

MỤC LỤC.

MỤC LỤC.....	1
CHƯƠNG I: TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU	6
1. Một số khái niệm cơ bản.....	6
1.1. Cơ sở dữ liệu.	6
1.2. Ví dụ về CSDL.	7
1.3. Cấu trúc của một hệ CSDL.....	8
1.4. Lược đồ và thể hiện.....	9
1.5. Tính độc lập dữ liệu, chia sẻ dữ liệu.	9
1.6. Người sử dụng CSDL.....	9
1.7. Sự cần thiết của CSDL.	10
1.8. Lịch sử phát triển các mô hình dữ liệu.....	10
1.9. Hệ quản trị CSDL.....	11
2. Nội dung và kiến trúc của hệ quản trị cơ sở dữ liệu (DBMS).	13
2.1. Các yêu cầu của một DBMS.	13
2.2. Các thành phần của DBMS.	13
2.2.1. Các mức của DBMS.....	13
2.2.2. Các chương trình khai báo cấu trúc.....	13
2.2.3. Các chương trình ứng dụng.	14
2.2.4. Từ điển dữ liệu.	14
2.3. Các ngôn ngữ của DBMS.....	14
2.3.1. Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL). 14	
2.3.2. Ngôn ngữ thực thi dữ liệu (Data Manipulation Language - DML). 14	
3. Câu hỏi.	14
CHƯƠNG II: CÁC MÔ HÌNH DỮ LIỆU.....	16
1. Mô hình thực thể kết hợp(Entity Relationship).	16
1.1. Một số khái niệm.....	16
1.2. Sơ đồ thực thể kết hợp.....	18
1.3. Các ràng buộc trên loại mối kết hợp.	19
1.3.1. Mối kết hợp một – một.....	19
1.3.2. Mối kết hợp nhiều – một.	20
1.3.3. Mối kết hợp nhiều – nhiều.....	20
1.4. Tính chất hàm trong mô hình thực thể kết hợp.	20

1.5. Thiết kế mô hình thực thể kết hợp (Entity Relationship).....	21
2. Mô hình quan hệ.....	21
2.1. Miền, thuộc tính.	22
2.2. Quan hệ.....	23
2.3. Lược đồ quan hệ.	24
2.4. Thể hiện.	24
2.5. Khóa của quan hệ và lược đồ quan hệ.....	24
2.5.1. Khóa.....	24
2.5.2. Khóa tối thiểu.	25
2.6. Chuyển mô hình thực thể kết hợp sang mô hình quan hệ.....	26
2.7. Các phép tính trên CSDL quan hệ.....	27
2.7.1. Chèn thêm một bộ.	27
2.7.2. Loại bỏ một bộ.....	28
2.7.3. Thay đổi giá trị thuộc tính của một bộ.	28
3. Câu hỏi và bài tập.....	29
CHƯƠNG III: NGÔN NGỮ ĐẠI SỐ QUAN HỆ.....	31
1. Các phép toán tập hợp.....	31
1.1. Phép hội.	31
1.2. Phép giao(set – intersection operation).	32
1.3. Phép trừ (set difference operation).....	33
1.4. Phép tích descartes.	33
2. Các phép toán quan hệ.	34
2.1. Phép chọn (Selection).....	34
2.2. Phép chiếu (Projection).	35
2.3. Phép kết nối (join).	36
2.4. Phép chia (Division).....	37
3. Bài tập.....	38
CHƯƠNG IV: NGÔN NGỮ TRUY VẤN SQL	42
1. Các lệnh định nghĩa dữ liệu.....	42
1.1. Lệnh tạo bảng (CREATE TABLE).....	42
1.2. Lệnh xóa bảng (DROP TABLE).....	44
1.3. Lệnh sửa chữa cấu trúc bảng (ALTER TABLE).	44
2. Truy vấn dữ liệu.	44
2.1. Lệnh SELECT.	45
2.2. Câu truy vấn đơn giản.	45
2.3. Tên bí danh, sử dụng “ * ” và loại bỏ dòng trùng.	45

2.4.	Phép toán trên chuỗi.	47
2.5.	Biểu thức số học.	47
2.6.	Sắp xếp khi hiển thị các bộ.	47
2.7.	Phép toán tập hợp (UNION).	48
2.8.	Các hàm thư viện.	48
2.9.	Hàm gom nhóm GROUP BY.	49
2.10.	Câu truy vấn lồng (Truy vấn con).	50
3.	Cập nhật dữ liệu.	52
3.1.	Thêm dữ liệu (INSERT INTO).	52
3.2.	Lệnh xóa dữ liệu (DELETE).	52
3.3.	Cập nhật dữ liệu (UPDATE).	53
4.	Tóm tắt câu truy vấn trong SQL.	53
5.	Bài tập.	54
CHƯƠNG V : RÀNG BUỘC TOÀN VỆN		63
1.	Khái niệm về ràng buộc toàn vẹn (Integrity Constraints).	63
1.1.	Khái niệm.	63
1.2.	Mục đích.	63
2.	Các đặc trưng của RBTV.	64
2.1.	Bối cảnh.	64
2.2.	Biểu diễn RBTV.	64
2.3.	Bảng tầm ảnh hưởng.	64
3.	Phân loại RBTV.	64
3.1.	RBTV có bối cảnh là một quan hệ (RBTV về miền giá trị, RBTV liên bộ, liên thuộc tính).	65
3.1.1.	RBTV về miền giá trị.	65
3.1.2.	RBTV liên bộ.	65
3.1.3.	RBTV liên thuộc tính.	65
3.2.	Ràng buộc toàn vẹn có bối cảnh là nhiều quan hệ cơ sở.	66
3.2.1.	RBTV tham chiếu.	66
3.2.2.	RBTV liên bộ – liên quan hệ.	66
3.2.3.	RBTV liên thuộc tính – liên quan hệ.	67
3.2.4.	RBTV do thuộc tính tổng hợp.	67
4.	Bài tập.	68
CHƯƠNG VI: LÝ THUYẾT VỀ THIẾT KẾ CSDL QUAN HỆ		72
1.	Phụ thuộc hàm.	72
1.1.	Khái niệm phụ thuộc hàm.	72

1.2. Hệ tiên đề cho phụ thuộc hàm.....	72
1.2.1. Hệ tiên đề Armstrong.	73
1.2.2. Bao đóng của một tập thuộc tính.....	74
1.2.3. Khóa của lược đồ quan hệ.	75
2. Phép tách các lược đồ quan hệ.....	76
2.1. Phép tách có kết nối không mất mát thông tin.	77
2.2. Kiểm tra phép kết nối không mất mát thông tin.....	77
3. Chuẩn hoá lược đồ quan hệ.....	79
3.1. Các dạng chuẩn.....	80
3.1.1. Dạng chuẩn thứ nhất(First Normal Form – 1NF).	80
3.1.2. Dạng chuẩn thứ hai (Second Normal Form – 2NF).	80
3.1.3. Dạng chuẩn thứ ba (Third Normal Form – 3NF).	81
4. Bài tập.....	81
THỰC HÀNH	85
A. GIỚI THIỆU VỀ SQL SERVER 2000.....	85
1. Khái niệm cơ bản về RDBMS (Relation Database Manager System). 85	
2. Các thành phần của SQL server 2000.....	85
3. Đối tượng cơ sở dữ liệu.....	86
3.1. Cơ sở dữ liệu Master.	86
3.2. Cơ sở dữ liệu model.	86
3.3. Cơ sở dữ liệu msdb.....	86
3.4. Cơ sở dữ liệu Tempdb.	87
3.5. Cơ sở dữ liệu pubs.....	87
3.6. Cơ sở dữ liệu Northwind.	87
4. Các đối tượng của một CSDL.....	87
4.1. Bảng (table).	87
4.2. Chỉ mục indexs.....	88
4.3. Bẫy lỗi: (Triggers).	88
4.4. Ràng buộc constraints.	88
4.5. Diagram – Lược đồ quan hệ.....	88
4.6. Khung nhìn (view).....	89
4.7. Thủ tục nội – stored procedure.....	89
5. Kiểu dữ liệu (Data type).....	89
B. CÁC CÔNG CỤ CHÍNH CỦA SQL SERVER 2000.....	91
1. Trợ giúp trực tuyến Books Online.	91

2. Tiện ích mạng Client/Server .	92
3. Trình Enterprise Manager.	93
4. Trình Query Analyzer.	94
C. CÁC PHÁT BIỂU CƠ BẢN CỦA T-SQL.	95
1. Phát biểu SELECT với mệnh đề FROM.	96
2. Phát biểu SELECT có dùng mệnh đề WHERE.	96
3. Phát biểu SELECT với mệnh đề ORDER BY.	98
4. Phát biểu SELECT với GROUP BY.	98
5. Các hàm thông dụng trong SQL server 2000.	98
5.1. Các hàm trong phát biểu GROUP BY.	98
5.2. Các hàm xử lý chuỗi.	99
5.3. Các hàm xử lý thời gian.	100
5.4. Các hàm về toán học.	101
5.5. Các hàm về chuyển đổi.	101
6. Phát biểu SELECT với AS.	102
7. Phát biểu SELECT với TOP N.	102
8. Phát biểu SELECT với DISTINCT.	102
9. Nhập dữ liệu bằng phát biểu INSERT.	102
10. Phát biểu cập nhật UPDATE.	103
11. Phát biểu xóa DELETE.	104
D. TẠO VÀ SỬA ĐỔI CÁC ĐỐI TƯỢNG.	104
1. Dùng lệnh.	104
1.1. Tạo bảng (CREATE TABLE).	104
1.2. Tạo view (bảng ảo).	105
2. Dùng các công cụ tạo.	105
2.1. Tạo bảng bằng công cụ Enterprise Manager.	105
2.2. Xóa bảng: (drop table).	106
2.3. Tạo cơ sở dữ liệu mới – new database.	107
2.4. Tạo kịch bản – creating scripts.	108
2.5. Tạo lược đồ quan hệ – Diagram trong SQL server 2000.	110
2.6. Tạo View (bảng ảo).	111
2.7. Tạo Stored Procedure (SP).	113
E. BÀI TẬP THỰC HÀNH.	114
Tài liệu tham khảo.	123

CHƯƠNG I: TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

Mục tiêu:

- + Giới thiệu khái niệm chung về cơ sở dữ liệu.
- + Trình bày các mô hình dữ liệu thông dụng.
- + Nêu lên các khái niệm về hệ quản trị cơ sở dữ liệu.

Ngày nay, nhu cầu tích lũy và xử lý các dữ liệu nảy sinh trong mọi công việc, trong mọi hoạt động của con người. Một bài toán nhỏ cũng cần đến dữ liệu, nhưng không nhất thiết phải quản lý các dữ liệu này theo các phương pháp khoa học. Tuy nhiên khi bài toán có kích thước lớn hơn hẳn và lượng số liệu cần xử lý tăng nhanh thì tầm bao quát của con người bình thường khó có thể quản lý hết được. Chúng đòi hỏi được quản lý tốt không chỉ vì kích thước mà còn vì sự phức tạp của bản thân chúng. Chính vì vậy việc xây dựng một hệ thống tập tin có cấu trúc là rất cần thiết để đáp ứng nhu cầu xử lý thông tin ngày càng cao và đa dạng của người dùng.

1. Một số khái niệm cơ bản.

1.1. Cơ sở dữ liệu.

- Dữ liệu: Là những số liệu rời rạc như tên mặt hàng, đơn giá bán, trị giá bán của một hóa đơn bán hàng.
- Cơ sở dữ liệu là một tập hợp các dữ liệu có cấu trúc, có mối quan hệ với nhau được lưu trữ trên các thiết bị của máy tính nhằm thỏa mãn đồng thời nhiều người sử dụng một cách có chọn lọc. Cơ sở dữ liệu, viết tắt là CSDL, tiếng Anh là Database.
- Đặc tính của môi trường cơ sở dữ liệu.
 - ✓ Giảm sự trùng lặp thông tin xuống mức thấp nhất và do đó bảo đảm được tính nhất quán và toàn vẹn dữ liệu.
 - ✓ Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau.
 - ✓ Khả năng chia sẻ thông tin cho nhiều người sử dụng và nhiều ứng dụng khác.
 - ✓ Giải quyết tình huống cạnh tranh trong truy xuất dữ liệu.

- ✓ Giải quyết tốt tình huống bảo mật trong chia sẻ dữ liệu.
- ✓ Phải có biện pháp phục hồi dữ liệu nhanh chóng khi xảy ra sự cố.

1.2. Ví dụ về CSDL.

- Trong mỗi một trường đại học thì mỗi năm học có rất nhiều sinh viên, có những sinh viên mới vào, có những sinh viên sắp ra trường và những sinh viên đang học, do đó để có thể quản lý tốt những sinh viên này đòi hỏi phải có một cơ sở dữ liệu lưu trữ tất cả những thông tin liên quan đến sinh viên như: họ tên, địa chỉ, môn học, năm học, khoa,...

Các thông tin về sinh viên được thể hiện trong các bảng như sau:

Tất cả những thông tin liên quan đến sinh viên, các môn học, điểm thi,... được lưu trữ trong các bảng trên. Tùy theo mức độ cho phép, người sử dụng có thể truy nhập vào cơ sở dữ liệu trên để tìm kiếm, cập nhật, chỉnh sửa dữ liệu liên quan tới sinh viên chẳng hạn như:

- Đối với phòng đào tạo: thêm một sinh viên mới vào CSDL, in ra danh sách những sinh viên đạt điểm 10, nhập điểm môn học vào CSDL, sửa lỗi sai do đánh máy,...
- Đối với sinh viên có thể truy nhập vào CSDL để xem kết quả điểm thi, xem mã môn học, số tín chỉ,...

Như vậy, để thực hiện được những công việc trên thì một hệ CSDL được tổ chức và hoạt động như thế nào?. Để hiểu hơn về cách thức tổ chức, hoạt động của một hệ CSDL ta sẽ tìm hiểu về cấu trúc của một hệ CSDL.

1.3. Cấu trúc của một hệ CSDL.

Một hệ CSDL có cấu trúc như hình dưới đây:

Cấu trúc chuẩn của một hệ CSDL gồm 3 mức:

- Mức vật lý (mức trong): Là mức quan tâm đến cách thức tổ chức vật lý của dữ liệu được lưu trữ trên phần cứng như thế nào. Mức này mô tả cách dùng kỹ thuật của các byte ở cấp độ máy. Thường thì hệ quản trị CSDL sẽ đảm nhận việc này.
- Mức quan niệm (mức logic): Đây là một mức trung gian, mô tả dữ liệu gì được lưu trữ trong CSDL dưới dạng thực thể và mối kết hợp (khái niệm thực thể và mối kết hợp được định nghĩa ở chương 2).
- Mức ngoài (View) Mô tả cách thức người sử dụng và ứng dụng nhìn thấy dữ liệu. Đây là mức sát với người sử dụng nhất và từ mức ngoài này người sử dụng có thể thực hiện các thao tác trên dữ liệu thông qua các hệ thống máy tính.

1.4. Lược đồ và thể hiện.

- Khi thiết kế thì người ta quan tâm đến những hoạch định trên CSDL, đó chính là lược đồ CSDL. Nhưng khi sử dụng thì quan tâm đến dữ liệu tồn tại trong cơ sở dữ liệu, đó chính là thể hiện của CSDL. Lược đồ thường không thay đổi trong khi đó các thể hiện thường xuyên thay đổi.
- Lược đồ (scheme): Lược đồ là bộ khung hay cấu trúc của CSDL, nó thường bao gồm một số danh mục, chỉ tiêu hoặc một số kiểu của thực thể trong CSDL.
- Thể hiện (Instance): Thể hiện CSDL là dữ liệu hiện có trong CSDL.

1.5. Tính độc lập dữ liệu, chia sẻ dữ liệu.

- Độc lập dữ liệu mức vật lý: Là khả năng mà khi ta thay đổi lược đồ trong mà không làm thay đổi lược đồ mức quan niệm hay các chương trình ứng dụng.
- Độc lập dữ liệu mức logic: Là khả năng mà khi ta thay đổi lược đồ mức quan niệm mà không phải thay đổi các lược đồ ngoài hay chương trình ứng dụng của chúng.

1.6. Người sử dụng CSDL.

Với sự phát triển của đời sống xã hội như hiện nay thì việc cập nhật, tra cứu, tìm kiếm thông tin là rất cần thiết và không thể thiếu, và mỗi người

đều có những nhu cầu khai thác thông tin khác nhau. Do đó để dễ dàng quản lý cơ sở dữ liệu người sử dụng được phân loại như sau:

- Người sử dụng cuối (End user): Những người này sử dụng CSDL để tìm kiếm, tra cứu thông tin,...
- Người thiết kế CSDL là những người có trách nhiệm thiết kế CSDL.
- Người quản trị CSDL có trách nhiệm quản lý hệ CSDL, là người có quyền hạn cao nhất đối với CSDL và có nhiệm vụ:
 - Định nghĩa và quản lý lược đồ quan niệm.
 - Định nghĩa các ứng dụng về khung nhìn người sử dụng.
 - Quản lý, bật/tắt các chức năng của hệ quản trị CSDL.
 - Tái định dạng CSDL khi cần thiết.
 - Có trách nhiệm về tính an toàn và tin cậy.
 - Những phân tích viên/lập trình viên ứng dụng: là những người sử dụng CSDL để thiết kế và sử dụng các giao dịch, đóng gói cho những người dùng thường xuyên.

1.7. Sự cần thiết của CSDL.

Hiện nay trong mọi lĩnh vực đời sống xã hội, các tổ chức cá nhân, tập thể, các công ty, xí nghiệp, trường học,...đều cần phải có một CSDL vì:

- Dữ liệu tạo thành một tài sản không thể thiếu của một tổ chức.
- Giảm tính dư thừa, trùng lặp thông tin.
- Đảm bảo tính nhất quán của dữ liệu.
- Có khả năng chia sẻ thông tin giữa nhiều người dùng nhiều ứng dụng khác nhau.
- Nâng cao tính an toàn và toàn vẹn dữ liệu.

1.8. Lịch sử phát triển các mô hình dữ liệu.

- Mô hình quan hệ (Relational Model) được đưa ra năm 1970 bởi E.F Codd (IBM) là một hệ thương mại đầu tiên năm 1981-1982, hiện nay xuất hiện trong các sản phẩm thương mại như: ORACLE, SYBASE, INFORMIX, CA-INGRES.

- Mô hình mạng (Network Model) đầu tiên được thực hiện bởi Honeywell vào năm 1964 -1965, mô hình này được ứng dụng mạnh nhờ có hỗ trợ của CODASYL(CODASYL-1971) sau đó được thực hiện trong rất nhiều hệ thống IDMS, DMS, 1100, IMAGE, VAX-DBMS. Đây là loại mô hình dữ liệu dựa trên đồ thị trong đó các mẫu tin là các nút, mối quan hệ giữa các mẫu tin là các cạnh.
- Mô hình dữ liệu phân cấp (Hierarchical datamodel) được thực hiện thông qua sự liên kết giữa IBM và North American Rockwell khoảng năm 1965. Kết quả xuất hiện trong các chuỗi hệ thống ISM là mô hình phổ biến nhất. Hệ thống khác sử dụng mô hình này là System 2K. Đây là mô hình dữ liệu dựa trên khái niệm cây.
- Mô hình hướng đối tượng (Object- oriented) Data Model (s): Nhiều mô hình được đưa ra để thực hiện trong hệ CSDL điển hình là ngôn ngữ lập trình hướng đối tượng như C⁺⁺, ngoài ra còn có các hệ như: O2, ORION, IRIS. Đây là mô hình dữ liệu dựa trên phương pháp hướng đối tượng.
- Mô hình quan hệ đối tượng (Object - Relational Models): Mô hình này đang có khuynh hướng phổ biến nhất hiện nay và đang được thí nghiệm trong các hệ như: ILLUSTRRA và UNISQL.

1.9. Hệ quản trị CSDL.

Một hệ quản trị CSDL (Database Mananger system - DBMS) là một tập các chương trình cho phép người sử dụng tạo ra và duy trì cơ sở dữ liệu. Hệ Quản trị CSDL là một phần mềm hệ thống cho phép định nghĩa, xây dựng và xử lý dữ liệu tức là:

- ✓ Khai báo “bộ khung” dữ liệu cùng với các mô tả chi tiết dữ liệu.
 - ✓ Lưu giữ dữ liệu lên các thiết bị lưu trữ thứ cấp.
 - ✓ Xử lý truy vấn, cập nhật và phát sinh báo cáo.
- Các chức năng chính:
- ✓ Cho phép khai báo cấu trúc dữ liệu, khai báo các mối liên hệ của dữ liệu và các qui tắc áp đặt lên dữ liệu đó.
 - ✓ Cho phép người sử dụng có thể thêm, xóa, sửa dữ liệu trong CSDL.

- ✓ Cho phép những người khai thác CSDL (chuyên nghiệp hoặc không chuyên) sử dụng để truy vấn các thông tin cần thiết trong CSDL.
- ✓ Có biện pháp bảo mật dữ liệu và phân quyền khai thác CSDL.
- ✓ Có cơ chế giải quyết tranh chấp dữ liệu. Mỗi hệ quản trị cơ sở dữ liệu có thể cài đặt một cơ chế riêng để giải quyết vấn đề này.
- Một số biện pháp sau đây được sử dụng
 - ✓ Cấp quyền ưu tiên cho từng người sử dụng.
 - ✓ Đánh dấu yêu cầu truy xuất dữ liệu, phân chia thời gian, người nào có yêu cầu trước thì có quyền truy xuất dữ liệu trước.
 - ✓ Có cơ chế sao lưu và phục hồi dữ liệu khi có sự cố xảy ra.
 - ✓ Hệ quản trị phải cung cấp một giao diện tốt để sử dụng, dễ hiểu cho những người sử dụng không chuyên.
 - ✓ Đảm bảo tính độc lập giữa dữ liệu và chương trình.
- Cho đến nay có khá nhiều hệ quản trị cơ sở dữ liệu được đưa ra thị trường như: DB2, Sysbase, VisualFoxpro, Microsoft Access, SQL Server, Oracle,...
- Các thao tác chủ yếu của người sử dụng đối với CSDL:
 - ✓ Tìm kiếm dữ liệu theo một chỉ tiêu nào đó.

Ví dụ: Công ty A cần tuyển dụng nhân viên vào làm việc sẽ tìm kiếm những thông tin cá nhân như: Ngày sinh, địa chỉ, nơi ở của những sinh viên đạt loại khá trở lên của một trường đại học B để chọn vào làm việc.
 - ✓ Bổ sung dữ liệu vào CSDL.

Ví dụ: Cần nhập những thông tin liên quan đến một nhân viên mới vào làm việc cho một công ty.
 - ✓ Loại bỏ dữ liệu ra khỏi CSDL.

Ví dụ: Phòng đào tạo của một trường đại học cần loại bỏ tất cả những thông tin liên quan đến những sinh viên đã bỏ học.
 - ✓ Sửa chữa dữ liệu trong CSDL.

Ví dụ: Khi tên của một sinh viên không đúng (do lỗi đánh máy hoặc nhập sai) hoặc một nhân viên của một công ty thay đổi nơi ở thì cần phải cập nhật lại.

- Các bước hoạt động của một hệ CSDL:

- ✓ Người sử dụng đưa ra các yêu cầu truy nhập dưới dạng các câu lệnh của một ngôn ngữ thao tác dữ liệu nào đó, DBMS nhận lời yêu cầu, phân tích cú pháp và chuyển cho mức logic.
- ✓ Mức logic tiến hành các truy nhập cơ sở dữ liệu mức vật lý và trả lại kết quả.
- ✓ DBMS hiển thị kết quả cho người sử dụng.

2. Nội dung và kiến trúc của hệ quản trị cơ sở dữ liệu (DBMS).

2.1. Các yêu cầu của một DBMS.

- Khả năng đáp ứng cao, luôn sẵn sàng thực hiện các yêu cầu của người sử dụng.
- Độ tin cậy cao, đảm bảo được tính đúng đắn của dữ liệu, không mất mát thông tin, điều khiển truy xuất đồng thời của nhiều người dùng.
- Lưu lượng cao.
- Thời gian phản hồi thấp.
- Chu kỳ sống lâu, đối với các hệ thống phức tạp không dễ bị thay thế (được thiết kế sao cho chúng có thể dễ dàng mở rộng một khi nhu cầu của tổ chức thay đổi).
- Có thể mã hóa dữ liệu để đảm bảo sự an toàn thông tin vì hệ thống có thể có nhiều người dùng truy cập.

2.2. Các thành phần của DBMS.

2.2.1. Các mức của DBMS.

Một DBMS gồm có 3 mức:

- Mức chương trình khai báo cấu trúc và chương trình ứng dụng.
- Mức mô tả CSDL, thao tác trên CSDL và mô tả từ điển dữ liệu.
- Mức CSDL.

2.2.2. Các chương trình khai báo cấu trúc.

- Sử dụng ngôn ngữ DDL (ngôn ngữ định nghĩa dữ liệu: Data Definition Language) của DBMS để thực hiện khai báo cấu trúc.
- Người thiết kế và quản trị CSDL sẽ thực hiện công việc khai báo cấu trúc logic (khai báo dữ liệu, ràng buộc trên dữ liệu).

2.2.3. Các chương trình ứng dụng.

- Sử dụng ngôn ngữ thao tác dữ liệu để truy xuất, cập nhật, khai thác dữ liệu.
- Người dùng sẽ sử dụng ngôn ngữ thao tác trực tiếp CSDL.

2.2.4. Từ điển dữ liệu.

Là một CSDL mà DBMS dùng để lưu trữ các mô tả về lược đồ và các thông tin khác như: các quyết định thiết kế, các mô tả chương trình ứng dụng, thông tin người sử dụng và bảo mật và an toàn dữ liệu.

2.3. Các ngôn ngữ của DBMS.

2.3.1. Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL).

Được sử dụng bởi những người thiết kế CSDL để chỉ rõ lược đồ mức quan niệm của một CSDL.

- ✓ Trong nhiều DBMS, DDL cũng được dùng để định nghĩa lược đồ trong và lược đồ ngoài (khung nhìn).
- ✓ Một số DBMS tách riêng ngôn ngữ định nghĩa lưu trữ (storage definition language - SDL) và ngôn ngữ định nghĩa khung nhìn (View Definition Language) được dùng để định nghĩa lược đồ trong và lược đồ ngoài.

2.3.2. Ngôn ngữ thực thi dữ liệu (Data Manipulation Language - DML).

- Được dùng để chỉ rõ việc rút trích và cập nhật dữ liệu.
 - ✓ Các câu lệnh DML có thể được nhúng trong một ngôn ngữ lập trình tổng quát chẳng hạn như: COBOL, PL/1 hay PASCAL.
 - ✓ Hoặc các câu lệnh DML có thể được dùng trực tiếp (Query Language).

3. Câu hỏi.

1. Nêu khái niệm về cơ sở dữ liệu.

2. Nêu khái niệm về hệ quản trị cơ sở dữ liệu và phân biệt với các hệ thống lập trình khác.
3. Nêu cấu trúc của một hệ cơ sở dữ liệu và phân biệt các mức của nó.
4. Nêu các thao tác chủ yếu của người dùng đối với cơ sở dữ liệu, cho ví dụ.
5. Hãy cho 5 ví dụ về cơ sở dữ liệu mà em biết.

CHƯƠNG II: CÁC MÔ HÌNH DỮ LIỆU

Mục tiêu:

- ✦ Trình bày khái niệm về mô hình thực thể kết hợp và mô hình quan hệ.
- ✦ Sử dụng mô hình thực thể kết hợp và mô hình quan hệ lập mô hình một số cơ sở dữ liệu trong thực tế.
- ✦ Chuyển đổi mô hình thực thể kết hợp sang mô hình quan hệ.

Như chúng ta đã biết cơ sở dữ liệu là rất cần thiết cho mọi hoạt động của các tổ chức cá nhân, tập thể, là phần không thể thiếu trong mọi lĩnh vực. Do đó việc thiết kế CSDL là rất quan trọng đòi hỏi người thiết kế phải có trình độ quan sát, có kiến thức về CSDL và có cái nhìn tổng quan về thế giới xung quanh,... để thiết kế CSDL một cách hợp lý đáp ứng được những yêu cầu của một CSDL hoàn chỉnh và nhiều thành phần người sử dụng.

Như đã giới thiệu ở chương một, có nhiều mô hình dữ liệu ra đời nhằm đáp ứng cho việc thiết kế cơ sở dữ liệu, trong chương này chúng ta sẽ tìm hiểu hai mô hình dữ liệu thông dụng đó là mô hình thực thể kết hợp và mô hình quan hệ.

1. Mô hình thực thể kết hợp(Entity Relationship).

1.1. Một số khái niệm.

- Thực thể (Entity) Là một đối tượng hay sự vật của thế giới thực tồn tại cụ thể hay tồn tại quan niệm mà có thể phân biệt được với các đối tượng khác. Ví dụ: Mỗi con người là một thực thể, mỗi chiếc xe máy là một thực thể,...
- Tập thực thể (Entity set): Là một nhóm bao gồm tất cả các thực thể tương tự nhau tạo ra một tập thực thể. Ví dụ: tất cả những người trong cùng một cơ quan, tất cả những người có tóc đỏ, tất cả những người có xe gắn máy,...đều là các tập thực thể.
- Loại thực thể (entity type): là một tập hợp các thực thể có cùng thuộc tính nhưng có giá trị khác nhau đối với thuộc tính này. Ví dụ loại thực thể nhân viên (NHANVIEN) có các thuộc tính như: mã nhân viên (MANV),

tên nhân viên (TENNV), ngày sinh (NGSINH), địa chỉ (DCHI), lương (LUONG). Mỗi nhân viên sẽ có một MANV, một TENNV, DCHI,... khác nhau nhưng đều có cùng các thuộc tính như trên.

- Thuộc tính (Attributes): Các đặc tính của tập thực thể gọi là các thuộc tính. Mỗi thuộc tính của tập thực thể lấy giá trị trên một miền dành cho thuộc tính đó. Thường thì miền giá trị đối với mỗi thuộc tính là một tập số nguyên, tập các số thực hoặc chuỗi ký tự nhưng cũng không loại trừ các kiểu giá trị khác. Ví dụ: Thuộc tính giới tính (GIOITINH),... của một nhân viên có miền giá trị là một chuỗi ký tự có chiều dài tối đa là 4 ký tự,...
- Khóa: Mỗi thuộc tính hay một tập các thuộc tính dùng để xác định một cách duy nhất mỗi thực thể trong tập thực thể gọi là khóa đối với tập thực thể đó. Về nguyên tắc mỗi thực thể có một khóa, bởi vì mỗi thực thể đều có thể phân biệt được với thực thể khác. Nếu không chọn được một tập các thuộc tính có chứa một khóa cho một tập thực thể thì không thể phân biệt được thực thể này với thực thể kia trong tập thực thể đó, trong trường hợp này thì các số đếm thường được gán làm thuộc tính khóa.

