan hệ  $Q(U)$  và tập phụ thuộc hàm  $F$

*Dữ liệu ra:* Phân rã  $p=\{Q_1, Q_2, \dots, Q_k\}$  đạt 3NF bảo toàn phụ thuộc hàm và bảo toàn thông tin.

Thuật toán:

*Bước 1.* Tìm phủ tối thiểu của  $F$

*Bước 2.* Nếu có một phụ thuộc hàm nào của  $F$  mà liên quan đến tất cả các thuộc tính của  $Q$  thì kết quả ra chính là  $Q$  ( $Q$  không thể phân rã)

*Bước 3.* Cứ mỗi phụ thuộc hàm  $X \rightarrow A \in F$  thì  $XA$  là một lược đồ cần tìm

*Bước 4.* Nếu không có lược đồ quan hệ nào trong  $p$  chứa một khóa của  $Q$  thì hãy tạo ra thêm một lược đồ quan hệ trong  $p$  chứa các thuộc tính tạo nên một khóa của  $R$

*Ví dụ 7.28.* Sử dụng lược đồ cho trong ví dụ 7.27. Ta có lược đồ quan hệ  $Q(ABCDEFGH)$  với tập phụ thuộc hàm:

$$F=\{AB \rightarrow D; EH \rightarrow G; G \rightarrow C; D \rightarrow C\}$$

Lược đồ này có 1 khóa là  $ABEH$ , theo thuật toán 7.10, thêm vào kết quả phân rã trong ví dụ 7.27 một lược đồ  $Q_5(ABEH)$  ta được một lược đồ CSDL đạt 3NF vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin:

$$Q_1(ABD), F_1 = \{AB \rightarrow D\}$$

$$Q_2(EHG), F_2 = \{EH \rightarrow G\}$$

$$Q_3(GC), F_3 = \{G \rightarrow C\}$$

$$Q_4(DC), F_4 = \{D \rightarrow C\}$$

$$Q_5(ABEH)$$

## VII.4.2. Các phụ thuộc đa trị và dạng chuẩn 4

### VII.4.2.1. Phụ thuộc đa trị

Một số lược đồ quan hệ dạng chuẩn 3NF hoặc BCNF vẫn còn nhiều vấn đề cần phải giải quyết, trường hợp thường thấy nhất là mối quan hệ giữa các thuộc tính không chỉ là phụ thuộc hàm mà còn có nhiều phụ thuộc khác, chẳng hạn như phụ thuộc đa trị.

*Ví dụ 7.29.* Xét tập lược đồ quan hệ sau:

GIANGDAY(MONHOC, GIAOVIEN, TAILIEU).

Ý nghĩa của các thuộc tính trong lược đồ như sau:

- Mỗi môn học có thể có nhiều giáo viên giảng dạy
- Mỗi môn học có thể có nhiều tài liệu tham khảo
- Giáo viên giảng dạy và tài liệu tham khảo không phụ thuộc lẫn nhau

Ta có một thể hiện của lược đồ GIANGDAY như sau:

MONHOC	GIAOVIEN	TAILIEU
CSDL	Nga	TL001
CSDL	Nga	TL002
CSDL	Tuấn	TL001
CSDL	Tuấn	TL002

Khóa của lược đồ là tổ hợp 3 thuộc tính { MONHOC, GIAOVIEN, TAILIEU }, rõ ràng đây là quan hệ dạng chuẩn BCNF. Ta nhận thấy có sự lặp các giá trị một cách dư thừa trong các bộ, chẳng hạn các giá trị ‘TL001’, ‘TL002’ của thuộc tính TAILIEU được lặp lại với mỗi giá trị của thuộc tính GIAOVIEN (và các giá trị ‘Nga’, ‘Tuấn’ được lặp lại với mỗi giá trị của thuộc tính TAILIEU) và dĩ nhiên ta không mong muốn có sự dư thừa đó. Vì vậy, chúng ta phải định nghĩa một dạng chuẩn 4 mạnh hơn BCNF và ngăn cấm các lược đồ quan hệ như GIANGDAY.

