

PHẦN I – NHẬP MÔN CƠ SỞ DỮ LIỆU QUAN HỆ

(Tổng số: 15 tiết, Lý thuyết: 15 tiết)

CHƯƠNG 1 – TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

(Tổng số: 2 tiết, Lý thuyết: 2 tiết)

§ 1. NHỮNG KHÁI NIỆM CƠ BẢN VỀ CƠ SỞ DỮ LIỆU**I. SỰ CẦN THIẾT CỦA CƠ SỞ DỮ LIỆU**

Trong những năm gần đây, thuật ngữ “Cơ sở dữ liệu” (CSDL - Database) đã trở nên khá quen thuộc không chỉ riêng với những người làm Tin học mà còn đối với cả những người làm trong nhiều lĩnh vực khác như Thống kê, Kinh tế, Quản lý Doanh nghiệp v.v... Các ứng dụng của Tin học vào công tác quản lý ngày càng nhiều hơn và càng đa dạng hơn. Có thể nói hầu hết các lĩnh vực kinh tế, xã hội, giáo dục, y tế v.v... đều đã ứng dụng các thành tựu mới của Tin học vào phục vụ công tác chuyên môn của mình. Chính vì lẽ đó mà ngày càng nhiều người quan tâm đến lĩnh vực thiết kế và xây dựng các CSDL.

Mục đích của Chương 1 chỉ đơn giản là cung cấp các khái niệm cơ bản về CSDL để các học viên có một cái nhìn ban đầu về một CSDL và một hệ quản trị CSDL. Trước hết chúng ta sẽ tìm hiểu lý do tại sao cần phải có một CSDL.

Hệ thống các tập tin cổ điển (File System):

Cho đến nay vẫn còn một số đơn vị kinh tế, hành chính sự nghiệp v.v... sử dụng mô hình hệ thống các tập tin cổ điển: Chúng được tổ chức riêng rẽ, phục vụ cho một mục đích của một đơn vị hay một đơn vị con trực thuộc cụ thể. Chẳng hạn, ta hãy xét ví dụ sau:

Ví dụ: Tại một công ty người ta trang bị máy vi tính cho tất cả các phòng, ban nghiệp vụ. Bộ phận Văn phòng sử dụng máy vi tính để soạn thảo văn bản bằng Microsoft Word do thủ trưởng yêu cầu về tình hình hoạt động của đơn vị, trong đó có chỉ tiêu về tổng số công nhân viên chức chia theo trình độ chuyên môn được đào tạo. Phòng Kế toán sử dụng máy vi tính để tính lương và in danh sách lương của từng bộ phận trong đơn vị dựa trên danh sách cán bộ viên chức cùng hệ số lương và các hệ số phụ cấp của họ do phòng Tổ chức cung cấp. Thông tin mà phòng Kế toán quản lý và khai thác là: Họ và Tên, Hệ số lương, Hệ số phụ cấp, Phụ cấp khác của các công nhân viên chức (CNVC) xếp theo từng phòng ban và sử dụng công cụ văn phòng là Microsoft Excel. Phòng Tổ chức quản lý thông tin lý lịch của CNVC chi tiết hơn gồm: Họ, Tên (để riêng thành một cột “Tên” để tiện sắp xếp Alphabet), Giới tính, Ngày sinh, Ngày tuyển dụng, Hoàn cảnh gia đình, Quá trình đào tạo, Hệ số lương, Hệ số phụ cấp, Ngày xếp lương trên... nhưng thiếu thông tin về Phụ cấp khác của CNVC. Phần mềm được sử dụng để quản lý là FoxPro for Windows.

Trong khi đó, tại Tổng công ty của họ, các phòng ban nghiệp vụ cũng được trang bị máy vi tính. Phòng Tổ chức cán bộ tại Tổng công ty sử dụng phần mềm Microsoft Access để quản lý CNVC gồm các cán bộ chủ chốt từ trường phó phòng, quản đốc và phó quản đốc xí nghiệp trở lên của các công ty con trực thuộc. Thông tin quản lý tại đây cũng giống như thông tin quản lý tại phòng tổ chức của công ty con.

Nhận xét:

Ưu điểm:

- Việc xây dựng hệ thống các tập tin riêng tại từng đơn vị quản lý ít tốn thời gian bởi khối lượng thông tin cần quản lý và khai thác là nhỏ, không đòi hỏi đầu tư vật chất và chất xám nhiều, do đó triển khai ứng dụng nhanh.

- Thông tin được khai thác chỉ phục vụ cho mục đích hẹp nên khả năng đáp ứng nhanh chóng, kịp thời.

Nhược điểm:

- Do thông tin được tổ chức ở mỗi phòng ban mỗi khác, cũng như phần mềm công cụ để triển khai mỗi nơi cũng rất khác nhau nên sự phối hợp tổ chức và khai thác ở các phòng ban là khó khăn. Thông tin ở phòng ban này không sử dụng được cho phòng ban khác, tại đơn vị con với đơn vị cấp trên. Cùng một thông tin được nhập vào máy tại nhiều nơi khác nhau gây ra lãng phí công sức nhập tin và không gian lưu trữ trên các vật mang tin. Sự trùng lặp thông tin có thể dẫn đến tình trạng không nhất quán dữ liệu. Chẳng hạn, nhân viên Nguyễn Văn Quang được ghi đầy đủ ở phòng Tổ chức, nhưng tại phòng Kế toán chỉ ghi tắt là Nguyễn V Quang.

- Thông tin được tổ chức ở nhiều nơi nên việc cập nhật cũng dễ làm mất tính nhất quán dữ liệu. Một cán bộ chủ chốt của công ty có thay đổi về hoàn cảnh gia đình (mới cưới vợ / lấy chồng, sinh thêm con...) có thể được cập nhật ngay tại đơn vị nhưng sau một thời gian mới được cập nhật tại Tổng công ty.

- Do hệ thống được tổ chức thành các hệ thống tập tin riêng lẻ nên thiếu sự chia sẻ thông tin giữa các nơi. Việc kết nối các hệ thống này hay việc nâng cấp ứng dụng sẽ là rất khó khăn.

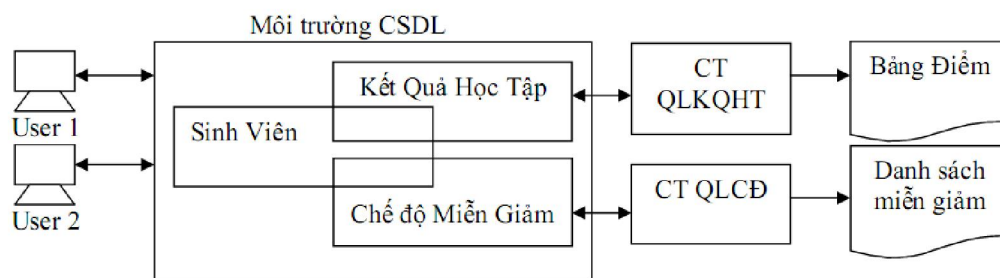
Qua phân tích trên chúng ta nhận thấy việc tổ chức dữ liệu theo hệ thống các tập tin có nhiều nhược điểm. Việc xây dựng một hệ thống tin đảm bảo được tính chất nhất quán dữ liệu, không trùng lặp thông tin mà vẫn đáp ứng được nhu cầu khai thác đồng thời của tất cả các phòng ban ở công ty và tổng công ty là thực sự cần thiết.

II. NHỮNG KHÁI NIỆM CƠ BẢN VỀ CƠ SỞ DỮ LIỆU

1. Khái niệm cơ sở dữ liệu

Cơ sở dữ liệu (CSDL) là sự tập hợp có tổ chức (có cấu trúc) các dữ liệu có liên quan luận lý với nhau, được lưu trữ trên các thiết bị lưu trữ thông tin (băng từ, đĩa từ...) để có thể thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng với nhiều mục đích khác nhau.

Ví dụ: Để quản lý hoạt động học tập trong một trường Đại học, chúng ta sẽ quản lý các dữ liệu là các thông tin về sinh viên, về các môn học, điểm thi... Do đó, chúng ta tổ chức các dữ liệu thành các bảng và lưu giữ chúng vào sổ sách hoặc sử dụng một phần mềm máy tính để lưu giữ chúng trên máy tính. Ta có một tập hợp các dữ liệu có liên quan đến nhau và mang nhiều ý nghĩa, đó là một CSDL.



Hình: CSDL quản lý sinh viên

Trước hết, chúng ta cần nhấn mạnh CSDL phải là một tập hợp các thông tin mang tính hệ thống chứ không phải là các thông tin rời rạc, không có mối quan hệ với nhau. Các thông tin này phải có cấu trúc và tập hợp các thông tin này phải có khả năng đáp ứng các nhu cầu khai thác của nhiều người sử dụng một cách đồng thời. Đó cũng chính là các đặc trưng của CSDL.

2. Các tính chất của một cơ sở dữ liệu

Một CSDL biểu thị một khía cạnh nào đó của thế giới thực như hoạt động của một công ty, một nhà trường, một ngân hàng... Những thay đổi của thế giới thực phải được phản ánh

một cách trung thực vào trong CSDL. Những thông tin được đưa vào trong CSDL tạo thành một không gian CSDL hoặc là một “thế giới nhỏ”.

Một CSDL là một tập hợp dữ liệu liên kết với nhau một cách logic và mang một ý nghĩa cố hữu nào đó. Một CSDL không phải là một tập hợp tùy tiện.

Một CSDL được thiết kế và được phổ biến cho một mục đích riêng. Nó có một nhóm người sử dụng có chủ định và có một số ứng dụng được xác định phù hợp với mối quan tâm của người sử dụng. Nói cách khác, một CSDL có một nguồn cung cấp dữ liệu, một mức độ tương tác với các sự kiện trong thế giới thực và một nhóm người quan tâm tích cực đến các nội dung của nó.

Một CSDL có thể có cỡ tùy ý và có độ phức tạp thay đổi. Có những CSDL chỉ gồm vài trăm bản ghi (như CSDL phục vụ việc quản lý lương ở một cơ quan nhỏ), và có những CSDL có dung lượng rất lớn (như các CSDL phục vụ cho việc tính cước điện thoại, quản lý nhân sự trên phạm vi của một tổng công ty lớn). Các CSDL phải được tổ chức quản lý sao cho những người sử dụng có thể tìm kiếm dữ liệu, cập nhật dữ liệu và lấy dữ liệu ra khi cần thiết. Một CSDL có thể được tạo ra và duy trì một cách thủ công và cũng có thể được Tin học hóa.

CSDL là tài nguyên thông tin dùng chung cho nhiều người. Bất kỳ người sử dụng nào trên mạng máy tính, tại các thiết bị đầu cuối, về nguyên tắc có quyền truy nhập khai thác toàn bộ hay một phần dữ liệu theo chế độ trực tuyến hay tương tác mà không phụ thuộc vào vị trí địa lý của người sử dụng với các tài nguyên đó.

CSDL được các hệ ứng dụng khai thác bằng ngôn ngữ con dữ liệu hoặc bằng các chương trình ứng dụng để xử lý, tìm kiếm, tra cứu, sửa đổi, bổ sung hay loại bỏ dữ liệu. Tìm kiếm và tra cứu thông tin là một trong những chức năng qua trọng và phổ biến nhất của các dịch vụ CSDL. Hệ quản trị CSDL (Database Management System - DBMS) là phần mềm điều khiển các hoạt động truy nhập CSDL. Khi người sử dụng đưa ra yêu cầu truy nhập bằng một ngôn ngữ con dữ liệu nào đó, HQTCSDL tiếp nhận và thực hiện các thao tác trên CSDL đã lưu trữ.

3. Ưu điểm của cơ sở dữ liệu

Tổ chức lưu trữ dữ liệu theo lý thuyết CSDL có những ưu điểm sau:

- Giảm bớt dư thừa dữ liệu trong lưu trữ: Trong các ứng dụng lập trình truyền thống, phương pháp tổ chức lưu trữ dữ liệu vừa tốn kém, lãng phí bộ nhớ và các thiết bị lưu trữ, vừa dư thừa thông tin lưu trữ. Nhiều chương trình ứng dụng khác nhau cùng xử lý trên các dữ liệu như nhau, dẫn đến sự dư thừa đáng kể về dữ liệu.

- Tổ chức lưu trữ dữ liệu theo lý thuyết CSDL sẽ tránh được sự không nhất quán trong lưu trữ dữ liệu và bảo đảm được tính toàn vẹn của dữ liệu: Nếu một thuộc tính được mô tả trong nhiều tệp dữ liệu khác nhau và lặp lại nhiều lần thì khi thực hiện việc cập nhật, sửa đổi, bổ sung dữ liệu sẽ không sửa hết nội dung các mục đó. Nếu dữ liệu càng nhiều thì sự sai sót khi cập nhật, bổ sung càng lớn. Khả năng xuất hiện mâu thuẫn, không nhất quán thông tin càng nhiều, dẫn đến không nhất quán dữ liệu trong lưu trữ. Tất yếu kéo theo sự dị thường thông tin, thừa, thiếu và mâu thuẫn thông tin.

- Tổ chức lưu trữ dữ liệu theo lý thuyết CSDL có thể triển khai đồng thời nhiều ứng dụng trên cùng một CSDL. Điều này có nghĩa là các ứng dụng không chỉ chia sẻ chung tài nguyên dữ liệu mà còn trên cùng một CSDL có thể triển khai đồng thời nhiều ứng dụng khác nhau tại các thiết bị đầu cuối khác nhau.

- Tổ chức dữ liệu theo lý thuyết CSDL sẽ thống nhất các tiêu chuẩn, thủ tục và các biện pháp bảo vệ, an toàn dữ liệu. Các hệ CSDL sẽ được quản lý tập trung bởi một người hay một nhóm người quản trị CSDL, bằng các hệ quản trị CSDL. Người quản trị CSDL có thể áp dụng thống nhất các tiêu chuẩn, quy định, thủ tục chung như quy định thống nhất về mẫu biểu báo cáo, thời gian bổ sung, cập nhật dữ liệu. Điều này làm dễ dàng cho công việc bảo trì dữ liệu.