Ví dụ: Một tập thực thể chỉ gồm các công dân Việt Nam có thể dùng số chứng minh thư (SOCM) làm khóa. Tuy nhiên nếu muốn xác định duy nhất các công dân trong nhiều quốc gia thì có thể có một số người có số có số chứng minh giống nhau. Vì vậy khóa của tập thực thể này sẽ bao gồm số chứng minh thư (SOCM) và mã quốc gia (MAQG).

- Phân loại thuộc tính
 - ✓ Thuộc tính đơn trị : Mỗi thực thể chỉ có một giá trị đối với các thuộc tính. Thuộc tính đơn không thể chia nhỏ được nữa. Ví dụ: thuộc tính MASV (mã sinh viên), PHAI (giới tính).
 - ✓ Thuộc tính đa trị: Là thuộc tính mà có nhiều giá trị cho một thực thể cụ thể. Ví dụ: thuộc tính BANGCAP của thực thể SINHVIEN có nhiều giá trị như: GIỎI, TRUNGBINH, KHA.
 - ✓ Thuộc tính đa hợp: Là thuộc tính có thể chia nhỏ thành nhiều thành phần con. Ví dụ: DCHI(SONHA, DUONG, QUAN, TP).
 - ✓ Giá trị null của thuộc tính: Được sử dụng để chỉ giá trị của thuộc tính của thực thể là không có giá trị tại một thuộc tính nào đó hoặc không biết giá trị của thuộc tính của một thực thể nào đó do có tồn tại nhưng thiếu sót hoặc không biết nó có tồn tại hay không.

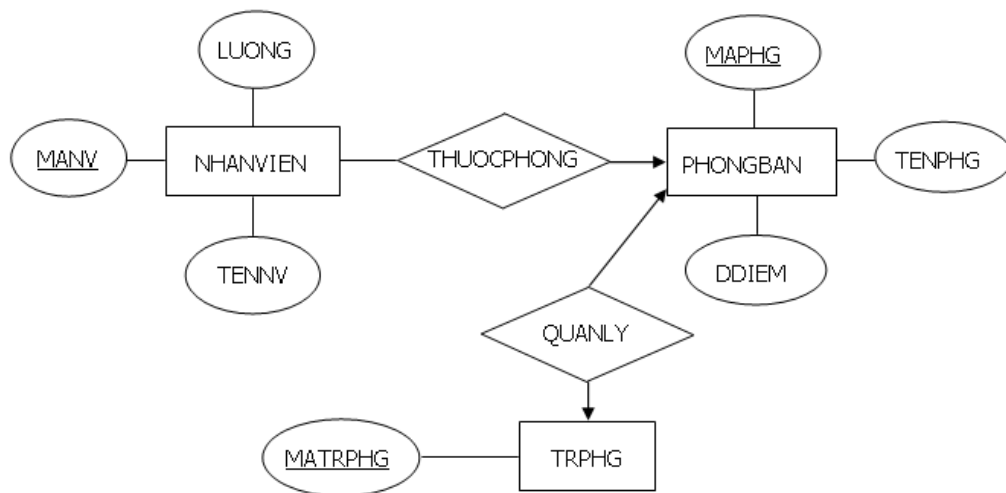
- **Mối kết hợp:** Một mối kết hợp (Relationship) là mối liên hệ giữa hai hay nhiều thực thể khác nhau mang một ý nghĩa nào đó. Ví dụ: nhân viên “Nguyễn Thanh Tùng” được phân công vào đề án “Sản phẩm z”. Hay nhân viên “Trần Hồng Quang” quản lý phòng ban “Điều hành”. Vậy ở đây có hai mối kết hợp là PHANCONG và QUANLY là mối liên hệ giữa “Nguyễn Thanh Tùng” và “Sản phẩm z”, giữa “Trần Hồng Quang” và phòng ban “Điều hành”.
- **Loại mối kết hợp (Relation type):** Những mối kết hợp có cùng loại trên các thực thể được nhóm lại hay phân loại thành một loại mối kết hợp. Ví dụ: Loại mối kết hợp PHANCONG trong đó có sự tham gia của NHANVIEN và đề án. Hay loại mối kết hợp QUANLY trong đó có sự tham gia của NHANVIEN và PHONGBAN.
- **Thuộc tính của mối kết hợp (Relationship attribute):**
 - ✓ Một loại mối kết hợp có thể có các thuộc tính để mô tả tính chất của mối kết hợp. Ví dụ: Nhân viên “Đinh Ba Tiến” tham gia vào đề án “Sản phẩm x” với thời gian là 32 giờ trong một tuần. Với THOIGIAN là thuộc tính của loại mối kết hợp.
 - ✓ Các thuộc tính này không thể gắn với các thực thể tham gia, nghĩa là nó chỉ có ý nghĩa trong ngữ cảnh của một mối kết hợp.
- **Lưu ý :** Mỗi kết hợp không có thuộc tính khoá chính. Việc xác định một mối kết hợp nào đó trong một tập mối kết hợp thông qua các khoá của thực thể tham gia.

1.2. Sơ đồ thực thể kết hợp.

Quy ước:

- Các hình chữ nhật biểu diễn các tập thực thể.
- Các vòng tròn biểu diễn các thuộc tính, chúng được liên kết với các tập thực thể bằng các cạnh (vô hướng). Đối với các thuộc tính là các thành phần của một khóa cho một tập thực thể thì sẽ được gạch dưới. Trường hợp đặc biệt nếu một tập thực thể chỉ có một thuộc tính thì có thể gọi tập thực thể đó bằng tên thuộc tính của tập. Khi đó tập thực thể sẽ là một vòng tròn chứ không phải là một hình chữ nhật và nó gắn kết với mối kết hợp mà tập đó hàm chứa.

- Các hình thoi biểu diễn các mối kết hợp, chúng được liên kết với các tập thành viên bởi các cạnh vô hướng hoặc có hướng.
- Ví dụ 1.1: Giả sử ta có ba tập thực thể NHANVIEN có các thuộc tính là: MANV, TENNV, PHONG; PHONGBAN có các thuộc tính MAPHG, TENPHG, DIADIEM; TRPHG có duy nhất một thuộc tính là MATRPHG. Ta có hai mối kết hợp liên kết các tập thực thể trên là THUOCPHONG và QUANLY.

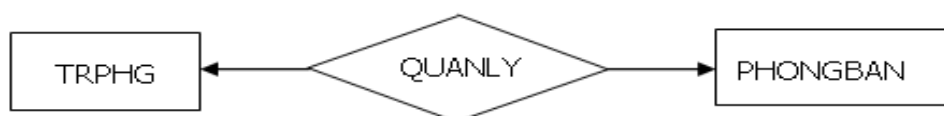


1.3. Các ràng buộc trên loại mối kết hợp.

Để mô hình hóa đầy đủ thế giới thực cần phải phân loại các mối kết hợp theo số lượng các thực thể từ mỗi tập tham gia vào trong mỗi kết hợp.

1.3.1. Mối kết hợp một – một.

- Một mối kết hợp một – một (one to one relationship) là một mối kết hợp mà với mỗi thực thể trong một tập thực thể này chỉ có nhiều nhất một phần tử được liên kết trong tập thực thể kia.
- Ví dụ 1.2: Mối kết hợp QUANLY giữa PHONGBAN và TRPHG trong ví dụ 1.1 ở trên được khai báo là mối kết hợp một – một.



1.3.2. Mỗi kết hợp nhiều – một.

- Một mối kết hợp giữa các tập thực thể E_1 và E_2 được gọi là mối kết hợp nhiều – một (many – one relationship) nếu mỗi thực thể trong tập thực thể E_2 có thể không liên kết với thực thể nào hoặc liên kết với một hay nhiều thực thể trong tập E_1 , nhưng mỗi thực thể trong tập thực thể E_1 chỉ liên kết nhiều nhất với một thực thể trong tập thực thể E_2 .
- Ví dụ: Mối kết hợp THUOCPHONG giữa NHANVIEN và PHONGBAN ở ví dụ trên là mối kết hợp nhiều – một có nghĩa là mỗi nhân viên chỉ làm việc trong một phòng, còn mỗi phòng thì có nhiều nhân viên.

**1.3.3. Mối kết hợp nhiều – nhiều.**

- Một mối kết hợp nhiều – nhiều (many – many relationship) là một mối kết hợp mà với mọi thực thể trong một tập thực thể này có thể không liên kết với thực thể nào hoặc liên kết với một hay nhiều thực thể trong tập thực thể kia.
- Ví dụ 1.3: Mối kết hợp CUNGCAP giữa tập thực thể NHACUNGCAP gồm các thuộc tính MANCC, TENNCC, DCHINCC và tập thực thể MATHANG gồm các thuộc tính MAMH, TENMH, MAUSAC, KHOILUONG là mối liên hệ nhiều – nhiều.

**1.4. Tính chất hàm trong mô hình thực thể kết hợp.**

- Sơ đồ thực thể kết hợp dùng các cung, đó là các cạnh có hướng chỉ ra bởi một mũi tên cho biết khi nào có mối kết hợp là nhiều – một hay một – một. Trong trường hợp mối kết hợp R là nhiều – một từ A đến B thì đặt một cung từ hình thoi R đến hình chữ nhật B. Chẳng hạn giả sử rằng mỗi nhân viên chỉ được phân công tối đa vào một phòng, điều đó giải thích mũi tên đi từ THUOCPHONG đến PHONGBAN như hình 2 ở trên.
- Tổng quát, nếu mối kết hợp R gồm nhiều hơn hai tập và thuộc loại nhiều – một vào một tập A nào đó thì ta sẽ vẽ một cung đi từ A đến R và các cạnh vô hướng đến các tập khác. Nếu R là mối kết hợp một – một giữa A

và B thì vẽ một mũi tên từ R đến cả A và B. Giả sử rằng, các trưởng phòng chỉ quản lý một phòng và mỗi phòng chỉ có một trưởng phòng. Điều đó giải thích cho các cung đi từ QUANLY đến PHONGBAN và TRPHG.

1.5. Thiết kế mô hình thực thể kết hợp (Entity Relationship).

- Mô hình thực thể kết hợp (Entity Relationship – ER) là mô hình ở mức quan niệm, việc thiết kế ER một cách thích hợp sẽ tạo điều kiện cho việc quản lý, truy xuất dữ liệu một cách chính xác, dễ dàng, đảm bảo đúng yêu cầu của người dùng. Thiết kế giản đồ là một việc làm lặp đi lặp lại và có thể cần có sự tinh chế.
 - ✓ Các thuộc tính của loại mối kết hợp (một – một) có thể được ghép vào một trong các thực thể tham gia.
 - ✓ Các thuộc tính của loại mối kết hợp (một – nhiều) hay (nhiều – một) có thể được ghép vào thực thể nhánh N.
 - ✓ Các thuộc tính của mối kết hợp (nhiều – nhiều) không thể được ghép chung.

2. Mô hình quan hệ.

Mô hình CSDL quan hệ là một mô hình được sử dụng rộng rãi trong đời sống xã hội của mọi tổ chức, cơ quan, xí nghiệp, doanh nghiệp,... nói chung là ở bất cứ cơ quan nào cần quản lý và xử lý các thông tin. Hãy xét một vài ví dụ minh họa.

- Ví dụ 2.1: Để thực hiện tốt công tác quản lý, bất cứ cơ quan nào cũng đều lưu trữ hồ sơ cán bộ dưới dạng sau:

- Ví dụ 2.2: Sổ theo dõi khách của một khách sạn có dạng như sau:

Trong hai ví dụ trên, tuy quản lý các thông tin có bản chất khác nhau, nhưng cả hai đều có chung một đặc thù là: dữ liệu được trình bày dưới dạng bảng. Mỗi bảng đều có dòng đầu chứa các tiêu đề cho các cột, các tiêu đề này được gọi là các thuộc tính. Chẳng hạn trong ví dụ 2.1 có các thuộc tính là: MS, TEN, NS, TRINHDO, QUEQUAN, GIOITINH, LUONG . Còn trong ví dụ 2.2 có các thuộc tính là: MK, DEN, DI, PHONG, TIEN.

Mỗi thuộc tính đều có một miền giá trị của nó, chẳng hạn thuộc tính NS có miền giá trị là các số nguyên: 1952, 1979, 1980, 1975,... Trong khi đó thuộc tính TEN lại có miền giá trị là các xâu ký tự như: An, Binh, Linh,...

Trong mỗi bảng đều có một số phần tử, mỗi phần tử là một dòng và mỗi bảng như vậy sau này được gọi là một quan hệ, mỗi dòng được gọi là một bộ.

2.1. Miền, thuộc tính.

- Thuộc tính là dữ liệu mô tả một đặc trưng của một thực thể. Miền là một tập hợp các giá trị nguyên tố (nguyên tố: không phân chia được). Mỗi miền có một tên, mô tả, kiểu dữ liệu và khuôn dạng.

- Ví dụ:

Tên miền	M_TENHV	M_SODT
Mô tả	Tập các tên của các học viên	Tập các số điện thoại trong nước
Kiểu dữ liệu	Chuỗi ký tự	Chuỗi ký tự
Khuôn dạng		(ddd)dddddd , d là ký số

2.2. Quan hệ.

Quan hệ được hiểu như là một tập con của tích Descartes của một hay nhiều miền. Như vậy về nguyên tắc mỗi quan hệ có thể xem là vô hạn nhưng ở đây ta luôn giả thiết quan hệ là hữu hạn.

- Mỗi dòng của quan hệ gọi là một bộ.
- Quan hệ là một tập con của tích Descartes $D_1 \times D_2 \times \dots \times D_n$ gọi là quan hệ n – ngôi. Mỗi bộ của quan hệ có n thành phần là các cột.
- Các cột của quan hệ gọi là các thuộc tính.

Định nghĩa:

Gọi $U = \{A_1, A_2, \dots, A_n\}$ là tập hữu các thuộc tính, mỗi thuộc tính A_i ($i=1, \dots, n$) có miền giá trị tương ứng là $\text{dom}(A_i)$. Người ta gọi r là quan hệ trên tập thuộc tính U nếu r là tập con của tích Descartes của n miền $\text{dom}(A_i)$: $r \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$.

- Chú ý: Quan hệ r có thể bị thay đổi theo thời gian do việc thực hiện các phép toán cập nhật trên các bộ của quan hệ (bổ sung, loại bỏ, sửa đổi, ...).
- Để chỉ một quan hệ r trên tập thuộc tính $\{A_1, A_2, \dots, A_n\}$ ta dùng ký hiệu $r(U)$ hay $r(A_1, A_2, \dots, A_n)$.
- Để mô tả quan hệ r gồm p bộ có n thuộc tính, người ta dùng một bảng gồm n cột và $p+1$ hàng, hàng thứ nhất là tên các thuộc tính, các hàng còn lại, mỗi hàng ứng với một bộ của quan hệ. Một quan hệ r được mô tả như bảng dưới đây:

- Ví dụ 2.3: xét lược đồ quan hệ LICHBAY(MACHUYEN, LOAI, SANBAYDI, SANBAYDEN, NGÀYBAY, GIOBAY, GIODEN) được thể hiện dưới bảng sau:

2.3. Lược đồ quan hệ.

Một lược đồ quan hệ (relation scheme) là một cặp có thứ tự $S = \langle U, F \rangle$. Trong đó U là tập hữu hạn các thuộc tính của quan hệ và F là tập các ràng buộc của quan hệ. Ở đây một ràng buộc trên tập các thuộc tính $\{A_1, A_2, \dots, A_n\}$ được hiểu là một tính chất trên tất cả các quan hệ xác định trên tập thuộc tính này.

- Chú ý:

- ✓ Để đơn giản khi nói tới một lược đồ quan hệ, khi mà không quan tâm tới các ràng buộc, ta ký hiệu một lược đồ quan hệ một cách đơn giản $r(U)$ hay $r(A_1, A_2, \dots, A_n)$ với $\{A_1, A_2, \dots, A_n\}$ là tập thuộc tính.
- ✓ Lược đồ quan hệ mô tả cấu trúc của quan hệ. Tại mỗi thời điểm, lược đồ quan hệ có một thể hiện quan hệ cụ thể.
- ✓ Trên mỗi lược đồ quan hệ có thể có nhiều quan hệ.

2.4. Thể hiện.

Một thể hiện (relation instance) của lược đồ quan hệ $S = \langle U, F \rangle$ là tập tất cả các bộ thỏa mãn tất cả các ràng buộc thuộc F .

2.5. Khóa của quan hệ và lược đồ quan hệ.

2.5.1. Khóa.

Định nghĩa:

- Cho r là một quan hệ xác định trên tập thuộc tính $U = \{A_1, A_2, \dots, A_n\}$, $K \subseteq U$, K được gọi là khóa của quan hệ r nếu: $\forall t_1, t_2 \in r$ thì $t_1[K] \neq t_2[K]$. ($t[K]$ là phép chiếu của t lên K). Hay nói cách khác: K là khóa của r nếu

và chỉ nếu không có hai bộ nào của r mà giá trị của chúng trên K là giống nhau.

- Ví dụ 2.4: Xét quan hệ r như sau:

Với $K_1=U$, $K_2=ABC$, $K_3=AB\dots$ là các khóa của r . Nhưng $X=BC$ không là khóa của r vì $t_2[X]=t_4[X]$ nhưng $t_2 \neq t_4$.

2.5.2. Khóa tối thiểu.

Định nghĩa:

- Cho r là quan hệ định nghĩa trên lược đồ quan hệ $\langle U, F \rangle$ với $U = \{A_1, A_2, \dots, A_n\}$, $K \subseteq U$. K được gọi là khóa tối thiểu của quan hệ r nếu và chỉ nếu:
 - ✓ K là khóa của r .
 - ✓ Bất kỳ tập con $K' \subset K$ nào đều không là khóa của r , nghĩa là K là khóa bé nhất hay khóa tối thiểu.
- Với ví dụ 2.4 ở phần trên thì $K_2=ABC$ không phải là khóa tối thiểu của r vì có $K_3 = AB \subseteq K_2$ cũng là khóa của r , $K_3 = AB$ là khóa tối thiểu.
- Ví dụ 2.5: cho quan hệ sinhvien như sau:

Trong quan hệ này thì Masv là khóa tối thiểu của quan hệ Sinhvien.

- Khóa của lược đồ quan hệ: Cho lược đồ quan hệ $S = \langle U, F \rangle$ với $U = \{A_1, A_2, \dots, A_n\}$, $K \subseteq U$, K được gọi là khóa tối thiểu của lược đồ quan hệ S

nếu K là khóa tối thiểu của mọi quan hệ r định nghĩa trên lược đồ quan hệ trên.

- Khóa chính: Trong số các khóa tối thiểu của một quan hệ chỉ có một khóa được chọn ra làm khóa chính, những khóa còn lại là những khóa dự bị hay khóa phụ.
- Khóa ngoại: Khóa ngoại không phải là khóa của quan hệ nhưng lại là khóa của một quan hệ khác.
- Ví dụ: Cho các quan hệ sau:

HOCVIEN (MA_HV,TEN_HV,DIACHI,LOP)

Có khóa chính là MA_HV, khóa ngoại là LOP

LOPHOC(MA_LOP,TEN_LOP,KHOA)

Có khóa là MA_LOP, khóa ngoại là KHOA.

KHOAHOC(MA_KHOA,TEN_KHOA)

Có khóa chính là MA_KHOA.

2.6. Chuyển mô hình thực thể kết hợp sang mô hình quan hệ.

Để chuyển đổi mô hình thực thể kết hợp sang mô hình quan hệ cần chuyển đổi các tập thực thể và các mối kết hợp có trong mô hình thực thể kết hợp thành các quan hệ.

- Một tập thực thể E có thể được biểu diễn bởi một quan hệ mà lược đồ quan hệ của nó chứa tất cả các thuộc tính của tập thực thể đó. Mỗi bộ của quan hệ biểu diễn một thực thể trong thể hiện hiện hành của E.

Ví dụ: Tập thực thể NHANVIEN trong ví dụ 1.1 hình 2 được biểu diễn bởi quan hệ NHANVIEN(MANV, TENNV, LUONG) trong đó MANV là khóa chính.

- Mỗi kết hợp R giữa các tập thực thể E_1, E_2, \dots, E_K được biểu diễn bởi một quan hệ có lược đồ quan hệ chứa các thuộc tính trong các khóa của mỗi tập E_1, E_2, \dots, E_K . Để tránh trùng tên có thể đặt lại tên cho các thuộc tính (nếu cần). Một bộ t trong quan hệ này biểu diễn cho một danh sách các thực thể e_1, e_2, \dots, e_k , trong đó e_i là một thực thể của tập thực thể E_i . Nghĩa là e_i là một thực thể duy nhất của tập thực thể E_i mà giá trị của thuộc tính khóa của E_i có trong thành phần của bộ t ở những thuộc tính này. Sự có mặt của bộ t trong quan hệ chỉ ra rằng danh sách các thực thể (e_1, e_2, \dots, e_k) là phần tử hiện hành của mối liên hệ R.

- Ví dụ: Hãy chuyển mô hình thực thể kết hợp ở ví dụ 1.1 sang mô hình quan hệ.

Trong mô hình thực thể kết hợp trên có 3 tập thực thể và 2 mối kết hợp. Dưới đây là các lược đồ quan hệ cho các tập thực thể, mỗi lược đồ xuất phát từ một tập thực thể cùng tên với quan hệ.

NHANVIEN(MANV, TENNV, LUONG).

PHONGBAN(MAPHG, TENPHG, DDIEM).

TRPHG(MATRPHG).

Trong các quan hệ trên, các thuộc tính của thực thể được lấy làm các thuộc tính của quan hệ. Trường hợp đặc biệt quan hệ TRPHG có một thuộc tính duy nhất. Các thuộc tính khóa chính được gạch dưới.

Bây giờ hãy xét đến các mối kết hợp, các mối kết hợp được thể hiện bằng các quan hệ sau:

QUANLY(MATRPHG, MAPHG).

THUOCPHONG(MANV, MAPHG).

2.7. Các phép tính trên CSDL quan hệ.

Các phép tính tác động lên một CSDL bao gồm: chèn, loại bỏ, thay đổi,...Trong mô hình CSDL quan hệ, các phép tính này được áp dụng cho từng bộ của các quan hệ được lưu trữ trên máy.

2.7.1. Chèn thêm một bộ.

- Phép chèn thêm một bộ vào quan hệ $r\{A_1, A_2, \dots, A_n\}$ như sau:

$INSERT(r, A_1=d_1, A_2=d_2, \dots, A_n=d_n)$

Hoặc có thể viết ngắn gọn : $INSERT(r, d_1, d_2, \dots, d_n)$

Trong đó: $A_i (i=1, \dots, n)$ là các thuộc tính và $d_i \in \text{dom}(A_i)$.

- Ví dụ 2.6: Cho quan hệ KHACHHANG như sau:

Thêm một bộ t_4 =(Nguyen Thi Hong, 5/12 Ly Thuong Kiet Q TB, DN) vào quan hệ KHACHHANG như sau:

INSERT (KHACHHANG, TEN_KH= 'Nguyen Thi Hong', DC_KH= '5/12 Ly Thuong Kiet Q TB', PH_KH='DN').

Lưu ý:

- Phép chèn có thể biểu diễn hình thức toán học như sau: $r:=r \cup t$.
- Mục đích của phép chèn là thêm một bộ vào một quan hệ, kết quả của phép tính này có thể gây ra một số sai sót vì những lý do như:
 - ✓ Bộ mới thêm vào không phù hợp với lược đồ cho trước.
 - ✓ Một giá trị của một thuộc tính nào đó nằm ngoài miền giá trị của thuộc tính đó.
 - ✓ Giá trị khóa của bộ mới có thể là giá trị đã có trong quan hệ đang lưu trữ.

2.7.2. Loại bỏ một bộ.

- Phép loại bỏ (DEL) là phép xóa một bộ ra khỏi một quan hệ cho trước. Giống như phép chèn, phép loại bỏ có dạng biểu diễn hình thức toán học là: $r = r - t$.
- Phép loại bỏ có dạng như sau: $DEL(r, A_1=d_1, A_2=d_2, \dots, A_n=d_n)$.
Hoặc có thể viết ngắn gọn : $DEL(r, d_1, d_2, \dots, d_n)$
- Ví dụ 2.7: Từ quan hệ KHACHHANG loại bỏ bộ t_2 .

$DEL(KHACHHANG, TEN_KH= 'Tran Thanh Nam', DC_KH= '20 Nguyen Trai Q1', PH_KH='TPHCM')$.

2.7.3. Thay đổi giá trị thuộc tính của một bộ.

- Khi cần điều chỉnh một số giá trị nào đó tại một số thuộc tính khi đó ta sẽ sử dụng phép thay đổi.
- Gọi tập $\{C_1, \dots, C_p\} \subseteq \{A_1, \dots, A_n\}$ là tập các thuộc tính mà tại đó các giá trị của bộ cần thay đổi. Ta có phép thay đổi như sau: $CH(r, A_1=d_1, A_2=d_2, \dots, A_n=d_n; C_1=e_1, C_2=e_2, \dots, C_p=e_p)$. Dạng biểu thức toán học của phép loại bỏ là:

$$r = r \setminus t \cup t'.$$

- Ví dụ 2.8: Trong ví dụ 2.6 để thay đổi địa chỉ của khách hàng Phan Van Hung ta làm như sau: CH(KHACHHANG, TEN_KH= 'Phan Van Hung', DC_KH='36 Le Thi Rieng Q5', PH_KH='BH'; DC_KH='61 Phan Dang Luu Q PN').

3. Câu hỏi và bài tập.

1. Nêu khái niệm về thực thể, tập thực thể, cho ví dụ.
2. Nêu khái niệm về khóa của tập thực thể, cho ví dụ.
3. Nêu các mối liên hệ giữa các tập thực thể, cho ví dụ.
4. Định nghĩa quan hệ, cho các ví dụ minh họa.
5. Nêu khái niệm về lược đồ quan hệ, thể hiện quan hệ, cho ví dụ.
6. Định nghĩa khóa, khóa tối thiểu của một quan hệ, cho ví dụ.
7. Nêu cách thức chuyển đổi mô hình thực thể kết hợp sang mô hình quan hệ.
8. Trình bày các phép toán trên cơ sở dữ liệu quan hệ, cho ví dụ minh họa.
9. Sử dụng mô hình thực thể kết hợp để mô tả dữ liệu cho hệ thống quản lý điểm của một lớp, sau đó vẽ sơ đồ thực thể kết hợp cho cơ sở dữ liệu đó.
10. Sử dụng mô hình thực thể kết hợp để mô tả dữ liệu cho hệ thống quản lý việc thuê truyện của một cửa hiệu cho thuê truyện, sau đó vẽ sơ đồ thực thể kết hợp cho cơ sở dữ liệu đó.
11. Sử dụng mô hình thực thể kết hợp để mô tả dữ liệu cho hệ thống quản lý việc mua bán vật tư của một cửa hàng vật tư và vẽ sơ đồ thực thể kết hợp cho cơ sở dữ liệu đó.
12. Sử dụng mô hình thực thể kết hợp để mô tả dữ liệu cho một hệ thống quản lý nhân sự của một phòng tổ chức, sau đó vẽ sơ đồ thực thể kết hợp cho cơ sở dữ liệu đó.
13. Hãy sử dụng mô hình thực thể kết hợp để mô tả dữ liệu cho hệ thống quản lý lương của một phòng tài vụ, sau đó vẽ sơ đồ thực thể kết hợp cho cơ sở dữ liệu đó.
14. Chuyển các mô hình thực thể kết hợp ở bài 9 đến bài 13 sang mô hình quan hệ.

CHƯƠNG III: NGÔN NGỮ ĐẠI SỐ QUAN HỆ

Mục tiêu:

- + Trình bày một số phép toán đại số quan hệ.
- + Vận dụng các phép toán đại số quan hệ để truy vấn dữ liệu.

Ngôn ngữ đại số quan hệ là cơ sở quan trọng của một ngôn ngữ bậc cao được sử dụng để thao tác trên các quan hệ. Ngôn ngữ này bao gồm hai nhóm phép toán: Các phép toán tập hợp (phép hội, phép giao, phép trừ, phép tích Descartes) và các phép toán trên quan hệ (Phép chọn, phép chiếu, phép kết nối, phép chia).

- Gọi r là quan hệ trên tập thuộc tính $\{A_1, A_2, \dots, A_n\}$. Ở đây luôn giả thiết rằng quan hệ r là tập hữu hạn các bộ.
- Toán hạng của đại số quan hệ là một quan hệ và kết quả của phép toán đại số quan hệ cũng là các quan hệ.
- Hai quan hệ được gọi khả hợp nếu chúng cùng bậc (có số thuộc tính bằng nhau) và thuộc tính thứ i của quan hệ này có miền giá trị bằng miền giá trị thuộc tính thứ i của quan hệ kia.

1. Các phép toán tập hợp.

1.1. Phép hội.

- Cho hai quan hệ r và s khả hợp, phép hội giữa hai quan hệ r và s (ký hiệu $r \cup s$) là tập tất cả các bộ thuộc r hoặc thuộc s .
- Phép hội được biểu diễn như sau:

$$r \cup s = \{t \mid t \in r \text{ hoặc } t \in s\}$$

- Ví dụ 1.1:

- Ví dụ 1.2: Cho hai quan hệ như sau:

Tìm những sinh viên học lớp 23THTH1 hoặc 23THTH2.

Quan hệ kết quả như sau:

1.2. Phép giao(set – intersection operation).

- Cho hai quan hệ r , s khả hợp, phép giao giữa hai quan hệ này ký hiệu $r \cap s$ là tập tất cả các bộ thuộc cả r và s .
- Phép giao được biểu diễn như sau: $r \cap s := \{ t / t \in r \text{ và } t \in s \}$.
- Ví dụ 1.3: Cho hai quan hệ r và s như ví dụ 1.1, giao của r và s là một quan hệ như sau:

- Ví dụ 1.4: Cho hai quan hệ 23THTH1 và DS_ĐOANVIEN như sau, tìm những sinh viên thuộc lớp 23THTH1 là đoàn viên.

Quan hệ kết quả như sau :

1.3. Phép trừ (set difference operation).

- Cho hai quan hệ r và s khả hợp, phép trừ giữa hai quan hệ r và s ký hiệu $r - s$ là tập tất cả các bộ thuộc r nhưng không thuộc s .
- Phép trừ được biểu diễn như sau: $r - s := \{ t \mid t \in r \text{ và } t \notin s \}$
- Ví dụ 1.5 : Cho hai quan hệ r và s như ví dụ 1.1, phép trừ giữa hai quan hệ r và s có kết quả như sau:

- Ví dụ 1.6 : Cho hai quan hệ 23THTH1 và DS_ĐOANVIEN như ví dụ 1.4, tìm những sinh viên thuộc lớp 23THTH1 chưa là đoàn viên.