Khái niệm phụ thuộc hàm không đủ mô tả mối quan hệ giữa các dữ liệu với nhau trong những trường hợp này, một giá trị của thuộc tính MONHOC không chỉ xác định một giá trị của thuộc tính GIAOVIEN mà xác định một tập các giá trị thuộc tính GIAOVIEN, mối quan hệ này được gọi là phụ thuộc đa trị. Phụ thuộc hàm là một trường hợp đặc biệt của phụ thuộc hàm đa trị.

#### ❖ Định nghĩa phụ thuộc đa trị

Giả thiết có một lược đồ quan hệ  $Q(U)$ ,  $X$  và  $Y$  là hai tập con của tập thuộc tính  $U$ . Một phụ thuộc đa trị (MVD), ký hiệu là  $X \twoheadrightarrow Y$  thỏa mãn trên  $Q$  nếu trong mọi thể hiện quan hệ  $R$  của lược đồ  $Q$ : nếu tồn tại 2 bộ bất kỳ  $r_1, r_2 \in R$  sao cho  $r_1[X] = r_2[X]$  thì khi đó cũng tồn tại 2 bộ  $s_1, s_2 \in R$  sao cho:

$$(1) s_1[X] = s_2[X] = r_1[X] = r_2[X]$$

$$(2) s_1[Y] = r_1[Y] \text{ và } s_1[U-X-Y] = r_2[U-X-Y]$$

$$(3) s_2[Y] = r_2[Y] \text{ và } s_2[U-X-Y] = r_1[U-X-Y]$$



Nghĩa là nếu  $r_1, r_2 \in R$ :

$$r_1 = (r_1[X], r_1[Y], r_1[U-X-Y])$$

$$r_2 = (r_2[X], r_2[Y], r_2[U-X-Y])$$

Sẽ tồn tại  $s_1, s_2 \in R$  sao cho:

$$s_1 = (r_2[X], r_1[Y], r_2[U-X-Y])$$

$$s_2 = (r_1[X], r_2[Y], r_1[U-X-Y])$$

Nghĩa là khi trao đổi các giá trị của  $Y$  hoặc của  $U-X-Y$  trong  $r_1, r_2$  thì được các bộ  $s_1, s_2$  cũng thuộc  $R$ .

Một phụ thuộc đa trị  $X \twoheadrightarrow Y$  được gọi phụ thuộc đa trị tầm thường nếu:

- $Y$  là một tập con của  $X$
- hoặc  $X \cup Y = R$

Một phụ thuộc đa trị không thỏa mãn một trong 2 điều kiện trên được gọi là một *phụ thuộc đa trị không tầm thường*. Nếu ta có một phụ thuộc đa trị không tầm thường trong một quan hệ, thì có thể phải lặp các giá trị một cách dư thừa trong các bộ.

Trong ví dụ 7.29,  $MONHOC \twoheadrightarrow GIAOVIEN$  và  $MONHOC \twoheadrightarrow TAILIEU$  là những phụ thuộc đa trị không tầm thường

#### VII.4.2.2. Dạng chuẩn 4 (4NF)

Một lược đồ quan hệ  $Q$  với một tập hợp các phụ thuộc  $F$  (gồm các phụ thuộc hàm và phụ thuộc đa trị) đạt dạng chuẩn 4 (4NF) nếu với mỗi phụ thuộc đa trị không tầm thường  $X \twoheadrightarrow Y$  trong  $F$  thì  $X$  là một siêu khóa của  $Q$ .

Như vậy, một lược đồ quan hệ không đạt 4NF nếu nó chứa các phụ thuộc hàm đa trị không tầm thường. Ví dụ, lược đồ quan hệ  $GIANGDAY$  ở ví dụ 7.28 không đạt 4NF bởi vì trong các phụ thuộc hàm đa trị không tầm thường  $MONHOC \twoheadrightarrow GIAOVIEN$  và  $MONHOC \twoheadrightarrow TAILIEU$ ,  $MONHOC$  không phải là một siêu khóa của  $GIANGDAY$ .