Người quản trị CSDL có thể bảo đảm việc truy nhập tới CSDL, có thể kiểm tra, kiểm soát các quyền truy nhập của người sử dụng. Ngăn chặn các truy nhập trái phép, sai quy định từ trong ra hoặc từ ngoài vào...

4. Những vấn đề cần giải quyết

Để đạt được hiệu quả khi sử dụng CSDL, ta cần giải quyết những vấn đề đặt ra sau đây:

- Tính chủ quyền của dữ liệu: Do tính chia sẻ của CSDL nên tính chủ quyền của dữ liệu có thể bị lu mờ và làm mờ nhạt tinh thần trách nhiệm, được thể hiện trên vấn đề an toàn dữ liệu, khả năng biểu diễn các mối liên hệ ngữ nghĩa của dữ liệu, và tính chính xác của dữ liệu. Điều này có nghĩa là người khai thác CSDL phải có nghĩa vụ cập nhật các thông tin mới nhất của CSDL.

- Tính bảo mật và quyền khai thác thông tin của người sử dụng: Do có nhiều người được phép khai thác CSDL một cách đồng thời nên cần phải có một cơ chế bảo mật và phân quyền hạn khai thác CSDL. Các hệ điều hành nhiều người sử dụng hay hệ điều hành mạng cục bộ (Novell Netware, Windows for Workgroup, WinNT...) đều có cung cấp cơ chế này.

- Tranh chấp dữ liệu: Nhiều người được phép truy nhập vào cùng một tài nguyên dữ liệu của CSDL với những mục đích khác nhau như xem, thêm, xóa hoặc sửa dữ liệu, do đó cần phải có một cơ chế ưu tiên khi truy nhập dữ liệu. Cơ chế ưu tiên có thể được thực hiện bằng việc cấp quyền ưu tiên cho từng người khai thác - người nào được cấp quyền hạn ưu tiên cao hơn thì được ưu tiên truy nhập dữ liệu trước; có thể dựa trên thời điểm truy nhập - ai có yêu cầu truy xuất trước thì có quyền truy nhập dữ liệu trước...

- Đảm bảo dữ liệu khi có sự cố: Việc quản lý dữ liệu tập trung có thể làm tăng khả năng mất mát hoặc sai lệch thông tin khi có sự cố như mất điện đột xuất, một phần đĩa lưu trữ CSDL bị hư hỏng v.v... Một số hệ điều hành mạng có cung cấp dịch vụ sao lưu ảnh đĩa cứng (cơ chế sử dụng đĩa cứng dự phòng - RAID), tự động kiểm tra và khắc phục lỗi khi có sự cố, Tuy nhiên, bên cạnh dịch vụ của hệ điều hành, để đảm bảo CSDL luôn luôn ổn định, một CSDL nhất thiết phải có một cơ chế khôi phục dữ liệu khi các sự cố bất ngờ xảy ra.

Giá phải trả cho ứng dụng CSDL

Như với bất kỳ quyết định kinh doanh nào, ứng dụng theo hướng CSDL cần trả giá và chịu một vài rủi ro như cần đến nhân sự mới và chuyên sâu, chi phí quản lý và thực hiện, giải quyết va chạm tâm lý...

- Nhu cầu về nhân sự mới, chuyên môn: Cần phải thuê hay tự đào tạo các chuyên gia công nghệ thông tin để phân tích, thiết kế, thực hiện CSDL trong tổ chức. Mặt khác, công nghệ ngày càng phát triển nhanh đã khiến các hệ thống CSDL thường xuyên cập nhật tri thức cho các cán bộ.

- Chi phí về quản lý, thực hiện: Hệ thống quản trị CSDL nhiều người dùng thường lớn và phức tạp, đòi hỏi nhiều người tham gia và bảo trì. Người ta cần tính đến chi phí thiết bị máy tính, chi phí phần mềm, chi phí truyền thông.

- Chi phí chuyển đổi hệ thống: Trước khi áp dụng CSDL, tổ chức đã xử lý dữ liệu theo phương pháp thủ công, hay sử dụng hệ thống xử lý tập tin. Để chuyển công nghệ xử lý dữ liệu sang cách tiếp cận mới, tổ chức cần có kinh phí chuyển các mô hình cũ sang mô hình mới, chuyển đổi dữ liệu cho phù hợp với môi trường mới.

- Va chạm về quan điểm sử dụng CSDL: Trước khi CSDL được áp dụng trong tổ chức, người ta cần thống nhất về mô tả dữ liệu, các khuôn dạng thể hiện dữ liệu... Sau khi hình thành CSDL, việc sử dụng dữ liệu của CSDL cần đạt hiệu quả. Cần khuyến khích các bộ phận phát huy tốt hệ thống CSDL vì rất có thể một số bộ phận nào đó muốn độc quyền thông tin hay e ngại dùng hệ thống tự động hoá, cho nên có xu hướng giảm hiệu quả của CSDL.

5. Các đối tượng sử dụng CSDL

- Người sử dụng CSDL không chuyên về lĩnh vực tin học và CSDL, do đó CSDL cần có các công cụ để cho những người sử dụng không chuyên có thể sử dụng để khai thác CSDL khi cần thiết.

- Chuyên viên Tin học biết khai thác CSDL: Những người này có thể xây dựng các ứng dụng khác nhau phục vụ cho các mục đích khác nhau trên CSDL.

- Người quản trị CSDL: Đây là những người hiểu biết về tin học, về các hệ quản trị CSDL và hệ thống máy tính. Họ là người tổ chức CSDL (khai báo cấu trúc CSDL, ghi nhận các yêu cầu bảo mật cho các dữ liệu cần bảo vệ...), do đó họ phải nắm rõ các vấn đề kỹ thuật về CSDL để có thể phục hồi dữ liệu khi có sự cố. Họ là những người cấp quyền hạn khai thác CSDL, do vậy họ có thể giải quyết được các vấn đề tranh chấp dữ liệu, nếu có.

6. Các ứng dụng của cơ sở dữ liệu

Hiện nay, hầu như CSDL gắn liền với mọi ứng dụng của Tin học; chẳng hạn như việc quản lý hệ thống thông tin trong các cơ quan nhà nước, việc lưu trữ và xử lý thông tin trong các doanh nghiệp, trong các lĩnh vực nghiên cứu khoa học, trong công tác giảng dạy, cũng như trong việc tổ chức thông tin đa phương tiện...

§ 2. KIẾN THỨC TỔNG QUÁT CỦA HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**I. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU – HỆ CƠ SỞ DỮ LIỆU****1. Khái niệm hệ quản trị cơ sở dữ liệu**

Như chúng ta đã biết, kích thước và độ phức tạp của CSDL rất khác nhau.

Ví dụ:

- Danh bạ điện thoại của một quốc gia, một thành phố... chứa tới hàng triệu số và những thông tin cần thiết về khách hàng.
- Trong trường Đại học có tới hàng ngàn sinh viên. Nhà trường phải quản lý tất cả những thông tin liên quan đến sinh viên như: Tên, ngày sinh, quê quán, địa chỉ, kết quả học tập...

Ta thấy, bộ nhớ là vấn đề cần phải được giải quyết. Tuy nhiên, vấn đề quan trọng hơn ở đây lại là cách thức tổ chức dữ liệu trong một CSDL để phục vụ cho việc truy cập, tìm kiếm, cập nhật... nhanh chóng và an toàn hơn.

Để giải quyết tốt tất cả các vấn đề đặt ra cho một CSDL như đã nêu trên, cần phải có một hệ thống các phần mềm chuyên dụng được gọi là hệ quản trị CSDL (Database Management System - DBMS). Đó là các công cụ hỗ trợ tích cực cho các nhà phân tích và thiết kế CSDL và những người khai thác CSDL.

Các khái niệm:

Một hệ quản trị CSDL là một tập hợp các chương trình giúp cho người sử dụng quản lý cấu trúc và dữ liệu của CSDL, điều khiển truy xuất dữ liệu trong CSDL, duy trì và khai thác một CSDL.

Định nghĩa một CSDL: Tạo cấu trúc dữ liệu tương ứng với mô hình dữ liệu được chọn, bao gồm việc đặc tả các kiểu dữ liệu, các cấu trúc và các ràng buộc cho các dữ liệu sẽ được lưu trữ trong CSDL.

Xây dựng một CSDL: Là quá trình lưu trữ các dữ liệu trên các phương tiện lưu trữ được hệ quản trị CSDL kiểm soát.

Thao tác một CSDL: Bao gồm các chức năng như truy vấn CSDL để lấy ra các dữ liệu cụ thể, cập nhật CSDL để phản ánh các thay đổi trong thế giới nhỏ và tạo ra các báo cáo từ các dữ liệu.

Cho đến nay có khá nhiều hệ quản trị CSDL mạnh được đưa ra thị trường như: Visual FoxPro, Microsoft Access, SQL Server, DB2, Sybase, Paradox, Informix, Oracle... với các chất lượng khác nhau.

2. Các chức năng của một hệ quản trị cơ sở dữ liệu

Một hệ quản trị CSDL thực hiện các chức năng sau:

a. Cung cấp môi trường tạo lập CSDL

Một hệ quản trị CSDL phải cung cấp một môi trường cho người dùng dễ dàng khai báo kiểu dữ liệu, các cấu trúc dữ liệu thể hiện thông tin và các ràng buộc trên dữ liệu. Với các hệ quản trị CSDL hiện nay, người dùng có thể tạo lập CSDL thông qua các giao diện đồ họa. Để thực hiện được chức năng này, mỗi hệ quản trị CSDL cung cấp cho người dùng một ngôn ngữ định nghĩa dữ liệu. Ta có thể hiểu ngôn ngữ định nghĩa dữ liệu thực chất là hệ thống các ký hiệu để mô tả CSDL.

b. Cung cấp môi trường cập nhật và khai thác dữ liệu

Ngôn ngữ để người dùng diễn tả yêu cầu cập nhật hay tìm kiếm, kết xuất thông tin được gọi là ngôn ngữ thao tác dữ liệu.

Thao tác dữ liệu gồm:

- Cập nhật: Nhập thêm, sửa, xoá dữ liệu.
- Khai thác: Tìm kiếm, kết xuất dữ liệu.

Trong thực tế, ngôn ngữ định nghĩa dữ liệu và ngôn ngữ thao tác dữ liệu là 2 thành phần của một ngôn ngữ CSDL duy nhất. Ngôn ngữ CSDL được sử dụng phổ biến hiện nay là SQL (Structured Query Language).

c. Cung cấp công cụ kiểm soát, điều khiển truy cập vào CSDL

Để góp phần đảm bảo được các yêu cầu đặt ra cho một hệ CSDL, hệ quản trị CSDL phải có các bộ chương trình thực hiện những nhiệm vụ sau:

- Phát hiện và ngăn chặn sự truy cập không được phép: Chức năng này góp phần đáp ứng yêu cầu an toàn và bảo mật thông tin.
- Duy trì tính nhất quán của dữ liệu.
- Tổ chức và điều khiển các truy cập đồng thời để bảo vệ các ràng buộc toàn vẹn và tính nhất quán.
- Khôi phục CSDL khi có sự cố ở phần cứng hay phần mềm.
- Quản lý các mô tả dữ liệu.

Nói chung, mọi hệ quản trị CSDL đều có thể cung cấp các chương trình nêu trên, nhưng các hệ quản trị CSDL khác nhau có chất lượng và khả năng khác nhau khi đáp ứng các nhu cầu thực tế. Chẳng hạn, một hệ quản trị CSDL dùng trên một máy tính cá nhân chỉ cung cấp những phương tiện bảo vệ dữ liệu, duy trì tính nhất quán dữ liệu, khôi phục dữ liệu một cách hạn chế. Trong khi đó, những hệ quản trị CSDL lớn, phục vụ cùng lúc cho nhiều người dùng, ngoài các phương tiện kể trên có thể có thêm những chức năng khác như xử lý các truy cập đồng thời...

Các hệ quản trị CSDL luôn phát triển theo hướng đáp ứng các đòi hỏi ngày càng cao hơn của người dùng, bởi vậy các chức năng của hệ quản trị CSDL ngày càng được mở rộng hơn.

3. Các thành phần của một hệ quản trị cơ sở dữ liệu

Các hệ CSDL đều có các thành phần sau:

- Ngôn ngữ giao tiếp giữa người sử dụng và CSDL, bao gồm:

- + Ngôn ngữ mô tả dữ liệu (Data Definition Language - DDL): Cho phép khai báo cấu trúc CSDL, khai báo các mối liên hệ của dữ liệu (Data Relationship) và các quy tắc (Rules, Constraint) quản lý áp đặt lên các dữ liệu đó.

- + Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML): Cho phép người sử dụng có thể thêm (Insert), xóa (Delete), sửa (Update) dữ liệu trong CSDL.

- + Ngôn ngữ truy vấn dữ liệu (Data Query Language - DQL): Cho phép những người khai thác (chuyên nghiệp hoặc không chuyên) sử dụng để tìm kiếm các thông tin cần thiết trong CSDL.

- + Ngôn ngữ quản lý dữ liệu (Data Control Language - DCL): Cho phép những người quản trị hệ thống khai báo bảo mật thông tin và cấp quyền hạn khai thác CSDL cho người sử dụng.

- Từ điển dữ liệu (Data Dictionary): Dùng để ghi nhận các thành phần cấu trúc của CSDL, mô tả các ánh xạ liên kết, mật mã, quyền hạn sử dụng, các chương trình ứng dụng...