Quan hệ kết quả như sau:

1.4. Phép tích Descartes.

- Gọi r là quan hệ xác định trên tập thuộc tính $\{A_1, A_2, \dots, A_n\}$ và s là quan hệ xác định trên tập thuộc tính $\{B_1, B_2, \dots, B_m\}$. Tích Descartes $r \times s$ của r và s là tập $(n+m)$ thuộc tính sao cho n thành phần đầu có dạng một bộ thuộc r và m thành phần sau có dạng của một bộ thuộc s .
- Nếu r có n_1 bộ giá trị và s có n_2 bộ giá trị thì $r \times s$ có $n_1 \times n_2$ bộ giá trị.

$r \times s = \{t \mid t \text{ có dạng } (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m) \text{ trong đó: } (a_1, a_2, \dots, a_n) \in r \text{ và } (b_1, b_2, \dots, b_m) \in s\}$.

- Ví dụ 1.7:

2. Các phép toán quan hệ.

2.1. Phép chọn (Selection).

- Phép chọn là phép tính để xây dựng một tập con các bộ của quan hệ đã cho thỏa mãn biểu thức P xác định. Biểu thức P được diễn tả bằng một tổ hợp các toán hạng, mỗi toán hạng là một phép so sánh đơn giản giữa hai biến là hai thuộc tính hoặc giữa một biến là một thuộc tính và một hằng, cho giá trị “đúng” hoặc “sai” đối với mỗi bộ đã cho khi kiểm tra riêng bộ ấy.
- Các phép toán trong biểu thức P gồm:
 - ✓ Các phép toán so sánh: $<, =, >, \geq, \leq, \neq$.
 - ✓ Các phép toán logic: \wedge (và), \vee (hoặc), \neg (không).
- Cho quan hệ r xác định trên tập thuộc tính $\{A_1, A_2, \dots, A_n\}$. Phép chọn trên quan hệ r với điều kiện chọn P là một quan hệ trên tập thuộc tính $\{A_1, A_2, \dots, A_n\}$, ký hiệu $\sigma_P(r)$.
- Phép chọn được định nghĩa như sau : $\sigma_P(r) := \{t \mid t \in r \wedge P(t) \text{ đúng}\}$. Với P(t) là giá trị của biểu thức P với bộ t.
- Ví dụ 2.1: Cho quan hệ r như sau:

- Ví dụ 2.2: Cho quan hệ NHANVIEN như sau:

Tìm những nhân viên thuộc phòng 4 và có lương > 30000.

Quan hệ kết quả như sau:

2.2. Phép chiếu (Projection).

Cho quan hệ r xác định trên tập thuộc tính $\{A_1, A_2, \dots, A_n\}$, $X \subseteq \{A_1, A_2, \dots, A_n\}$, phép chiếu trên quan hệ r với tập thuộc tính X là một quan hệ trên tập thuộc tính X , ký hiệu là: $\Pi_X(r)$ và được định nghĩa như sau: $\Pi_X(r) = \{t[X] \mid t \in r\}$

Trong đó $t[X]$ là giá trị của bộ t trên tập thuộc tính X .

- Ví dụ 2.3: Cho quan hệ r như sau:

- Ví dụ 2.4: Cho quan hệ NHANVIEN như ở ví dụ 2.2, cho biết tên và lương của các nhân viên.

Quan hệ kết quả như sau:

2.3. Phép kết nối (join).

- Gọi θ là một trong các phép so sánh $\{=, >, >=, <, <=, \neq\}$. Phép kết nối θ của quan hệ r đối với thuộc tính A và quan hệ s đối với thuộc tính B là những bộ t của tích Descartes $r \times s$ sao cho $t[A] \theta t[B]$. Phép kết được định nghĩa như sau:

$$r \underset{A \theta B}{\bowtie} s = \{t \mid t \in r \times s \text{ và } t[A] \theta t[B]\}.$$

- Khi phép so sánh θ là phép “=” thì phép kết nối đó được gọi là kết nối bằng. Trường hợp kết nối bằng đối với hai thuộc tính cùng tên là A và kết quả thu được loại bỏ đi một trong hai cột $R.A$ hoặc $S.A$ thì phép kết nối đó gọi là kết nối tự nhiên và sử dụng ký hiệu “*” thay cho \bowtie .
- Ví dụ 2.5: Cho hai quan hệ r, s và phép kết θ giữa r và s như sau:

- Ví dụ 2.6: Cho hai quan hệ r, s và phép kết tự nhiên giữa r và s như sau:

2.4. Phép chia (Division).

- Cho hai quan hệ r, s tương ứng trên các lược đồ R, S , phép chia giữa hai quan hệ ký hiệu $r \div s$ được thực hiện với điều kiện các thuộc tính trong S phải là một tập thuộc tính con của R nghĩa là $S \subseteq R$. Mục tiêu của phép chia này là đưa ra các bộ trong một quan hệ r sao cho khớp với tất cả các bộ trong quan hệ khác là s .

$$R = (A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_n)$$

$$S = (B_1, B_2, \dots, B_n)$$

- Kết quả của phép chia $r \div s$ là một quan hệ trên lược đồ $R - S = (A_1, A_2, \dots, A_m)$, kết quả này bao gồm một tập hợp các bộ lấy từ r được định nghĩa trên các thuộc tính $R - S$ sao cho khớp với việc kết hợp mọi bộ trong s .

$$r \div s := \{t \mid t \in \prod_{R-S}(r) \wedge \forall u \in s: tu \in r\}.$$

Nghĩa là $r \div s$ với các thuộc tính A_1, A_2, \dots, A_m là một tập hợp chứa tất cả các bộ t sao cho với mọi bộ u trong s thì có một bộ tu trong r .

- Phép chia được áp dụng cho các câu truy vấn có từ tất cả.

Ví dụ 2.7 Cho hai quan hệ r, s , phép chia giữa r và s như sau:

3. Bài tập.

1. Cho hai quan hệ r và s như sau:

- a. Tính $r \cup s$, $r \cap s$, $r - s$ và $s - r$
- b. Giả sử $X = \{A, B\}$, $Y = \{A, C, D\}$
Tính $\Pi_X(r)$, $\Pi_X(s)$, $\Pi_X(r \cup s)$, $\Pi_Y(r \cap s)$

c. Tính $\sigma_{(A=1) \vee (D=0)}(r)$, $\sigma_{(A=1) \vee (D=0)}(r \cup s)$

2. Cho 3 quan hệ

Tìm :

a. $r \cup s$, $r \cap s$, $r - s$.

b. $r \times u$, $\Pi_{A,B}(r)$, $\sigma_{(A=a1) \vee (E=e2)}(u)$.

c. $\sigma_{(A=a1) \vee (E=e2)}(\Pi_{A,E}(r \times u))$.

3. Cho 3 quan hệ

DS(sbd, hoten, ngaysinh, gioitinh, quequan)

SBD_PH(sbd, sophach)

DTM(sophach, diem)

Trong đó:

- sbd: Số báo danh.
- Hoten: Họ và tên sinh viên.
- Ngaysinh : Ngày sinh.

- Gioitinh : Giới tính.
- Quequan : Quê quán.
- Sophach : Số phách.
- Diem : Điểm thi môn học.

Hãy dùng các phép toán đại số quan hệ trả lời các câu hỏi sau :

- a. Cho biết họ và tên, ngày sinh, giới tính, quê quán, điểm thi của mỗi sinh viên.
- b. Cho biết họ và tên, ngày sinh, giới tính, điểm thi của những sinh viên có điểm thi > 5 .
- c. Cho biết họ và tên, ngày sinh, điểm thi của những sinh viên có giới tính là nữ và quê ở 'Đồng Nai'.
- d. Cho biết họ và tên, ngày sinh, điểm thi của những sinh viên có giới tính là nữ và quê ở 'Bình định' hoặc ở 'Hải phòng' và có điểm thi < 3 hoặc điểm thi > 8 .

4. Cho 3 quan hệ:

Mathang(Mamh, Tenmh, Mausac, Donvitinh)

Khachhang(Makh, Tenkh, Dchi, DT, Gioitinh)

Muaban(Mamh, Makh, Muaban, Ngaymb, Soluong, Dongia)

Trong đó :

- Mamh : Mã mặt hàng.
- Tenmh : Tên mặt hàng.
- Mausac : Màu sắc.
- DVT : Đơn vị tính.
- Makh : Mã khách hàng.
- Tenkh : Tên khách hàng.
- Dchi : Địa chỉ khách hàng.
- DT : Điện thoại.
- Gioitinh : Giới tính.
- Muaban : Mua bán trong đó mua thì ghi là F, bán thì ghi là T.
- Ngaymb : Ngày mua hoặc bán.
- Soluong : Số lượng.
- Dongia : Đơn giá.

Hãy dùng các phép toán đại số quan hệ trả lời các câu hỏi sau:

- a. Cho biết Makh, Tenkh, Tenmh của những khách hàng đã bán mặt hàng có Mamh = 'MH001' hoặc Mamh = 'MH002'.
- b. Cho biết Makh của những khách hàng đã bán mặt hàng màu đỏ với số lượng > 100 .
- c. Cho biết Tenkh, Dchi, DT của những khách hàng đã bán mặt hàng màu 'vàng' hoặc 'xanh' với số lượng > 200 .
- d. Cho biết Tenkh, Tenmh của những khách hàng có giới tính là 'Nam' đã bán mặt hàng màu 'đỏ' và mua mặt hàng màu 'xanh' với số lượng $100 < \text{Soluong} < 200$.
- e. Cho biết Makh chưa tham gia mua bán lần nào.

CHƯƠNG IV: NGÔN NGỮ TRUY VẤN SQL

Mục tiêu:

- + Giới thiệu các lệnh cơ bản, các hàm và một số phép toán được dùng trong ngôn ngữ truy vấn SQL chuẩn.
- + Sử dụng ngôn ngữ SQL để truy vấn dữ liệu.
- Ngôn ngữ truy vấn SQL (Structured Query Language) là ngôn ngữ CSDL chuẩn được phát triển từ IBM (Sequel, 70) với phiên bản chuẩn được chấp nhận bởi ANSI/ISO (SQL- 86 hay SQL1). Phiên bản hiện nay là SQL3.
- Ngôn ngữ SQL bao gồm các phần như: Ngôn ngữ định nghĩa dữ liệu (DDL), ngôn ngữ thực thi dữ liệu tương tác (DML), DML nhúng, định nghĩa khung nhìn, phân quyền và bảo mật, ràng buộc toàn vẹn, điều khiển giao tác.
- SQL là ngôn ngữ phi thủ tục, chuẩn mực và điển hình. Do vậy hiện nay rất nhiều sản phẩm phần mềm đều được cài đặt SQL như ORACLE, DB2,...
- Phép toán cơ bản trong SQL là phép ánh xạ được miêu tả như một khối SELECT-FROM-WHERE.
- SQL thường sử dụng thuật ngữ: Bảng thay thế quan hệ, cột thay cho thuộc tính, dòng thay cho bộ.

1. Các lệnh định nghĩa dữ liệu.

1.1. Lệnh tạo bảng (CREATE TABLE).

- Sử dụng để tạo một bảng mới.
- Để định nghĩa một bảng cần chỉ ra:
 - + Tên bảng.
 - + Các thuộc tính.
 - + Kiểu dữ liệu.
 - + Các ràng buộc trên thuộc tính (NULL, NOT NULL, khoá chính, khóa ngoại,...).
- Cú pháp:

```
CREATE TABLE <tên bảng>
(
    <tên cột 1> <kiểu dữ liệu> [not null],
    <tên cột 2> <kiểu dữ liệu> [not null],
    ...
);
```

Trong đó:

- <tên bảng>, <tên cột> là xâu ký tự bất kỳ bắt đầu bằng một chữ cái sau đó là chữ cái hoặc chữ số, không chứa ký hiệu trống, không trùng với từ khóa. Trong một bảng tên cột là duy nhất. Thứ tự của cột trong bảng là không quan trọng.
- <kiểu dữ liệu>: tùy từng ngôn ngữ cụ thể sẽ có những kiểu dữ liệu nhất định.
- Đối với các ràng buộc về khóa chính, khóa ngoại,... có thể được chỉ ra trong CREATE TABLE hoặc có thể bổ sung sau bằng lệnh ALTER TABLE.

- Ví dụ 1: **CREATE TABLE** PHONGBAN

```
(
    TENPHG VARCHAR(10) NOT NULL,
    MAPHG INTEGER NOT NULL,
    MA_NQL CHAR(9),
    NG_NHANCHUC DATE TIME,
    PRIMARY KEY (MAPHG),
    FOREIGN      KEY(MA_NQL)      REFERENCES
NHANVIEN(MANV)
).
```

- Ý nghĩa của các ràng buộc trong tạo bảng:

- ✓ NOT NULL là không cho phép có giá trị NULL.
- ✓ Khóa chính: PRIMARY KEY<thuộc tính>: Ràng buộc toàn vẹn thuộc tính, khóa chính phải luôn khác NULL.
- ✓ Tính duy nhất: UNIQUE <thuộc tính>: Ràng buộc toàn vẹn thuộc tính, ràng buộc UNIQUE có thể có giá trị NULL.

- ✓ Kiểm tra: CHECK <điều kiện>, với <điều kiện> là một điều kiện đơn giản, không chứa câu truy vấn hay tham chiếu tới quan hệ khác.
- ✓ Ràng buộc khóa ngoại (FOREIGN KEY): khóa ngoại phải tham chiếu đến khóa chính trong quan hệ được tham chiếu. Ràng buộc khóa ngoại được biểu diễn như sau:

FOREIGN KEY <thuộc tính> REFERENCE <quan hệ> [thuộc tính].

1.2. Lệnh xóa bảng (DROP TABLE).

- Khi lệnh này thực thi thì tất cả các thông tin về bảng bao gồm cấu trúc và dữ liệu đang có trong bảng đều được xóa.
- Cú pháp: DROP TABLE <tên bảng>
- Ví dụ: DROP TABLE NHANVIEN

DROP TABLE PHONGBAN

1.3. Lệnh sửa chữa cấu trúc bảng (ALTER TABLE).

Lệnh này khi thực thi sẽ thay đổi cấu trúc và ràng buộc trên bảng bao gồm:

- Thêm cột: ALTER TABLE <tên bảng> ADD <thuộc tính> <kiểu dữ liệu>.

Ví dụ: ALTER TABLE NHANVIEN ADD NG_NGHIEP VARCHAR(12)

- Xóa cột: ALTER TABLE <tên bảng> DROP <thuộc tính>.

Ví dụ: ALTER TABLE NHANVIEN DROP DCHI

- Có thể dùng ALTER TABLE để bổ sung, thay đổi hoặc loại bỏ các ràng buộc toàn vẹn.

ALTER TABLE <tên bảng> DROP/ADD CONSTRAINT fk_<tên bảng>.

2. Truy vấn dữ liệu.

- Một trong những cấu trúc tiêu biểu nhất trong ngôn ngữ SQL là khối SELECT được mô tả về cú pháp như một khối SELECT – FROM – WHERE.

- Một cách tổng quát, khối SELECT bao gồm ba mệnh đề chính:

SELECT: Xác định nội dung của các cột cần đưa ra kết quả.

FROM: Xác định các bảng cần lấy thông tin ra.

WHERE: Xác định các bản ghi thỏa yêu cầu chọn lọc để đưa ra kết quả.

Ngoài ra, để mở rộng thêm khả năng của ngôn ngữ khối SELECT – FROM – WHERE còn được bổ sung thêm các mệnh đề GROUP BY, HAVING, ORDER BY, các hàm mẫu, ... sẽ được trình bày trong các phần sau.

2.1. Lệnh SELECT.

Đây là một lệnh cơ bản để rút trích thông tin từ CSDL và không hoàn toàn giống với phép chọn trong đại số quan hệ.

- ✓ SQL cho phép một bảng (quan hệ) có hai hay nhiều bộ có giá trị giống nhau trên mọi thuộc tính hay gọi cách khác là các bộ trùng nhau.
- ✓ Quan hệ bảng trong SQL là đa bộ chứ không phải tập hợp các bộ nên có thể có nhiều bộ trùng nhau.

2.2. Câu truy vấn đơn giản.

- Truy vấn rút trích dữ liệu không điều kiện từ một bảng

```
SELECT <danh sách các thuộc tính>  
FROM <tên bảng>
```

Ví dụ : **SELECT** MANV,TENNV,DCHI
 FROM NHANVIEN

- Truy vấn rút trích dữ liệu có điều kiện từ một bảng

```
SELECT <danh sách các thuộc tính>  
FROM <tên bảng>  
WHERE <điều kiện>
```

Ví dụ : **SELECT** MANV,TENNV,DCHI
 FROM NHANVIEN
 WHERE LUONG > 30.000

2.3. Tên bí danh, sử dụng “ * ” và loại bỏ dòng trùng.

- Tên bí danh Alias (AS).

- ✓ Khi một câu truy vấn tham chiếu tới hai hay nhiều thuộc tính cùng tên thì phải nêu ra rõ ràng bằng cách dùng tên quan hệ trước tên thuộc tính.

Ví dụ: NHANVIEN.MANV.

- ✓ Nếu câu truy vấn cần tham chiếu tới cùng một quan hệ hai lần thì dùng tên bí danh cho quan hệ.
- ✓ SQL cho phép đổi tên quan hệ và tên thuộc tính được thực hiện bằng mệnh đề AS như sau: <tên cũ> AS <tên mới>.

Ví dụ: Tìm tên nhân viên làm việc ở phòng 4.

```
SELECT TENNV AS NV_P4  
FROM NHANVIEN  
WHERE MAPHG = 4
```

Ví dụ: Với mỗi nhân viên, tìm tên nhân viên và tên người quản lý

```
SELECT A.TENNV, B.TENNV  
FROM NHANVIEN AS A, NHANVIEN AS B  
WHERE A.MA_NQL=B.TENNV
```

- Sử dụng '*': Để lấy tất cả các thuộc tính của quan hệ ta không cần phải liệt kê ra hết mà chỉ cần sử dụng ký tự * ở câu lệnh SELECT.

Ví dụ : cho biết tất cả các thông tin về nhân viên ở phòng số 5

```
SELECT *  
FROM NHANVIEN  
WHERE MAPHG=5
```

- Sử dụng DISTINCT để loại bỏ dòng trùng:

SQL cho phép các bộ trùng nhau trong kết quả của câu truy vấn vì các lý do sau:

- ✓ Thao tác loại bỏ các dòng trùng nhau có chi phí thời gian đáng kể.
- ✓ Người sử dụng đôi lúc muốn thấy những bộ trùng nhau trong quan hệ kết quả.
- ✓ Khi áp dụng các hàm kết hợp (aggregation function) tự động loại bỏ các dòng trùng có thể cho kết quả sai.
- ✓ Để loại bỏ các dòng trùng ta thêm từ khóa DISTINCT sau SELECT.

Ví dụ: Tìm tất cả các lương của nhân viên (có loại bỏ dòng trùng)

```
SELECT DISTINCT LUONG
FROM NHANVIEN
```

2.4. Phép toán trên chuỗi.

Dùng LIKE để trích chuỗi cần tìm kiếm dựa trên một số ký tự đã biết.

Cú pháp:

```
LIKE "<chuỗi ký tự>%" hoặc
LIKE "%<chuỗi ký tự>" hoặc
LIKE "%<chuỗi ký tự>%"
```

Ví dụ : Cho biết danh sách những nhân viên có địa chỉ ở TPHCM.

```
SELECT    MANV, TENNV
FROM      NHANVIEN.
WHERE     DCHI LIKE '%TPHCM%'
```

2.5. Biểu thức số học.

Sau lệnh select có thể chứa các biểu thức sử dụng các phép tính số học như +, -, *, /

Ví dụ : Tìm mã nhân viên, tên nhân viên và lương tăng thêm 10% của các nhân viên được phân công vào đề án sản phẩm X.

```
SELECT    MANV, TENNV, LUONG*1.1
FROM      NHANVIEN, DEAN, PHANCONG
WHERE     MANV = MANVIEN AND
          SODAN = MADA AND
          TENDA = 'sản phẩm X'
```

2.6. Sắp xếp khi hiển thị các bộ.

- Mệnh đề ORDER BY được dùng để sắp xếp các bộ trong kết quả câu truy vấn dựa trên giá trị các thuộc tính. Có 2 cách sắp xếp:
 - ✓ ASC : Sắp xếp theo thứ tự tăng dần (mặc định).
 - ✓ DESC: Sắp xếp theo thứ tự giảm dần.
- Nếu giá trị của các thuộc tính là các ký tự dạng “chữ” thì sẽ sắp xếp theo thứ tự alphabet với ASC và ngược lại với DESC. Nếu giá trị các thuộc tính là các ký tự dạng số thì sẽ sắp xếp theo thứ tự tăng từ nhỏ đến lớn với ASC hoặc giảm từ lớn đến nhỏ với DESC.

- Ví dụ: Liệt kê tên nhân viên (TENNV), lương của mỗi nhân viên (LUONG) trong đó lương sắp xếp theo thứ tự giảm dần.

```
SELECT TENNV , LUONG
FROM NHANVIEN
ORDER BY LUONG DESC
```

2.7. Phép toán tập hợp (UNION).

- Các phép toán tập hợp trong SQL được thực hiện trên các quan hệ khả hợp tương ứng với các phép toán ĐSQH.
- Khi thực hiện các phép toán này SQL sẽ tự động loại bỏ các dòng trùng.
- Ví dụ: Cho biết danh sách những đề án (TENDA) và nhân viên tham gia đề án có họ là “đình” hoặc những đề án (TENDA) mà người trưởng phòng chủ trì đề án có họ là “đình”.

```
(SELECT TENDA
FROM DEAN , PHANCONG , NHANVIEN
WHERE MADA = SODA AND
MANV = MA NVIEN AND
HONV = “đình”)

UNION
(SELECT TENDA
FROM DEAN , PHONGBAN , NHANVIEN
WHERE PHONG = MAPHG AND
TRPHG = MANV AND HONV = “đình”)
```

2.8. Các hàm thư viện.

- Các hàm thư viện bao gồm:
 - ✓ AVG: Tính giá trị trung bình.
 - ✓ MIN: Tính giá trị nhỏ nhất.
 - ✓ MAX: Tính giá trị lớn nhất.
 - ✓ SUM: Giá trị tổng.
 - ✓ COUNT: Đếm các bộ.
- Các hàm thư viện thao tác trên tập giá trị và trả về một trị và không tự loại bỏ dòng trùng.

- Ví dụ: Tìm tổng lương, lương lớn nhất, lương ít nhất và lương trung bình của các nhân viên phòng “Nghiên cứu”.

```
SELECT SUM (LUONG), MAX (LUONG), MIN(LUONG)  
AVG (LUONG)  
FROM NHANVIEN , PHONGBAN  
WHERE PHONG = MAPHG AND TENPHG = ‘nghiên cứu’
```

- Ví dụ: Cho biết có bao nhiêu giá trị lương nhân viên khác nhau.

```
SELECT COUNT (DISTNCT LUONG)  
FROM NHANVIEN
```

2.9. Hàm gom nhóm GROUP BY.

- Hàm gom nhóm được áp dụng trên các nhóm bộ có cùng thuộc tính trong quan hệ, mỗi nhóm bộ bao gồm tập hợp các bộ có cùng giá trị trên các thuộc tính gom nhóm.
- Lưu ý: Sử dụng mệnh đề GROUP BY để chỉ ra các thuộc tính gom nhóm, các thuộc tính trong mệnh đề SELECT nằm ngoài một hàm thư viện thì phải xuất hiện trong mệnh đề GROUP BY.

- Cú pháp trong SQL

```
SELECT < (các) thuộc tính [ các hàm kết hợp]>  
FROM < bảng 1 > , < bảng 2 > , ...  
WHERE < điều kiện chọn >  
GROUP BY < ( các ) thuộc tính gom nhóm >  
[ HAVING < điều kiện trên nhóm > ]
```

- Mệnh đề HAVING lấy các giá trị của hàm gom nhóm chỉ trên những nhóm nào thỏa mãn điều kiện nhất định.
- Mệnh đề HAVING dùng để chỉ điều kiện lọc trên các nhóm không phải điều kiện lọc trên từng bộ.
- Điều kiện trong mệnh đề HAVING được thực hiện sau khi gom nhóm các bộ.
- SQL xử lý câu truy vấn chứa mệnh đề GROUP BY và HAVING theo cách như sau:

(1) Chọn tất cả các dòng thỏa mãn điều kiện được chỉ ra trong mệnh đề WHERE.

- (2) Từ các dòng này hình thành các nhóm tương ứng với mệnh đề GROUP BY.
 - (3) Bỏ qua tất cả các nhóm không thỏa mãn điều kiện trong mệnh đề HAVING.
 - (4) Áp dụng các hàm thư viện cho mỗi nhóm.
 - (5) Rút trích các giá trị đối với các cột và hàm kết hợp được liệt kê trong mệnh đề SELECT.
- Ví dụ: Cho biết tên phòng và lương trung bình của các nhân viên trong phòng ban lớn hơn 40.000.

```
SELECT  TENPHG , AVG ( LUONG )  
FROM    NHANVIEN  
WHERE   PHG = MAPHG  
GROUP BY TENPHG  
HAVING AVG (LUONG) > 40000
```

- Ví dụ: Với mỗi đề án có trên hai nhân viên tham gia, tìm mã đề án (MADA), tên đề án (TENDA) và số lượng nhân viên tham gia đề án đó (bài tập).
- Ví dụ: Tìm những phòng ban có lương trung bình của nhân viên cao nhất (bài tập).

2.10. Câu truy vấn lồng (Truy vấn con).

- Ở các ví dụ trên, các điều kiện trong mệnh đề WHERE chỉ bao gồm các thuộc tính đơn hay các phép toán so sánh hằng số. Trong phần này chúng ta tìm hiểu cách biểu diễn câu truy vấn con trong mệnh đề WHERE.
 - ✓ Câu truy vấn con là một cấu trúc SELECT – FROM – WHERE lồng trong một câu truy vấn khác.
 - ✓ Câu truy vấn con thường được dùng để kiểm tra các tập hợp thành viên, so sánh tập hợp và định bản số.
 - ✓ Khi sử dụng truy vấn con trong mệnh đề WHERE của một câu truy vấn khác, mệnh đề SELECT trong truy vấn con phải phù hợp với số thuộc tính và kiểu dữ liệu của mệnh đề WHERE trong truy vấn ngoài.
- Các mệnh đề thường được sử dụng với truy vấn con:

✓ IN / NOT IN

Cú pháp: <biểu thức>[not] IN (<truy vấn con>)

Ví dụ: Tìm tên và địa chỉ của các nhân viên nam làm việc ở phòng “Nghiên cứu”.

```
SELECT TENNV, DCHI
FROM NHANVIEN
WHERE PHG IN ( SELECT MAPHG
                FROM PHONGBAN
                WHERE TENPHG= "nghiencuu") AND
                PHAI="nam"
```

✓ Mệnh đề ANY

Cú pháp: <biểu thức><phép toán so sánh> ANY (<truy vấn con>).

Ví dụ: Tìm những nhân viên có lương lớn hơn ít nhất một nhân viên ở phòng số 5.

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE LUONG ≥ ANY ( SELECT LUONG
                    FROM NHANVIEN
                    WHERE PHG=5 )
```

✓ Mệnh đề SOME (tương tự như ANY).

✓ Mệnh đề ALL.

Cú pháp: <biểu thức><phép toán so sánh> ALL (<truy vấn con>).

Ví dụ: Tìm những nhân viên có lương lớn hơn tất cả những nhân viên phòng số 5.

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE LUONG ≥ ALL ( SELECT LUONG
                    FROM NHANVIEN
                    WHERE PHG=5 )
```

✓ Kiểm tra sự tồn tại [NOT]EXISTS

Cấu trúc: [NOT]EXISTS (<truy vấn con>).

Ví dụ: Tìm những nhân viên có người thân cùng tên và cùng giới tính.

```
SELECT N.TENNV
FROM NHANVIEN N
```

```
WHERE EXISTS ( SELECT *
                FROM THANNHAN
                WHERE MA_NVIEN=N.MANV AND
                      TENTN=TENNV AND
                      PHAI=N.PHAI)
```

3. Cập nhật dữ liệu.

Các lệnh cập nhật dữ liệu gồm: INSERT, UPDATE, DELETE.

3.1. Thêm dữ liệu (INSERT INTO).

Khi lệnh này thực thi thì có thể thêm vào bảng một hay nhiều bộ.

- Thêm một bộ vào quan hệ.

Cú pháp:

```
INSERT INTO < tên bảng>[<thuộc tính1><thuộc tính2>
]
VALUES ( <giá trị 1>,<giá trị 2>          )
```

Ví dụ: **INSERT INTO** NHANVIEN (TENNV, DCHI)
VALUES ('Tiến', 'TPHCM')

- Thêm nhiều bộ vào quan hệ và các bộ này là kết quả lấy từ một câu truy vấn.

Cú pháp: **INSERT INTO** <bảng> [<thuộc tính1> <thuộc tính 2>]

```
SELECT          FROM          WHERE
```

Ví dụ: **CREATE TABLE** T_PHONG

```
( TENP VARCHAR(10)
  SONV INTEGER
  LUONG TC INTEGER )
```

```
INSERT INTO T_PHONG (TENP, SONV, LUONGTC)
SELECT TENPHG, COUNT(*), SUM(LUONG)
FROM PHONGBAN, NHANVIEN
WHERE MAPHG=PHG
GROUP BY TENPHG
```

3.2. Lệnh xóa dữ liệu (DELETE).

- Lệnh này có thể xóa khỏi quan hệ một hay nhiều bộ.

✓ Cú pháp: **DELETE** [FROM] <tên bảng>
WHERE <điều kiện>

- ✓ Với <tên bảng> là bảng được chọn để xóa, <điều kiện> là điều kiện để chọn các bộ sẽ được xóa.
- ✓ Số lượng các bộ bị xóa phụ thuộc vào số lượng các bộ trong quan hệ thỏa mãn điều kiện trong mệnh đề WHERE.
- ✓ Nếu không chỉ ra điều kiện trong mệnh đề WHERE, thì tất cả các bộ trong quan hệ sẽ bị xóa.