Giả sử ta tách thể hiện của lược đồ quan hệ  $GIANGDAY$  trong ví dụ 7.28 thành 2 quan hệ như sau:

<b>GIAOVIEN_GD</b>	<b>MONHOC</b>	<b>GIAOVIEN</b>
	CSDL	Nga
	CSDL	Nga

TAILIEU_GD	MONHOC	TAILIEU
	CSDL	TL001
	CSDL	TL002

Hai lược đồ GIAOVIEN\_GD và TAILIEU\_GD tương ứng với 2 quan hệ này đạt 4NF vì các phụ thuộc đa trị MONHOC  $\twoheadrightarrow$  GIAOVIEN và MONHOC  $\twoheadrightarrow$  TAILIEU là các phụ thuộc đa trị tầm thường. Trong hai lược đồ quan hệ này không có các phụ thuộc đa trị không tầm thường cũng như không có các phụ thuộc hàm.

### VII.4.3. Các phụ thuộc nối và dạng chuẩn 5

#### VII.4.3.1. Phụ thuộc nối

Khi nghiên cứu các tính chất của phép tách – kết nối các lược đồ quan hệ bảo toàn thông tin, Aho và Nicolas đã chứng minh rằng nếu chỉ dừng lại ở các khái niệm phụ thuộc hàm và phụ thuộc đa trị trong một số trường hợp vẫn chưa đủ giải quyết sự dư thừa và tách – kết nối không tổn thất thông tin. Vấn đề đặt ra là khi tách lược đồ quan hệ thành các lược đồ con sao cho khi kết nối lại với nhau thu được lược đồ quan hệ ban đầu, không làm sai lệch thông tin.

#### ❖ Định nghĩa phụ thuộc nối:

Cho lược đồ quan hệ  $Q(U)$  và  $R$  là một thể hiện bất kỳ của  $Q$ . Một phụ thuộc nối giữa các lược đồ  $Q_1, Q_2, \dots, Q_n$ , ký hiệu  $^*(Q_1, Q_2, \dots, Q_n)$  là một ràng buộc nối không mất trên mọi thể hiện  $R$  của  $Q$ , có nghĩa là:

$$^*(\pi_{Q_1}(R), \pi_{Q_2}(R), \dots, \pi_{Q_n}(R)) = R$$

#### VII.4.3.2. Dạng chuẩn 5 (5NF)

Một lược đồ quan hệ  $Q$  đạt dạng chuẩn 5 (5NF) khi và chỉ khi tất cả các phụ thuộc nối được thực hiện trên các khóa của lược đồ quan hệ.

Việc phát hiện các phụ thuộc nối trong các cơ sở dữ liệu thực tế với hàng trăm thuộc tính là một điều rất khó khăn. Vì vậy, thực tiễn thiết kế cơ sở dữ liệu hiện nay thường không chú ý đến nó.

Nói chung, trong thực tế thiết kế cơ sở dữ liệu, người ta chỉ chuẩn hóa các lược đồ đạt đến 3NF, BCNF là đủ.

## VII.5. Tổng kết chương và câu hỏi ôn tập

### VII.5.1. Tổng kết chương

Chương này đã trình bày các dạng chuẩn để đánh giá chất lượng của một lược đồ CSDL: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF cũng như các phương pháp để chuẩn hóa một lược đồ quan hệ từ dạng chuẩn thấp sang dạng chuẩn cao, cụ thể là phương pháp tách về dạng chuẩn 3NF và dạng chuẩn BCNF bảo toàn thông tin và bảo toàn phụ thuộc hàm. Các vấn đề này liên quan đến các phụ thuộc hàm, phụ thuộc đa trị, khóa, phủ tối thiểu.