- Có biện pháp phân quyền và bảo mật tốt khi có yêu cầu bảo mật.
- Cơ chế giải quyết vấn đề tranh chấp dữ liệu. Mỗi hệ quản trị CSDL cũng có thể cài đặt một cơ chế riêng để giải quyết các vấn đề này. Một số biện pháp sau đây được sử dụng:
 - + Cấp quyền ưu tiên cho từng người sử dụng (người quản trị CSDL thực hiện).
 - + 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.
- Hệ quản trị CSDL cũng phải có cơ chế sao lưu (Backup) và phục hồi (Restore) dữ liệu khi có sự cố xảy ra.
- Hệ quản trị CSDL phải cung cấp một giao diện (Interface) tốt, dễ sử dụng, dễ hiểu cho những người sử dụng không chuyên.
- Hệ quản trị CSDL phải bảo đảm tính độc lập giữa dữ liệu và chương trình: Khi có sự thay đổi dữ liệu (như sửa đổi cấu trúc lưu trữ các bảng dữ liệu...) thì các chương trình ứng dụng đang chạy trên CSDL đó vẫn không cần phải được viết lại, hay cũng không làm ảnh hưởng đến những người sử dụng khác.

4. Hệ cơ sở dữ liệu

Người ta gọi CSDL và hệ quản trị CSDL bằng một thuật ngữ chung là hệ CSDL. Môi trường của một hệ CSDL được mô tả bằng hình vẽ dưới đây:



Hình: Hệ cơ sở dữ liệu

Hệ CSDL là một hệ thống gồm người dùng, các phần mềm ứng dụng, hệ quản trị CSDL và CSDL. Trong đó, người dùng chỉ can thiệp trực tiếp với các chương trình ứng dụng, các chương trình ứng dụng sẽ thông qua hệ quản trị CSDL để truy cập CSDL và thực hiện các công việc bên dưới, đáp ứng yêu cầu của người dùng.

II. MÔ HÌNH CƠ SỞ DỮ LIỆU

1. Khái niệm mô hình cơ sở dữ liệu

Các loại cấu trúc CSDL và mối liên hệ giữa chúng đóng vai trò rất lớn trong việc xác định tính hiệu quả của hệ quản trị CSDL. Vì vậy, thiết kế CSDL trở thành hoạt động chính trong môi trường CSDL.

Việc thiết kế CSDL được thực hiện đơn giản hơn nhiều khi ta sử dụng các mô hình. Các mô hình là sự trừu tượng đơn giản của các sự kiện trong thế giới thực. Các trừu tượng như vậy cho phép ta khảo sát các đặc điểm của các thực thể và các mối liên hệ được tạo ra giữa các thực thể đó. Việc thiết kế các mô hình tốt sẽ đưa ra các CSDL tốt và trên cơ sở đó sẽ có các ứng dụng tốt.

Một mô hình CSDL là một tập hợp các khái niệm, các luật hay quy tắc dùng để biểu diễn, mô tả một cách trừu tượng cấu trúc của CSDL. Cấu trúc của một CSDL là các kiểu dữ liệu,

các mối liên kết và các ràng buộc phải tuân theo trên các dữ liệu. Nhiều mô hình còn có thêm một tập hợp các phép toán cơ bản để đặc tả các thao tác trên CSDL.

Thông qua mô hình dữ liệu, người thiết kế sẽ mô tả toàn cảnh CSDL được thiết kế bao gồm:

- Các đối tượng, thực thể được quản lý
- Các mối quan hệ giữa các đối tượng
- Các ràng buộc dữ liệu thể hiện các quy tắc quản lý ảnh hưởng đến các đối tượng quản lý.

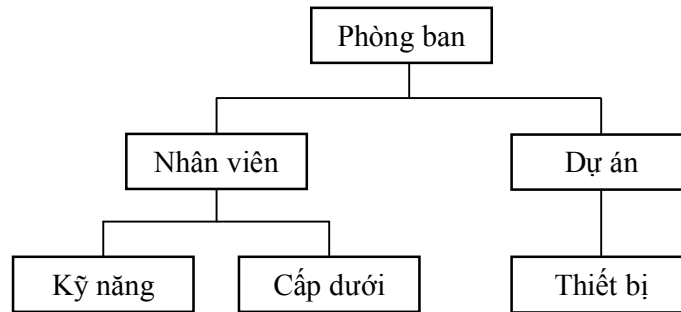
2. Các mô hình cơ sở dữ liệu

Có nhiều mô hình được đề nghị, ở đây chúng ta sẽ khảo sát các mô hình đặc trưng nhất:

a. Mô hình phân cấp

Mô hình phân cấp hay mô hình CSDL dạng cây được tổ chức theo cấu trúc từ trên xuống dưới giống như cây lộn ngược. Mỗi nút tương ứng với một kiểu dữ liệu, có thể có một hoặc nhiều trường, mô tả thực thể và một nhánh cây tạo nên một liên kết giữa kiểu dữ liệu này với kiểu dữ liệu khác. Mỗi nút đều có một nút cha và nhiều nút con, trừ nút gốc không có nút cha.

Ví dụ mô hình phân cấp quản lý nhân sự của một Công ty:



Hình: Mô hình phân cấp

Tuy nhiên, mô hình chỉ thể hiện được quan hệ 1-n (một-nhiều), tức là chỉ mô tả được trường hợp nút cha có nhiều nút con như một phòng thì có thể có nhiều nhân viên hay một phòng có thể có nhiều dự án, còn trường hợp ngược lại thì không. Chẳng hạn nếu một dự án mà thuộc về nhiều phòng thì dự án phải được lưu ở nhiều nơi khác nhau. Điều này gây dư thừa dữ liệu và lãng phí không gian lưu trữ.

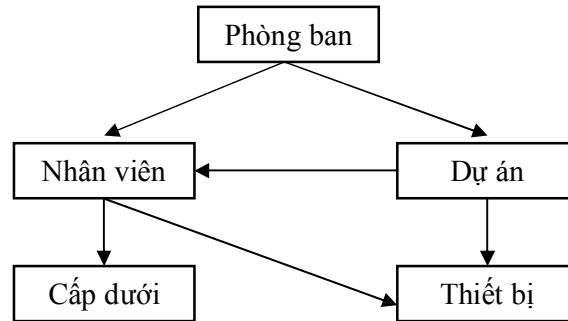
Điểm nổi bật trong các thủ tục truy xuất đến một đối tượng trong mô hình phân cấp là đường dẫn đi từ gốc đến phần tử cần xét trong cây phân cấp.

b. Mô hình mạng

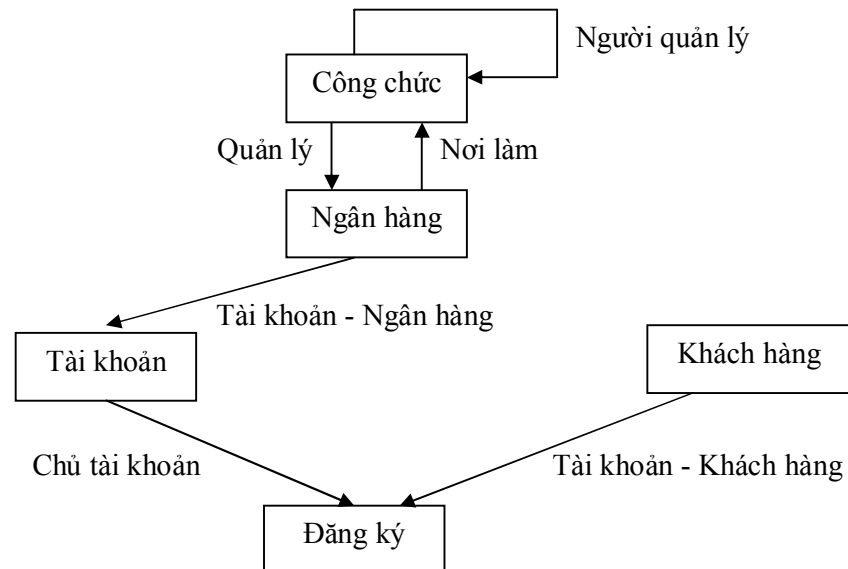
Mô hình mạng được biểu diễn như một đồ thị có hướng. Dữ liệu được biểu diễn bởi một tập các bản ghi, các mối quan hệ được biểu diễn bởi các mối nối (links) có thể được xem như những con trỏ. Xuất phát từ một đối tượng (biểu diễn bằng một bản ghi), có thể nối với một nút bất kỳ để biểu diễn một liên kết 1-n thông qua con trỏ liên kết, và có thể có nhiều mối quan hệ đến những đối tượng khác

Sự khác nhau chính giữa hệ thống mạng và hệ thống phân cấp là mô hình mạng không ràng buộc về số và hướng của các liên kết thiết lập giữa các nút.

Ví dụ 1: Mô hình quản lý nhân sự của một Công ty



Ví dụ 2:



c. Mô hình quan hệ

Cơ sở dữ liệu quan hệ được xây dựng theo lý thuyết do E.F.Codd giới thiệu năm 1970. Thuật ngữ “quan hệ” là do bảng dữ liệu 2 chiều được Codd gọi là bảng quan hệ. Mô hình quan hệ từ năm 1980 đã trở thành mô hình được dùng rộng rãi để phát triển hệ quản trị CSDL.

Đặc điểm của mô hình dữ liệu quan hệ:

- Dữ liệu được thể hiện trong các bảng. Các bảng gọi là các “quan hệ”. Mỗi bảng gồm các dòng và các cột, mỗi cột có một tên duy nhất. Các dòng gọi là các “bộ”, cột là “thuộc tính”.
- Mỗi dòng cho thông tin về một đối tượng cụ thể, bao gồm một bộ các giá trị tương ứng với các cột, mỗi giá trị thể hiện thông tin về một thuộc tính của đối tượng đó, tên thuộc tính này chính là tên cột mà giá trị đó được hiển thị.
- Mỗi liên kết giữa các đối tượng được thể hiện bằng mối liên kết giữa các bảng nhờ vào sự xuất hiện trùng lặp của một số thuộc tính ở hơn một bảng.

Ví dụ: Ta có các bảng dữ liệu trong CSDL sau:

KHACH_HANG	MaKH	HoTen	Tuoi	ĐịaChí
	Khách hàng 1	Nguyễn Văn Quang	16	68, Nguyễn Du
	Khách hàng 2	Hồ Hải Hà	18	23/10, Nguyễn Trãi
	Khách hàng 3	Dương Quốc Toàn	21	32/7, Nguyễn Trãi

DON_HANG	KhachHang	DonHang
	Khách hàng 1	Đơn hàng 1
	Khách hàng 1	Đơn hàng 2
	Khách hàng 2	Đơn hàng 3
	Khách hàng 2	Đơn hàng 4
	Khách hàng 3	Đơn hàng 5

Cơ sở dữ liệu quan hệ dùng các thuộc tính để liên kết dữ liệu giữa các bảng khác nhau thay vì dùng con trỏ để liên kết tập bản ghi như trong mô hình mạng. Chẳng hạn thuộc tính **MaKH** của bảng KHACH_HANG và thuộc tính **KhachHang** của bảng DON_HANG là 2 thuộc tính dùng để liên kết hai bảng quan hệ này. Đi sâu vào chi tiết của mô hình quan hệ sẽ được giới thiệu trong Chương sau: “Mô hình cơ sở dữ liệu quan hệ”.

CHƯƠNG 2 – MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ

(Tổng số: 3 tiết, Lý thuyết: 3 tiết)

§ 1. NHỮNG KHÁI NIỆM CƠ BẢN

I. MÔ HÌNH QUAN HỆ LÀ GÌ?

Mô hình CSDL quan hệ (gọi tắt là mô hình quan hệ) do Edgar Frank Codd đề xuất năm 1970. Mô hình này bao gồm:

- Một hệ thống các ký hiệu để mô tả dữ liệu dưới dạng dòng và cột như quan hệ, bộ, thuộc tính, khóa chính, khóa ngoại...
- Một tập hợp các phép toán thao tác trên dữ liệu: Phép toán tập hợp, phép toán quan hệ.
- Ràng buộc toàn vẹn quan hệ.

Các hệ quản trị CSDL quan hệ ngày nay được xây dựng dựa vào lý thuyết của mô hình quan hệ.

Chẳng hạn với bài toán quản lý điểm thi của sinh viên sau:

MASV	HOTEN	MONHOC	TENKHOA	DIEMTHI
99001	TRAN DAN THU	CO SO DU LIEU	CONG NGHE THONG TIN	3.0
99002	NGUYEN HA DA THAO	CO SO DU LIEU	CONG NGHE THONG TIN	8.0
99001	TRAN DAN THU	TIN HOC VAN PHONG	CONG NGHE THONG TIN	6.0
99005	LE THANH TRUNG	TIN HOC VAN PHONG	ANH VAN	5.0

Nếu lưu trữ dữ liệu theo dạng bảng với các cột MASV, HOTEN, MONHOC, TENKHOA, DIEMTHI thì các giá trị của các cột HOTEN, MONHOC, TENKHOA sẽ bị trùng lặp. Sự trùng lặp này gây nên một số vấn đề:

- Ta không thể lưu trữ một sinh viên mới khi sinh viên này chưa có điểm thi
- Khi cần sửa đổi họ tên sinh viên thì ta phải sửa tất cả các dòng có liên quan đến sinh viên này. Điều này dễ gây ra tình trạng dữ liệu thiếu nhất quán.
- Khi có nhu cầu xóa điểm thi của một sinh viên kéo theo khả năng xóa luôn họ tên sinh viên đó.

Để lưu trữ đúng với mô hình quan hệ ta phải thay MONHOC bằng MAMH, thay TENKHOA bằng MAKHOA, tách một bảng dữ liệu lớn đó ra thành nhiều bảng con, như mô hình dưới.

MASV	HOTEN	MAKHOA
99001	TRAN DAN THU	CNTT
99002	NGUYEN HA DA THAO	CNTT
99001	TRAN DAN THU	CNTT
99005	LE THANH TRUNG	AV

MAKHOA	TENKHOA
CNTT	CONG NGHE THONG TIN
AV	ANH VAN

MASV	MAMH	DIEMTHI
99001	CSDL	3.0
99002	CSDL	8.0
99001	THVP	6.0
99005	THVP	5.0

MAMH	TENMH	SOTIET
CSDL	CO SO DU LIEU	90
THVP	TIN HOC VAN PHONG	90

II. NHỮNG KHÁI NIỆM CƠ BẢN CỦA MÔ HÌNH QUAN HỆ

Mô hình quan hệ biểu thị CSDL như một tập các quan hệ. Mỗi quan hệ có thể được biểu diễn như một bảng giá trị, mỗi một dòng trong bảng biểu thị một tập hợp các giá trị dữ liệu liên quan với nhau. Tên bảng và tên các cột dùng để giúp giải thích ý nghĩa của các giá trị trong mỗi hàng. Mọi giá trị trong một cột đều cùng một kiểu dữ liệu.