- Ví dụ:

```
DELETE FROM NHANVIEN  
WHERE MANV='123'
```

```
DELETE FROM NHANVIEN  
WHERE PHG IN ( SELECT MAPHG  
               FROM PHONGBAN  
               WHERE TENPHG='nguyên cứu')
```

3.3. Cập nhật dữ liệu (UPDATE).

- Lệnh cập nhật khi thực thi sẽ làm thay đổi nội dung của quan hệ theo một điều kiện nào đó.
- Cú pháp: UPDATE <tên bảng>

```
SET      <thuộc tính 1>=<giá trị 1>  
        <thuộc tính 2>=<giá trị 2>
```

```
[WHERE <điều kiện>]
```

- Ví dụ:

```
UPDATE PHONGBAN  
SET  TENPHG = 'Quản trị'  
WHERE MAPHG= 5
```

4. Tóm tắt câu truy vấn trong SQL.

- Một câu truy vấn SQL có thể được tóm tắt một cách tổng quát như sau:

```
SELECT <danh sách các thuộc tính>  
FROM <danh sách các bảng>  
[WHERE <điều kiện>]  
[GROUP BY <các thuộc tính gom nhóm>]  
[HAVING <điều kiện gom nhóm>]  
[ORDER BY <danh sách các thuộc tính>]
```

- Trong đó hai mệnh đề đầu là bắt buộc, còn các mệnh đề sau có thể có hoặc không.

5. Bài tập

1. Cho cơ sở dữ liệu có 3 bảng sau:

DS(sbd, hoten, ngaysinh, gioitinh, quequan).

SBD_PH(sbd, sophach).

DTM(sophach, diem).

Trong đó:

- Sbd: Số báo danh.
- Hoten: Họ và tên sinh viên.
- Ngaysinh: Ngày sinh.
- Gioitinh: Giới tính.
- Quequan: Quê quán.
- Sophach: Số phách.
- Diem: Điểm thi môn học.

Trả lời các câu hỏi sau bằng SQL.

- Xóa sinh viên có số báo danh là 211.
- Cho biết họ tên, giới tính, ngày sinh của mỗi sinh viên.
- Cho biết họ tên, giới tính, ngày sinh của mỗi sinh viên có điểm thi > 8.
- Cho biết những nữ sinh viên quê ở “bến tre” và có điểm > 5.
- Cho biết họ tên, ngày sinh, điểm thi của những sinh viên có giới tính là nam quê ở “Hải Dương” và sắp xếp theo thứ tự giảm dần của điểm.
- Cho biết họ tên, ngày sinh, điểm thi của những sinh viên có quê ở “Hải Phòng” hoặc “Hưng Yên” và có điểm thi < 5 hoặc > 8. Danh sách đưa ra sắp xếp theo chiều giảm dần của điểm.
- Cho biết họ tên của những sinh viên không đi thi.

- h. Cho biết số lượng sinh viên và điểm trung bình của các sinh viên ở mỗi tỉnh.
- i. Cho biết tổng số sinh viên nữ và điểm trung bình của các sinh viên nữ ở mỗi tỉnh.
- j. Tăng thêm 2 điểm cho các học sinh quê ở Sơn La.

2. Cho các lược đồ quan hệ sau:

Monhoc(MSMH, TENMH, SOTINCHI, TINHCHAT)

MSMH: Mã số môn học.

TENMH: Tên môn học.

SOTINCHI: Số lượng tín chỉ.

TÍNH CHẤT: Bằng 1 nếu đó là môn học bắt buộc, bằng 0 nếu đó là môn học không bắt buộc.

Sinhvien(MSSV, HOTEN, NGAYSINH, LOP)

MSSV: Mã số sinh viên.

HOTEN: Họ tên sinh viên.

NGAYSINH: Ngày sinh.

LOP: Lớp.

Diem(MSSV, MSMH, DIEMTHI)

DIEMTHI: Điểm thi

Hãy dùng lệnh SQL để thực hiện các câu lệnh sau:

- a. Cho biết họ tên sinh viên (HOTEN) và điểm thi(DIEMTHI) của các sinh viên.
- b. Hãy cho biết những môn học bắt buộc có số tín chỉ là 3.
- c. Hãy cho biết những môn học bắt buộc có SOTINCHI cao nhất.
- d. Hãy liệt kê danh sách gồm MSSV, HOTEN, LOP, DIEMTHI của những sinh viên thi môn học CSDL.
- e. Hãy cho biết những sinh viên có điểm thi cao nhất.
- f. Hãy liệt kê danh sách gồm MSSV, HOTEN, ĐIỂM TRUNG BÌNH của các sinh viên.
- g. Hãy liệt kê danh sách gồm MSSV, HOTEN, LOP, điểm trung bình của các sinh viên có điểm trung bình các môn dưới 5, theo thứ tự LOP, HOTEN.
- h. Hãy cho biết những sinh viên có điểm trung bình các môn cao nhất.

3. Cho các lược đồ CSDL sau:

Khach_H(MAKH, HOTEN, DIACHI, DIENTHOAI)

MAKH: Mã khách hàng.

HOTEN: Họ tên.

DIACHI: Địa chỉ.

DIENTHOAI: Điện thoại.

Hoadon(SOHD, NGAYLAPHD, NGAYBAN, MAKH)

SOHD: Số hóa đơn.

NGAYLAPHD: Ngày lập hóa đơn.

NGAYBAN: Ngày bán.

Nhom(MANHOM, TENNHOM)

MANHOM: Mã nhóm.

TENNHOM: Tên nhóm.

Hang(MAHANG, TENHANG, DONGIA, DVT, MANHOM)

MAHANG: Mã hang.

TENHANG: Tên hang.

DONGIA: Đơn giá.

DVT: Đơn vị tính.

DongHoaDon(SOHD, MAHANG, SLBAN)

SLBAN: Số lượng bán.

Hãy thực hiện các câu hỏi sau bằng SQL

- Danh sách các khách hàng đã mua hàng trong ngày d. Yêu cầu các thông tin MAKH, HOTEN, DIACHI, DIENTHOAI.
- Danh sách các mặt hàng trong sổ hóa đơn (SOHD) là x. Yêu cầu các thông tin MAHANG, TENHANG, SLBAN, DONGIA, THANHTIEN ($THANHTIEN = SLBAN * DONGIA$; THANHTIEN là thuộc tính tự đặt). Yêu cầu sắp xếp tăng dần theo cột TENHANG.
- Danh sách các mặt hàng thuộc mã nhóm hàng là A có đơn giá cao nhất. Yêu cầu các thông tin: MAHANG, TENHANG, DONGIA.
- Đếm số lượng mặt hàng của mỗi nhóm hàng. Yêu cầu các thông tin: MANHOM, TENNHOM, SOLUONG. (Trong đó SOLUONG là thuộc tính tự đặt).
- Danh sách các khách hàng đã mua các mặt hàng có mã nhóm hàng là A trong ngày d. Yêu cầu các thông tin MAKH, HOTEN, DIACHI, DIENTHOAI, TENHANG.
- Thống kê việc mua hàng trong năm 2002 của khách hàng có mã khách hàng là Kh01 (theo từng hóa đơn). Yêu cầu các thông tin MAKH, HOTEN, SOHD, TRIGIAHD trong đó TRIGIAHD là tổng số tiền trong một hóa đơn (TRIGIAHD là thuộc tính tự đặt).

4. Cho cơ sở dữ liệu quản lý nhân viên gồm các lược đồ như sau:

NHANVIEN(HONV, TENNV, MANV, NGSINH, DCHI, PHAI, LUONG, MA_NQL, PHG)

PHONGBAN(TENPHG, MAPHG, TRPHG, NG_NCHUC)

DIADIEM(MAPHG, DDIEM)

THANNHAN(MA_NV, TENTN, PHAI, NGSINH, QUANHE)

DEAN(TENDA, MADA, DDIEM_DA, PHONG)

PHANCONG(MA_NV, SODA, THOIGIAN)

- a. Tìm những nhân viên làm việc ở phòng số 4.

Select * from nhanvien where phg = '4'

- b. Tìm những nhân viên có mức lương trên 30.000.

Select * from nhanvien where luong > 30 000

- c. Tìm các nhân viên có mức lương > 25.000 ở phòng số 4 hoặc các nhân viên có mức lương > 30.000 ở phòng số 5.

Select * from nhanvien where (luong > 25000 and phg = '4') or (luong > 30000 and phg = '5')

- d. Cho biết họ và tên của các nhân viên ở TPHCM.

Select honv, tennv from nhanvien where dchi = 'TPHCM'

- e. Cho biết ngày sinh và địa chỉ của nhân viên “Trần Thanh Nam”.

Select ngaysinh, diachi from nhanvien where honv = 'Tran Thanh' and Tennv = 'Nam'

- f. Với mỗi phòng ban cho biết tên phòng ban và địa điểm phòng.

Select tenphong, diadiem from phongban, diadiem where phongban.maphong = diadiem.maphong

- g. Tìm những người trưởng phòng ở từng phòng ban.

- h. Tìm tên và địa chỉ của các nhân viên phòng “Nghiên cứu”.

- i. Tìm tên những nữ nhân viên và tên người thân của họ.

- j. Với mỗi đề án ở “Hà nội” liệt kê các mã số đề án (MADA), mã số phòng ban chủ trì đề án (PHONG), họ tên trưởng phòng cũng như đại chỉ và ngày sinh của người đó.

- k. Với mỗi nhân viên cho biết họ tên nhân viên và họ tên người quản lý trực tiếp của nhân viên đó.

- l. Với mỗi đề án, liệt kê tên đề án và tổng số giờ làm việc một tuần của tất cả các nhân viên tham gia đề án đó.

- m. Với mỗi nhân viên cho biết họ tên và số người thân của nhân viên đó.

- n. Với mỗi phòng ban liệt kê tên phòng ban và lương trung bình của những nhân viên làm việc cho phòng ban đó.
- o. Cho biết danh sách những nhân viên (MANV, TENNV) có trên hai thân nhân.
- p. Cho biết danh sách những nhân viên (MANV, TENNV) không có thân nhân.

5. Cho lược đồ quan hệ sau

TINH_TP(T_TP, DIENTICH, DANSO, TENTINH)

Mô tả: Một tỉnh (thành phố) có 1 mã số phân biệt với các tỉnh (thành phố) khác, và ta biết được tên tỉnh, diện tích, dân số của tỉnh đó.

MIEN_TTP(T_TP,MIEN)

Mô tả: Một tỉnh (thành phố) sẽ xác định được miền (Bắc, Nam, Trung,Tâynguyên) của tỉnh (thành phố) đó.

BIENGIOI(NUOC,T_TP)

Mô tả: Một nước sẽ có nhiều tỉnh (T

P) nằm ở vùng biên giới, và 1 tỉnh có thể có giáp ranh biên giới của nhiều nước.

LANGGIENG(T_TP,T_TP_LG)

Mô tả: Một tỉnh (TP) sẽ có nhiều tỉnh (TP) khác là láng giềng (nằm cạnh nhau).

- a. Xác định khóa của các quan hệ.
- b. Hãy trả lời câu hỏi sau bằng ngôn ngữ SQL.

1/ Cho biết dân số cùng tên tỉnh của các tỉnh (TP) có diện tích > 7000km².

SELECT TENTINH, DANSO FROM TINH_TP WHERE DIENTICH > 7000

2/ Cho biết dân số cùng tên tỉnh của các tỉnh miền Bắc.

SELECT DANSO, TENTINH FROM TINH_TP, MIEN_TTP WHERE TINH_TP.T_TP = MIEN_TTP.T_TP AND MIEN LIKE 'BAC'

3/ Cho biết mã các nước biên giới của các tỉnh miền Nam.

```
SELECT NUOC FROM BIENGIOI, MIEN WHERE BIENGIOI.TTP =  
MIEN.TTP AND MIEN LIKE 'NAM'
```

4/ Cho biết diện tích trung bình của các tỉnh (TP).

```
SELECT AVG(DIENTICH) FROM TINH_TP
```

5/ Cho biết mật độ dân cư cùng với tên tỉnh (TP) của tất cả các tỉnh (TP).

```
SELECT TENTINH,  DANSO/DIENTICH AS  MATDO  FROM  
TINH_TP
```

6/ Cho biết tên những tỉnh (TP) có một diện tích lớn hơn tất cả các tỉnh (TP) láng giềng của nó.

7/ Cho biết tên những tỉnh (TP) có diện tích lớn nhất.

```
SELECT TENTINH FROM TINH_TP WHERE DIENTICH = (SELECT  
MAX(DIENTICH) FROM TINH_TP)
```

8/ Cho biết tên những tỉnh (TP) giáp với 2 nước biên giới khác nhau.

9/ Cho biết danh sách các miền cùng với số tỉnh (TP) trong các miền đó.

```
SELECT MIEN, COUNT(*) FROM MIEN GROUP BY MIEN
```

10/ Cho biết tên tỉnh (TP) có nhiều láng giềng nhất.

```
SELECT  TENTINH  FROM  TINH_TP,  LANGGIENG  WHERE  
TINH_TP.MATINH = LANGGIENG.MATINH GROUP BY TINH_TP  
HAVING COUNT (LANGGIENG.T_TP) = (SELECT TOP 1 COUNT(*)  
FROM LANGGIENG GROUP BY TINH_TP ORDER BY COUNT(*)  
DESC)
```

11/ Cho biết tên những tỉnh (TP) có diện tích nhỏ hơn diện tích trung bình của tất cả các tỉnh (TP).

```
SELECT TENTINH FROM TINH_TP WHERE DIENTICH < (SELECT  
AVG(DIENTICH) FROM TINH_TP)
```

12/ Cho biết tên những tỉnh (TP) giáp với các tỉnh (TP) ở miền trung và không phải là miền trung.

```
SELECT TENTINH FROM TINH_TP WHERE MATINH IN (SELECT
    T_TP_LG FROM LANGGIENG WHERE T_TP IN (SELECT T_TP
    FROM MIEN WHERE MIEN LIKE 'MIENTRUNG') AND
    T_TP_LG NOT IN (SELECT T_TP FROM MIEN WHERE MIEN
    LIKE 'MIENTRUNG'))
```

13/ Cho biết tên các tỉnh thành phố với tổng diện tích, tổng dân số của từng miền.

14/ Cho biết miền nào có diện tích lớn nhất.

6. Cho lược đồ quan hệ sau:

KHACH_HANG(MAKH, TENKH, DIACHIKH, LOAIKH)

Mô tả: Một khách hàng có 1 mã khách hàng (MAKH) duy nhất, 1 tên khách hàng, 1 địa chỉ khách hàng và 1 loại khách hàng. Loại khách hàng gồm có: “CÁ NHÂN, DOANH NGHIỆP”.

DIEN_THOAI(SODT, MAKH, LOAIDT, MAQUAN, MATINH, SOHD)

Mô tả: Một điện thoại có 1 số điện thoại để phân biệt với các điện thoại khác thuộc về 1 khách hàng, loại điện thoại gồm có: “VÔ TUYẾN”, ”DÂY CÁP”. Một số điện thoại nằm ở 1 quận của 1 tỉnh, được lắp đặt theo 1 hợp đồng (SOHD).

TINH(MATINH, TENT, MANUOC)

Mô tả: Một tỉnh - thành phố có 1 mã tỉnh - thành phố (MATINH) duy nhất, 1 tên tỉnh - thành phố (TENTINH) thuộc về 1 nước (MANUOC).

QUAN(MAQUAN, MATINH, TENQUAN, TONGSO_DT)

Mô tả: Một quận có 1 mã quận để phân biệt với các quận khác trong cùng 1 tỉnh và có 1 tên quận và tổng số điện thoại được lắp trong quận.

NUOC(MANUOC, TENNUOC)

Mô tả: Một nước có 1 mã nước duy nhất và 1 tên nước.

DICHVU(MADV, TENDV)

Mô tả: Một dịch vụ có 1 mã dịch vụ duy nhất và 1 tên dịch vụ.

DANGKY(MADV, SODT)

Mô tả: Một số điện thoại đăng ký nhiều dịch vụ, 1 dịch vụ cũng được nhiều số điện thoại đăng ký.

KHANANG_DK(LOAIKH,MADV)

Mô tả: Một loại khách hàng được cho phép đăng ký nhiều dịch vụ, 1 dịch vụ cũng được nhiều số điện thoại đăng ký.

1. Xác định khóa của quan hệ trên.
2. Trả lời các câu hỏi sau bằng ngôn ngữ đã học:

a) Danh sách các điện thoại loại “VÔ TUYẾN”.

```
SELECT * FROM DIENTHOAI WHERE LOAIDT LIKE 'VO TUYEN'
```

b) Danh sách khách hàng thuộc loại khách hàng ‘DOANH NGHIỆP’.

```
SELECT * FROM KHACHHANG WHERE LOAIKH LIKE 'DOANH NGHIEP'
```

c) Danh sách tên khách hàng lắp điện thoại loại “DÂY CÁP” của tỉnh có tên gọi là :”LONG AN”.

```
SELECT TENKH FROM KHACHHANG, DIENTHOAI, TINH WHERE
KHACHHANG.MAKH = DIENTHOAI.MAKH AND
DIENTHOAI.MATINH = TINH.MATINH AND LOAIDT LIKE 'DAY CAP' AND TENTINH LIKE 'LONG AN'
```

d) Liệt kê tất cả các tỉnh hiện có lắp hơn 1000 điện thoại.

```
SELECT TINH.MATINH, TINH.TENTINH FROM TINH, QUAN
WHERE TINH.MATINH = QUAN.MATINH GROUP BY
TINH.MATINH, TINH.TENTINH HAVING SUM (TONGSO_DT) >
10000
```

e) Danh sách các khách hàng có đăng ký dịch vụ có tên “Dịch vụ 108” thuộc loại khách hàng “CÁ NHÂN”.

```
SELECT KHACHHANG.* FROM KHACHHANG, KHANANG_DK,
DICHVU WHERE KHACHHANG.LOAIKH = KHANANG_DK.LOAIKH
AND KHANANG_DK.MADV = DICHVU.MADV AND TENDV LIKE
'DICHVU 108' AND LOAIKH LIKE 'CA NHAN'
```

f) Cho biết dịch vụ nào chưa hề được đăng ký.

```
SELECT * FROM DICHVU WHERE MADV NOT IN (SELECT  
DISTINCT MADV FROM DANGKY)
```

g) Cho biết dịch vụ nào được đăng ký nhiều nhất.

```
SELECT DICHVU.* FROM DICHVU, DANGKY WHERE  
DICHVU.MADV = DANGKY.MADV GROUP BY DICHVU.* HAVING  
COUNT (DANGKY.*) = (SELECT TOP 1 COUNT(*) FROM DANGKY  
GROUP BY MADV ORDER BY COUNT(*) DESC)
```

CHƯƠNG V : RÀNG BUỘC TOÀN VỆ

Mục tiêu:

- ✦ Tìm hiểu vai trò của ràng buộc toàn vẹn trong một cơ sở dữ liệu.
- ✦ Trình bày một số đặc trưng của ràng buộc toàn vẹn.
- ✦ Phân loại ràng buộc toàn vẹn.

1. Khái niệm về ràng buộc toàn vẹn (Integrity Constraints).

1.1. Khái niệm.

- Ràng buộc toàn vẹn (viết tắt là RBTV) là các quy tắc, điều kiện bất biến không được vi phạm trong một cơ sở dữ liệu.
- Trong một cơ sở dữ liệu, luôn luôn tồn tại rất nhiều mối liên hệ ảnh hưởng qua lại lẫn nhau giữa các thuộc tính của một quan hệ, giữa các bộ giá trị trong một quan hệ và giữa các thuộc tính của các bộ giá trị trong các quan hệ với nhau. Các mối quan hệ phụ thuộc lẫn nhau này chính là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong cơ sở dữ liệu đều phải thỏa mãn ở bất kỳ thời điểm nào nhằm đảm bảo cho CSDL luôn luôn ở trạng thái an toàn và nhất quán về dữ liệu.
- Việc kiểm tra RBTV có thể được tiến hành vào một trong các thời điểm sau:
 - ✓ Kiểm tra ngay khi thực hiện một thao tác cập nhật CSDL (thêm, sửa, xóa). Thao tác cập nhật được xem là hợp lệ nếu như nó không vi phạm bất cứ một RBTV nào.
 - ✓ Kiểm tra định kỳ hay đột xuất, nghĩa là việc kiểm tra RBTV được tiến hành một cách độc lập đối với thao tác cập nhật dữ liệu. Đối với những trường hợp vi phạm RBTV, hệ thống sẽ có những xử lý ngầm định hoặc yêu cầu người sử dụng sửa chữa sai sót một cách tường minh.

1.2. Mục đích.

- Đảm bảo tính kết dính của các bộ phận cấu thành nên CSDL.
- Đảm bảo tính nhất quán của dữ liệu.

- Đảm bảo CSDL luôn biểu diễn đúng ngữ nghĩa thực tế.

2. Các đặc trưng của RBTV.

2.1. Bối cảnh.

Bối cảnh của một ràng buộc toàn vẹn là một hay nhiều quan hệ mà RBTV có thể bị vi phạm khi thực hiện các thao tác thêm, xóa, sửa dữ liệu trên quan hệ đó.

2.2. Biểu diễn RBTV.

Điều kiện của RBTV có thể được biểu diễn bằng:

- Một biểu thức toán học.
- Một phép so sánh trên tập hợp.
- Một biểu thức vị từ trên các quan hệ cần được kiểm tra RBTV.
- Một thuật toán kiểm tra điều kiện (có thể sử dụng ngôn ngữ tự nhiên, mã giả, ...)

2.3. Bảng tầm ảnh hưởng.

- Bảng tầm ảnh hưởng là bảng được dùng để ghi nhận các ảnh hưởng của RBTV khi có các thao tác thêm, xóa, sửa trên các quan hệ.
- Bảng tầm ảnh hưởng có dạng như sau:

- ✓ Trong đó dấu “+” có nghĩa là có thể bị vi phạm ràng buộc toàn vẹn và cần phải được kiểm tra.
- ✓ Dấu “-” có nghĩa là không bị vi phạm ràng buộc toàn vẹn.

3. Phân loại RBTV.

Có hai loại RBTV là: RBTV có bối cảnh là một quan hệ và RBTV có bối cảnh là nhiều quan hệ.

3.1. RBTV có bối cảnh là một quan hệ (RBTV về miền giá trị, RBTV liên bộ, liên thuộc tính).

3.1.1. RBTV về miền giá trị.

- Là RBTV có liên quan đến miền giá trị của các thuộc tính trong một quan hệ hay nói cách khác là nó quy định miền giá trị của một thuộc tính.
- Ví dụ: R_1 : “ Số tín chỉ của một môn học phải > 0 ”

Bối cảnh : MHOC

Biểu diễn : $\forall t \in \text{MHOC}$

$t.\text{TINCHI} > 0$

END \forall

Bảng tầm ảnh hưởng:

3.1.2. RBTV liên bộ.

- Là sự ràng buộc giữa các bộ trong cùng một quan hệ, sự tồn tại của một hay nhiều bộ phụ thuộc vào sự tồn tại của một hay nhiều bộ khác trong quan hệ. Ví dụ : R_2 : “Mỗi sinh viên có một mã số sinh viên duy nhất và khác nhau”.

Bối cảnh : SVIEN

Biểu diễn : $\forall t_1, t_2 \in \text{SVIEN}$

$t_1.\text{MASV} \neq t_2.\text{MASV}$

END \forall

Bảng tầm ảnh hưởng:

3.1.3. RBTV liên thuộc tính.

- Là mối liên hệ giữa các thuộc tính khác nhau trong cùng một quan hệ.
- Ví dụ: Cho quan hệ THISINH (SOBAODANH, HOTEN, DCHI, NGSINH, NGTHI).

R_3 ” thí sinh đăng ký thi không quá 35 tuổi ”

Bối cảnh: THISINH

Biểu diễn: $\forall t \in \text{THISINH}$

$t.\text{NGSINH} - t.\text{NGTHI} \leq 35$

END \forall

Bảng tầm ảnh hưởng:

3.2. Ràng buộc toàn vẹn có bối cảnh là nhiều quan hệ cơ sở.

Bao gồm: RBTV tham chiếu, RBTV liên bộ – liên quan hệ, RBTV liên thuộc tính – liên quan hệ, RBTV do thuộc tính tổng hợp, RBTV do sự hiện diện của chu trình trên đồ thị.

3.2.1. RBTV tham chiếu.

- RBTV tham chiếu nhằm đảm bảo giá trị xuất hiện trong một quan hệ của một tập thuộc tính cho trước phải xuất hiện trong một tập thuộc tính trong một quan hệ khác. RBTV này bảo đảm tính hợp lệ của các tham chiếu thông qua “các khóa ngoại”.
- Ví dụ: Cho hai quan hệ

NHANVIEN (MANV, TENNV, NGSINH, DCHI, PHAI,
LUONG, PHG)
PHONGBAN (MAPHG, TENPHG, TRPHG, NG_NC)

R_4 : “nếu một mã phòng xuất hiện trong quan hệ NHANVIEN thì mã phòng đó phải xuất hiện trong quan hệ PHONGBAN”.

Bối cảnh: NHANVIEN, PHONGBAN.

Biểu diễn: $\forall t \in \text{NHANVIEN}$

$\exists u \in \text{PHONGBAN}$

$u[\text{MAPHG}] = t[\text{PHG}]$

Hay $\text{NHANVIEN}[\text{PHG}] \subseteq \text{PHONGBAN}[\text{MAPHG}]$

Bảng ảnh hưởng :

3.2.2. RBTV liên bộ – liên quan hệ.

Áp dụng đối với từng nhóm các bộ của nhiều quan hệ bối cảnh khác nhau (thông thường là hai quan hệ).

- Ví dụ: Cho hai quan hệ sau:

HOADON (SOHD, NGLAP, SODH, TRIGIA, NGXUAT)
CTHOADON (SOHD, MAHH, SOLUONG, DONGIA)

R₅ : “Mỗi hóa đơn bán hàng phải có ít nhất một mặt hàng”.

Bối cảnh : $\forall t \in \text{HOADON}$
 $\exists u \in \text{CTHOADON}$
 $u.\text{SOHD} = t.\text{SOHD}$

Bảng tầm ảnh hưởng :

3.2.3. RBTV liên thuộc tính – liên quan hệ.

- Là mối liên hệ giữa các thuộc tính trên nhiều quan hệ khác nhau.
- Ví dụ: Cho hai quan hệ.

NHANVIEN (MANV, TENNV, NGSINH, DCHI, PHAI, LUONG, MA_NQL, PHG)

PHONGBAN (MAPHG, TENPHG, TR_PHG, NG_NHANCHUC)

R₆: “Ngày sinh của trưởng phòng phải nhỏ hơn ngày nhận chức”.

Bối cảnh: NHANVIEN, PHONGBAN.

Biểu diễn: $\forall t \in \text{PHONGBAN}, \forall u \in \text{NHANVIEN}$
 $t.\text{TR_PHG} = u.\text{MANV}$
 $\Rightarrow t.\text{NG_NHANCHUC} > u.\text{NGSINH}$

Bảng tầm ảnh hưởng :

3.2.4. RBTV do thuộc tính tổng hợp.

- Ví dụ: Cho hai quan hệ

SINHVIEN (MASV, TENSX, DTB)

DIEM_SV (MASV, MAMON, DIEM)

R₇: “Điểm trung bình của sinh viên phải bằng điểm bình quân của các môn mà sinh viên đó học”.

Bối cảnh: $\forall t \in \text{SINHVIEN}$
 $t.\text{DTB} = \text{AVG}_{u \in D} u.\text{DIEM}.$
 Trong đó $D \leftarrow_{t.\text{MASV}=u.\text{MASV}} \text{DIEM_SV}$

Bảng tầm ảnh hưởng:

4. Bài tập.

Bài 1. Mô tả các ràng buộc toàn vẹn về miền giá trị, ràng buộc toàn vẹn liên thuộc tính cho lược đồ cơ sở dữ liệu sau:

Để quản lý lịch dạy của các giáo viên và lịch học của các lớp, một trường tổ chức như sau:

Mỗi giáo viên có một mã số giáo viên (MAGV) duy nhất, mỗi MAGV xác định các thông tin như: họ và tên giáo viên (HOTEN), số điện thoại (DTGV). Mỗi giáo viên có thể dạy nhiều môn cho nhiều khoa nhưng chỉ thuộc sự quản lý hành chính của một khoa nào đó.

Mỗi môn học có một mã số môn học (MAMH) duy nhất, mỗi môn học xác định tên môn học(TENMH). Ứng với mỗi lớp thì mỗi môn học chỉ được phân cho một giáo viên.

Mỗi phòng học có một số phòng học (PHONG) duy nhất, mỗi phòng có một chức năng(CHUCNANG), chẳng hạn như phòng lý thuyết, phòng thực hành máy tính, phòng nghe nhìn, xưởng thực tập cơ khí,...

Mỗi khoa có một mã khoa (MAKHOA) duy nhất, mỗi khoa xác định các thông tin như: tên khoa(TENKHOA), điện thoại khoa(DTKHOA).

Mỗi lớp có một mã lớp (MALOP) duy nhất, mỗi lớp có một tên lớp (TENLOP), sĩ số lớp (SISO). Mỗi lớp có thể học nhiều môn của nhiều khoa nhưng chỉ thuộc sự quản lý hành chính của một khoa nào đó. Hàng tuần, mỗi giáo viên phải lập lịch báo giảng cho biết giáo viên đó sẽ dạy những lớp nào, ngày nào (NGAYDAY), môn gì, tại phòng nào, từ tiết (TUTIET) nào đến tiết (DENTIET) nào, tựa đề bài dạy (BAIDAY), những ghi chú (GHICHU) về các tiết dạy này, đây là giờ dạy lý thuyết(LYTHUYET) hay thực hành - giả sử nếu LYTHUYET=1 thì đó là giờ dạy thực hành và nếu LYTHUYET=2 thì đó là giờ lý thuyết, một ngày có 16 tiết, sáng từ tiết 1 đến tiết 6, chiều từ tiết 7 đến tiết 12, tối từ tiết 13 đến 16.

Giả sử ta có lược đồ cơ sở dữ liệu để quản lý bài toán trên như sau:

Giaovien(MAGV,HOTEN,DTGV,MAKHOA)

Monhoc(MAMH, TENMH)

Phonghoc(PHONG, CHUCNANG)

Khoa(MAKHOA, TENKHOA, DTKHOA)

Lop(MALOP, TENLOP, SISO, MAKHOA)

Lichday(MAGV, MAMH, PHONG, MALOP, NGÀYDAY, TUTIET, DENTIET, BAIDAY, LYTHUYET, GHICHU)

Bài 2. Mô tả tất cả các ràng buộc toàn vẹn cho lược đồ cơ sở dữ liệu dưới đây và lập bảng tầm ảnh hưởng của các ràng buộc toàn vẹn.

Cho một lược đồ cơ sở dữ liệu quản lý việc thuê mướn phòng tại một khách sạn. Lược đồ cơ sở dữ liệu gồm các lược đồ quan hệ như sau:

Q1: Phong(MAPH, SO_NGUOI, DACDIEM, GIA_PHONG)

Tên từ: Các phòng của khách sạn được phân biệt với nhau qua MAPH. SO_NGUOI là khả năng chứa tối đa của phòng. DACDIEM mô tả số đặc điểm của phòng. GIA_PHONG là giá cả thuê phòng trong 1 ngày.