Trong thực tế, tùy thuộc vào từng yêu cầu cụ thể mà ta thiết kế lược đồ ở dạng nào là phù hợp. Chẳng hạn, ta không quan tâm nhiều đến vấn đề về không gian lưu trữ (gây ra bởi sự trùng lặp thông tin) hay chi phí kiểm tra các ràng buộc toàn vẹn, vấn đề là làm sao có thể truy vấn dữ liệu nhanh nhất. Với yêu cầu này, lược đồ CSDL chỉ cần dừng lại ở dạng chuẩn 1.

### VII.5.2. Câu hỏi ôn tập

1. Phụ thuộc hàm là gì? Chúng ta có thể suy ra phụ thuộc hàm từ một thể hiện của một lược đồ quan hệ không?
2. Vì sao qui tắc suy diễn của Armstrong là quan trọng?
3. Bao đóng của tập phụ thuộc hàm là gì? Có dễ dàng tìm bao đóng của tập phụ thuộc hàm không? Vì sao?
4. Bài toán tìm bao đóng của tập phụ thuộc hàm chuyển thành bài toán đơn giản hơn là gì?
5. Phủ tối thiểu là gì?
6. Một lược đồ quan hệ chỉ có duy nhất một khóa, đúng hay sai?
7. Định nghĩa các dạng chuẩn: 1NF, 2NF, 3NF, BCNF?
8. Vì sao cần phải chuẩn hóa lược đồ CSDL?

### VII.5.3. Bài tập

1. Cho lược đồ quan hệ  $Q(ABCD)$ ,  $R$  là một thể hiện quan hệ trên  $Q$

A	B	C	D
a1	b1	c1	d1

a1	b2	c1	d1
a1	b3	c2	d1
a2	b2	c2	d2

Phụ thuộc hàm nào sau đây không thoả R?

- a.  $D \rightarrow A$ ;
- b.  $AC \rightarrow D$ ;
- c.  $CD \rightarrow A$ ;
- d.  $D \rightarrow B$ ;

2. Cho lược đồ quan hệ  $Q(ABCD)$  và tập phụ thuộc hàm

$$F = \{ A \rightarrow D; D \rightarrow A; AB \rightarrow C \}$$

- a. Tính  $AC^+$
- b. Chứng minh  $BD \rightarrow C \in F^+$

3. Cho lược đồ quan hệ  $Q$  và tập phụ thuộc hàm  $F$

$$F = \{ AB \rightarrow E; AG \rightarrow I; BE \rightarrow I; E \rightarrow G; GI \rightarrow H \}$$

Chứng minh rằng  $AB \rightarrow GH, AE \rightarrow IH$

4. Cho tập phụ thuộc hàm

$$F = \{ AB \rightarrow C; B \rightarrow D; CD \rightarrow E; CE \rightarrow GH; G \rightarrow A \}$$

Chứng minh rằng  $AB \rightarrow E; AB \rightarrow G$

5. Cho lược đồ quan hệ  $Q(ABCD)$  với tập phụ thuộc hàm  $F$

$$F = \{ AB \rightarrow C; D \rightarrow B; C \rightarrow AD \}$$

Tìm tất cả các khóa của lược đồ quan hệ.

6. Cho lược đồ quan hệ  $SINHVIEN$  với tập phụ thuộc hàm  $F$

$SINHVIEN(MASV, HOTEN, CMND, MONHOC, TINCHI, LANTHI, DIEMHI)$

$$F = \{ MASV \rightarrow HOTEN, CMND;$$

$$CMND \rightarrow HOTEN, MASV;$$

$$MONHOC \rightarrow TINCHI;$$

$$MASV, MONHOC, LANTHI \rightarrow DIEMHI \}$$

Tìm tất cả các khóa của lược đồ trên?