Theo thuật ngữ mô hình quan hệ, mỗi hàng được gọi là một bộ, mỗi đầu cột được gọi là một thuộc tính, và bảng được gọi là một quan hệ. Kiểu dữ liệu mô tả các kiểu của dữ liệu xuất hiện trong mỗi cột gọi là một miền.

1. Thực thể (Entity)

Thực thể là một đối tượng cụ thể hay trừu tượng trong thế giới thực mà nó tồn tại và có thể phân biệt được với các đối tượng khác. Tên thực thể là các danh từ.

Ví dụ:

- Bạn *Nguyễn Văn A* là một thực thể cụ thể.
- *Sinh viên* là một thực thể trừu tượng.

2. Thuộc tính (Attribute)

Thuộc tính là các thông tin riêng biệt cần lưu trữ của các đối tượng.

Chẳng hạn, với bài toán quản lý điểm thi của sinh viên:

- Thực thể *Sinh viên* có một số thuộc tính: Họ tên, ngày sinh, giới tính, tỉnh thường trú, học bổng, lớp đang theo học... các đặc trưng này gọi là thuộc tính của đối tượng sinh viên.
- Mỗi môn học có một số thuộc tính: Mã môn, Tên môn, Số đơn vị học trình.

Một thuộc tính hay một tập thuộc tính mà giá trị của nó xác định duy nhất mỗi thực thể trong tập các thực thể được gọi là khóa đối với tập thực thể này. (*Khái niệm khóa sẽ được trình bày sau*)

Mỗi thuộc tính có các thành phần: Tên gọi, kiểu dữ liệu, miền giá trị.

a. Tên gọi

Thường được đặt một cách ngắn gọn, gợi nhớ.

Ví dụ: Đối tượng *SinhVien*, và *GiangVien* đều có thuộc tính tên thì đặt là *TenSV*, *TenGV*.

Quy ước: Trong lý thuyết, nếu không cần lưu ý đến ngữ nghĩa, thì tên của các thuộc tính sẽ được ký hiệu bằng các chữ cái in hoa đầu tiên trong bảng chữ cái La-tinh: A, B, C, D...

Trong cùng một đối tượng không thể có 2 thuộc tính cùng tên.

Tên thuộc tính phải được đặt một cách gọi nhớ, không nên đặt tên thuộc tính quá dài (vì như thế sẽ làm cho việc viết các câu lệnh truy vấn trở nên vất vả hơn), nhưng cũng không nên đặt tên thuộc tính quá ngắn (vì nó sẽ không cho thấy ngữ nghĩa của thuộc tính).

Không đặt trùng tên 2 thuộc tính mang ngữ nghĩa khác nhau thuộc 2 đối tượng khác nhau.

b. Kiểu dữ liệu

Các kiểu dữ liệu thường được sử dụng trong các hệ quản trị CSDL:

- Text (hoặc Character, Char, String): Kiểu văn bản.
- Number (hoặc Numeric, Float): Kiểu số
- Yes/No (hoặc Boolean): Kiểu luận lý
- Date/Time: Kiểu thời gian (Ngày/tháng/năm + Giờ:phút:giây)
- Memo (hoặc VarChar): Kiểu văn bản có độ dài thay đổi.

Chẳng hạn, với sinh viên Nguyễn Văn Thành thì các thuộc tính Họ và tên, Mã số sinh viên thuộc kiểu chuỗi, thuộc tính Ngày sinh thuộc kiểu ngày tháng, Hộ khẩu thường trú kiểu chuỗi, thuộc tính Hình ảnh kiểu hình ảnh...

Mỗi hệ quản trị CSDL có thể gọi tên các kiểu dữ liệu nói trên bằng các tên gọi khác nhau, ngoài ra còn bổ sung thêm một số kiểu dữ liệu riêng của mình. Chẳng hạn, Microsoft Access có kiểu dữ liệu OLE để chứa các đối tượng nhúng như hình ảnh, âm thanh, video... ORACLE có kiểu dữ liệu LONG cho phép chứa dữ liệu có kích thước lớn tới 2 tỷ bytes.

c. Miền giá trị (Domain)

Tùy thuộc vào quy tắc quản lý, mỗi thuộc tính có thể chỉ chứa một số giá trị nào đó của kiểu dữ liệu. Tập giá trị hợp lệ của một thuộc tính được gọi là miền giá trị của thuộc tính.

Một số ví dụ về định nghĩa miền:

- Họ tên: Tập hợp các dãy chữ cái có độ dài tối đa 30.
- Tuổi: Tập các số nguyên nằm trong khoảng từ 1 đến 80.
- Điểm học tập: Có miền giá trị từ 0 đến 10.
- Giới tính: Tập hợp gồm 2 giá trị “Nam”, “Nữ”.

Trong nhiều hệ quản trị CSDL, người ta thường đưa thêm vào miền giá trị của các thuộc tính một giá trị đặc biệt gọi là giá trị rỗng (NULL). Tùy theo ngữ cảnh mà giá trị này có thể đặc trưng cho một giá trị không thể xác định được hoặc một giá trị chưa được xác định ở vào thời điểm nhập tin nhưng có thể được xác định vào một thời điểm khác.

d. Loại giá trị của thuộc tính

- Đơn trị: Các thuộc tính có giá trị duy nhất cho một thực thể. (VD: Số CMND, Mã SV...)
- Đa trị: Các thuộc tính có một tập giá trị cho cùng một thực thể. (VD: Bảng cấp)
- Suy diễn được: Các thuộc tính có giá trị được suy ra từ các thuộc tính khác. (VD: Thuộc tính Tuổi được suy ra từ thuộc tính Năm sinh...)

3. Lược đồ quan hệ (Relation Schema)

Tập tất cả các thuộc tính cần quản lý của một đối tượng cùng với mối liên hệ giữa chúng được gọi là lược đồ quan hệ. Lược đồ quan hệ Q với tập thuộc tính $\{A_1, A_2, \dots, A_n\}$ được viết là $Q(A_1, A_2, \dots, A_n)$. Tập các thuộc tính của Q được ký hiệu là Q^+ .

Thường khi thành lập một lược đồ, người thiết kế luôn gán cho nó một ý nghĩa nhất định, ý nghĩa đó gọi là tên từ của lược đồ quan hệ đó. Khi phát biểu tên từ cho một lược đồ quan hệ,

người thiết kế cần phải mô tả đầy đủ ý nghĩa để người khác tránh hiểu nhầm. Dựa vào tên từ người ta xác định được tập thuộc tính khóa của lược đồ quan hệ. (Khái niệm khoá sẽ được trình bày sau)

Chẳng hạn, đối với lược đồ quan hệ sinh viên (được đặt tên là Sv):

Sv(MASV, HOSV, TENSU, NU, NGAYSINH, MALOP, HOCBONG, TINH)

+ Tên quan hệ: Sv

+ Tập thuộc tính: {MASV, HOSV, TENSU, NU, NGAYSINH, MALOP, HOCBONG, TINH}, ký hiệu Sv⁺

+ Tên từ của lược đồ quan hệ sinh viên là: “Mỗi sinh viên có một mã sinh viên (MASV) duy nhất. Mỗi mã sinh viên xác định tất cả các thuộc tính của sinh viên đó như họ tên (HOTEN), nam/nữ (NU), ngày sinh (NGAYSINH), lớp đang theo học (MALOP), học bổng (HOCBONG), tỉnh cư trú (TINH).

+ Bậc của lược đồ quan hệ sinh viên là: 8

Tập hợp các lược đồ quan hệ cùng nằm trong một hệ thống quản lý được gọi là một lược đồ CSDL. Trong một lược đồ CSDL, các tên lược đồ quan hệ là duy nhất.

Ví dụ: Lược đồ CSDL quản lý điểm sinh viên có thể gồm các lược đồ quan hệ sau:

Sv(MASV, HOSV, TENSU, NU, NGAYSINH, MALOP, TINH, HOCBONG)

Lop(MALOP, TENLOP, SISO, MAKHOA)

Kh(MAKHOA, TENKHOA, SOCBGD)

Mh(MAMH, TENMH, SOTIET)

Kq(MASV, MAMH, DIEMTHI)

4. Quan hệ (Relation)

Sự thể hiện của lược đồ quan hệ Q ở một thời điểm nào đó được gọi là quan hệ. Tại những thời điểm khác nhau thì quan hệ sẽ có những thể hiện khác nhau. Như vậy, trên một lược đồ quan hệ có thể định nghĩa rất nhiều quan hệ.

Thường ta dùng các ký hiệu như R, S, Q để chỉ các lược đồ quan hệ, còn quan hệ được định nghĩa trên nó tương ứng được ký hiệu là r, s, q.

Một quan hệ có n thuộc tính được gọi là quan hệ n ngôi.

Để chỉ quan hệ r xác định trên lược đồ quan hệ Q ta có thể viết r(Q).

Mỗi quan hệ (hay bảng quan hệ) như là một bảng 2 chiều gồm các dòng và các cột

Mỗi dòng của bảng là một bộ (còn gọi là bản ghi, hoặc mẫu tin).

Mỗi cột của bảng là một thuộc tính (còn gọi là trường).

Ví dụ: Xét một thể hiện của lược đồ quan hệ SINHVIEN như sau:

MASV	HOTEN	MAMH	MAKHOA	DIEMTHI
99001	TRAN DAN THU	CSDL	CNTT	3.0
99002	NGUYEN HA DA THAO	CSDL	CNTT	8.0
99001	TRAN DAN THU	THVP	CNTT	6.0
99005	LE THANH TRUNG	THVP	AV	5.0

+ Quan hệ SINHVIEN là quan hệ 5 ngôi.

+ Mỗi một cột là một thuộc tính của quan hệ SINHVIEN.

+ Mỗi một dòng là một Sinh viên.

Chú ý:

- Các thuộc tính của một quan hệ đều có tên khác nhau.
- Trong một quan hệ, tất cả các giá trị của một thuộc tính có cùng một miền giá trị.
- Quan hệ là một bảng không chứa các hàng giống hệt nhau. Mỗi hàng là duy nhất, không thể có 2 hàng có cùng các giá trị ở tất cả vùng tin.
- Thứ tự của các hàng và các cột là không quan trọng. Thứ tự các giá trị trong một hàng là quan trọng.
- Mỗi quan hệ là một bảng, nhưng không phải mọi bảng đều là quan hệ.

5. Bộ (Tuple)

Bộ là tập mỗi giá trị liên quan của tất cả các thuộc tính của một lược đồ quan hệ.

Thường người ta dùng các chữ cái thường (như t, p, q...) để biểu diễn các bộ. Chẳng hạn để nói bộ t thuộc quan hệ r ta viết: $t \in r$.

Chẳng hạn quan hệ SINHVIEN sau có 4 bộ:

MASV	HOTEN	MAMH	MAKHOA	DIEMTHI
99001	TRAN DAN THU	CSDL	CNTT	3.0
99002	NGUYEN HA DA THAO	CSDL	CNTT	8.0
99001	TRAN DAN THU	THVP	CNTT	6.0
99005	LE THANH TRUNG	THVP	AV	5.0

Ta có: $t_1 = (99001, \text{TRAN DAN THU}, \text{CSDL}, \text{CNTT}, 3.0)$ là một bộ của quan hệ SINHVIEN.

§ 2. KHÁI NIỆM VỀ KHOÁ

I. SIÊU KHOÁ – KHÓA

1. Siêu khoá (Super Key)

Một quan hệ được định nghĩa như một tập hợp các bộ. Theo định nghĩa tập hợp, các phần tử của một tập hợp là khác nhau, vì vậy mọi bộ trong quan hệ phải khác nhau. Điều đó có nghĩa là: Không thể có 2 bộ có cùng một tổ hợp giá trị cho tất cả các thuộc tính của chúng.

Thông thường, có tồn tại các tập con của các thuộc tính của một lược đồ quan hệ có tính chất là không có 2 bộ nào ở trong mọi trạng thái quan hệ r của R có cùng một tổ hợp giá trị cho các thuộc tính của nó.

Giả sử chúng ta ký hiệu một tập con như vậy là S , khi đó với 2 bộ khác nhau bất kỳ t_1 và t_2 trong một trạng thái quan hệ r của lược đồ quan hệ R chúng ta có ràng buộc là:

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

Tập hợp S có các thuộc tính như vậy được gọi là một siêu khoá của lược đồ quan hệ R . Một siêu khoá S xác định rõ một ràng buộc về tính duy nhất, phát biểu rằng không có 2 bộ khác nhau trong một trạng thái r của R có cùng một giá trị cho S .

Một quan hệ có thể có nhiều siêu khoá.

Mỗi quan hệ có ít nhất một siêu khoá mặc định, đó là tập hợp tất cả các thuộc tính của nó.

2. Khóa (Key)

Một khoá K của một lược đồ quan hệ R là một siêu khoá của R với tính chất là nếu bỏ đi bất kỳ thuộc tính A nào ra khỏi K thì sẽ còn lại một tập K không phải là siêu khoá của R . Như vậy, một khoá là một siêu khoá tối thiểu (siêu khoá chứa ít thuộc tính nhất), nghĩa là đó là một siêu khoá mà ta không thể vứt bỏ thuộc tính nào ra khỏi nó mà vẫn giữ được ràng buộc về tính duy nhất.

Ví dụ: Xét quan hệ SINHVIEN như sau:

Mã số	Họ tên	Ngày sinh	Giới tính	Địa chỉ
4515202	Lê Văn	12/09/84	Nữ	Hà Nội
4516802	Hoàng Tùng	21/03/84	Nam	Bắc Ninh
4620503	Trương Định	15/05/85	Nam	Hà Nam
4612203	Phạm An	16/04/85	Nam	Nam Định
4521402	Đỗ Cung	20/01/84	Nam	Nghệ An

+ Thuộc tính {Mã số} là một khoá của SINHVIEN bởi vì không có 2 bộ sinh viên có cùng một giá trị cho Mã số.