Q2: Tien_nghi(LOAI_TN, TEN_TN)

Tên từ: Ngoài các vật dụng tối thiểu, khách sạn có thể trang bị thêm một số tiện nghi khác cho các phòng như: điện thoại, tivi, tủ lạnh, ... LOAI_TN là mã số để phân biệt từng loại tiện nghi. TEN_TN là tên gọi của loại tiện nghi.

Q3: Tai_san(LOAI_TN, STT, MAPH, NGÀY_TB).

Tên từ: Mỗi loại tiện nghi, khách sạn có thể mua một số lượng lớn và STT dùng để phân biệt các vật dụng trong cùng loại tiện nghi. Một vật dụng có thể được sắp xếp trang bị cho nhiều phòng khác nhau nhưng trong một ngày vật dụng chỉ trang bị cho một phòng. MAPH là phòng được trang bị và NGÀY_TB là ngày bắt đầu trang bị.

Lưu ý: Mỗi khi một vật dụng được thay đổi phòng thì cập nhật lại MAPH và NGÀY_TB của vật dụng đó.

Q4: Thue_phong(MAPH, HOTEN, NGÀYBD, NGÀYKT, NGÀYTRA, LOAIDV, NGÀYDV, TIENDV).

Tên từ: HOTEN là họ tên của khách thuê phòng MAPH. Giả sử rằng họ tên các khách thuê phòng trong cùng một phòng trong một ngày luôn luôn khác nhau. NGAYBD và NGAYKT là ngày bắt đầu và ngày kết thúc (dự kiến) thuê phòng. NGAYTRA là ngày trả thật sự. Giả sử rằng không có trường hợp khách trả phòng và thuê lại chính phòng đó trong cùng một ngày. Số tiền thuê phòng được chia đều cho số khách thuê trong cùng phòng. Khách thuê phòng có thể sử dụng thêm các dịch vụ (gọi điện thoại đường dài, thuê xe, thủ tục hành chính,...) LOAI_DV là mã số của loại dịch vụ sử dụng. NGAYDV ngày dịch vụ thực hiện. TIENDV là số tiền khách thuê phải trả cho dịch vụ.

Nếu trong cùng một ngày khách thuê phòng sử dụng 1 dịch vụ nhiều lần thì tiền dịch vụ được cộng dồn lại thành một lần và tạo thành một bộ (ví dụ trong ngày gọi điện thoại 3 cuộc với số tiền phải trả lần lượt là: 5000ĐVN, 4500ĐVN, 2000ĐVN thì sẽ được tính chung một lần là 11500ĐVN). Các dịch vụ được tính riêng đối với từng khách. Nếu là dịch vụ chung cho một số khách thì sẽ tính tiền cho một đơn vị khách đại diện nào đó.

Bài 3. Mô tả các ràng buộc toàn vẹn cho lược đồ cơ sở dữ liệu sau và lập bảng tầm ảnh hưởng.

Cho một lược đồ cơ sở dữ liệu quản lý việc cho mượn sách tại một thư viện (xem tại chỗ hoặc mang về nhà). Lược đồ cơ sở dữ liệu gồm các lược đồ quan hệ như sau:

Q1: The_loai(MATL, TENTL).

Tên từ: Sách của thư viện được phân chia theo thể loại để bạn đọc dễ dàng tra cứu. MATL là mã số của từng thể loại và dùng để phân biệt giữa các thể loại. TENTL là tên gọi của thể loại.

Q2: Sach(MASH, TENSH, NGUYEN_TAC, TAC_GIA, MATL).

Tên từ: MASH dùng để phân biệt các quyển sách. TENSH là tên (tựa) bằng tiếng Việt của sách và NGUYEN_TAC là tên nguyên tác (tiếng Việt hoặc tiếng nước ngoài). TAC_GIA là tên tác giả (hay nhóm các tác giả) của sách. Nếu sách có nhiều tập hay nhiều bản thì cũng xem như các đầu sách khác nhau và có mã số khác nhau. MATL là mã thể loại của sách.

Q3: Phieu_muon(MADG, TENDG, DCHI, NGAYCAP, MASH, NGAYMUON, NGAYTRA, TAI_CHO).

Tân từ: Mỗi độc giả chỉ có một phiếu mượn sách với mã số là MADG để phân biệt với các độc giả khác. Các thuộc tính TENDG, DCHI là tên và địa chỉ của độc giả. NGAYCAP là ngày cấp thẻ cho độc giả. MASH là mã số của sách mượn. Giả sử không có trường hợp mượn rồi trả lại cùng 1 quyển sách trong cùng 1 ngày. Nếu sách mượn đọc tại chỗ thì thuộc tính TAI_CHO có giá trị True và NGAYMUON=NGAYTRA. Nếu sách mượn về nhà thì thuộc tính TAI_CHO có giá trị False và NGAYTRA sẽ có giá trị trống cho đến khi sách được mang trả lại cho thư viện. Mỗi độc giả chỉ được giữ tại nhà tối đa 3 quyển sách và mỗi quyển sách chỉ được giữ tại nhà tối đa 30 ngày (không cần lưu ý đến biện pháp xử lý nếu khách vi phạm nội qui).

Q4: Le_phi(MADG, NAM, NGAY_NOP, SOTIEN).

Tân từ: Độc giả phải đóng lệ phí hằng năm (NAM) để gia hạn thẻ mới được mượn sách.

NGAY_NOP, SOTIEN là ngày và số tiền nộp lệ phí cho NAM.

CHƯƠNG VI: LÝ THUYẾT VỀ THIẾT KẾ CSDL QUAN HỆ

Mục tiêu:

- ✦ Giới thiệu khái niệm phụ thuộc hàm và một số thuật toán liên quan.
- ✦ Ứng dụng thuật toán giải một số bài toán tìm khóa, tìm bao đóng.
- ✦ Trình bày phép tách có kết nối không mất mát thông tin.
- ✦ Giới thiệu một số dạng chuẩn hóa lược đồ quan hệ.

Thiết kế CSDL là một việc làm rất quan trọng để tạo ra một CSDL tốt và hoàn chỉnh. Vậy một CSDL được gọi là tốt nếu đáp ứng được các điều kiện như:

- Không dư thừa dữ liệu.
- Dữ liệu phải nhất quán.
- Không xảy ra hiện tượng dị thường khi thêm hoặc xóa dữ liệu.

1. Phụ thuộc hàm.

1.1. Khái niệm phụ thuộc hàm.

- **Định nghĩa:** Cho R là một lược đồ quan hệ với $U = \{A_1, A_2, \dots, A_n\}$ là tập thuộc tính. X và Y là các tập con của U . Ta nói rằng $X \rightarrow Y$ (đọc là X xác định Y hay Y phụ thuộc hàm vào X) nếu với bất kỳ quan hệ r nào đó là một thể hiện của lược đồ R và $\forall t_1, t_2 \in r$ mà $t_1[X] = t_2[X]$ thì $t_1[Y] = t_2[Y]$.
- Ví dụ: Trong quan hệ SINHVIEN (MASV, TENSX, DCHI)

Ta có: $MASV \rightarrow TENSX, MASV \rightarrow DCHI$

1.2. Hệ tiên đề cho phụ thuộc hàm.

- Gọi F là tập tất cả các phụ thuộc hàm đối với lược đồ quan hệ R và $X \rightarrow Y$ là một phụ thuộc hàm với $X, Y \subset U$. Ta nói rằng $X \rightarrow Y$ được suy diễn logic từ F nếu mỗi quan hệ r thỏa mãn các phụ thuộc hàm của F thì cũng thỏa $X \rightarrow Y$.

- Ví dụ: cho $F = \{A \rightarrow B, B \rightarrow C\}$ thì $A \rightarrow C$ được suy ra từ F .
- Gọi F^+ là bao đóng của F tức là tất cả các phụ thuộc hàm được suy diễn logic từ F . Nếu $F = F^+$ thì F là họ đầy đủ của các phụ thuộc hàm.
- Để có thể xác định các khóa của một lược đồ quan hệ và hiểu được các phép suy diễn logic cho các phụ thuộc hàm cần tính được F^+ từ F , hoặc khẳng định được $X \rightarrow Y$ có thuộc F^+ hay không nếu biết phụ thuộc hàm từ $X \rightarrow Y$ và tập phụ thuộc của hàm F . Do đó đòi hỏi phải có các qui tắc suy diễn để suy ra một hay nhiều phụ thuộc hàm từ các phụ thuộc hàm khác. Để đáp ứng điều này năm 1974 Armstrong đưa ra một hệ tiên đề gọi là hệ tiên đề Armstrong (hay hệ luật dẫn Armstrong).

1.2.1. Hệ tiên đề Armstrong.

Cho R là một lược đồ quan hệ với $U = \{A_1, A_2, \dots, A_n\}$ là tập các thuộc tính của nó và $X, Y, Z, W \subseteq U$.

- Các qui tắc của hệ tiên đề Armstrong bao gồm:

- ✓ Tính phản xạ:

Nếu $Y \subseteq X$ thì $X \rightarrow Y$.

- ✓ Tính tăng trưởng:

Nếu $Z \subset U$ và $X \rightarrow Y$ thì $XZ \rightarrow YZ$.

- ✓ Tính bất cầu:

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

Và người ta đã chứng minh rằng hệ tiên đề Armstrong là đúng đắn và đầy đủ thông qua 3 bổ đề sau đây:

Bổ đề 1: Hệ tiên đề Armstrong là đúng nghĩa là với F là tập phụ thuộc hàm đúng trên quan hệ R , nếu $X \rightarrow Y$ là một phụ thuộc hàm được suy diễn logic từ F nhờ hệ tiên đề Armstrong, thì $X \rightarrow Y$ cũng đúng trên quan hệ R .

Bổ đề 2: Từ hệ tiên đề Armstrong suy ra một số luật bổ sung sau đây:

- ✓ Tính phân rã (hoặc luật tách):

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

- ✓ Tính hợp (hoặc luật hợp):

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

✓ Tính tựa bắc cầu, hoặc bắc cầu giả:

Nếu $X \rightarrow Y$ và $YZ \rightarrow W$ thì $XZ \rightarrow W$.

- Ví dụ: Cho lược đồ quan hệ $R(ABCD)$ với các phụ thuộc hàm $A \rightarrow C$ và $B \rightarrow D$. Chứng minh rằng $AB \rightarrow ABCD$.

Ta có : $A \rightarrow C$ (giả thiết).

$\Rightarrow AB \rightarrow ABC$ (luật tăng trưởng).

Và $B \rightarrow D$ (giả thiết).

$\Rightarrow ABC \rightarrow ABCD$ (luật tăng trưởng).

Vậy $AB \rightarrow ABCD$ (luật bắc cầu).

1.2.2. Bao đóng của một tập thuộc tính.

- **Định nghĩa:** Cho F là tập các phụ thuộc hàm trên tập thuộc tính U , $X \subseteq U$. Bao đóng (closure) của tập các thuộc tính (X) trên tập các phụ thuộc hàm F ký hiệu là X_F^+ được định nghĩa như sau:

$$X_F^+ = \{A / X \rightarrow A \in F^+\}.$$

- Nghĩa là X_F^+ là tập tất cả các thuộc tính A mà phụ thuộc hàm $X \rightarrow A$ có thể được suy diễn logic từ F nhờ hệ tiên đề Armstrong.

Bổ đề 3: $X \rightarrow Y$ suy diễn từ hệ tiên đề Armstrong khi và chỉ khi $Y \subseteq X_F^+$.

- Để tiện ta có thể viết X^+ thay cho X_F^+

- Thuật toán tìm bao đóng:

Thuật toán này được áp dụng để tìm bao đóng của tập các thuộc tính đối với tập các phụ thuộc hàm.

Việc tính toán bao đóng F^+ của tập các phụ thuộc hàm F trong trường hợp tổng quát là rất khó khăn và tốn kém thời gian bởi vì tập các phụ thuộc hàm thuộc F^+ rất lớn cho dù F có thể là rất nhỏ. Nhưng việc tính X^+ , bao đóng của tập thuộc tính X lại không khó. Tính bao đóng X^+ được thể hiện qua thuật toán sau đây:

Thuật toán: Tính bao đóng của tập các thuộc tính đối với một tập các phụ thuộc hàm.

Vào: Tập U hữu hạn các thuộc tính, tập các phụ thuộc hàm F trên U và $X \subseteq U$

Ra: X^+ , bao đóng của X đối với F .

Phương pháp: Tính liên tiếp tập các thuộc tính X_0, X_1, \dots theo qui tắc:

$$1. X_0 = X$$

$$2. X_{i+1} = X_i \cup A \text{ sao cho } \exists (Y \rightarrow Z) \in F, A \in Z \text{ và } Y \subseteq X_i.$$

Vì rằng $X = X_0 \subseteq \dots \subseteq U$, U là hữu hạn cho nên sẽ tồn tại một chỉ số i nào đó mà $X_i = X_{i+1}$ khi đó $X^+ = X_i$.

- Ví dụ: Cho F là tập các phụ thuộc hàm sau đây:

$AB \rightarrow C, D \rightarrow EG, ACD \rightarrow B, C \rightarrow A, BE \rightarrow C, CE \rightarrow AG, BC \rightarrow D, CG \rightarrow BD.$

$X = BD$. Tìm bao đóng X^+ của X .

Áp dụng thuật toán tìm bao đóng trên ta đặt $X_0 = BD$. Chọn phụ thuộc hàm có vế trái chứa B, D hoặc BD , có $D \rightarrow EG$. Khi đó có $X_1 = BDEG$.

Tìm phụ thuộc hàm số có vế trái chứa trong X_1 (có $D \rightarrow EG, BE \rightarrow C$).

Tính tiếp ta được: $X_2 = BCDEG$

$$X_3 = X_4 = ABCDEG.$$

$$\text{Vậy } X^+ = (BD)^+ = ABCDEG.$$

1.2.3. Khóa của lược đồ quan hệ.

- **Định nghĩa:** Cho lược đồ quan hệ R với các thuộc tính $U = \{A_1, A_2, \dots, A_n\}$ và tập phụ thuộc hàm $F, X \subseteq U$. Ta nói X là một khóa của R nếu:

a. $X \rightarrow U \in F^+$ nghĩa là X xác định hàm tất cả các thuộc tính (các phụ thuộc hàm này thuộc F hoặc được suy diễn logic từ F).

b. $\neg \exists Y \subset X, Y \neq \emptyset$ mà $Y \rightarrow U \in F^+$ (đọc là không tồn tại một tập con $Y \neq \emptyset$ nào của X mà Y xác định hàm tất cả các thuộc tính).

- Muốn chứng minh X là khóa của R , phải thực hiện các bước sau:

$$\checkmark \text{ Chứng minh } X \rightarrow U \in F^+ \Leftrightarrow X^+ = U.$$

$$\checkmark \text{ Chứng minh } \forall Y \subset X, Y \neq \emptyset \text{ ta có } Y^+ \neq U.$$

$$\Leftrightarrow A \subset X, X - A \neq \emptyset \text{ và } \{X - A\}^+ \neq U.$$

- Muốn tìm khóa của một lược đồ quan hệ ta dựa vào thuật toán tìm khóa.

- **Thuật toán tìm khóa:**

✓ Vào: Lược đồ quan hệ R với tập thuộc tính U và tập phụ thuộc hàm F .

✓ Ra: Tập K là khóa của R .

- **Phương pháp:**

✓ Đặt $K=U$.

✓ Lặp lại quá trình loại khỏi K thuộc tính A mà $\{K - A\}^+ = U$.

- Mô tả thuật toán:

Begin

$K:=U$;

FOR each A IN K DO

IF $\{K - A\}^+ = U$ THEN $K:=K - A$

END.

- Thuật toán trên tìm được một khóa của R . Muốn tìm được các khóa khác (nếu có) ta có thể thay đổi thứ tự loại bỏ các thuộc tính trong K .
- Ví dụ: Cho lược đồ quan hệ R trên tập thuộc tính $U=\{A,B,C,D\}$ với các phụ thuộc hàm: $A \rightarrow C$ và $B \rightarrow D$. Hãy tìm khóa của lược đồ quan hệ trên.
- Áp dụng thuật toán trên ta đặt $K=U$ và lần lượt loại bỏ các thuộc tính trong K như sau:

Loại bỏ D : Ta có $ABC^+ = U$ (vì $A \rightarrow C, B \rightarrow D$) nên $K=ABC$.

Loại bỏ C : Ta có $AB^+ = U$ (vì $A \rightarrow C, B \rightarrow D$) nên $K=AB$.

Loại bỏ B : Ta có $A^+ = AC \neq U$ (vì $A \rightarrow C$) nên $K=AB$.

Loại bỏ A : Ta có $B^+ = BD \neq U$ (vì $B \rightarrow D$) nên $K=AB$.

Vậy $K=AB$ là khóa của R .

- Từ thuật toán trên ta có nhận xét sau:

- + Các thuộc tính không xuất hiện trong cả vế trái và vế phải của tập phụ thuộc hàm F phải có trong khóa.
- + Các thuộc tính chỉ xuất hiện bên vế trái của các phụ thuộc hàm trong F cũng phải thuộc khóa.

2. Phép tách các lược đồ quan hệ.

- Phép tách một lược đồ quan hệ $R = \{A_1, \dots, A_n\}$ là việc thay thế lược đồ quan hệ R bằng các tập lược đồ R_1, \dots, R_k , trong đó $R_i \subseteq R, i = 1, 2, \dots, k$ và $R = R_1 \cup R_2 \cup R_3 \dots \cup R_k$.
- Ở đây không đòi hỏi các lược đồ R_i phải là phân biệt. Mục tiêu của phép tách chủ yếu là loại bỏ các dị thường dữ liệu gây ra.

- Ví dụ: Cho lược đồ quan hệ nhà cung cấp như sau:

NCC(TEN,DCHI,HANG,GIA)

Và giả sử có các phụ thuộc hàm $TEN \rightarrow DCHI$; $TEN, HANG \rightarrow GIA$.

Lược đồ NCC có thể được thay thế bởi hai lược đồ khác là:

$R_1(TEN, DCHI)$ và $R_2(TEN, HANG, GIA)$.

- Vấn đề đặt ra là liệu một cơ sở dữ liệu r thỏa trên R khi tách ra R_1, \dots, R_k có còn phù hợp không. Để nghiên cứu các điều kiện cho kết quả phép kết nối tự nhiên nêu trên là duy nhất và bằng quan hệ ban đầu, ta cần đưa thêm một số khái niệm sau đây.

2.1. Phép tách có kết nối không mất mát thông tin.

- Nếu R là một lược đồ quan hệ được tách thành các lược đồ con R_1, R_2, \dots, R_k và F là tập các phụ thuộc dữ liệu, nói rằng phép tách là tách – kết nối không mất mát thông tin đối với F nếu với mỗi quan hệ r trên R thỏa F :

$$r = \prod_{R_1}(r) * \prod_{R_2}(r) * \dots * \prod_{R_k}(r).$$

Tức là r được tạo nên từ phép kết nối tự nhiên của các hình chiếu của nó trên các R_i , $i = 1, 2, \dots, k$.

- Tính chất:

Tập các lược đồ $\rho = (R_1, R_2, \dots, R_k)$ được thay thế cho lược đồ R . Gọi m_ρ là ánh xạ xác định $m_\rho(r) = * \prod_{R_i}(r)$, $i=1, \dots, k$ có nghĩa là $m_\rho(r)$ là kết nối của các phép chiếu của r trên các lược đồ con trong ρ . Điều kiện để kết nối không mất mát thông tin đối với F được biểu diễn như sau:

Với mọi r thỏa F , $r = m_\rho(r)$.

2.2. Kiểm tra phép kết nối không mất mát thông tin.

Liệu một phép tách có phép kết nối không mất mát thông tin hay không đối với tập các phụ thuộc hàm được kiểm tra qua thuật toán sau đây.

- Thuật toán:

Vào: Lược đồ quan hệ $R = \{A_1, \dots, A_n\}$, tập các phụ thuộc hàm F và phép tách $\rho = (R_1, R_2, \dots, R_k)$.

Ra: Kết luận phép tách ρ có phải là không mất mát thông tin.

- Phương pháp:

Thiết lập một bảng với n cột và k hàng, cột thứ j ứng với thuộc tính A_j , hàng thứ i ứng với lược đồ R_i . Tại hàng i và cột j điền ký hiệu a_j nếu $A_j \in R_i$, nếu không điền ký hiệu b_{ij} .

Bây giờ xem xét đến các phụ thuộc hàm từ F áp dụng cho bảng vừa thiết lập được. Giả sử xem xét $(X \rightarrow Y) \in F$. Xét đến các hàng và nếu có giá trị bằng nhau trên thuộc tính X (ví dụ t_1, t_2 mà $t_1[X] = t_2[X]$) thì làm bằng giá trị của chúng trên Y (ví dụ $t_1[Y] = t_2[Y]$). Chú ý là khi làm bằng giá trị của chúng trên Y , nếu một trong hai giá trị là a_j thì ưu tiên làm bằng ký hiệu là a_j . Nếu không làm bằng chúng bằng một trong các ký hiệu là b_{ij} .

Tiếp tục áp dụng các phụ thuộc hàm cho bảng (kể cả việc lập lại phụ thuộc hàm đã được áp dụng) cho tới khi không còn áp dụng được nữa.

Xem xét bảng kết quả: Nếu xuất hiện một hàng gồm các ký hiệu a_1, a_2, \dots, a_n thì phép kết nối là không mất mát thông tin. Trường hợp ngược lại là kết nối mất mát thông tin.

- Ví dụ: Xem xét ở quan hệ nhà cung cấp.

NCC (NCC, DCHI, HANG, GIA)

Được tách thành hai quan hệ $R_1(\text{NCC}, \text{DCHI})$ và $R_2(\text{NCC}, \text{HANG}, \text{GIA})$ với các phụ thuộc hàm:

$\text{NCC} \rightarrow \text{DCHI}; \text{NCC}, \text{HANG} \rightarrow \text{GIA}$

Bảng ban đầu được thiết lập như sau:

NCC	DCHI	HANG	GIA
a_1	a_2	b_{13}	b_{14}
a_1	b_{22}	a_3	a_4

Áp dụng phụ thuộc hàm $\text{NCC} \rightarrow \text{DCHI}$ cho hai hàng của bảng. Hai hàng bằng nhau trên cột NCC (đều bằng a_1) nên ở cột DIACHI chúng được làm bằng và bằng a_2 .

Bảng kết quả là:

NCC	DIACHI	MAHANG
GIA		

a_1	a_2	b_{13}	b_{14}
a_1	a_2	a_3	a_4

Bảng kết quả có dòng thứ hai có giá trị toàn là a, do đó nó là kết nối không mất mát thông tin.

3. Chuẩn hoá lược đồ quan hệ.

Nếu một lược đồ quan hệ thiết kế không tốt sẽ gây ra những dị thường dữ liệu như: dư thừa dữ liệu, và do việc cập nhật dữ liệu (qua các phép tính chèn, loại bỏ hay thay đổi dữ liệu). Để tránh dị thường dữ liệu lược đồ quan hệ cần thiết phải được biến đổi thành các dạng phù hợp. Quá trình đó được gọi là quá trình chuẩn hoá lược đồ quan hệ. Lược đồ quan hệ được chuẩn hoá là lược đồ quan hệ mà trong đó miền của mỗi thuộc tính chỉ chứa những giá trị đơn tức là không phân nhỏ được nữa. Ngược lại gọi là lược đồ quan hệ không chuẩn hoá. Quá trình chuẩn hoá một lược đồ quan hệ có thể thành một hoặc nhiều lược đồ quan hệ chuẩn hoá khác và có kết nối không mất mát thông tin. Một lược đồ R là chuẩn hoá thì quan hệ r của nó cũng là chuẩn hoá.

Ví dụ:

Trước khi mô tả các dạng chuẩn hoá, cần biết một số khái niệm sau:

- Cho một lược đồ quan hệ R với tập thuộc tính $U = \{A_1, A_2, \dots, A_n\}$. Thuộc tính A_i ($i=1, \dots, n$) được gọi là thuộc tính khoá nếu A_i thuộc vào một khoá nào đó của R. Ngược lại thì A_i không phải là thuộc tính khoá, hay là thuộc tính không khoá.

- Ví dụ: Cho lược đồ quan hệ R trên tập thuộc tính $U=\{A, B, C, D\}$ với các phụ thuộc hàm $A \rightarrow C; B \rightarrow D$. Qua thuật toán tìm khoá ta xác định được AB là khoá của lược đồ R, Khi đó các thuộc tính A,B là các thuộc tính khoá còn C,D là các thuộc tính không khoá.
- Cho lược đồ quan hệ R xác định trên tập thuộc tính U. $X, Y \subseteq U$ và phụ thuộc hàm $X \rightarrow Y$, ta nói Y phụ thuộc hàm đầy đủ vào X nếu và chỉ nếu không tồn tại tập con W của X để $W \rightarrow Y$. Nếu có tập con W của X mà $W \rightarrow Y$ thì ta nói Y phụ thuộc hàm một phần vào X (hay phụ thuộc hàm không đầy đủ).
- Ví dụ: Cho lược đồ quan hệ R trên tập thuộc tính $U=\{A, B, C, D\}$ và các phụ thuộc hàm $AB \rightarrow C, AB \rightarrow D, A \rightarrow D$. Trong các phụ thuộc hàm này ta nói C phụ thuộc hàm đầy đủ vào AB, còn D phụ thuộc một phần vào AB.

3.1. Các dạng chuẩn.

- Để đánh giá một cách cụ thể chất lượng thiết kế của một lược đồ cơ sở dữ liệu, lúc ban đầu E.F.Codd tác giả của mô hình dữ liệu quan hệ đã đưa ra ba dạng chuẩn và sau đó hai ông R.F.Boyce và E.F.Codd cải tiến dạng chuẩn thứ ba gọi là dạng chuẩn BCNF.
- Các dạng chuẩn được định nghĩa dựa trên khái niệm phụ thuộc hàm.

3.1.1. Dạng chuẩn thứ nhất (First Normal Form – 1NF).

- **Định nghĩa:** Một lược đồ quan hệ R với tập thuộc tính U được gọi là ở dạng chuẩn thứ nhất (1NF) nếu miền của mỗi thuộc tính trong U chỉ chứa những giá trị đơn.
- Từ định nghĩa này cho thấy bất kỳ một lược đồ quan hệ chuẩn hoá nào cũng đều ở dạng chuẩn 1NF và quan hệ r của nó cũng ở dạng chuẩn này.

3.1.2. Dạng chuẩn thứ hai (Second Normal Form – 2NF).

- **Định nghĩa:** Một lược đồ quan hệ R với tập thuộc tính U được gọi là ở dạng chuẩn thứ hai (2NF) nếu R ở dạng chuẩn thứ nhất và mọi thuộc tính không khoá của R đều phụ thuộc hàm đầy đủ vào thuộc tính khoá. Hay nói cách khác mọi thuộc tính không khoá của R không phụ thuộc hàm vào một tập con nào của khoá.

- Ví dụ: Cho lược đồ quan hệ $R(ABCD)$ với các phụ thuộc hàm: $AC \rightarrow D$ và $A \rightarrow B$. R không phải ở dạng chuẩn 2NF.

Thật vậy, ta có AC là khoá của R và B, D là thuộc tính không khoá, đặt $X=AC, Y=A, Y \subset X$ (khoá) mà $Y \rightarrow B$ do đó R không ở dạng chuẩn 2NF.

3.1.3. Dạng chuẩn thứ ba (Third Normal Form – 3NF).

- **Định nghĩa:** Một lược đồ quan hệ R với tập thuộc tính U được gọi là ở dạng chuẩn thứ ba (3NF) nếu R ở dạng chuẩn thứ hai và không có một tập con X nào của U mà X không phải là khoá lại xác định hàm các thuộc tính không khoá của R . Hay nói cách khác, mọi thuộc tính không khoá đều không phụ thuộc bất cầu vào thuộc tính khoá.
- Ví dụ: Cho lược đồ quan hệ $R(BDW)$ với phụ thuộc hàm $BD \rightarrow W$, ta có BD là khoá của R , và W là thuộc tính không khoá cho thấy R ở dạng chuẩn 3NF.
- Ví dụ: Cho lược đồ $R(SIDM)$ – 1NF và các phụ thuộc hàm $SI \rightarrow D, SD \rightarrow M$, ta có SI là khoá, thuộc tính không khoá là D, M , cho thấy R ở dạng chuẩn 2NF nhưng không phải 3NF vì $SD \rightarrow M$.

4. Bài tập.

1. Nêu định nghĩa, tính chất phụ thuộc hàm, bao đóng F^+ của tập phụ thuộc hàm F , bao đóng X^+ của tập thuộc tính X , cho ví dụ.
2. Nêu định nghĩa khoá của lược đồ quan hệ, cho ví dụ.
3. Nêu thuật toán tìm khoá, thuật toán tìm bao bao đóng, cho ví dụ.
4. Nêu định nghĩa các dạng chuẩn 1NF, 2NF, 3NF, cho ví dụ.
5. Trình bày phép tách lược đồ quan hệ có kết nối không mất mát thông tin, cho ví dụ.
6. Cho quan hệ sau:

r(A B C D E)				
a1	b1	c1	d1	e1
a1	b2	c2	d2	d1
a2	b1	c3	d3	e1
a2	b1	c4	d3	e1
a3	b2	c5	d1	e1

Phụ thuộc hàm nào sau đây thỏa r:

$A \rightarrow D, AB \rightarrow D, C \rightarrow BDE, E \rightarrow A, A \rightarrow E.$

7. Cho lược đồ quan hệ và tập các phụ thuộc hàm:

a. $F = \{AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$ trên R. Chứng minh rằng: $AB \rightarrow GHI.$

$AB \rightarrow E, E \rightarrow G \Rightarrow AB \rightarrow G$ (BẮT CẦU)

$AB \rightarrow A$ (PHẢN XẠ)

$\Rightarrow AB \rightarrow AG$ (TÍNH HỢP)

$AG \rightarrow I$ (GIẢ THIẾT)

$\Rightarrow AB \rightarrow I$ (BẮT CẦU)

$\Rightarrow AB \rightarrow GI$ (TÍNH HỢP)

$GI \rightarrow H$ (GIẢ THIẾT)

$\Rightarrow AB \rightarrow H$ (BẮT CẦU)

$\Rightarrow AB \rightarrow GHI$ (TÍNH HỢP)

b. $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$ trên R. Chứng minh rằng: $AB \rightarrow E, AB \rightarrow G.$

• $AB \rightarrow E$

▪ $AB \rightarrow B$

▪ $B \rightarrow D$

▪ $AB \rightarrow D$

▪ $AB \rightarrow C$ (GIẢ THIẾT)

▪ $AB \rightarrow CD$ (HỢP)

▪ $CD \rightarrow E$ (GIẢ THIẾT)

▪ $AB \rightarrow E$ (BẮT CẦU)

• $AB \rightarrow G$

▪ $AB \rightarrow C$ (GT)

▪ $AB \rightarrow E$ (TỪ CÂU A)

▪ $AB \rightarrow CE$ (HỢP)

▪ $CE \rightarrow GH$ (GT)

▪ $AB \rightarrow GH$ (BẮT CẦU)

▪ $AB \rightarrow G$ (PHÂN RÃ)

8. Cho lược đồ quan hệ R(ABCD) và tập phụ thuộc hàm F, phụ thuộc hàm nào sau đây được suy dẫn từ F.

a. $F = \{A \rightarrow B, A \rightarrow C\}$ và các phụ thuộc hàm $A \rightarrow D, C \rightarrow D, BC \rightarrow A, A \rightarrow BC.$

b. $F = \{A \rightarrow B, BC \rightarrow D\}$ và các phụ thuộc hàm $C \rightarrow D, A \rightarrow D, AD \rightarrow C, BC \rightarrow A, B \rightarrow CD.$

9. Cho lược đồ quan hệ $R(ABCDEF)$ và tập phụ thuộc hàm $F=\{AB \rightarrow C, C \rightarrow D, D \rightarrow E, DE \rightarrow F\}$. Hãy tìm khoá của R .