7. Cho lược đồ quan hệ  $Q(A,B,C,D)$  và tập phụ thuộc hàm

$$F = \{C \rightarrow A; A \rightarrow C; AD \rightarrow B; BC \rightarrow D; AB \rightarrow D; CD \rightarrow B\}$$

Hãy tìm phủ tối thiểu của  $F$ ?

8. Cho lược đồ quan hệ

DONHANG (SODH, NGAYDH, KHACHHANG, MAHH, SOLUONG)

$$F = \{SODH \rightarrow NGAYDH, KHACHHANG; \\ SODH, MAHH \rightarrow SOLUONG\}$$

Xác định dạng chuẩn cho lược đồ quan hệ DONHANG?

9. Xác định dạng chuẩn của các lược đồ sau:

a.  $Q(ABCD)$ , với  $F = \{CA \rightarrow D; A \rightarrow B\}$

b.  $Q(NGPMGV)$  với  $F = \{NGP \rightarrow M; M \rightarrow GV\}$

c.  $Q(SNDTX)$  với  $F = \{S \rightarrow N; S \rightarrow D; S \rightarrow T; S \rightarrow X\}$

d.  $Q(ABCD)$  với  $F = \{AB \rightarrow C; D \rightarrow B; C \rightarrow ABD\}$

10. Kiểm tra sự bảo toàn thông tin của phép phân rã sau:

$$Q(ABCDE), F = \{A \rightarrow C; B \rightarrow C; C \rightarrow D; DE \rightarrow C; CE \rightarrow A\}$$

$$p = (R_1(AD); R_2(AB); R_3(BE); R_4(CDE); R_5(AE))$$

11. Cho lược đồ quan hệ  $Q(ABCD)$ , tập phụ thuộc hàm  $F = \{A \rightarrow B; B \rightarrow C; A \rightarrow D; D \rightarrow C\}$  và một phép phân rã sau:

$$p = \{Q_1(A,B); Q_2(AC); Q_3(BD)\}$$

a.  $p$  có bảo toàn thông tin?

b.  $p$  có bảo toàn phụ thuộc hàm?

12. Cho  $Q(ABCDEG)$  và  $F = \{A \rightarrow D; A \rightarrow E; A \rightarrow G; B \rightarrow C; DEG \rightarrow B\}$

a. Hãy xác định dạng chuẩn của  $Q$ .

b. Nếu  $Q$  chưa đạt dạng chuẩn 3, hãy phân rã  $Q$  thành lược đồ cơ sở dữ liệu đạt tối thiểu dạng chuẩn 3?

c. Hãy xác định khoá và tập phụ thuộc hàm cho từng lược đồ quan hệ con.

13. Cho lược đồ quan hệ  $Q(SIDM)$

$$F = \{SI \rightarrow DM; SD \rightarrow M; D \rightarrow M\}$$

- a. Tìm bao đóng:  $D^+$  ;  $SD^+$  ;  $SI^+$
- b. Tìm tất cả các khoá của Q?
- c. Tìm phủ tối thiểu của F
- d. Xác định dạng chuẩn cao nhất của Q?
- e. Nếu Q chưa đạt dạng chuẩn 3, hãy phân rã Q thành lược đồ CSDL dạng chuẩn 3 vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin?
- f. Nếu Q chưa đạt dạng chuẩn BCNF, hãy phân rã Q thành lược đồ CSDL dạng BCNF
- g. Kiểm tra phép phân rã Q thành các lược đồ con (SID, SIM) có bảo toàn thông tin?
- h. Kiểm tra phép phân rã Q thành các lược đồ con (SID, SIM) có bảo toàn phụ thuộc hàm ?

14. Phân rã lược đồ thành dạng BC

- a. Q(SDIM) với  $F=\{SI \rightarrow D; SD \rightarrow M\}$
- b. Q(ABCD) với  $F=\{A \rightarrow B; B \rightarrow C; D \rightarrow B\}$
- c. Q(CSZ) với  $\{CS \rightarrow Z; Z \rightarrow C\}$