+ Mọi tập hợp thuộc tính có chứa Mã số, ví dụ: {Mã số, Họ tên, Ngày sinh}, đều là một siêu khoá. Tuy nhiên, siêu khoá {Mã số, Họ tên, Ngày sinh} không phải là khoá bởi vì nếu bỏ đi thuộc tính Họ tên hoặc Ngày sinh hoặc cả 2 thì nó vẫn còn là một siêu khoá.

Giá trị của một thuộc tính khoá có thể được sử dụng để xác định một cách duy nhất mỗi bộ trong một quan hệ. Ví dụ: Giá trị 4515202 của Mã số xác định một cách duy nhất bộ giá trị tương ứng với sinh viên Lê Văn trong quan hệ SINHVIEN.

Một khoá được xác định từ ý nghĩa của các thuộc tính và tính chất là bất biến, tính chất đó phải thỏa mãn khi chúng ta chèn thêm các bộ mới vào quan hệ. Ví dụ: Ta không thể chỉ định thuộc tính Họ tên của quan hệ SINHVIEN là khoá bởi vì có thể tồn tại 2 sinh viên có cùng Họ tên khi chúng ta chèn thêm thông tin của sinh viên mới vào.

Một lược đồ quan hệ R phải có ít nhất một khoá và có thể có nhiều khoá.

Các thuộc tính thuộc một khóa được gọi là thuộc tính khóa (prime attribute), các thuộc tính còn lại trong lược đồ quan hệ được gọi là các thuộc tính không khóa (nonprime attribute).

Quy ước:

- Trong một bộ bất kỳ của quan hệ, các thuộc tính khóa không chứa giá trị rỗng.
- Các thuộc tính tạo nên khóa của một lược đồ quan hệ được gạch dưới.

Ví dụ: Xét quan hệ Sv(MASV, HOSV, TENSU, NU, NGAYSINH, MALOP)

+ Thuộc tính khóa: MASV

+ Thuộc tính không khóa: HOSV, TENSU, NU, NGAYSINH, MALOP

Trong thực tế, đối với các loại thực thể tồn tại khách quan (Ví dụ: Sinh viên, Giảng viên, Nhân viên, Hàng hoá...) người thiết kế CSDL thường gán thêm cho các lược đồ quan hệ này một thuộc tính giả gọi là Mã số để làm khóa (Ví dụ: Mã số Sinh viên, Mã số Giảng viên, Mã số Nhân viên, Mã số Hàng hoá...). Trong khi đó, các lược đồ quan hệ biểu diễn cho sự trừu tượng hoá thường có khóa là một tổ hợp của 2 hay nhiều thuộc tính của nó.

Ví dụ: Xét lược đồ quan hệ: Kq(MASV, MAMH, DIEMTHI) quản lý điểm thi của sinh viên ứng (MASV) với từng môn học (MAMH) cụ thể. Một sinh viên có thể thi nhiều môn, và nhiều sinh viên có thể cùng thi chung một môn. Như vậy, ta chọn khóa là tập hợp 2 thuộc tính: {MASV, MAMH}.

II. KHÓA DỰ TUYỂN – KHÓA CHÍNH – KHÓA NGOẠI

1. Khóa dự tuyển (Candidate Key) – Khóa chính (Primary Key)

Một lược đồ quan hệ có thể có nhiều hơn một khóa. Trong trường hợp đó, mỗi một khóa được gọi là một khóa dự tuyển.

Một trong các khóa dự tuyển được chọn làm khóa tiêu biểu, khóa này được gọi là khóa chính. Ta phải chỉ định một trong các khóa dự tuyển làm khóa chính của quan hệ.

Một quan hệ chỉ có một khóa chính và có thể có nhiều khóa dự tuyển. Chú ý rằng khi một lược đồ quan hệ có nhiều khóa dự tuyển, việc lựa chọn một khóa dự tuyển để làm khóa chính là tùy ý, tuy nhiên tốt nhất là chọn khóa chính gồm một thuộc tính hoặc có số lượng các thuộc tính ít nhất.

Khóa chính được chọn để cài đặt trong một hệ quản trị CSDL.

Khi chọn khóa chính ta phải chú ý các tính chất sau:

- Khóa có tính áp dụng khi nó không bỏ sót bất kỳ trường hợp nào của vấn đề.
- Khóa phải có tính duy nhất dùng để phân biệt bộ này với bộ kia trong quan hệ.
- Khóa có tính nhỏ nhất: Khi ta bỏ qua bất kỳ thuộc tính nào của nó thì nó không còn tính duy nhất nữa.

- Khóa có tính ổn định khi giá trị của khóa không thay đổi.

Ví dụ: Xét quan hệ SINHVIEN sau:

SINHVIEN(MaSV, SoCMND, TenSV, Ngaysinh, Gioitinh, QueQuan)

Ta xét các tập thuộc tính sau:

K1={MaSV, TenSV} → Là siêu khóa

K2={Ngaysinh, QueQuan} → Không là khóa

K3={NgaySinh, TenSV, QueQuan} → Không là khóa

$K4 = \{MaSV\}$ → Là siêu khóa (khóa)

$K5 = \{MaSV, TenSV, Ngaysinh, Gioitinh, QueQuan\}$ → Là siêu khóa

$K6 = \{MaSV, Ngaysinh\}$ → Là siêu khóa

$K7 = \{SoCMND\}$ → Là siêu khóa (khóa)

$K4 = \{MaSV\}, K7 = \{SoCMND\}$ → Là các khóa dự tuyển

$K4 = \{MaSV\}$ hoặc $K7 = \{SoCMND\}$ → Là khóa chính (tùy theo sự lựa chọn trong hệ quản trị CSDL)

2. Khóa ngoại (Foreign Key)

Một thuộc tính được gọi là khóa ngoại nếu nó là thuộc tính của một lược đồ quan hệ này nhưng lại là khóa chính của lược đồ quan hệ khác.

Xét 2 lược đồ quan hệ R và S, gọi F là tập thuộc tính khóa ngoại của R tham chiếu đến khóa chính của S. Các thuộc tính khóa ngoại của R phải có cùng miền giá trị với thuộc tính khóa chính của S.

Quy ước: Các thuộc tính khóa ngoại được gạch dưới theo kiểu không liền nét.

Ví dụ: Xét lược đồ CSDL gồm các lược đồ quan hệ sau:

Sv(MASV, HOSV, TENSv, NU, NGAYSINH, MALOP)

Lop(MALOP, TENLOP, SISO, MAKHOA)

+ MASV là khóa chính của lược đồ quan hệ Sv.

+ MALOP là khóa chính của lược đồ quan hệ Lop.

+ MALOP là khóa ngoại của lược đồ quan hệ Sv.

Chú ý:

- Trong một lược đồ quan hệ, một thuộc tính có thể vừa tham gia vào khóa chính, vừa tham gia vào khóa ngoại.

- Khóa ngoại có thể tham chiếu đến khóa chính trong cùng một lược đồ quan hệ.

- Có thể có nhiều khóa ngoại tham chiếu đến cùng một khóa chính.

CHƯƠNG 3 – NGÔN NGỮ THAO TÁC DỮ LIỆU

(Tổng số: 5 tiết, Lý thuyết 5 tiết)

§ 1. ĐẠI SỐ QUAN HỆ

I. GIỚI THIỆU

1. Các khái niệm

Truy vấn CSDL: Trên một CSDL cho trước, người dùng đưa ra các câu hỏi yêu cầu lấy tin (query) cho hệ CSDL, hệ CSDL xử lý câu hỏi và gửi các câu trả lời cho người dùng.

Ngôn ngữ truy vấn: Là một ngôn ngữ được quy ước riêng, cho phép người dùng cập nhật và rút trích dữ liệu được lưu trong một CSDL.

Ngôn ngữ truy vấn đại số quan hệ (Relational Algebra) là cơ sở của các ngôn ngữ truy vấn được cài đặt trên các hệ CSDL (ví dụ như SQL); là ngôn ngữ mang tính thao tác, rất có ưu thế trong việc biểu diễn kế hoạch thi hành truy vấn.

Đại số quan hệ (ĐSQH) được trình bày xem như một phương pháp để mô hình hoá các phép toán trên CSDL quan hệ. Đồng thời đây cũng là một trong những ưu điểm của mô hình dữ liệu quan hệ, đó là các phép toán trên các quan hệ của một CSDL cho trước nhằm xây dựng ngôn ngữ khai thác, xử lý dữ liệu. Nhìn chung các phép toán của đại số quan hệ là khá đơn giản, nhưng nó khá mạnh và là một đại số có tính đầy đủ, phi thủ tục. Tuy nhiên đây là một cơ sở cho việc thiết lập các ngôn ngữ con dữ liệu bậc cao hơn.

Truy vấn được áp dụng cho các quan hệ, mọi phép toán quan hệ nhận đầu vào là một hay nhiều quan hệ và tạo ra kết quả nhận về là một quan hệ.

Một chuỗi các phép toán ĐSQH được gọi là một biểu thức ĐSQH mà kết quả cũng là một quan hệ.

2. Các phép toán đại số quan hệ

Các phép toán đại số quan hệ được chia thành 2 nhóm: Một nhóm bao gồm các phép toán được xây dựng đặc biệt cho các CSDL quan hệ, nhóm kia bao gồm các phép toán tập hợp lấy từ lý thuyết tập hợp toán học.

Các phép toán đại số quan hệ

- Phép chọn (Selection): σ
- Phép chiếu (Projection): Π

Các phép toán được hình thành từ lý thuyết tập hợp toán học

- Phép hợp (Union): $\cup$
- Phép giao (Intersection): $\cap$
- Phép trừ (Difference): $-$
- Phép kết (Join): $\bowtie$
- Phép tích Descartes (Cartesian product): $\times$
- Phép chia (Division/Quotient): $\div$

II. CÁC PHÉP TOÁN ĐẠI SỐ QUAN HỆ

1. Phép chọn (Selection)

- Mục đích:* Phép chọn là phép chọn lọc ra một tập con các bộ của quan hệ R đã cho, thỏa biểu thức điều kiện cho trước.

b. Định nghĩa:

Cho quan hệ $r(R)$ và điều kiện F , với F là một biểu thức luận lý có giá trị là đúng (TRUE) hoặc sai (FALSE).

Phép chọn trên r theo điều kiện F , ký hiệu là $\sigma_F(r)$, cho kết quả là một quan hệ gồm các bộ của r sao cho thỏa mãn điều kiện F là TRUE.

$$\sigma_F(r) = \{ u \mid u \in r \text{ và } F(u) = \text{TRUE} \}$$

Biểu thức F là điều kiện chọn, được định nghĩa là một tổ hợp các phép toán logic của các biểu thức so sánh, mỗi biểu thức so sánh là một phép toán so sánh giữa 2 toán hạng. Hai toán hạng được so sánh với nhau có thể là 2 thuộc tính hoặc giữa một thuộc tính và một giá trị hằng.

Ví dụ:

+ Điều kiện: $\text{DIEM_THI} < \text{DIEM_GIUA_HP}$ (thuộc tính điểm thi $\text{DIEM_THI} <$ thuộc tính điểm kiểm tra giữa học phần DIEM_GIUA_HP)

+ Điều kiện: $\text{DIEM_THI} \geq 5$ (thuộc tính điểm thi $\text{DIEM_THI} \geq 5$)

Các phép toán luận lý (logic) trong biểu thức F : NOT ($\neg$), AND ($\wedge$), OR ($\vee$)

Các phép toán so sánh trong biểu thức F : $=, \neq, <, \leq, >, \geq$

Biểu thức F cho giá trị đúng (TRUE) hoặc sai (FALSE) đối với mỗi bộ đã cho của quan hệ khi kiểm tra riêng bộ đó. $F(u)$ là điều kiện có được bằng cách thay thế các tên thuộc tính trong điều kiện F bởi các giá trị tương ứng trong bộ u .

Kết quả trả về là một quan hệ mới có cùng danh sách thuộc tính với r , và có số bộ luôn ít hơn hoặc bằng số bộ của r .

Phép chọn là phép toán một ngôi, nghĩa là nó được áp dụng cho một quan hệ.

Phép chọn có tính chất giao hoán:

$$\sigma_{F1}(\sigma_{F2}(r)) = \sigma_{F2}(\sigma_{F1}(r)) = \sigma_{F1 \wedge F2}(r)$$

Ta có thể kết hợp một loạt các phép chọn thành một phép chọn đơn giản bằng cách sử dụng phép toán AND ($\wedge$). Ví dụ:

$$\sigma_{F1}(\sigma_{F2}(r)) = \sigma_{F2 \text{ AND } F1}(r)$$

Ví dụ: Cho quan hệ SINHVIEN như sau:

MASV	HOTEN	MAMH	MAKHOA	DIEMTHI
99001	TRAN DAN THU	CSDL	CNTT	3.0
99002	NGUYEN HA DA THAO	CSDL	CNTT	8.0
99001	TRAN DAN THU	THVP	CNTT	6.0
99005	LE THANH TRUNG	THVP	AV	5.0

Yêu cầu: Tìm các sinh viên học môn CSDL và có điểm thi trên 5

$$\sigma_{(MAMH = \text{CSDL} \wedge \text{DIEMTHI} > 5)}(\text{SINHVIEN})$$

Kết quả:

MASV	HOTEN	MAMH	MAKHOA	DIEMTHI
99002	NGUYEN HA DA THAO	CSDL	CNTT	8.0

2. Phép chiếu (Projection)

a. Mục đích:

Phép chiếu trên một quan hệ thực chất là loại bỏ đi một số thuộc tính của quan hệ đó.