- ✓ $K = U(ABCDEF)$
- ✓ Loại $F, (abcde)^+ = U$ (do $DE \rightarrow F$). $k = abcde$
- ✓ Loại $e, (abcd)^+ = U$ (do $d \rightarrow e, de \rightarrow F$). $K = abcd$
- ✓ Loại $D, (abc)^+ = U$ (do $c \rightarrow d, d \rightarrow e, de \rightarrow f$). $K = abc$
- ✓ Loại $C, (ab)^+ = U$ (do $AB \rightarrow C, \dots$). $K = ab$
- ✓ Loại $B, (a)^+ \neq U$. $K = ab$
- ✓ Loại $A, (b)^+ \neq U$. $K = ab$
- ✓ vậy khóa của R là $K = AB$.

10. Cho lược đồ quan hệ $R(ABCD)$ và tập phụ thuộc hàm $F=\{AB \rightarrow C, D \rightarrow B, C \rightarrow ABD\}$. Hãy tìm khoá của R .

- ✓ Đặt $K = U(ABCD)$
- ✓ Loại $D, (ABC)^+ = U$ (do $C \rightarrow ABD$). $K = (ABC)$
- ✓ Loại $C, (AB)^+ = U$ (do $AB \Rightarrow C, C \rightarrow ABD$). $K = (AB)$
- ✓ Loại B : Không được
- ✓ Loại A : Không được
- ✓ Vậy $K = AB$

11. Cho lược đồ quan hệ $R(ABCDEFG)$ và tập phụ thuộc hàm $F=\{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C, CG \rightarrow BD, E \rightarrow G\}$. Tìm khoá của R . chứng minh $ab \rightarrow g$. bao đóng ab

- ✓ Đặt $K = U(ABCDEFG)$
- ✓ Loại $G, (ABCDE)^+ = U$ (do $E \rightarrow G$). $K = ABCDE$.
- ✓ Loại $E, (ABCD)^+ = U$ (do $D \rightarrow EG, D \rightarrow E$ và $D \rightarrow G$). $K = ABCD$.
- ✓ Loại $D, (ABC)^+ = U$ (do $BC \rightarrow D, D \rightarrow \dots$). $K = ABC$
- ✓ Loại $C, (AB)^+ = U$ (do $AB \rightarrow C, BC \rightarrow D, \dots$). $K = AB$
- ✓ Loại $B, (A)^+ \neq U$. $K = AB$
- ✓ Loại $A, (B)^+ \neq U$. $K = AB$

12. Cho lược đồ quan hệ R và tập phụ thuộc hàm F . Các phép tách sau có mất mát thông tin không? Với:
- $R = \{A, B, C\}$ và $F = \{A \rightarrow B\}$, $\rho = (R_1, R_2)$, $R_1 = \{A, B\}$, $R_2 = \{A, C\}$.
 - $R = \{A, B, C\}$ và $F = \{A \rightarrow B, A \rightarrow C\}$.
 - $\rho = \{R_1, R_2\}$, $R_1 = \{A, B\}$, $R_2 = \{A, C\}$.
13. Cho lược đồ quan hệ R và tập phụ thuộc hàm F . Phép tách sau có mất mát thông tin không? Với:
- $R = \{A, B, C, D, E, G, H, I, K, L\}$ và $F = \{AB \rightarrow D, DE \rightarrow H, IK \rightarrow L, LB \rightarrow C\}$, $\rho = \{R_1, R_2, R_3\}$, $R_1 = \{A, B, C\}$, $R_2 = \{C, D, E, H\}$, $R_3 = \{E, H, I, K, L\}$.
 - $R = \{A, B, C, D, E, G\}$ và $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E, DE \rightarrow F\}$, $\rho = \{R_1, R_2, R_3, R_4\}$, $R_1 = \{A, B, C\}$, $R_2 = \{C, D\}$, $R_3 = \{D, E\}$, $R_4 = \{D, E, F\}$.
14. Cho lược đồ quan hệ $R(ABCD)$, $F = \{AB \rightarrow C, D \rightarrow B, C \rightarrow ABD\}$ và phép tách $\rho = (ABC, BCD)$. Hãy kiểm tra phép tách trên có kết nối mất mát thông tin không?
15. Cho lược đồ quan hệ $R(ABCDEFG)$ và $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C, CG \rightarrow BD, E \rightarrow G\}$, $\rho = (ABC, BCD, DEG)$. Hãy kiểm tra phép tách trên có kết nối mất mát thông tin không?.

THỰC HÀNH**A. GIỚI THIỆU VỀ SQL SERVER 2000.**

SQL server 2000 ra đời tối ưu hơn nhiều cơ sở dữ liệu quan hệ, khắc phục được những thiếu sót và những vấn đề chưa giải quyết một cách triệt để trong các phiên bản SQL trước đây, đồng thời bổ sung thêm một số tính năng mới đáp ứng được nhu cầu truy xuất ngày càng đa dạng của người sử dụng.

1. Khái niệm cơ bản về RDBMS (Relation Database Manager System).

RDBMS – hệ thống quản lý cơ sở dữ liệu quan hệ, không những lưu trữ dữ liệu mà còn quản trị hệ CSDL bằng cách kiểm soát những dữ liệu nào được cập nhật vào và những dữ liệu nào có thể truy xuất ra khỏi hệ thống. RDBMS cho phép lưu trữ CSDL cùng với những nguyên tắc ràng buộc dữ liệu do người dùng hay hệ thống tự định nghĩa.

2. Các thành phần của SQL server 2000.

RDBMS cũng như SQL server chứa đựng nhiều đối tượng bao gồm :

- ✓ Database : cơ sở dữ liệu của SQL server.
- ✓ Tập tin log: tập tin lưu trữ những chuyển tác của SQL.
- ✓ Tables: bảng dữ liệu.
- ✓ File groups: tập tin nhóm.
- ✓ Diagrams : sơ đồ quan hệ.
- ✓ Views: khung nhìn.
- ✓ Stored procedure: thủ tục và hàm nội.
- ✓ Users defined function: hàm do người dùng tự định nghĩa.
- ✓ User: người sử dụng CSDL.
- ✓ Roles những qui tắc.
- ✓ Defaults : các giá trị mặc nhiên.

- ✓ User – defined data types : kiểu dữ liệu do người dùng tự định nghĩa.
- ✓ Full – text catalogs: tập phân loại dữ liệu text.

3. Đối tượng cơ sở dữ liệu.

Cơ sở dữ liệu là đối tượng có ảnh hưởng cao nhất khi ta làm việc với SQL server. Tuy nhiên những đối tượng con của cơ sở dữ liệu mới là thành phần chính của cơ sở dữ liệu.

- Khi cài đặt SQL server ta có sáu cơ sở dữ liệu mặc định như sau:

- ✓ Master.
- ✓ Model.
- ✓ Msdb.
- ✓ Tempdb.
- ✓ Pubs.
- ✓ Northwind.

3.1. Cơ sở dữ liệu Master.

- Bất kỳ hệ SQL server nào cũng đều có CSDL master còn được gọi là master file. Cơ sở dữ liệu này chứa đựng tất cả các bảng dữ liệu đặc biệt (bảng hệ thống). Chúng kiểm soát tất cả các hoạt động của SQL server.
- Ví dụ: Khi người dùng tạo một CSDL mới trong SQL server, thêm hay xóa một stored procedure, tất cả những thông tin này đều được lưu trữ trong cơ sở dữ liệu master của hệ thống.

3.2. Cơ sở dữ liệu model.

- Cơ sở dữ liệu này chứa đựng tất cả các template dùng làm mẫu để tạo cơ sở dữ liệu mới. Khi ta tạo mới một cơ sở dữ liệu thì SQL server lấy tất cả các mẫu (bao gồm table, view,...) từ cơ sở dữ liệu model này. Do đó ta không thể xóa cơ sở dữ liệu này.
- Khi một cơ sở dữ liệu mới được tạo ra thì ít nhất cũng bằng và giống cơ sở dữ liệu model.

3.3. Cơ sở dữ liệu msdb.

Cơ sở dữ liệu msdb hỗ trợ quá trình SQL agent lưu trữ tất cả các tác vụ xảy ra trong SQL server. Ví dụ như khi ta tạo một lập trình cho backup dữ

liệu hay lịch trình để thực thi stored procedure. Tất cả các tác vụ này đều được lưu trữ vào msdb. Nếu CSDL này bị xóa ta phải cài đặt lại nó khi cần dùng hoặc khi hệ thống yêu cầu.

3.4. Cơ sở dữ liệu Tempdb.

Đây là một trong những CSDL chính của SQL server. CSDL này cho phép người dùng tạo những tham khảo hay thực tập trước khi vào CSDL thực sự. Ngoài ra nó còn dùng làm bộ đệm CSDL cho các CSDL khác trong SQL server và giúp thực hiện những thao tác về CSDL mỗi khi SQL server khởi động.

3.5. Cơ sở dữ liệu pubs.

CSDL này chứa hầu hết nội dung về hướng dẫn, trợ giúp và cả sách tham khảo về SQL server. Ta có thể xóa CSDL này mà không ảnh hưởng đến hệ thống.

3.6. Cơ sở dữ liệu Northwind.

Cũng giống như CSDL pubs, đây là CSDL mẫu cho người dùng tham khảo. CSDL này chứa đựng những đối tượng mẫu và một số dữ liệu nhằm giúp cho việc xử lý, thực nghiệm trên SQL server cũng như thông qua các ứng dụng khác như Visual Basic, Java, C⁺⁺.

4. Các đối tượng của một CSDL.

4.1. Bảng (table).

- Bảng là phần chính của CSDL, là đối tượng lưu trữ dữ liệu thực, khi cần giao tiếp với CSDL khác thì bảng là đối tượng căn bản nhất trong bất kỳ loại CSDL nào, chúng được coi như một miền dữ liệu. Mỗi bảng được định nghĩa nhiều trường (field hay column name). Mỗi trường ứng với một loại kiểu dữ liệu. Dữ liệu nhập vào có thể được chấp nhận hoặc từ chối tùy thuộc vào nguyên tắc ràng buộc dữ liệu hoặc loại dữ liệu tương thích do người sử dụng định nghĩa.
- Khi định nghĩa bảng số liệu ta cần quan tâm đến các yếu tố sau:
 - Key: Trường đó có khóa hay không.
 - ID: Trường có thuộc tính identity hay không.
 - Column name: Tên của trường (cột).

- Data type: Loại dữ liệu cho trường tương ứng.
- Size: Kích thước trường dữ liệu.
- Allow null: Cho phép có giá trị rỗng (null) lưu trong trường hay không.
- Default : Giá trị mặc nhiên cho trường.
- Identity: Nếu cần sử dụng một trường có giá trị tự động, trường này not null và identity là yes (on).
- Identity seed: Nếu trường này là identity thì số bắt đầu phải là số 1 hoặc 2.
- Identity increament: Số nhảy cho mỗi lần tăng.

4.2. Chỉ mục indexes.

Index hay còn gọi là chỉ mục, đối tượng này chỉ tồn tại trong bảng hay khung nhìn (view). Chỉ mục có ảnh hưởng lớn đến tốc độ truy cập số liệu nhất là khi tìm kiếm thông tin trên bảng do vậy chỉ mục giúp tăng tốc cho việc tìm kiếm.

4.3. Bẫy lỗi: (Triggers).

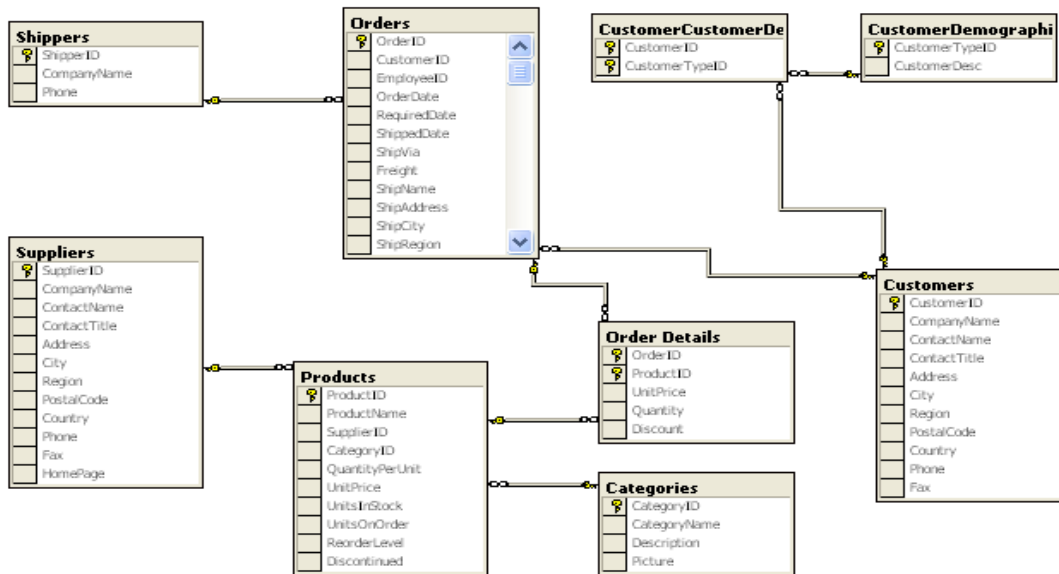
Trigger là một đoạn mã chương trình, chỉ tồn tại trong bảng và tự động thực thi khi có một hành động nào đó xảy ra đối với dữ liệu trong bảng như: insert, update, delete hoặc kiểm tra dữ liệu theo một tiêu chuẩn nào đó.

4.4. Ràng buộc constraints.

Constraints là một đối tượng ràng buộc dữ liệu trong một bảng hoặc nhiều bảng với nhau và tuân theo một qui tắc nào đó.

4.5. Diagram – Lược đồ quan hệ.

Diagram là công cụ giúp ta kết nối các bảng với nhau.



4.6. Khung nhìn (view).

View là một khung nhìn (hay bảng ảo) của bảng, cũng giống như bảng nhưng view không thể chứa dữ liệu. Bản thân view có thể tạo thêm trường mới dựa vào những phép toán, biểu thức của SQL. Ngoài ra view còn có thể kết nối nhiều bảng lại với nhau theo quan hệ nhất định cùng với những tiêu chuẩn nhằm tạo ra một bảng dữ liệu theo yêu cầu của người dùng. Trong SQL server 2000, mục đích của view là kiểm soát tất cả những gì mà người sử dụng muốn thấy. Nó bao gồm hai ảnh hưởng chính đó là tính bảo mật và dễ sử dụng.

4.7. Thủ tục nội – stored procedure.

Stored procedure (spoc) cho phép khai báo biến, nhận tham số cũng như thực thi các phát biểu có điều kiện. Ta có thể gọi sử dụng stored procedure theo cách gọi của thủ tục hay hàm trong các ngôn ngữ lập trình truyền thống và có thể sử dụng lại khi có yêu cầu.

5. Kiểu dữ liệu (Data type).

Bảng dữ liệu:

Kiểu dữ liệu	Loại	Bytes	Diễn giải
Bit	Integer	1	Kiểu dữ liệu bit có giá trị 0 hay 1. tương tự như trong các CSDL khác. 0 là FALSE và 1 là TRUE.
BigInt	Integer	8	Kiểu dữ liệu số nguyên có giá trị từ -2^{63} đến $2^{63} - 1$, thông thường khi cần lưu trữ dữ liệu có dạng số

			nguyên thật lớn thì nên dùng kiểu dữ liệu bigint
Int	Integer	4	Kiểu dữ liệu số nguyên có giá trị từ -2,147,483,648 đến 2,147,483,647. thông thường khi cần lưu trữ dữ liệu dạng số nguyên không lớn lắm thì ta nên dùng kiểu dữ liệu int.
SmallInt	Integer	2	Kiểu dữ liệu số nguyên có giá trị từ -32,168 đến 32,167. thông thường khi cần lưu trữ dữ liệu dạng số nguyên vừa và nhỏ thì ta nên dùng kiểu dữ liệu smallint.
TinyInt	Integer	1	Kiểu dữ liệu số nguyên có giá trị từ 0 đến 255. thông thường khi cần lưu trữ dữ liệu dạng số nguyên nhỏ thì ta nên dùng kiểu dữ liệu tinyint.
Decimal or Numeric	Decimal /Numeric	Varies	Kiểu dữ liệu số từ $-10^{38} - 1$ đến $10^{38} - 1$
Money	Money	8	Kiểu dữ liệu từ -2^{63} đến 2^{63} thêm vào đó 4 số lẻ
SmallMoney	Money	4	Kiểu dữ liệu tiền từ -214,748.3648 đến 214,748.3647
Float	Approximate Numeric	Varies	Kiểu dữ liệu số từ: $-1.79E+308$ đến $1.79E+308$
DateTime	Date/Time	8	Kiểu dữ liệu ngày tháng xuất hiện đầu tiên trong SQL server 7.0 cho phép có giá trị null, có giá trị từ January 1, 1753 đến December 31, 9999
Small DateTime	Date/Time	4	Kiểu dữ liệu ngày tháng xuất hiện đầu tiên trong SQL server 7.0 cho phép có giá trị null, có giá trị từ January 1, 1900 đến December 31, 2079
Cursor	Kiểu số đặc biệt	1	Con trỏ đến cursor, trong khi trỏ đến chúng nhận một byte, khả năng trỏ đến phụ thuộc vào số lượng mẫu tin trong bảng
Unique Identity	Số đặc biệt (binary)	16	Nhận dạng toàn cục duy nhất theo không gian và thời gian

Char	Character	Varies	Kiểu dữ liệu ký tự Non – Unicode với chiều dài cố định, chiều dài cho phép là 8000 ký tự.
Varchar	Character	Varies	Kiểu dữ liệu ký tự Non – Unicode với chiều dài biến đổi, chiều dài cho phép là 8000 ký tự.
Nchar	Unicode	Varies	Kiểu dữ liệu ký tự Unicode với chiều dài cố định, chiều dài cho phép là 4000 ký tự.
Nvarchar	Unicode	Varies	Kiểu dữ liệu ký tự Unicode với chiều dài biến đổi, chiều dài cho phép là 4000 ký tự.
Text	Character	Varies	Kiểu dữ liệu ký tự Non – Unicode với không gian chứa 8 kb số trang, chiều dài cho phép là 2,147,483,647 ký tự.
Ntext	Unicode	Varies	Kiểu dữ liệu ký tự Non – Unicode với không gian chứa 8 kb số trang, chiều dài biến đổi cho phép 1,073,741,823 ký tự.
Binary	Binary	Varies	Dữ liệu nhị phân có chiều dài cố định tối đa 8000 bytes
VarBinary	Binary	Varies	Dữ liệu nhị phân có chiều dài biến đổi tối đa 8000 bytes
Image	Binary	Varies	Dữ liệu nhị phân có chiều dài biến đổi tối đa 2,147,483,647 bytes, không gian chứa 8kb trang
Table	Khác	Đặc biệt	Đây là kiểu dữ liệu hoàn toàn mới trong SQL server 2000, kiểu dữ liệu đặc biệt như là bảng dữ liệu, dạng cơ sở dữ liệu đối tượng. Nó cho phép lưu bảng trong bảng (Hay các bảng có thể lưu lồng nhau)
SQL_Variant	Khác	Đặc biệt	Kiểu dữ liệu này tương tự kiểu Variant trong Visual Basic.

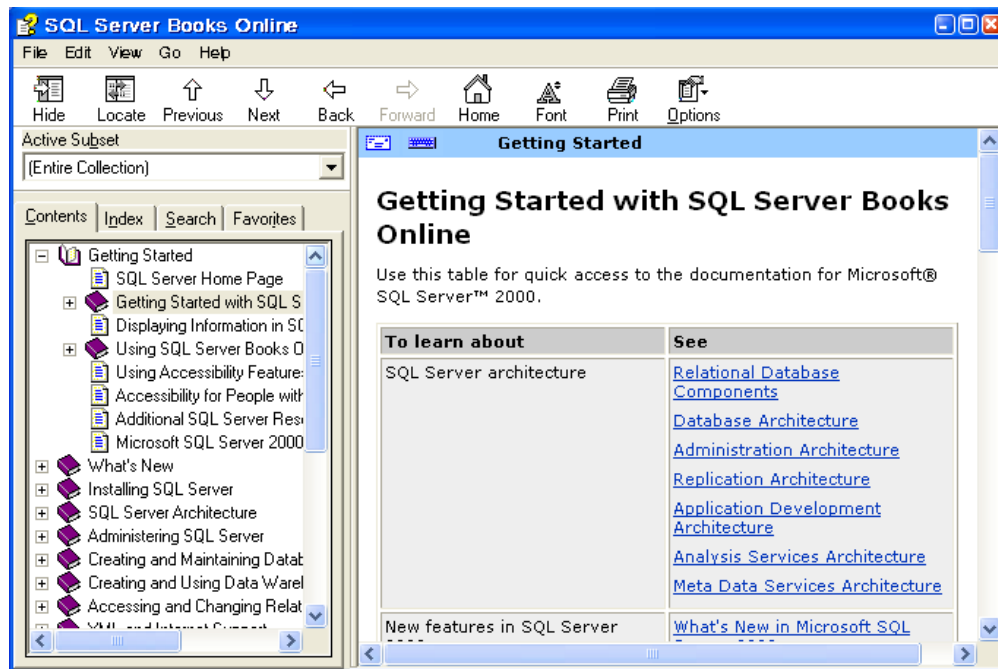
B. CÁC CÔNG CỤ CHÍNH CỦA SQL SERVER 2000

1. Trợ giúp trực tuyến Books Online.

Books Online là công cụ trợ giúp trực tuyến rất hữu ích và cần thiết cho người sử dụng SQL server. Books Online giúp ta truy tìm những vấn đề có

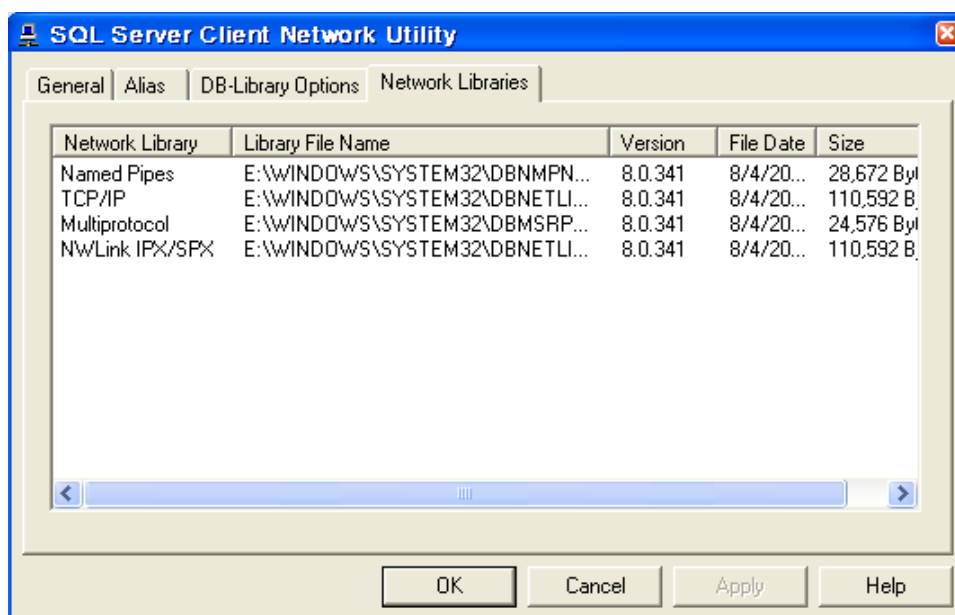
Thực hành trên SQL server

liên quan đến CSDL SQL server. Để vào Books Online ta vào start → program → microsoft SQL server → books online.



2. Tiện ích mạng Client/Server .

Để nhiều máy PC có thể kết nối và sử dụng CSDL SQL server trong hệ thống mạng ta cần cấu hình các tiện ích trên máy trạm giống như cấu hình của server. Để gọi tiện ích cấu hình mạng ta vào: start → program → microsoft SQL server → client network utility.

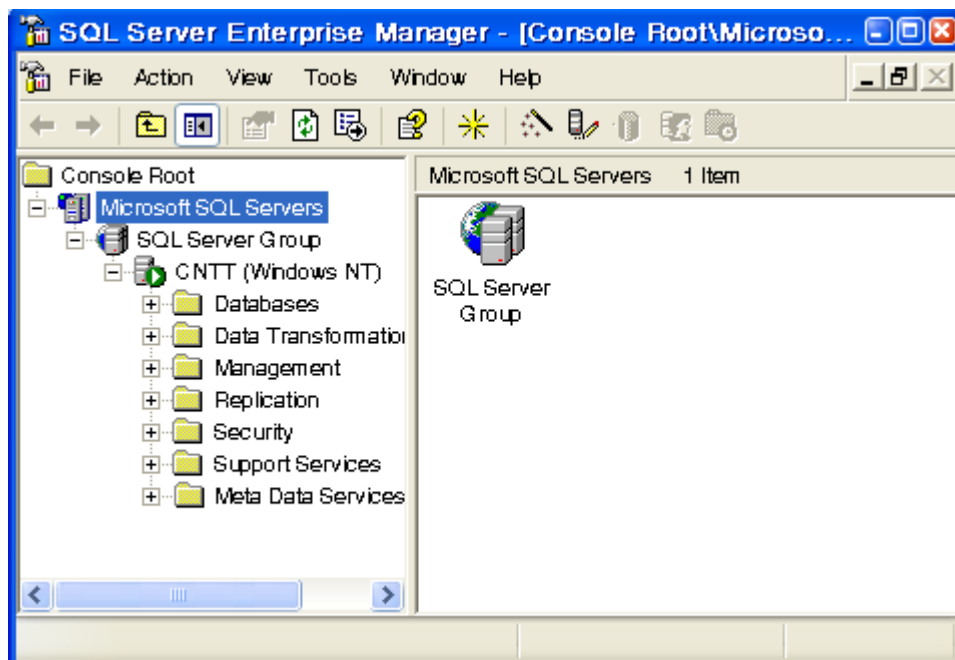


SQL server cung cấp một bộ thư viện Net – Libraries hay còn gọi là NetLibs bao gồm nhiều nghi thức (protocol) kết nối cho phép các máy trạm có thể truy cập CSDL của SQL server. Các nghi thức bao gồm:

- Multiprotocol.
- NWLink IPX/SPX.
- Apple Talk.
- Banyan VINES.
- Shared memory
- VIA.

3. Trình Enterprise Mananger.

Enterprise Mananger gọi tắt là EM, là màn hình điều khiển khi quản trị SQL server. Em cung cấp cho người quản trị nhiều chức năng để quản lý SQL server bằng giao diện đồ họa. Vào Enterprise Mananger như sau: start → program → microsoft SQL server → Enterprise Mananger.



Các chức năng trong Enterprise Mananger:

- Tạo, cập nhật, xóa CSDL và đối tượng của chúng.
- Tạo, cập nhật, xóa các gói Data Transformation packages.
- Quản lý lịch trình backup dữ liệu.

- Quản lý người dùng đang truy cập SQL server và CSDL.
- Cấu hình job management.
- Tạo, cập nhật, xóa quyền login user.
- Thiết lập và quản lý mail server.
- Tạo và quản lý tìm kiếm.
- Cấu hình cho server.
- Tạo và cấu hình cho replication.

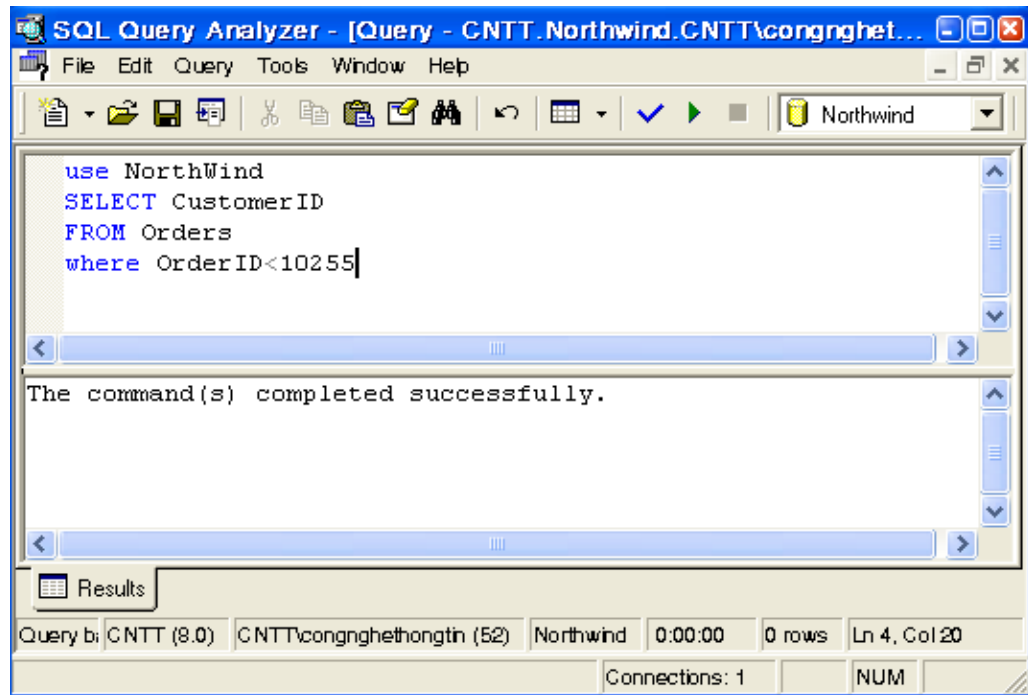
Ngoài ra Enterprise Manager còn có một vài chức năng khác như: đăng ký nhiều SQL server khác, SQL server profiler, Query Analyzer,...