Nếu ta coi một quan hệ như một bảng thì phép chọn thực hiện chọn ra một số hàng của bảng thoả mãn điều kiện chọn và bỏ qua các hàng không thoả mãn điều kiện chọn. Phép chiếu là phép toán chọn ra một số cột của bảng. Nếu chúng ta chỉ quan tâm đến một số thuộc tính của quan hệ, chúng ta dùng phép chiếu để chiếu lên các thuộc tính đó.

b. Định nghĩa:

Cho lược đồ quan hệ $R(A_1, A_2, \dots, A_n)$ và một tập con các thuộc tính X , với $X \subset \{A_1, A_2, \dots, A_n\}$.

Gọi t là một bộ thuộc R , A là một thuộc tính, $t[A]$ là giá trị của bộ t tại thuộc tính A .

Giả sử $X = \{B_1, B_2, \dots, B_m\}$. Khi đó, $t[X]$ là giá trị của bộ t trên tập thuộc tính X , $t[X] = \{t[B_1], t[B_2], \dots, t[B_m]\}$

Phép chiếu $\pi_X(R)$ trên một tập thuộc tính X , ký hiệu là $\Pi_X(r)$ được định nghĩa như sau:

$$\Pi_X(r) = \{t[X] / t \in r\}$$

Phép chiếu của quan hệ R trên tập thuộc tính X là một tập các bộ, được xây dựng bằng cách loại bỏ đi từ các bộ t trong quan hệ R những thuộc tính không nằm trong X .

Kết quả trả về là một quan hệ có m thuộc tính nằm trong X và có cùng thứ tự như thứ tự của chúng có trong danh sách.

Số lượng các bộ kết quả luôn nhỏ hơn hoặc bằng số lượng các bộ trong R . Nếu X không bao gồm các thuộc tính khoá của R thì quan hệ kết quả có thể có những bộ trùng nhau. Phép chiếu loại bỏ mọi bộ trùng lặp, và như vậy, kết quả của phép chiếu là một tập hợp các bộ và là một quan hệ đúng đắn.

Nếu X_2 chứa tất cả các thuộc tính có trong X_1 thì:

$$\Pi_{X_1}(\Pi_{X_2}(r)) = \Pi_{X_1}(r)$$

Phép chiếu không có tính giao hoán.

Ví dụ: Cho quan hệ SINHVIEN như sau:

MASV	HOTEN	MAMH	MAKHOA	DIEMTHI
99001	TRAN DAN THU	CSDL	CNTT	3.0
99002	NGUYEN HA DA THAO	CSDL	CNTT	8.0
99001	TRAN DAN THU	THVP	CNTT	6.0
99005	LE THANH TRUNG	THVP	AV	5.0

Yêu cầu: Lập danh sách cho biết MASV, HOTEN, MAKHOA của các sinh viên thuộc khoa CNTT (Công nghệ Thông tin).

$$\sigma_{(MAKHOA = CNTT)}(\Pi_{MASV, HOTEN, MAKHOA}(SINHVIEN))$$

Kết quả:

MASV	HOTEN	MAKHOA
99001	TRAN DAN THU	CNTT
99002	NGUYEN HA DA THAO	CNTT

III. CÁC PHÉP TOÁN TẬP HỢP

Nhóm tiếp theo của các phép toán đại số quan hệ là các phép toán toán học thông thường trên các tập hợp. Các phép toán này là các phép toán hai ngôi, nghĩa là mỗi phép toán được áp dụng cho hai tập hợp. Khi áp dụng các phép toán này cho CSDL quan hệ, hai quan hệ tham gia vào một trong các phép toán trên phải có kiểu của các bộ như nhau, hay nói cách khác, chúng phải có cùng một cấu trúc. Điều kiện này được gọi là *khả hợp*.

Khái niệm khả hợp

Cho r_1 và r_2 là hai quan hệ, khi đó r_1 và r_2 được gọi là khả hợp khi và chỉ khi chúng được xác định trên cùng tập các thuộc tính và các thuộc tính có cùng miền giá trị.

Ví dụ: Cho $A = \{A_1, A_2, \dots, A_n\}$ là tập các thuộc tính của r_1 , và $B = \{B_1, B_2, \dots, B_m\}$ là tập các thuộc tính của r_2 , khi đó r_1 và r_2 khả hợp tương đương với $m = n$ và $\text{Dom}(A_1) = \text{Dom}(B_1), \text{Dom}(A_2) = \text{Dom}(B_2), \dots, \text{Dom}(A_n) = \text{Dom}(B_m)$.

Cho r, s là 2 quan hệ khả hợp, ta có các phép toán tập hợp:

1. Phép hợp (Union)

Cho hai quan hệ r và s trên cùng một lược đồ quan hệ R .

Phép hợp của hai quan hệ r và s , ký hiệu $r \cup s$, là một quan hệ gồm các bộ của r hay của s .

$$r \cup s = \{u \mid u \in r \text{ hay } u \in s\}$$

Phép hợp có tính giao hoán.

$$r \cup s = s \cup r$$

Ví dụ:

R	A	B	C
	1	2	3
	4	5	6
	1	2	7
	8	4	5

S	A	B	C
	1	2	3
	3	5	7
	6	2	9

$R \cup S$	A	B	C
	1	2	3
	4	5	6
	1	2	7
	8	4	5
	3	5	7
	6	2	9

2. Phép trừ (Difference)

Cho hai quan hệ r và s trên cùng một lược đồ quan hệ R .

Phép trừ (hiệu) của quan hệ r cho s , ký hiệu $r - s$, cho kết quả là một quan hệ gồm các bộ của r không có trong s .

$$r - s = \{u \mid u \in r \text{ và } u \notin s\}$$

Phép hiệu không có tính giao hoán.

$$r - s \neq s - r$$

Ví dụ:

R	A	B	C
	1	2	3
	4	5	6
	1	2	7
	8	4	5

S	A	B	C
	1	2	3
	3	5	7
	6	2	9

$R - S$	A	B	C
	4	5	6
	1	2	7
	8	4	5

3. Phép giao (Intersection)

Cho hai quan hệ r và s trên cùng một lược đồ quan hệ R .

Phép giao của hai quan hệ r và s , ký hiệu $r \cap s$, cho kết quả là một quan hệ gồm các bộ xuất hiện đồng thời trong cả r và s .

$$r \cap s = \{ u \mid u \in r \text{ và } u \in s \}$$

Phép giao của hai quan hệ r và s có thể được tính từ phép trừ:

$$r \cap s = r - (r - s) = s - (s - r)$$

Phép giao có tính giao hoán.

$$r \cap s = s \cap r$$

Ví dụ 1:

R	A	B	C
	1	2	3
	4	5	6
	1	2	7
	8	4	5

S	A	B	C
	1	2	3
	3	5	7
	6	2	9

$R \cap S$	A	B	C
	1	2	3

Ví dụ 2: Cho 2 quan hệ NHAHANG và KHACHSAN như sau:

NHAHANG	TEN	DIACHI	PHUONG	QUAN
	Huy Hoàng	504 Điện Biên Phủ	14	3
	Tương Lai	229 Ngô Chí Thanh	6	5
	Sinh Đôi	300 Lý Thái Tổ	6	10
	Đông Nam Á	230 Ngô Gia Tự	10	1

KHACHSAN	TEN	DIACHI	PHUONG	QUAN
	Viễn Đông	2 Lê Lợi	5	1
	Huy Hoàng	504 Điện Biên Phủ	14	3
	Ánh Hồng	151 Đặng Dung	2	PN
	Tương Lai	229 Ngô Chí Thanh	6	5

Tìm các nơi vừa là khách sạn, vừa là nhà hàng:

$$NH_KS = NHAHANG \cap KHACHSAN$$

Kết quả:

NH_KS	TEN	DIACHI	PHUONG	QUAN
	Huy Hoàng	504 Điện Biên Phủ	14	3
	Tương Lai	229 Ngô Chí Thanh	6	5

4. Phép tích Descartes (Cartesian product)

Tích Descartes (Đề-các) còn gọi là tích hỗn hợp (cross product) hoặc là nối hỗn hợp (cross join), được ký hiệu là $\times$. Đó cũng là một phép toán hai ngôi nhưng những quan hệ mà nó áp dụng trên đó không phải là tương thích đồng nhất. Phép toán này được sử dụng để nối các bộ của hai quan hệ vào một kiểu kết hợp.

Cho quan hệ r trên lược đồ quan hệ $R (A_1, A_2, \dots, A_m)$ và s trên lược đồ quan hệ $S (B_1, B_2, \dots, B_n)$.

Phép tích Descartes (Cartesian product) của hai quan hệ r và s , ký hiệu là $r \times s$, là một quan hệ trên lược đồ $T(A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_n)$ gồm các bộ u sao cho m thành phần đầu tiên là một bộ của r và n thành phần cuối cùng là một bộ của s .

$$r \times s = \{ (u_1, \dots, u_m, u_{m+1}, \dots, u_{m+n}) \mid (u_1, \dots, u_m) \in r \text{ và } (u_{m+1}, \dots, u_{m+n}) \in s \}$$

Như vậy, nếu R có m bộ và S có n bộ thì $R \times S$ có $m \cdot n$ bộ. Phép toán này nếu áp dụng một mình thì thông thường không có ý nghĩa. Nó chỉ có lợi khi kết hợp tiếp theo bằng một phép chọn các giá trị tương thích của các thuộc tính xuất phát từ các quan hệ thành phần. Tích Đề-các kết hợp với một phép chọn cho ta một phép nối (join).

Phép tích Descartes có tính giao hoán.

$$r \times s = s \times r$$

Ví dụ: Ta có hai quan hệ R và S như sau:

R	A	B	C
	1	2	3
	4	5	6
	1	2	7
	8	4	5

S	T	D
	1	5
	3	7

Kết quả phép tích Descartes: $R \times S$

$R \times S$	A	B	C	T	D
	1	2	3	1	5
	1	2	3	3	7
	4	5	6	1	5
	4	5	6	3	7
	1	2	7	1	5
	1	2	7	3	7
	8	4	5	1	5
	8	4	5	3	7

5. Phép kết (Join)

a. Phép kết θ (Theta)

Phép kết được dùng để kết hợp các bộ có liên hệ với nhau từ hai quan hệ thành một bộ. Phép toán này rất quan trọng đối với CSDL quan hệ có nhiều bảng bởi vì nó cho phép ta xử lý các mối liên kết giữa các quan hệ. Dạng tổng quát của phép kết trên hai quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$ là:

$$R \bowtie_{\langle \text{Điều kiện kết} \rangle} S$$

Kết quả của phép kết là một quan hệ $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ có $n+m$ thuộc tính. Mỗi bộ của Q là một sự kết nối giữa một bộ của R và một bộ của S khi chúng thỏa mãn điều kiện kết.

Sự khác nhau giữa tích Đề-các và phép kết là ở chỗ trong phép kết, chỉ có các bộ thỏa mãn điều kiện kết mới xuất hiện trong kết quả, trong khi đó trong tích Đề-các, mọi tổ hợp của các bộ đều có trong kết quả. Điều kiện kết được chỉ ra trên các thuộc tính của hai quan hệ R và S và được tính toán cho mỗi tổ hợp các bộ. Mọi tổ hợp bộ mà điều kiện kết là đúng được chứa trong quan hệ kết quả Q như là một bộ đơn.

Một điều kiện kết tổng quát có dạng:

$$\langle \text{Điều kiện} \rangle \text{ AND } \langle \text{Điều kiện} \rangle \text{ AND } \dots \text{ AND } \langle \text{Điều kiện} \rangle$$

Trong đó, mỗi điều kiện có dạng $A_i \theta B_j$, A_i là một thuộc tính của R , B_j là một thuộc tính của S , A_i và B_j có cùng miền và θ là một trong các dấu phép toán so sánh $\{<, <=, =, >=, >, \neq\}$. Một phép toán kết với điều kiện tổng quát như vậy gọi là một phép kết θ (Tê-ta). Các bộ có các thuộc tính kết là null không xuất hiện trong kết quả.

Nói một cách ngắn gọn, phép kết θ là phép tích Đề-các có áp dụng điều kiện.

Ví dụ, giả sử ta có hai quan hệ R và S như sau:

R	<i>A</i>	<i>B</i>	<i>C</i>
	1	2	3
	4	5	6
	1	2	7
	8	4	5

T	<i>A</i>	<i>D</i>
	1	5
	3	7

Kết quả phép kết θ : $R \bowtie_{R.A > T.A} T$

<i>R.A</i>	<i>B</i>	<i>C</i>	<i>T.A</i>	<i>D</i>
4	5	6	1	5
4	5	6	3	7
8	4	5	1	5
8	4	5	3	7

b. *Phép kết trong (Inner Join/Equijoin)*

Phép kết trong là phép kết θ với điều kiện là bằng nhau ($=$) giữa khóa chính và khóa ngoại.

$$R \bowtie_{<R.Khóa\ chính = S.Khóa\ ngoại>} S$$

Ví dụ: Cho 2 quan hệ SINHVIEN và KHOA như sau:

SINHVIEN	<i>MASV</i>	<i>HOTEN</i>	<i>MAKHOA</i>
	99001	TRAN DAN THU	CNTT
	99002	NGUYEN HA DA THAO	CNTT
	99001	TRAN DAN THU	CNTT
	99005	LE THANH TRUNG	AV

KHOA	<i>MAKHOA</i>	<i>TENKHOA</i>
	CNTT	CONG NGHE THÔNG TIN
	AV	ANH VAN

Kết quả phép kết trong: $SINHVIEN \bowtie_{SINHVIEN.MAKHOA = KHOA.MAKHOA} KHOA$

MASV	HOTEN	MAKHOA	MAKHOA	TENKHOA
99001	TRAN DAN THU	CNTT	CNTT	CONG NGHE THÔNG TIN
99002	NGUYEN HA DA THAO	CNTT	CNTT	CONG NGHE THÔNG TIN
99001	TRAN DAN THU	CNTT	CNTT	CONG NGHE THÔNG TIN
99005	LE THANH TRUNG	AV	AV	ANH VAN

c. *Phép kết tự nhiên (Natural Join)*

Trong kết quả của phép kết trong (Inner Join), chúng ta thấy luôn luôn có một hoặc nhiều cặp thuộc tính có các giá trị như nhau trong mỗi bộ. Việc có các cặp thuộc tính có giá trị như

nhau là thừa, vì vậy người ta đề nghị một phép kết mới gọi là kết tự nhiên. Phép kết tự nhiên nhằm loại bỏ thuộc tính thứ hai (thuộc tính thừa) trong điều kiện nối bằng.