4. Trình Query Analyzer.

Query Analyzer là công cụ được cung cấp bởi SQL server, là trình soạn thảo và thực thi câu lệnh SQL hay stored procedure. Để sử dụng trình này đòi hỏi ta phải kết nối vào SQL server. Mỗi khi kích hoạt chúng thì phải cung cấp tên server của SQL server muốn kết nối. Đồng thời cung cấp username và password để đăng nhập.



Khi cung cấp username và password, chúng phụ thuộc vào sự xác nhận của hai yếu tố: Nếu username và password đăng nhập vào kế thừa username và password của window NT/2000 thì Authentication là windows Authentication, ngược lại thì chọn SQL server Authentication. Nếu kích hoạt Query Analyzer từ Enterprise Manager thì yêu cầu trên không cần thiết. Còn nếu kích hoạt từ menu start → program → microsoft SQL server → Query Analyzer thì yêu cầu trên xuất hiện. Màn hình soạn thảo như sau:



- Để thực hiện kiểm tra một câu lệnh SQL ta ấn phím Ctrl+F5 hoặc nhấp chuột vào công cụ  trên thanh công cụ.
- Để thực thi câu lệnh SQL ta ấn phím F5 hoặc sử dụng công cụ  trên thanh công cụ.
- Sau khi soạn thảo xong, để lưu lại, ta vào File→Save hoặc Save As thì nội dung văn bản được lưu lại dưới một file dạng text (có đuôi .sql).
- Để mở một file đã lưu trước đó ta nhấp chuột vào công cụ  trên thanh công cụ, chọn đường dẫn và file cần mở.

C. CÁC PHÁT BIỂU CƠ BẢN CỦA T-SQL.

Cú pháp của câu truy vấn SQL:

```

SELECT<danh sách các cột>
FROM<danh sách các bảng>
[WHERE<điều kiện>]
[GROUP BY<tên cột hay biểu thức sử dụng cột trong
SELECT>]
[HAVING<điều kiện dựa trên GROUP BY > ]
[ORDER BY<danh sách cột>]

```

1. Phát biểu SELECT với mệnh đề FROM.

- Phát biểu SELECT dùng để đọc thông tin từ CSDL theo những trường qui định hay những biểu thức cho trường đó. Ta có thể lấy thông tin với tất cả các trường trong bản bằng cách dùng ‘*’.
- Mệnh đề FROM chỉ ra tên của một hoặc nhiều bảng có quan hệ cần lấy thông tin.
- Ví dụ 1: **USES** northwind

```

SELECT OrderID,CustomerID
FROM Orders

```

- Ví dụ 2: **USES** northwind

```

SELECT *
FROM Orders

```

2. Phát biểu SELECT có dùng mệnh đề WHERE.

- Mệnh đề WHERE được dùng trong câu truy vấn để làm điều kiện lọc mẫu tin theo một tiêu chuẩn nào đó. Thông thường WHERE dùng cột (trường) để so sánh với giá trị cột khác, hay biểu thức chứa cột bất kỳ trong bảng.
- Ví dụ 3: **USES** northwind

```

SELECT *
FROM suppliers
WHERE suppliers > 20

```

❖ Các phép toán thường dùng trong mệnh đề WHERE.

Các phép toán so sánh:

- > : Lớn hơn.
- < : Nhỏ hơn.
- >= : Lớn hơn hoặc bằng.
- <= : Nhỏ hơn hoặc bằng.
- = : Bằng.
- != : Khác.
- <> : Khác.
- !> : Không lớn hơn.
- !< : Không nhỏ hơn.

Các phép toán logic:

- And : Phép toán và
 Ví dụ: `USES northwind`
`SELECT * FROM suppliers`
`WHERE suppliers > 20 and country='France'`
- Or : Phép hoặc.
 Ví dụ: `USES northwind`
`SELECT * FROM products`
`WHERE productID > 20 or supplier != 24`
- Not : Phép toán phủ định.
 Ví dụ: `USES northwind`
`SELECT * FROM employeeID`
`WHERE region IS NOT NULL`
- Between : Nằm trong miền.
 Ví dụ: `WHERE employee between 4 and 7`
- Like : Phép toán trên chuỗi.
 Ví dụ: `WHERE Lastname Like '%n'`
- In : Phép toán so sánh trong một tập hợp.
 Ví dụ: `WHERE EmployeeID in ('1','3','6')`
- Exists : Kiểm tra mẫu tin có tồn tại hay không.

Ví dụ: `if exists(SELECT * FROM Employees WHERE firstname='Steven') print('true')`

3. Phát biểu SELECT với mệnh đề ORDER BY.

ORDER BY: sắp xếp theo thứ tự tăng (ASC), giảm (DESC).

Cú pháp: ORDER BY <tên cột 1>, <tên cột 2>.... <tên cột k> ASC

ORDER BY <tên cột 1> + <tên cột 2> +....+ <tên cột k> ASC

ORDER BY <tên cột 1> + <tên cột 2>, <tên cột k> ASC

Tương tự cho sắp xếp giảm dần DESC.

4. Phát biểu SELECT với GROUP BY.

Mệnh đề GROUP BY được dùng để gom nhóm các bộ có cùng giá trị tại một thuộc tính nào đó.

5. Các hàm thông dụng trong SQL server 2000.

5.1. Các hàm trong phát biểu GROUP BY.

- Hàm AVG

Hàm này trả về giá trị trung bình của cột hay trường.

Ví dụ: `SELECT AVG (ReceiptAmount) FROM tbReceipt`

- Hàm MIN

Trả về giá trị nhỏ nhất của cột hay trường được chọn ra bởi mệnh đề SELECT.

Ví dụ: `SELECT MIN (ReceiptAmount) FROM tbReceipt`

- Hàm MAX

Trả về giá trị lớn nhất của cột hay trường được chọn ra bởi mệnh đề SELECT trong câu truy vấn.

Ví dụ: `SELECT MAX (ReceiptAmount) FROM tbReceipt`

- Hàm COUNT

Trả về số lượng mẫu tin trong câu truy vấn trên bảng.

Ví dụ: `SELECT COUNT(CustomerNo) FROM tbReceipt`

Có thể dùng COUNT(*) để đếm các bộ (dòng) trong bảng.

Ví dụ: **SELECT** COUNT(*) **FROM** Orders.

- Hàm **SUM**

Trả về tổng giá trị của trường, cột trong câu truy vấn.

Ví dụ: **SELECT SUM** (ReceiptAmount) **FROM** tbReceipt.

5.2. Các hàm xử lý chuỗi.

- Hàm ASCII

Hàm này trả về giá trị mã ASCII của ký tự bên trái chuỗi.

Ví dụ: print ASCII ('TOI')

Kết quả trả về là: 84.

- Hàm CHAR

Hàm này chuyển đổi kiểu mã ASCII từ số nguyên sang dạng chuẩn.

Ví dụ: print CHAR(35)

Kết quả trả về là #.

- Hàm UPPER

Chuyển đổi chuỗi sang kiểu chữ hoa.

Ví dụ: print UPPER ('Lam')

Kết quả trả về là: LAM

- Hàm LOWER

Chuyển đổi chuỗi sang kiểu chữ thường

Ví dụ: print LOWER ('Lam')

Kết quả trả về là: lam.

- Hàm LEN

Trả về chiều dài của chuỗi.

Ví dụ: print LEN ('SQL server')

Kết quả trả về là 10.

- Hàm LEFT

Trả về chuỗi bên trái tính từ đầu cho đến vị trí n

Ví dụ: `print LEFT ('Mycrosoft',4)`

Kết quả trả về là: Mycr

- RIGHT (tương tự như hàm LEFT)

- Hàm CHARINDEX

Trả về vị trí bắt đầu của chuỗi con trong chuỗi cần xét.

Ví dụ: `print CHARINDEX ('Lam', 'Ngo Tung Lam')`

Kết quả trả về là 10

- Thủ tục LTRIM

Loại bỏ khoảng trắng bên trái chuỗi.

Ví dụ: `print LTRIM('Nguyen')`

Kết quả trả về là 'Nguyen'

- Thủ tục RTRIM

Loại bỏ khoảng trắng bên phải chuỗi.

Ví dụ: `print RTRIM('Nguyen ')`

Kết quả trả về là 'Nguyen'

5.3. Các hàm xử lý thời gian.

- Hàm Getdate()

Trả về ngày tháng năm của hệ thống

Ví dụ: `SELECT 'Ngày tháng năm:'=Getdate()`

- Hàm DATEPART

Hàm này trả về một phần của chuỗi dạng ngày tháng đầy đủ.

Ví dụ: `SELECT 'Hom nay ngay:'=DATEPART(d,Getdate())`.

HÀM DATEPART	THAM SỐ
Year	yy, yyyy
Quarter	qq, q
Month	mm, m

Dateofyear	dy, y
Date	dd, d
Week	wk, ww
Weekday	dw
Hour	hh
Minute	mi, n
Second	ss, s
Milisecond	ms

- Hàm Date :

Trả về ngày thứ mấy trong tháng.

Ví dụ: SELECT 'Hom nay ngay'=Date(Getdate())

- Hàm Month

Trả về tháng thứ mấy trong năm.

Ví dụ: SELECT 'Hom nay thang:'=Month(Getdate())

- Hàm Year

Trả về năm hiện tại.

Ví dụ: SELECT 'Nam nay la:'=Year(Getdate())

5.4. Các hàm về toán học.

- Hàm SQUARE trả về bình phương của một biểu thức.

Ví dụ: print SQUARE(3)

Kết quả trả về là 9.

- Hàm SQRT trả về căn bậc hai của một biểu thức.

Ví dụ: print SQRT(9)

Kết quả trả về là 3

- Hàm ROUND trả về kết quả là số làm tròn của một biểu thức.

Ví dụ: print ROUND(748.58,-1).

Kết quả trả về là: 750.00

5.5. Các hàm về chuyển đổi.

- Hàm CAST trả về giá trị có kiểu dữ liệu theo định nghĩa.

Ví dụ: `print CAST(Getdate() AS VARCHAR(11))`

Kết quả trả về: oct 18 2001.

- Hàm `convert(int,'12')`

Kết quả trả về là số nguyên có giá trị là 12.

6. Phát biểu SELECT với AS.

Khi cần thiết phải thay đổi tên trường hay cột nào đó trong câu truy vấn ta dùng AS để thay đổi tên hay giá trị chưa có thành tên mới.

Ví dụ: `SELECT COUNT(*) AS SOLUONG
FROM ORDERS`

7. Phát biểu SELECT với TOP N.

TOP N cho phép ta lấy ra từ bảng dữ liệu N mẫu tin đầu tiên, ta có thể kết hợp TOP N với các mệnh đề WHERE, ORDER BY.

Ví dụ: `SELECT TOP 10 FROM ORDERS`

8. Phát biểu SELECT với DISTINCT.

Khi ta có một hay nhiều bảng kết nối với nhau, kết quả thông tin nhận được từ câu truy vấn có thể có nhiều dòng trùng lặp với nhau, để khắc phục điều này thì ta dùng DISTINCT như ví dụ dưới đây.

Ví dụ: `SELECT DISTINCT LUONG`

`FROM NHANVIEN`

Sau khi dùng DISTINCT thì các dòng trùng được loại bỏ.

9. Nhập dữ liệu bằng phát biểu INSERT.

- INSERT vào bảng lấy giá trị cụ thể.

Cú pháp: `INSERT INTO <tên bảng>[<danh sách các cột>]`

`VALUES (các giá trị cần chèn)`

Ví dụ: `INSERT INTO SINHVIEN (MASV, TENSX, NAM,
MAKH)`

`VALUES(2, Tung, 3, CNTT)`

- INSERT vào bảng lấy giá trị từ bảng khác.

Cú pháp: `INSERT INTO <tên bảng 1>[<danh sách các cột>]`

```
SELECT [danh sách các cột]
FROM <tên bảng 2>
WHERE <điều kiện>
```

Ví dụ: cho hai quan hệ:

```
SINHVIEN(MASV, TENSX, NAM, KHOA)
```

```
HOCVIEN(MAHV, TENHV, NAM, NGANH, NGÀYNHẬP
)
```

Thêm vào quan hệ SINHVIEN những HOCVIEN có MAHV>12

```
INSERT INTO SINHVIEN(MASV, TENSX, NAM, KHOA)
SELECT MAHV, TENHV, NAM, NGANH
FROM HOCVIEN
WHERE MAHV >12.
```

10. Phát biểu cập nhật UPDATE.

Phát biểu UPDATE dùng cập nhật lại dữ liệu đã tồn tại trong bảng.

- Cập nhật giá trị cụ thể.

Cú pháp: UPDATE <tên bảng>
 SET<tên cột >=<giá trị >
 [WHERE <điều kiện>]

Ví dụ: UPDATE HOCVIEN
 SET TENHV='THU'
 WHERE MAHV='11'

- Cập nhật giá trị từ một bảng khác.

Cú pháp: UPDATE <tên bảng>
 SET<tên cột >=<SELECT.. FROM... WHERE >
 [WHERE <điều kiện>]

Ví dụ: UPDATE HOCVIEN
 SET TENHV= (SELECT TENSX FROM SINHVIEN
 WHERE MASV=5)

WHERE MAHV='11'

11. Phát biểu xóa DELETE.

Xóa nội dung trong bảng với điều kiện được chỉ định trong mệnh đề WHERE.

Cú pháp: DELECT FROM <tên bảng>

WHERE <điều kiện>

Ví dụ: DELECT FROM SINHVIEN

WHERE MASV='3'

D. TẠO VÀ SỬA ĐỔI CÁC ĐỐI TƯỢNG.

1. Dừng lệnh.

1.1. Tạo bảng (CREATE TABLE).

Khi tạo bảng ta phải chú ý là: chọn tên bảng ngắn gọn và dễ hiểu, không có khoảng trắng, không trùng với những từ khóa, có ý nghĩa để người sử dụng dễ dàng nhận biết.

Cú pháp:

CREATE TABLE <tên_bảng> (<tên_cột> <kiểu dữ liệu> [null | not null]....);

- Trong đó:

<tên_bảng>: là chuỗi ký tự bất kỳ không có ký hiệu trống và không trùng với các từ khóa

<tên_cột>: là chuỗi ký tự bất kỳ không chứa ký hiệu trống, trong một bảng tên cột là duy nhất. Thứ tự các cột trong bảng là không quan trọng.

- Giải thích :

- ✓ Tên bảng: Là tên của table mới. Tên của Table không được trùng với tên table khác đang tồn tại trong database (giới hạn trong 128 ký tự).
- ✓ Tên cột : Tên của column, tồn tại duy nhất trong bảng.

(Xem ví dụ ở bài giảng chương 4: Ngôn ngữ truy vấn CSDL quan hệ SQL)

1.2. Tạo view (bảng ảo)

Có hai cách tạo view nhưng trong phần này ta làm quen với cách tạo view bằng phát biểu SQL.

Cú pháp tạo view như sau:

```
CREATE VIEW<View name>  
AS  
SELECT STATEMENT.
```

Khi tạo một view tương tự như các đối tượng khác của SQL server, ta cần đặt tên view, thường qui định tên view có dạng như sau vw<Tên view>. Ví dụ như vwbangphu.

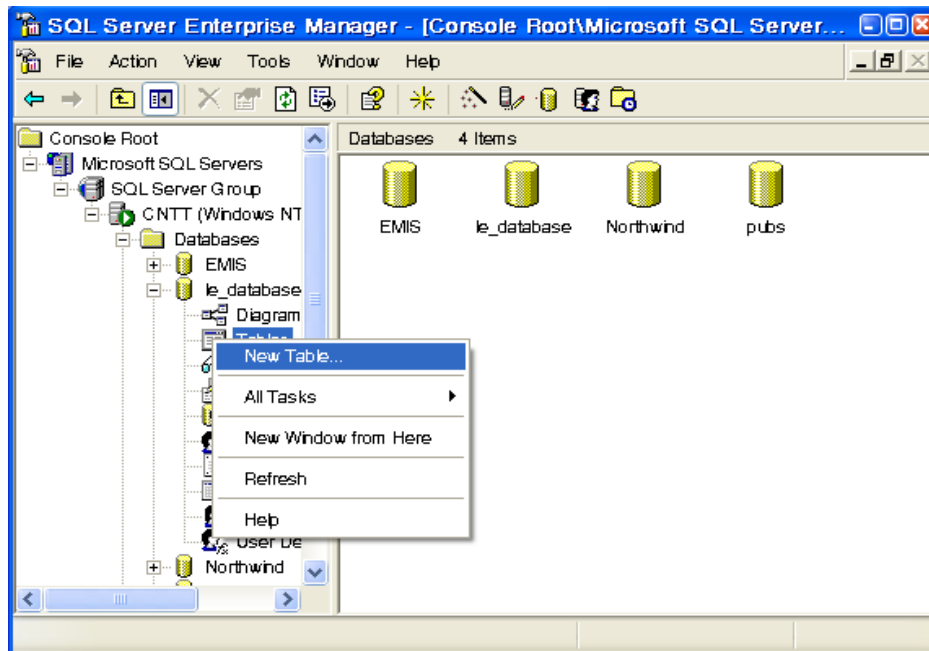
Ví dụ: Tạo view đơn giản trên một bảng như sau:

```
USE Account  
GO  
CREATE VIEW vwcust  
AS  
SELECT CustNo, CustName, CustEmail, Active  
FROM tblcustomer  
WHERE Active=1
```

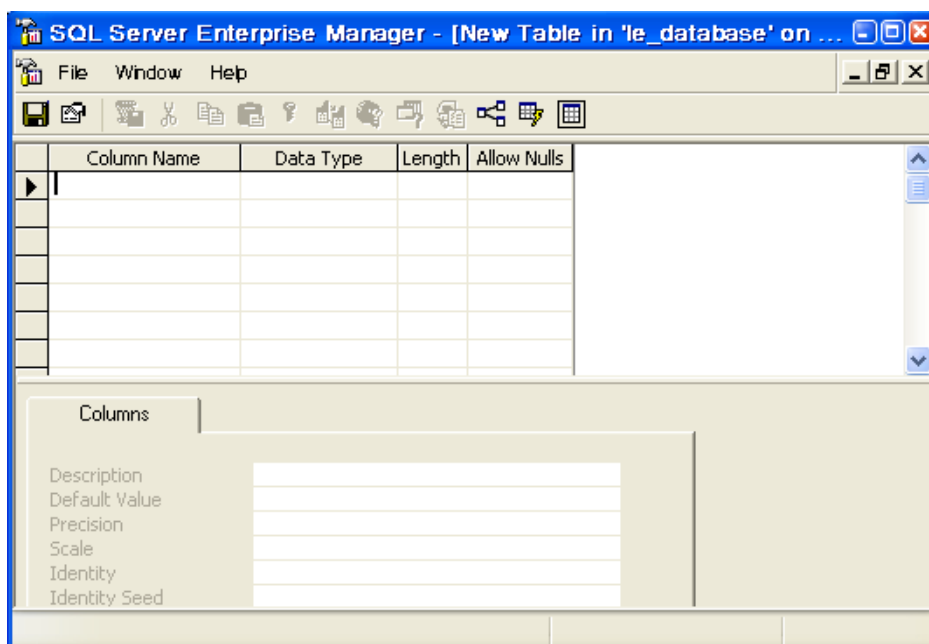
2. Dùng các công cụ tạo.

2.1. Tạo bảng bằng công cụ Enterprise Manager.

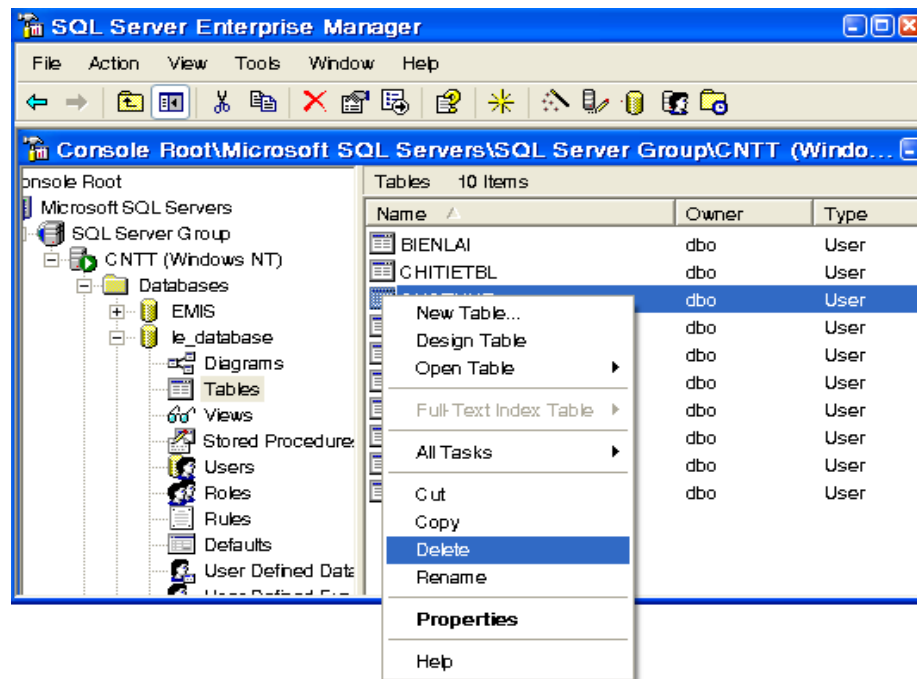
Từ màn hình Enterprise Manager nhấp chuột vào database, sau đó chọn database cần tạo bảng rồi nhấp phải vào table chọn new table như hình sau: (hoặc có thể vào menu action chọn new table).



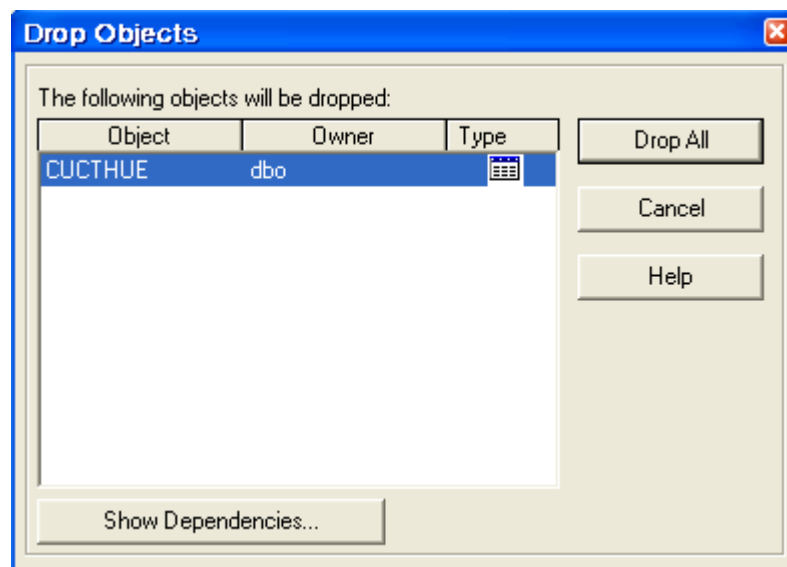
Sau khi chọn new table sẽ xuất hiện màn hình tạo bảng như sau:



2.2. Xóa bảng: (drop table).

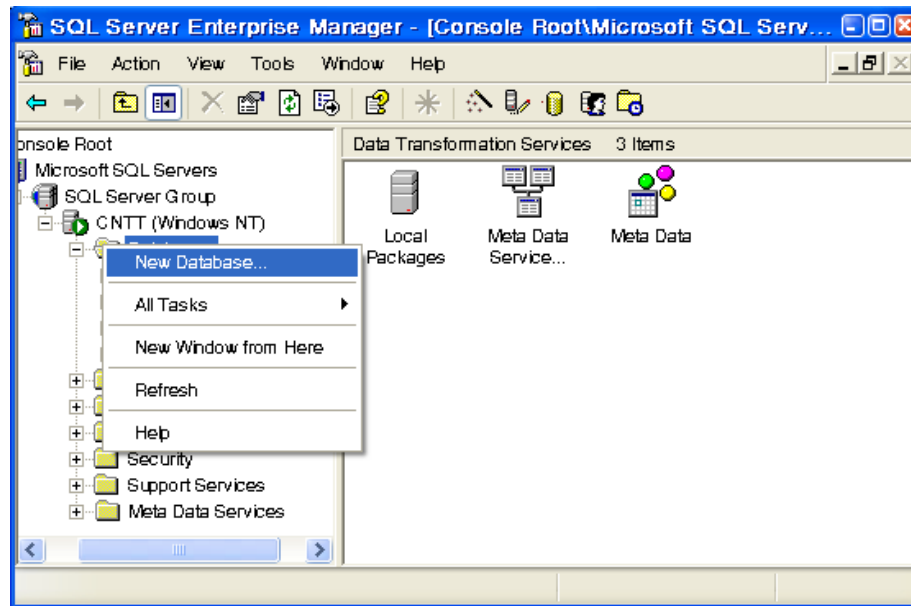


sau đó xuất hiện hộp thoại sau và chọn Drop All

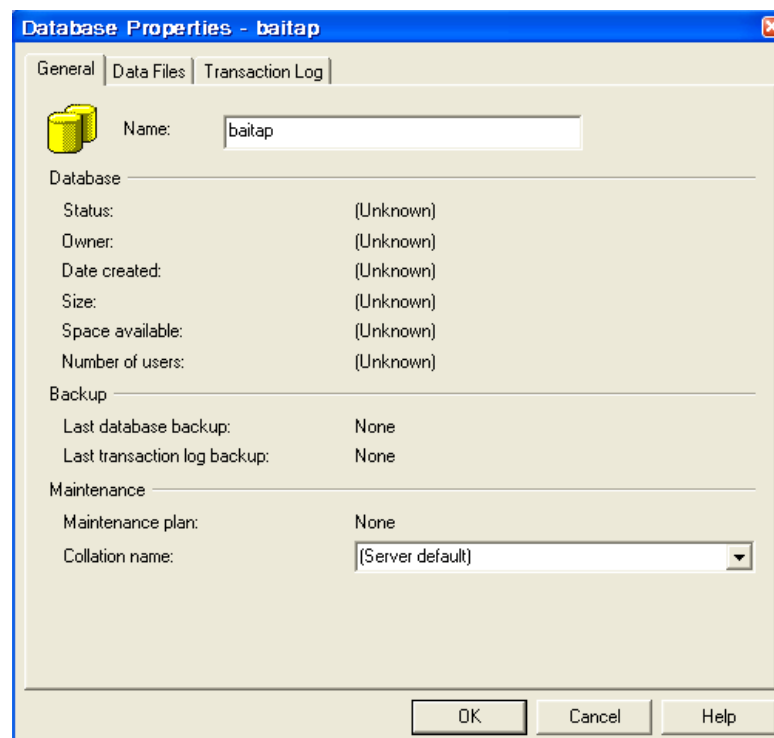


2.3. Tạo cơ sở dữ liệu mới – new database.

Nhấp phải vào folder database rồi chọn new database.



Sau khi chọn new database thì xuất hiện hộp thoại sau:



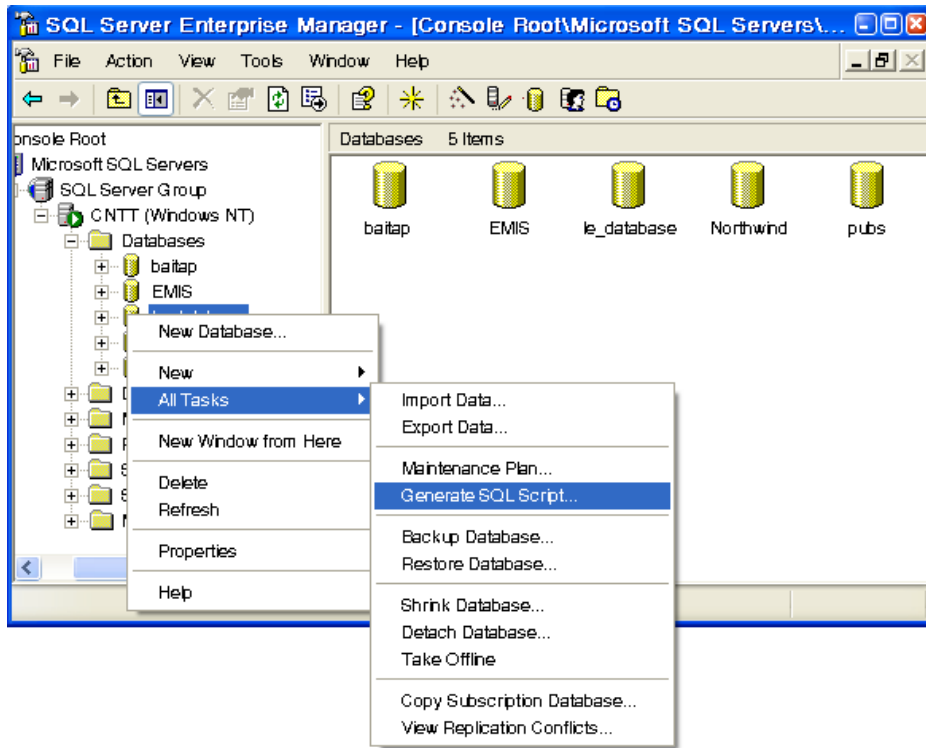
Đặt tên cho database rồi click ok để hoàn tất việc tạo database.

2.4. Tạo kịch bản – creating scripts.

Thông thường trong khi xây dựng CSDL đôi khi ta cần chuyển CSDL từ máy này sang máy khác, hay từ khu vực này sang khu vực khác, có nhiều cách để thực hiện điều này và một trong những cách đó là tạo kịch bản scripts. Đó là công cụ tái tạo lại CSDL mới từ một kịch bản của CSDL gốc.

Các bước tiến hành tạo một csdl từ một kịch bản gốc như sau:

- Đầu tiên nhấp chuột phải vào database nào cần tạo kịch bản và chọn All Tasks → Generate SQL Script.



Sau khi chọn generate SQL script, cửa sổ mới hiện ra trong đó có 3 tab và một số lựa chọn khác.

➤ Tab General:

- ✓ Script All Object: cho phép hiện ra danh sách các đối tượng.
- ✓ All Table: cho phép hiện ra danh sách tất cả các bảng.
- ✓ All stored Procedures: hiện ra danh sách tất cả các SP.
- ✓ All Defaults: hiện ra danh sách tất cả các giá trị mặc nhiên.
- ✓ All Rules: hiện ra danh sách tất cả các qui luật định trước.
- ✓ Preview: cho phép xem những ví dụ của kịch bản tạo ra.