Phép kết tự nhiên hai quan hệ R và S, ký hiệu là $R \bowtie S$, cho kết quả là quan hệ chứa các bộ là sự phối hợp của bộ u trong quan hệ R với tất cả các bộ v trong quan hệ S sao cho các trị trên miền thuộc tính chung của hai bộ này là như nhau.

Phép kết tự nhiên không đòi hỏi hai tập thuộc tính R và S giao nhau khác rỗng.

Nếu $R \cap S = \emptyset$ thì $R \bowtie S$ là phép tích Descartes của R và S.

Phép kết tự nhiên của hai quan hệ R và S có thể được tính từ phép tích Đề-các, phép chọn và phép chiếu:

$$R \bowtie S = \Pi_T \sigma_F (R \times S)$$

Trong đó: + F là biểu thức $R.A_1 = S.A_1 \wedge R.A_2 = S.A_2 \wedge \dots \wedge R.A_m = S.A_m$.

+ $R \cap S = \{A_1, A_2, \dots, A_m\}$.

+ $T = R \cap S$

Phép kết tự nhiên có tính giao hoán: $R \bowtie S = S \bowtie R$

Ví dụ 1: Sử dụng phép kết tự nhiên hai quan hệ SINHVIEN và KHOA trong ví dụ trên:

$SINHVIEN \bowtie KHOA$

MASV	HOTEN	MAKHOA	TENKHOA
99001	TRAN DAN THU	CNTT	CONG NGHE THÔNG TIN
99002	NGUYEN HA DA THAO	CNTT	CONG NGHE THÔNG TIN
99001	TRAN DAN THU	CNTT	CONG NGHE THÔNG TIN
99005	LE THANH TRUNG	AV	ANH VAN

Ví dụ 2: Cho bảng Lịch thi và bảng Môn học, hãy tổng hợp để in ra bảng kết quả gồm Mã môn, Tên môn thi, Ngày, Phòng.

LICHTHI	MaM	Ngay	MONHOC	MaM	TenM
	01	12/07/1994		01	Phân tích hệ thống
	02	14/07/1994		02	Cơ sở dữ liệu
	03	16/07/1994		03	Lập trình C
	04	17/07/1994		04	Lập trình Web

Kết quả phép kết tự nhiên $MONHOC \bowtie LICHTHI$:

KQ	MaM	TenM	Ngay
	01	Phân tích hệ thống	12/07/1994
	02	Cơ sở dữ liệu	14/07/1994
	03	Lập trình C	16/07/1994
	04	Lập trình Web	17/07/1994

§ 2. NGÔN NGỮ SQL

I. GIỚI THIỆU

Mỗi hệ quản trị CSDL đều phải có ngôn ngữ giao tiếp giữa người sử dụng với CSDL. Những năm 1975 - 1976, IBM lần đầu tiên đưa ra hệ quản trị CSDL kiểu quan hệ mang tên SYSTEM-R với ngôn ngữ giao tiếp CSDL là SEQUEL (Structured English Query Language). Năm 1976, ngôn ngữ SEQUEL được cải tiến thành SEQUEL-2. Khoảng năm 1978 - 1979, SEQUEL-2 được cải tiến và đổi tên thành ngôn ngữ truy vấn có cấu trúc (SQL - Structured Query Language). Năm 1986 viện tiêu chuẩn quốc gia Mỹ (American National Standards Institute - ANSI) đã công nhận và chuẩn hoá ngôn ngữ SQL và sau đó tổ chức tiêu chuẩn thế giới (International Standards Organization - ISO) cũng đã công nhận ngôn ngữ này. Đó là chuẩn SQL-86. Cho tới nay SQL đã qua 3 lần chuẩn hoá (1989, 1992, 1996) để mở rộng các phép toán và tăng cường khả năng bảo mật và tính toàn vẹn dữ liệu.

Trong chương này chúng ta chỉ nghiên cứu về ngôn ngữ SQL.

II. CÁC LỆNH TRUY VẤN LỰA CHỌN DỮ LIỆU

1. Câu lệnh tổng quát

Ngôn ngữ truy vấn SQL có tập lệnh khá phong phú để thao tác trên CSDL. Chẳng hạn lệnh CREATE để tạo các bảng quan hệ, lệnh UPDATE để cập nhật dữ liệu, lệnh DELETE để xoá dữ liệu, lệnh INSERT để thêm dữ liệu...

Trong phần này, chúng ta tìm hiểu câu lệnh quan trọng nhất của SQL đó là câu lệnh truy vấn lựa chọn, tìm kiếm dữ liệu SELECT. Kết quả của lệnh SELECT là một quan hệ, quan hệ kết quả này có thể kết xuất ra màn hình, máy in, hoặc là trên các thiết bị lưu trữ thông tin khác. Mỗi câu lệnh SQL có thể được viết trên nhiều dòng và kết thúc lệnh bởi dấu chấm phẩy (;), tuy nhiên từ khoá, tên hàm, tên thuộc tính, tên bảng, tên đối tượng thì không được phép viết tách xuống hàng. Trong thực tế, từ khoá, tên thuộc tính, tên bảng, tên đối tượng được viết chữ in hoa hay chữ thường là như nhau.

Cú pháp tổng quát của câu lệnh SELECT như sau:

SELECT DISTINCT <Danh sách thuộc tính> <Biểu thức>...

FROM <Danh sách các quan hệ>

WHERE <Biểu thức>

GROUP BY <Danh sách thuộc tính gom nhóm>

HAVING <Điều kiện nhóm>

ORDER BY <Danh sách các thuộc tính> [ASC | DESC]

Trong đó:

<Biểu thức> là sự kết hợp một cách hợp lệ giữa các thuộc tính, các toán tử và các hàm.

Sau đây sẽ là các toán tử và hàm thông dụng nhất:

Các toán tử số học: ^ (lũy thừa), * (nhân), / (chia), MOD (phần dư), + (cộng), - (trừ)

Các toán tử luận lý: NOT (phủ định), AND (phép hội), OR (phép tuyển)

Các toán tử tập hợp: IN (danh sách các giá trị), LIKE, BETWEEN ... AND ..., NOT LIKE, UNION (phép hợp), INTERSECT (phép giao), MINUS (phép trừ)

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

Lưu ý: Trong một số phiên bản của SQL, phép toán <> có thể được viết dạng !=

Các hàm sử dụng trong truy vấn:

DATE(): Trả về ngày/tháng/năm của hệ thống.

TIME(): Trả về giờ:phút:giây của hệ thống.

DAY(Biểu thức ngày): Trả về giá trị ngày của biểu thức ngày/tháng/năm.

MONTH(Biểu thức ngày): Trả về giá trị tháng của biểu thức ngày/tháng/năm.

YEAR(Biểu thức ngày): Trả về giá trị năm của biểu thức ngày/tháng/năm.

LEN(Biểu thức chuỗi): Trả về chiều dài của chuỗi.

Sau đây ta sẽ lần lượt tìm hiểu kỹ hơn các mệnh đề của câu lệnh.

Ví dụ lược đồ CSDL sau minh họa cho các truy vấn dữ liệu:

Sinhvien(MASV, HOTENSV, NU, NGAYSINH, NOISINH, TINH, MALOP)

Lop(MALOP, TENLOP, MAKHOA)

Khoa(MAKHOA, TENKHOA)

Monhoc(MAMH, TENMH, DONVIHT)

Giangvien(MAGV, HOTENGV, HOCVI, CHUYENNGANH, MAKHOA)

Ketqua(MASV, MAMH, LANTHI, DIEMTHI)

Phancong(MALOP, MAMH, MAGV)

2. Tìm thông tin từ các cột của bảng - Mệnh đề SELECT

SELECT DISTINCT * <Danh sách thuộc tính> <Biểu thức>...

FROM <Danh sách các quan hệ>

Những thuộc tính được liệt kê trong mệnh đề SELECT sẽ là các thuộc tính có trong quan hệ đích.

Ký hiệu * theo sau từ khóa SELECT dùng để chỉ tất cả các thuộc tính của quan hệ nguồn sẽ là thuộc tính của quan hệ đích. Danh sách các thuộc tính cách nhau bởi dấu phẩy và thứ tự này cũng là thứ tự của các thuộc tính trong quan hệ đích.

Mệnh đề FROM:

Những quan hệ liên quan đến câu truy vấn được liệt kê sau mệnh đề FROM, các quan hệ này cách nhau bởi dấu phẩy, thứ tự của các quan hệ được chỉ ra ở đây là không quan trọng.

Cần chú ý rằng khi mệnh đề FROM chỉ ra từ 2 quan hệ trở lên, nếu có một thuộc tính ở mệnh đề SELECT là thuộc tính của nhiều hơn một quan hệ thì cần phải chỉ rõ thuộc tính đó thuộc về quan hệ nào theo cú pháp <Tên quan hệ>.<Tên thuộc tính>. Đây là lỗi thường gặp khi thực hành với câu lệnh truy vấn SQL.

Ví dụ 1: Lập danh sách sinh viên gồm MASV, HOTENSV, NU, NGAYSINH, TINH

```
SELECT MASV, HOTENSV, NU, NGAYSINH, TINH
FROM Sinhvien;
```

Khi cần lấy thông tin về tất cả các cột của bảng, chúng ta có thể sử dụng dấu sao (*) thay cho việc liệt kê các tên cột của bảng. Nếu áp dụng cách viết này thì câu lệnh trên tương đương với câu lệnh sau:

```
SELECT * FROM Sinhvien;
```

Cú pháp này chỉ được sử dụng khi câu truy vấn chỉ liên quan đến một quan hệ.

Ví dụ 2: Lập danh sách bao gồm các thông tin về giảng viên như mã số giảng viên, họ và tên giảng viên, học vị, chuyên ngành.

```
SELECT MAGV, HOTENGV, HOCVI, CHUYENNGANH
FROM Giangvien;
```

Câu lệnh tìm kiếm thông tin từ các cột của bảng ở trên là cài đặt của phép chiếu trên 4 thuộc tính MAGV, HOTENGV, HOCVI, CHUYENNGANH của quan hệ Giangvien.

Nếu chúng ta muốn đặt tên khác cho tên của các cột của bảng (còn gọi là bí danh - Alias), việc này được thực hiện bằng cách thêm từ khóa AS và theo sau là một tên mới. Nếu tên có chứa các ký tự đặc biệt và/hoặc khoảng trắng thì viết tên đó trong cặp dấu ngoặc vuông ([]).

Chẳng hạn Ví dụ 2 có thể viết lại là:

```
SELECT MAGV AS [Mã Giảng viên], HOTENGV AS [Họ tên], HOCVI AS
[Trình độ], CHUYENNGANH AS [Chuyên ngành]
FROM Giangvien;
```

Câu lệnh SELECT không chỉ thực hiện việc trích thông tin từ các cột đơn lẻ của bảng mà còn có thể thực hiện các tính toán theo công thức hay biểu thức bất kỳ dựa trên giá trị của các cột trên từng bản ghi của bảng.

Từ khóa DISTINCT nhằm loại bỏ bớt các bộ trùng nhau trong bảng kết quả của lệnh truy vấn (chỉ giữ lại một bộ đại diện cho các bộ giống nhau).

Ví dụ 3: Hãy cho biết danh sách các chuyên ngành của các giảng viên trong trường.

```
SELECT DISTINCT CHUYENNGANH
FROM Giangvien;
```

Kết quả của câu lệnh này là tất cả những chuyên ngành mà các giảng viên trong trường có thể đảm nhận, mỗi chuyên ngành chỉ xuất hiện một lần trong kết quả truy vấn được.

3. Chọn các dòng của bảng - Mệnh đề WHERE

SELECT DISTINCT * <Danh sách thuộc tính> <Biểu thức>...

FROM <Danh sách các quan hệ>

WHERE <Biểu thức điều kiện>

Trong đó <Biểu thức điều kiện> có giá trị là hoặc đúng (TRUE) hoặc sai (FALSE). Biểu thức điều kiện là sự cài đặt của phép chọn trong ngôn ngữ đại số quan hệ. Nếu điều kiện này chỉ liên quan đến một quan hệ thì gọi là điều kiện chọn, nếu điều kiện liên quan đến từ hai quan hệ trở lên thì gọi là điều kiện kết. Các điều kiện chọn và điều kiện kết có thể phối hợp với nhau bởi các toán tử logic (AND, OR, NOT) để tạo nên những biểu thức logic phức tạp hơn. Chú ý rằng thứ tự của các điều kiện ở đây là quan trọng. Nếu có thể thì nên thực hiện điều kiện chọn trước khi thực hiện điều kiện kết.

Sau đây là một số ví dụ:

Ví dụ 4: Lập danh sách thông tin những môn học có số đơn vị học trình từ 4 trở lên. Danh sách cần các thông tin MAMH, TENMH, DONVIHT.

```
SELECT MAMH, TENMH, DONVIHT
FROM Monhoc
WHERE DONVIHT >=4;
```

Ví dụ 5: Lập danh sách các sinh viên có mã lớp là CDTH2A, CDTH2B, CDTH2C.

```
SELECT * FROM Sinhvien
WHERE MALOP="CDTH2A" OR MALOP="CDTH2B" OR MALOP="CDTH2C";
```

Cũng có thể viết cách khác như sau:

```
SELECT * FROM Sinhvien
WHERE MALOP IN ("CDTH2A", "CDTH2B", "CDTH2C");
```

Ví dụ 6: Lập danh sách gồm Mã sinh viên, điểm thi của mã lớp CDTH2A có điểm thi lần 1 mã môn CSDL từ 6 đến 8

```
SELECT MASV, DIEMTHI FROM Ketqua
WHERE LANTHI=1 AND DIEMTHI>=6 AND DIEMTHI<=8 AND MAMH="CSDL";
```

Toán tử so sánh tương đối: LIKE

Mẫu so sánh trong phép toán LIKE là một giá trị kiểu text, đó là một dãy ký tự bất kỳ, trong đó có hai ký tự có ý nghĩa đặc biệt sau đây:

? đại diện cho một ký tự bất kỳ tại vị trí có dấu chấm hỏi.