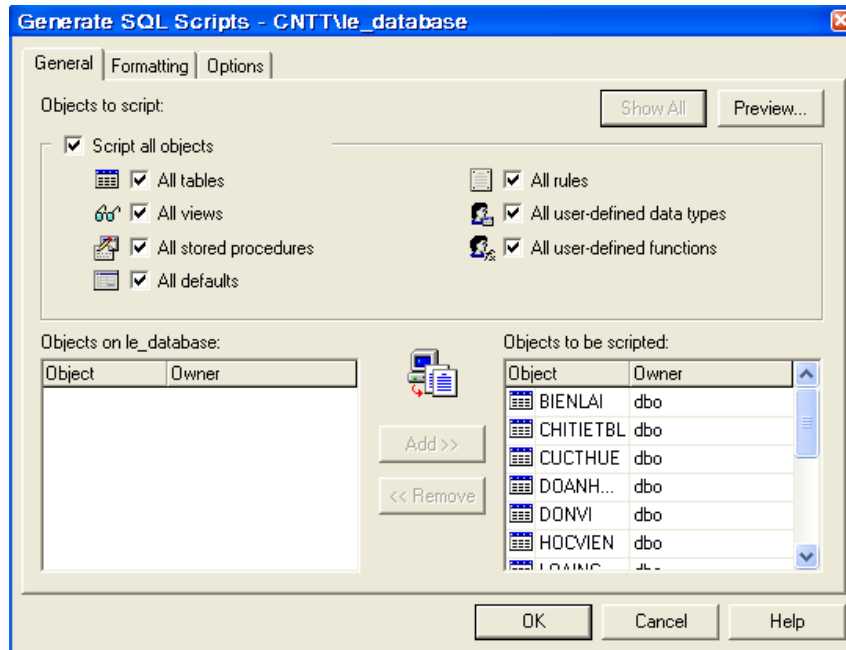
➤ Tab Formatting.

Tab này dùng định dạng script tạo ra như cú pháp cho sẵn, có phát biểu crop table xóa bảng trùng tên nếu tồn tại trong CSDL trước khi sử dụng, create table tạo lại bảng mới.

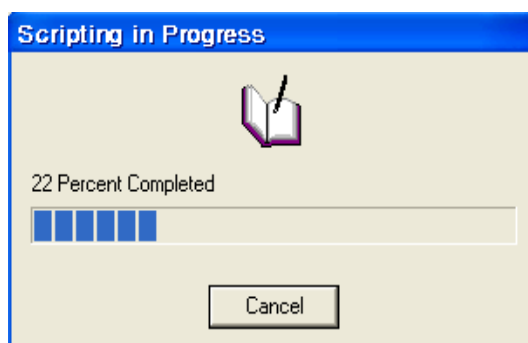
➤ Tab options.

Chọn một số đặc tính như:

- ✓ Primary Key: nếu bảng mới có primary key như bảng gốc.
- ✓ Foreign key: nếu bảng mới có foreign key như bảng gốc.
- ✓ Rules: tạo bảng có các qui luật đi theo.
- ✓ Default: tạo giá trị Default cho bảng mới.

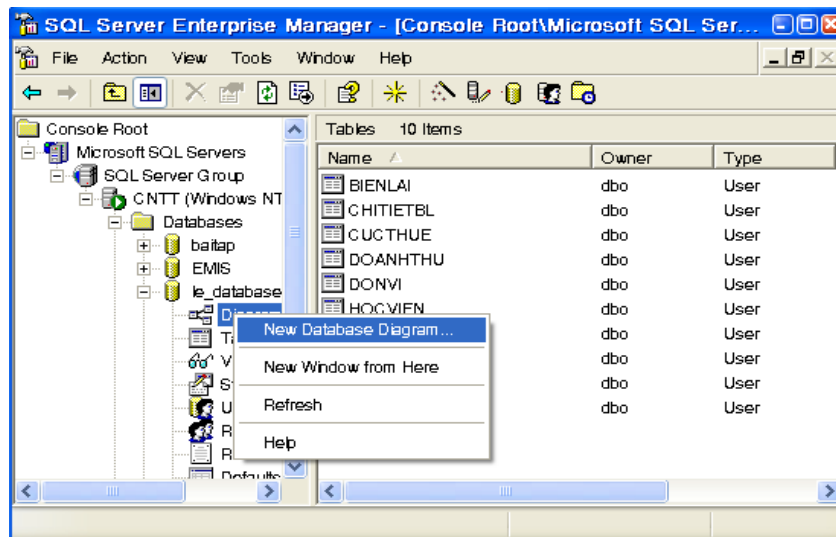


Sau khi chọn các tham số xong nhấp ok để hoàn tất và kịch bản được lưu bởi một file dạng text có đuôi mở rộng là .sql

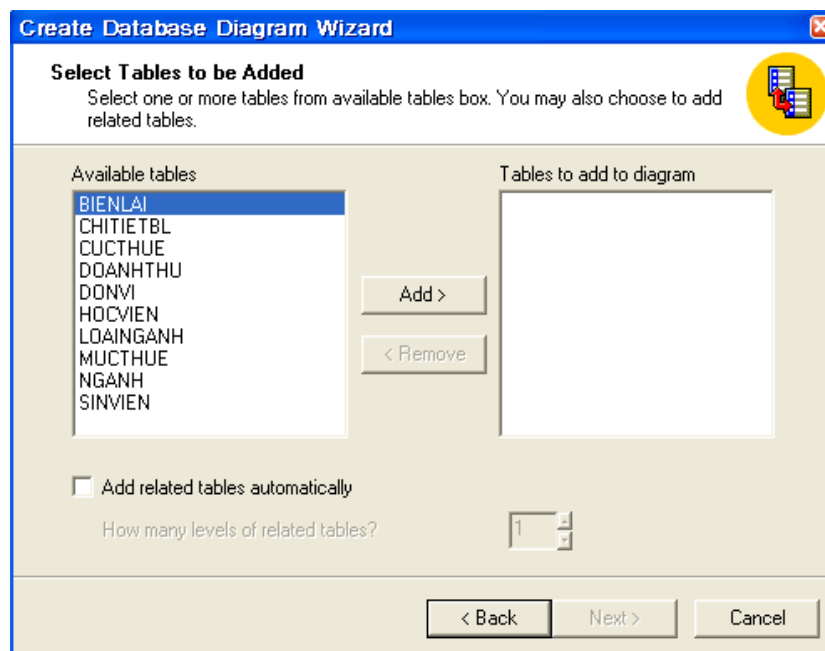


2.5. Tạo lược đồ quan hệ – Diagram trong SQL server 2000.

Nhấp phải chuột vào Diagram, chọn new diagram.



Xuất hiện hộp thoại sau:



Chọn các bảng cần đưa vào lược đồ quan hệ trong khung Available tables sau đó nhấp chọn Add → next → finish để kết thúc.

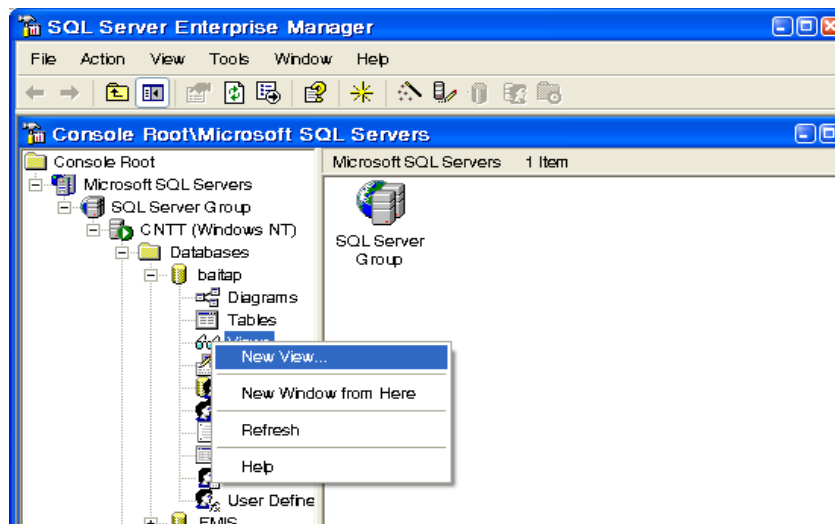
2.6. Tạo View (bảng ảo).

Như đã trình bày ở phần trước View là một đối tượng của CSDL. Để tạo mới một view chỉ cần chọn ngăn view, sau đó nhấp action hay nhấp chuột phải vào mục view trong CSDL trong màn hình rồi chọn new view. Sau khi màn hình tạo view hiển thị ta có thể chọn một hay nhiều bảng có liên quan để tạo view. Trong cửa sổ view có bốn phần khác nhau:

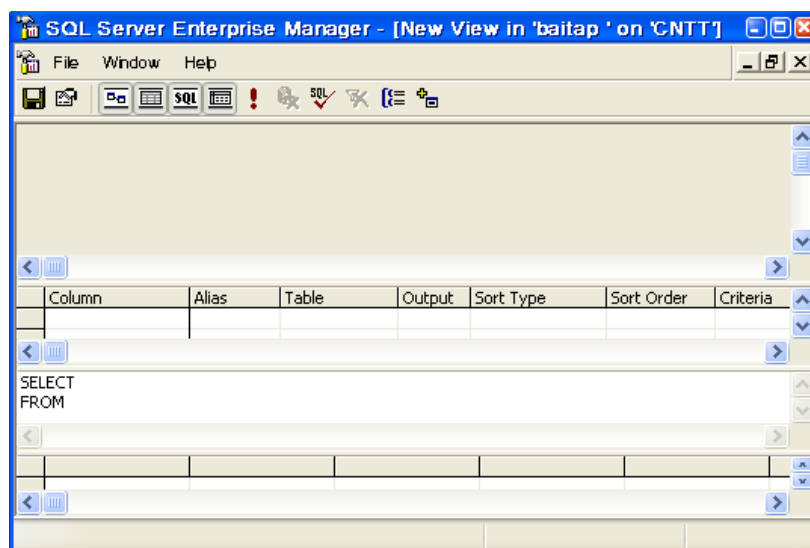
- Sơ đồ – Diagram pane.
- Khung lưới – Grid pane.
- Khung phát biểu SQL – SQL pane.
- Khung xuất kết quả – Result pane.

Trong phần sơ đồ chứa danh sách các bảng, chọn những cột từ các bảng để tạo view.

Phần phát biểu SQL, sau khi thực hiện việc chọn các cột thì phát biểu SQL sẽ được tạo ra. Nếu muốn biết kết quả chỉ cần nhấp chuột vào dấu chấm than trên thanh toolbar (!). Màn hình tạo view như sau:



Sau khi nhấp chuột vào new view sẽ xuất hiện màn hình như sau:

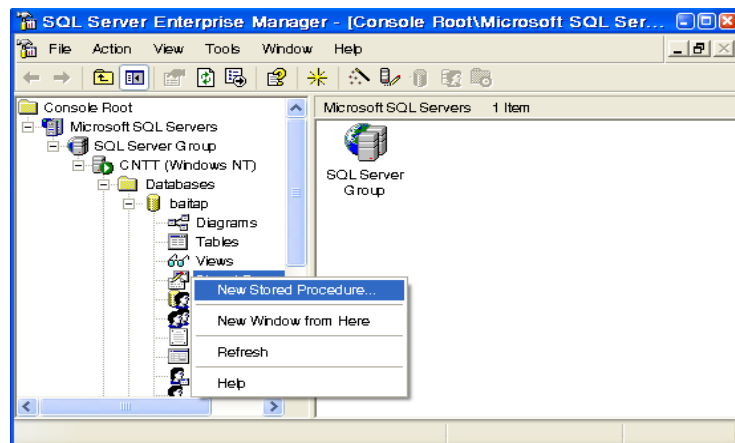


Chọn các bảng xong, bước tiếp theo là tạo quan hệ trong các bảng nếu có. Tiếp theo chọn những cột từ các bảng đưa vào khung lưới. Những cột đưa vào khung lưới là những cột sẽ xuất hiện trong bảng ảo mới khi tham chiếu đến view. Quan hệ có dạng như trong hình sau:

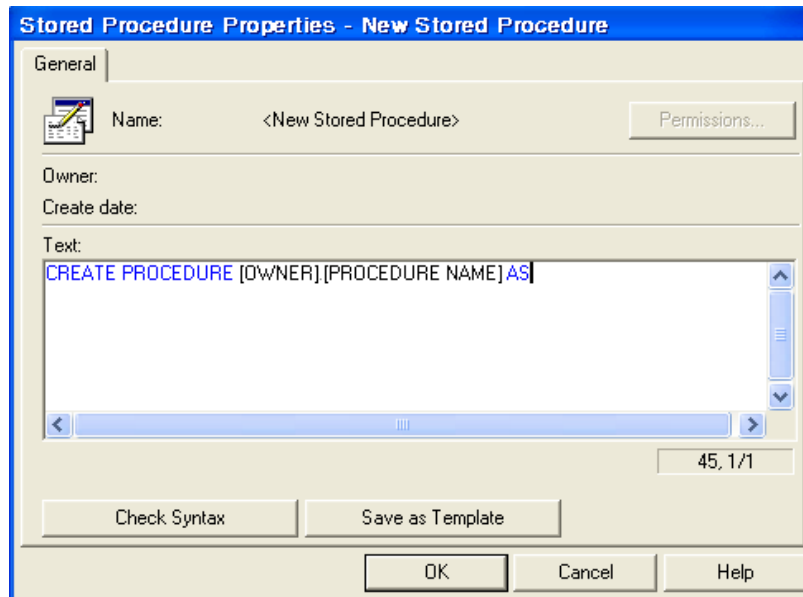
Sau khi chọn cột từ bảng, có thể chọn một số tiêu chuẩn như phát biểu Where và có thể sử dụng phát biểu SQL do view tạo ra.

2.7. Tạo Stored Procedure (SP).

Như đã biết, stored procedure cũng là một đối tượng của CSDL. Để tạo stored procedure ta nhấp chuột phải vào mục stored procedure trong CSDL cần tạo.



Sau khi nhấp chuột phải vào new stored procedure, màn hình soạn thảo nội dung stored procedure hiển thị như sau:



E. BÀI TẬP THỰC HÀNH.

Bài tập thực hành 1.

Cho lược đồ quan hệ sau:

1.SACH(MASACH,TUASACH,TACGIA,MANXB).

Mô tả: 1 quyển sách có 1 mã MASACH duy nhất để phân biệt và có 1 tựa sách, biết được sách do tác giả nào viết, và do nhà xuất bản nào xuất bản.

2.NHAXB(MANXB,TENNXB,DIACHIXB,TPHOXB).

Mô tả: 1 nhà xuất bản có 1 mã MANXB để phân biệt, có tên nhà xuất bản, địa chỉ nhà xuất bản, thành phố nơi có nhà xuất bản.

3.DOCGIA(SOTHE,HOTEN,DIACHIDG,TPHODG).

Mô tả: 1 độc giả sẽ có 1 số thẻ mượn sách riêng để phân biệt với các độc giả khác, có họ tên độc giả, có 1 địa chỉ và thành phố nơi độc giả ở.

4.MUONSACH(SOTHE,MASACH,NGAYMUON,NGAYTRA).

Mô tả: 1 độc giả có thể mượn được nhiều sách và 1 quyển sách có thể được nhiều người mượn đến, cứ mỗi lần mượn sách sẽ được ghi nhận ngày mượn sách (NGAYMUON), khi trả sách sẽ ghi nhận ngày trả (NGAYTRA).

MASACH	TUASACH	TACGIA	MANXB
--------	---------	--------	-------

TH1	Nhập môn cơ sở dữ liệu	Lê Tiến Vương	TK
TH2	Hướng dẫn triển khai CCDL SQL Server 7.0 và Visual Basic 6.0	Michal Otex	TK
VH3	Bay qua thời gian	Nguyễn thị Châu Giang	HNV
TH3	Cơ sở dữ liệu	Đỗ Trung Tuấn	GD

MANXB	TENNXB	DIACHIXB	TPHOXB
TK	Nhà xuất bản Thống Kê	16 Nguyễn Huệ Q1	TP HCM
HNV	Nhà Xuất bản Hội nhà Văn	65 Nguyễn Du Hà Nội	TP Hà Nội
GD	Nhà Xuất bản Giáo dục	Hà Nội	TP Hà Nội

SOTHE	HOTEN	DIACHIDG	TPHODG
T1	Nguyễn Xuân Mai	25 NTMK	TP HCM
T2	Nguyễn Ngọc Linh	15 Nguyễn Du	TPHCM
T3	Nguyễn Thị Nhỏ	20 Lê Lợi	TP Đà Nẵng

SOTHE	MASACH	NGAYMUON	NGAYTRA
T1	TH1	12/02/2002	15/02/2002
T1	TH2	15/02/2002	16/02/2002
T2	VH3	17/02/2002	19/02/2002
T2	TH1	20/02/2002	25/02/2002
T3	TH1	22/02/2002	28/02/2002
T3	VH3	25/02/2002	30/02/2002

A. Cài đặt CSDL trên hệ QTCSDL cụ thể SQL Server.

B. Hãy trả lời các câu hỏi sau bằng ngôn ngữ SQL.

1/ Hãy cho biết các số thẻ và mã sách đã mượn sau ngày ‘10/02/2002’.

2/ Hãy cho biết tên và địa chỉ tất cả các nhà xuất bản ở thành phố ‘Tp.HCM’.

- 3/ Hãy cho biết tất cả các độc giả (họ tên, địa chỉ) có mượn sách trong khoảng từ ngày '20/02/2002' đến ngày '25/02/2002'.
- 4/ Hãy cho biết các quyền sách (mã sách, tựa sách) của tác giả 'Lê Tiến Vương' được xuất bản bởi nhà xuất bản 'Nhà xuất bản thống kê'.
- 5/ Cho biết tên các độc giả mượn sách có tựa sách 'Nhập môn CSDL quan hệ'.
- 6/ Cho biết tên các độc giả ở thành phố 'TP Đà Nẵng' có mượn sách sau ngày '15/02/2002'.
- 7/ Cho biết tựa sách của các sách mà độc giả có tên 'Nguyễn Thị Nhỏ' đã mượn đọc.
- 8/ Cho biết các quyền sách (mã sách, tựa sách) có ít nhất 1 lần được mượn.
- 9/ Hãy cho biết tên tác giả nào được độc giả mượn sách đọc nhiều nhất.
- 10/ Hãy cho biết quyền sách (mã sách) nào mà chưa được mượn đọc.

Bài tập thực hành 2

Quản lý học viên.

Một trung tâm tin học muốn tin học hóa công tác quản lý ghi danh và kết quả học tập của học viên. Kết quả phân tích thiết kế ứng dụng trên như sau.

KHOAHOC(MAKH, TENKH, BD,KT)

Mô tả: Mỗi một khóa học có một mã số (MAKH) duy nhất để phân biệt với các khóa học khác, có tên khóa học (TENKH), ngày bắt đầu (BD) và ngày kết thúc khóa học (KT).

HOCVIEN(MAHV,HO,TEN,NGAYSINH,DCHI)

Mô tả: Mỗi một học viên có một mã số duy nhất (MAHV) để phân biệt với các học viên khác có họ (HO), tên (TEN), ngày tháng năm sinh (NGAYSINH), địa chỉ (DCHI).

GIAOVIEN(MAGV, HOTEN, NGAYSINH, DIACHI)

Mô tả: Mỗi một giáo viên có một mã số (MAGV) duy nhất để phân biệt với các giáo viên khác, có họ tên (HOTEN), ngày tháng năm sinh (NGAYSINH), địa chỉ (DIACHI).

LOPHOC(MALOP,TENLOP, MAKH, MAGV, SISODK, TRG, PHHOC)

Mô tả: Mỗi một lớp học có một mã số (MALOP), có tên lớp học (TENLOP). Mỗi lớp học thuộc về một khóa học duy nhất (MAKH) và do một giáo viên (MAGV) duy nhất phụ trách, và có một lớp trưởng (LTRG). Vào đầu khóa học người ta dự định một số dự kiến (SISODK) cho mỗi lớp, và một phòng học (PHHOC) sử dụng trong suốt khóa học.

**BIENLAI(SOBL,MAKH,MALOP,MAHV,
DIEM,KQUA,XEPLOAI,TIENNOP)**

Mô tả: Mỗi học viên ghi danh vào một lớp (MALOP), sẽ có một số biên lai duy nhất (SOBL) trong khóa đó, trong đó có ghi nhận số tiền học viên thực nộp (TIENNOP). Nếu học viên được miễn học phí thì TIENNOP = 0 hoặc null. Cuối mỗi khóa học, mỗi học viên sẽ có một kết quả đánh giá, bao gồm : điểm kiểm tra (DIEM), xếp loại (XEPLOAI) và kết quả đậu rớt (KQUA). Điểm kiểm tra được tính trên thang mười, xếp loại có 4 mức: Giỏi, Khá, TB, Yếu.

KHOAHOC			
MAKH	TENKH	BD	KT
THA102	Tin học A khóa 1/2002	15/02/2002	15/05/2002
THA202	Tin học A khóa 2/2002	30/05/2002	30/08/2002
THB202	Tin học B khóa 2/2002	30/05/2002	30/08/2002

LOPHOC						
MALOP	TENLOP	MAKH	MAGV	SISODK	LTRG	PHHOC
01	Tin học A 1.1	THA102	1	20	9	101
02	Tin học A 2.1	THA202	2	20	8	201
03	Tin học B 2.1	THB202	3	15	17	202
04	Tin học B 2.2	THB202	4	15		203

GIAOVIEN			
MAGV	HOTEN	NGAYSINH	DIACHI

1	Văn Thế Thành	05/05/1978	25 Nguyễn Du
2	Nguyễn Lộc	01/01/1975	11 Nguyễn thị Nhỏ
3	Ngô thị Khoai	02/05/1976	12/25 Trần Đại Nghĩa
4	Lê Hồ	03/07/1970	30 Lê Lợi

HOCVIE N				
MAHV	HO	TEN	NGAYSIN H	DCHI
0	Hồ Thanh	Sơn	01/02/1982	15 Hồ Tùng Mậu
1	Trần Hồ	Anh	03/04/1983	14 Phạm Ngũ Lão
2	Nguyễn thị	Ngọc Anh	05/04/1980	25/58 Lê Văn Sĩ
3	Lê Hoàng	Thư	01/07/1982	12/14 Ngô Quyền
4	Nguyễn Minh	Đảng	05/07/1982	12 Nguyễn Trọng Tuyền
5	Nguyễn Thị	Hồng Hoa	03/04/1981	24 Lê lợi
6	Nguyễn Kiên	Giang	05/04/1982	35 Nguyễn Thị Minh Khai
7	Lâm Ngọc	Châu	03/07/1980	15 Paster

BIENLAI							
SOBL	MAKH	MALOP	MAHV	DIEM	KETQUA	XEPLOAI	TIENNO
1	THA102	01	1	7	DAU	KHA	100000
2	THA102	01	2	6	DAU	TBKHA	100000
3	THA102	01	3	5	DAU	TB	100000
4	THA102	01	4	8	DAU	KHA	100000
1	THA202	02	1	4.5	KHONGDAU	YEU	100000
2	THA202	02	2	6	DAU	TBKHA	100000
3	THA202	02	3	8	DAU	KHA	100000
4	THA202	02	5	5	DAU	TBKHA	100000
5	THA202	02	6	9	DAU	GIOI	0
1	THB202	04	1	8	DAU	KHA	100000
2	THB202	04	2	9	DAU	KHA	100000
3	THB202	04	5	4.5	KHONGDAU	YEU	100000
4	THB202	04	6	5	DAU	TB	100000

5	THB202	04	7	5	DAU	TB	100000
---	--------	----	---	---	-----	----	--------

Yêu cầu:

A/ Xác định khóa chính, khóa ngoại của lược đồ quan hệ trên.

B/ Cài đặt CSDL trên hệ QTCSDDL cụ thể : SQL Server,...

C/ Xây dựng các truy vấn sau bằng ngôn ngữ SQL:

1. Cho biết kết quả cuối khóa (điểm, kết quả, xếp loại) của các học viên do giáo viên “VAN THE THANH” hoặc “NGUYEN LOC” dạy trong khóa có mã số “THA102”.
2. Cho biết danh sách lớp học và số lượng học viên thực sự của lớp đó
3. Cho biết họ tên, ngày sinh, địa chỉ của học viên có điểm cao nhất trong khóa có mã “THA197”.
4. Cho biết tên các lớp học và điểm trung bình, điểm cao nhất của tất cả các học viên trong lớp.
5. Cho biết tên lớp học và lượng số học viên xếp loại khá hoặc giỏi trong lớp đó
6. Cho biết họ tên , địa chỉ của những học viên là lớp trưởng.
7. Cho biết họ tên giáo viên và số lượng học viên mà lớp đó dạy.
8. Cho biết họ tên và kết quả học tập (điểm, xếp loại, kết quả) của những học viên được miễn học phí.
9. Cho biết các tên lớp học, tên giáo viên phụ trách, sĩ số dự kiến của lớp và phòng học của các lớp học đang diễn ra vào ngày ‘17/04/2002’.
10. Cho biết tên lớp học và lượng số học viên “KHONG DAU”, số lượng học viên xếp loại trung bình hoặc yếu trong lớp đó.

Bài thực hành 3**CHO CƠ SỞ DỮ LIỆU EMPLOYMENT NHƯ SAU:**

DEPTNO	DEPTNAME	LOC	MGR	EXP_BUDG	REV_BUDG
10	Accounting	Dallas	200	10000	

30	Research	San Francisco	105	125000	
40	Sales	Boston	109	280000	800000
50	Manufacturing	Houston	210	130000	
60	Shipping	Houston	215	90000	

Quan hệ: EMPLOYEE (Nhân viên)

EMPNO	NAME	JOB	SALARY	COMM	DEPTNO	SEX
100	Wilson	Clrk	1700		10	M
101	Smith	Slsn	2500	1300	40	F
103	Reed	Anlt	3500		30	M
105	Watson	Mngr	4500	0	30	M
109	Allen	Mngr	3800	8000	40	F
110	Turner	Clrk	1800		50	M
200	Chen	Mngr	2900		10	F
210	Ramirez	Mngr	3600		50	M
213	McDonnell	Clrk	1625		60	M
214	Simpson	Drvr	825		60	M
215	Di Salvo	Spvr	2700		60	M
220	Schwartz	Slsn	4250	5300	40	F

Quan hệ: JOB (Công việc)

JOB	JOBNAME	MINSALRY	MAXSALRY	MGRFLAG
Mngr	Manager	2500	5500	Y
Clrk	Clerk	950	1800	N
Slsa	Sales Assist	950	2000	N
Amgr	Assist Manager	1500	3000	Y
Drvr	Driver	1050	1700	N
Slsn	Salesman	750	5000	N
Spvr	Supervisor	1500	2000	Y
Anlt	Analyst	1300	3500	N
Secy	Secretary	800	2200	N
Newh	New Hire	500	800	N

Quan hệ: EMPLHIST (Quá trình tuyển dụng).

EMPNO	SEQ	DATE_BEG	DATE_END	SALARY	FRJOB	TOJOB	PROMO	FRDEPT	TODEPT
101	1	01/03/81	30/06/81	800	Newh	Slsa	N		40
101	2	01/07/81	30/06/82	1125	Slsa	Slsa	N	40	40
101	3	01/07/82	30/06/84	1900	Slsa	Slsn	Y	40	40
101	4	01/07/84		2500	Slsn	Slsn	N	40	40
109	1	01/04/80	30/09/80	725	Newh	Slsa	N		40
109	2	01/10/80	30/06/82	1500	Slsa	Slsn	Y	40	40
109	3	01/07/82	31/03/83	2600	Slsn	Amgr	Y	40	40
109	4	01/04/83	30/06/85	3200	Amgr	Mngr	Y	40	40

109	5	01/07/85		3800	Mngr	Mngr	N	40	40
220	1	01/05/82	31/08/82	1900	Newh	Mngr	N		60
220	2	01/09/82	30/06/83	1600	Mngr	Slsn	N	60	40
220	3	01/07/83	31/12/83	2200	Slsn	Slsn	N	40	40
220	4	01/01/84	30/06/84	3400	Slsn	Slsn	N	40	40
220	5	01/07/84		4250	Slsn	Slsn	N	40	40

1. Tạo và nhập dữ liệu cho các bảng trên.
2. Liệt kê danh sách nhân viên hiện đang làm việc cho Công ty.
3. Liệt kê các thông tin EMPNO, NAME, JOB, SALARY, COMM, DEPNO và SEX của các nhân viên theo các phòng Ban.
4. Như trên, nhưng trong từng Phòng Ban thì sắp tên nhân viên theo thứ tự Alphabet .
5. Cho danh sách nhân viên theo mức lương giảm dần.
6. Cho Tên nhân viên, Công việc đang làm và mức lương của các nhân viên, theo thứ tự giảm dần theo Tên nhân viên.
7. Cho biết địa điểm của các Phòng của Công ty.
8. Cho biết bảng lương của từng công việc được áp dụng cho các nhân viên của Công ty.
9. Liệt kê bảng lương theo thứ tự Alphabet của tên đầy đủ của các công việc.
10. In bảng lương theo thứ tự giảm dần của mức lương cao nhất của các công việc.
11. Liệt kê danh sách nhân viên của Công ty theo mức lương từ cao đến thấp.
12. Sắp danh sách nhân viên của Công ty theo giới tính và Alphabet của tên.
13. Cho danh sách nhân viên của Công ty theo thứ tự loại công việc (JOB), trong từng côngviệc sắp xếp theo thứ tự giảm dần của tiền lương (SALARY).
14. Cho biết công ty có các loại việc (JOB) nào.

15. Cho danh sách nhân viên của Phòng có mã số 40. (Mệnh đề WHERE).
16. Cho danh sách các giám đốc (manager) có lương trên \$4000. (Phép toán AND).
17. Cho danh sách nhân viên của phòng 40 và những người có lương trên \$4000. (OR).
18. Cho danh sách nhân viên không thuộc phòng số 30. (Phép toán NOT).
19. Cho danh sách các nhân viên thuộc các phòng 10, 30 và 50. (Phép toán IN).
20. Cho danh sách các nhân viên có mức lương trong khoảng 3500 đến 4500 Dollar.
21. Cho biết các nhân viên có tên bắt đầu bằng chữ W. (Phép toán LIKE).

Tài liệu tham khảo

1. Giáo trình cơ sở dữ liệu – Bộ giáo dục và đào tạo.
2. Giáo trình cơ sở dữ liệu quan hệ, Nguyễn văn Tâm, Nguyễn Hữu Bình – Nhà xuất bản thống kê.
3. Nhập môn cơ sở dữ liệu, Lê Tiến Vương – Nhà xuất bản thống kê năm 2000.
4. Ngôn ngữ cơ sở dữ liệu SQL, Ngô Trung Việt – Nhà xuất bản giao thông vận tải năm 1998.
5. Lập trình ứng dụng SQL SERVER 2000 – Phạm Hữu Khang.
6. Microsoft SQL server.