* đại diện cho một nhóm ký tự bất kỳ tại vị trí đó.

Ví dụ 7: Lập danh sách sinh viên có họ Nguyễn đang học tại lớp có mã lớp là CDTH2A

```
SELECT * FROM Sinhvien
WHERE MALOP="CDTH2A" AND HOTENSV LIKE "Nguyễn*";
```

Lưu ý: Một số hệ quản trị CSDL cho phép sử dụng dấu nháy đơn (') hoặc nháy kép (") bao quanh các giá trị ở dạng chuỗi văn bản.

Toán tử BETWEEN ... AND

Toán tử BETWEEN ... AND ... lấy ra một miền dữ liệu nằm giữa 2 giá trị kiểu số, chuỗi văn bản hoặc ngày tháng.

Ví dụ 6 ở trên có thể viết cách khác:

```
SELECT MASV, DIEMTHI
FROM Ketqua
WHERE LANTHI=1 AND DIEMTHI BETWEEN 6 AND 8 AND MAMH="CSDL"
```

4. Kết nối nhiều bảng dữ liệu

Để truy vấn thông tin từ nhiều bảng dữ liệu, ta sử dụng phép kết INNER JOIN giữa các bảng trong mệnh đề FROM.

```
SELECT <Thuộc tính>
FROM <Danh sách các quan hệ>
WHERE <Điều kiện kết>;
```

Hoặc có thể viết cách khác:

```
SELECT <Thuộc tính>
FROM <Quan hệ 1> INNER JOIN <Quan hệ 2> ON <Điều kiện kết>;
```

Ví dụ kết nối 2 bảng Sinhvien và Lop:

```
SELECT *
FROM Sinhvien, Lop
WHERE Sinhvien.MALOP=Lop.MALOP;
```

Hoặc:

```
SELECT *
FROM Sinhvien INNER JOIN Lop ON Sinhvien.MALOP=Lop.MALOP;
```

Ví dụ 7: Lập danh sách các sinh viên của mã lớp *CDTH2A* có điểm thi từ 8.0 trở lên của môn học có mã môn học "869"

```
SELECT Sinhvien.MASV, HOTENSV, NU, NGAYSINH, DIEMTHI
FROM Sinhvien, Ketqua
WHERE Sinhvien.MASV=Ketqua.MASV AND MALOP="CDTH2A"
AND MAMH="869" AND DIEMTHI>=8.0;
```

Chú ý: Do thuộc tính MASV xuất hiện ở cả 2 quan hệ Sinhvien và Ketqua, nên khi liệt kê nó ở mệnh đề SELECT cần chỉ rõ ra nó thuộc quan hệ nào.

5. Sắp xếp các dòng của bảng - Mệnh đề ORDER BY

Bảng kết quả truy vấn có thể được sắp xếp tăng/giảm theo một (hoặc nhiều) thuộc tính nào đó bằng cách sử dụng mệnh đề ORDER BY (độ ưu tiên giảm dần từ trái sang phải), từ khóa DESC (Descending) được dùng nếu muốn sắp xếp giảm dần, nếu không có DESC, mặc định dữ liệu sẽ được sắp xếp tăng dần ASC (Ascending) theo các thuộc tính đã chỉ ra.

Ví dụ 8: Lập danh sách tất cả thông tin các lớp. Danh sách được sắp xếp theo MAKHOA.

```
SELECT MALOP, TENLOP, MAKHOA
FROM Lop
ORDER BY MAKHOA
```

6. Các lệnh truy vấn lồng nhau

Là những câu lệnh mà trong thành phần WHERE có chứa thêm một câu lệnh SELECT khác nữa. Câu lệnh này thường gặp khi dữ liệu cần thiết phải duyệt qua nhiều lần. Đây là một trong những vấn đề khó khăn nhất khi truy vấn dữ liệu.

Ví dụ 10: Lập danh sách những sinh viên của mã lớp *CDTH2A* có điểm thi lần 1 cao nhất môn học có mã "CSDL".

Với câu lệnh này, nếu dùng các ngôn ngữ lập trình không có ngôn ngữ hỏi cấu trúc sẽ rất dài dòng. Đầu tiên phải tìm cho ra số điểm lớn nhất thỏa mãn điều kiện trên, sau đó phải duyệt dữ liệu thêm một lần nữa để chọn ra những bộ thỏa đề bài.

```
SELECT Sinhvien.MASV, HOTENSV, NU, NGAYSINH, DIEMTHI
FROM Sinhvien, Ketqua
WHERE MAMH='CSDL' AND LANTHI=1 AND
Sinhvien.MASV=Ketqua.MASV AND
DIEMTHI>=ALL
(SELECT DIEMTHI
FROM Ketqua, Sinhvien
WHERE Sinhvien.MASV=Ketqua.MASV
AND MAMH='CSDL' AND LANTHI=1);
```

Ví dụ 11: Lập danh sách những giảng viên cùng khoa với giảng viên có họ tên "Nguyễn Văn Thanh"

```
SELECT * FROM Giangvien
WHERE Makhoa IN
(SELECT MAKHOA FROM Giangvien
WHERE HOTENGV="Nguyễn Văn Thanh");
```

Kết quả của câu hỏi con được sử dụng trong phép so sánh với một giá trị khác trong biểu thức điều kiện của câu hỏi bao nó. Các phép so sánh có dạng:

<Phép so sánh> <Lượng từ> (SELECT Câu hỏi con)

Trong đó phép so sánh có thể là phép so sánh số học hoặc phép so sánh trên tập hợp.

<Lượng từ> có thể là IN, NOT IN hoặc <>ALL.

7. Gom nhóm dữ liệu - Mệnh đề GROUP BY

Khi cần tính toán trên các bộ theo một nhóm, một thuộc tính nào đó, ta dùng mệnh đề GROUP BY, chẳng hạn cần tính điểm trung bình chung tất cả các môn học cho tất cả các sinh viên, hay là cần tính số lượng sinh viên cho mỗi lớp, mỗi khoa, đếm số lượng sinh viên nữ của mỗi khoa, đếm số lượng sinh viên của mỗi tỉnh...

Mệnh đề GROUP BY <Thuộc tính gom nhóm> dùng để phân nhóm dữ liệu. Những bộ của bảng có cùng giá trị trên các thuộc tính này sẽ tạo thành một nhóm.

Lưu ý những thuộc tính có tham gia vào mệnh đề GROUP BY để phân nhóm phải được liệt kê trong danh sách thuộc tính theo sau từ khóa SELECT.

Ví dụ 12: Lập bảng điểm trung bình lần 1 các môn học của các sinh viên thuộc mã lớp CDTH2A. Danh sách cần: MASV, HOTENSV, DIEMTB. DIEMTB là thuộc tính tự đặt.

```
SELECT Ketqua.MASV, HOTENSV, AVG(DIEMTHI) AS DIEMTB
FROM Sinhvien, Ketqua
WHERE Sinhvien.MASV=Ketqua.MASV
      AND MALOP="CDTH2A" AND LANTHI=1
GROUP BY Ketqua.MASV, HOTENSV
```

Mệnh đề HAVING <Điều kiện trên nhóm>

Nếu cần kiểm tra điều kiện của một nhóm thì dùng mệnh đề HAVING, chẳng hạn như cho biết những sinh viên nào có điểm trung bình các môn ≥ 8 , những khoa nào có nhiều hơn 100 sinh viên nữ... Mệnh đề HAVING <điều kiện trên nhóm> được sử dụng như là phép chọn phối hợp với việc phân nhóm dữ liệu.

Ví dụ 13: Lập bảng điểm trung bình lần 1 lớn hơn hoặc bằng 8.0 các môn đã thi của các sinh viên có mã lớp CDTH2A. Danh sách cần: MASV, HOTENSV, DIEMTB. Trong đó DIEMTB là thuộc tính tự đặt. Giống như ở Ví dụ 12 nhưng ta có thêm điều kiện là điểm trung bình các môn đã thi lớn hơn hoặc bằng 8.0.

```
SELECT KETQUA.MASV, HOTENSV, AVG(DIEMTHI) AS DIEMTB
FROM SINHVIEN, KETQUA, LOP
WHERE MALOP="CDTH2A" AND LANTHI=1
      AND SINHVIEN.MASV=KETQUA.MASV
GROUP BY KETQUA.MASV, HOTENSV
HAVING AVG(DIEMTHI) >=8.0;
```

8. Hàm tính toán theo nhóm

Các hàm tính toán theo nhóm (đầu vào là một tập giá trị và trả về một giá trị đơn):

SUM(Thuộc tính): tính tổng giá trị của các bộ theo thuộc tính đã chỉ ra.

MAX(Thuộc tính): cho biết giá trị lớn nhất của các bộ theo thuộc tính đã chỉ ra.

MIN(Thuộc tính): cho biết giá trị nhỏ nhất của các bộ theo thuộc tính đã chỉ ra.

AVG(Thuộc tính): cho biết giá trị trung bình của các bộ theo thuộc tính đã chỉ ra.

COUNT *: Đếm tất cả các bộ.

COUNT(Thuộc tính): chỉ đếm những bộ mà giá trị của thuộc tính là khác nhau.

Chú ý: Cách sử dụng các toán tử và các hàm này còn tùy thuộc vào câu lệnh SELECT được sử dụng.

Ví dụ: Cho biết số lượng sinh viên trong toàn trường

```
SELECT COUNT(MASV)
FROM Sinhvien;
```

Ví dụ: Cho biết số lượng sinh viên theo từng mã lớp

```
SELECT    MALOP, COUNT(MASV)
FROM      Sinhvien
GROUP BY  MALOP;
```

Thứ tự dịch một lệnh truy vấn tổng hợp là như sau:

FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

III. CÁC LỆNH TRUY VẤN CẬP NHẬT – XÓA DỮ LIỆU

1. Truy vấn cập nhật dữ liệu

Câu lệnh UPDATE trong SQL được sử dụng để cập nhật dữ liệu trong các bảng.

UPDATE <Tên bảng>

SET <Tên cột 1> = <Biểu thức 1>, ...<Tên cột k> = <Biểu thức k>

WHERE <Điều kiện>;

Sau UPDATE là tên của bảng cần cập nhật dữ liệu. Một câu lệnh UPDATE có thể cập nhật dữ liệu cho nhiều cột bằng cách chỉ định danh sách tên cột và biểu thức tương ứng sau từ khoá SET. Mệnh đề WHERE trong câu lệnh UPDATE được sử dụng để chỉ định các dòng dữ liệu chịu tác động của câu lệnh. Nếu không chỉ định, phạm vi tác động của câu lệnh là toàn bộ các dòng trong bảng.

Ví dụ: Cập nhật lại số đơn vị học trình thành 4 cho môn có tên “Nhập môn máy tính”.

```
UPDATE Monhoc
SET      DONVIHT = 4
WHERE    TENMH = 'Nhập môn máy tính';
```

2. Truy vấn xóa dữ liệu

Để xóa dữ liệu trong một bảng, ta sử dụng câu lệnh DELETE. Cú pháp như sau:

DELETE FROM <Tên bảng>

FROM <Danh sách bảng>

WHERE <Điều kiện>;

Trong đó, tên của bảng cần xóa dữ liệu được chỉ định sau DELETE FROM.

Mệnh đề WHERE trong câu lệnh được sử dụng để chỉ định điều kiện đối với các dòng dữ liệu cần xóa. Nếu câu lệnh DELETE không có mệnh đề WHERE thì toàn bộ các dòng trong bảng đều bị xóa.

Mệnh đề FROM chỉ định danh sách các bảng có dữ liệu liên quan đến việc xóa dữ liệu.

Ví dụ: Xóa sinh viên có mã A01 ra khỏi bảng Sinhvien

```
DELETE FROM Sinhvien
WHERE MASV = 'A01';
```

Ví dụ: Xóa các sinh viên thuộc lớp có tên “Tin học 4” ra khỏi bảng Sinhvien

```
DELETE FROM Sinhvien
FROM Lop
WHERE Sinhvien.MALOP = Lop.MALOP AND TENLOP = 'Tin học 4';
```

BÀI TẬP NGÔN NGỮ SQL

1/- Cho lược đồ CSDL quản lý phân công nhân viên:

Congtrinh(MACT, TENCT, DIADIEM, NGAYCAPGP, NGAYKC, NGAYHT)

Nhanvien(MANV, HOTEN, NGAYSINH, PHAI, DIACHI, MAPB)

Phongban(MAPB, TENPB)

Phancong(MACT, MANV, SLNGAYCONG)

Hãy thực hiện các câu hỏi sau bằng SQL:

a/- Danh sách những nhân viên có tham gia vào công trình có mã công trình(MACT) là X. Yêu cầu các thông tin: MANV, HOTEN, SLNGAYCONG, trong đó MANV được sắp tăng dần.

b/- Đếm số lượng ngày công của mỗi công trình. Yêu cầu các thông tin: MACT, TENCT, TONGNGAYCONG (TONGNGAYCONG là thuộc tính tự đặt).

c/- Danh sách những nhân viên có sinh nhật trong tháng 08. yêu cầu các thông tin: MANV, TENNV, NGAYSINH, DIACHI, TENPB, sắp xếp quan hệ kết quả theo thứ tự tuổi giảm dần.

d/- Đếm số lượng nhân viên của mỗi phòng ban. Yêu cầu các thông tin: MAPB, TENPB, SOLUONG. (SOLUONG là thuộc tính tự đặt).

2/- Cho lược đồ CSDL quản lý phân công giảng dạy:

Giaovien(MAGV, HOTEN, MAKHOA)

Monhoc(MAMH, TENMH)

Phonghoc(PHONG, CHUCNANG)

Khoa(MAKHOA, TENKHOA)

Lop(MALOP, TENLOP, MAKHOA)

Lichday(MAGV, MAMH, PHONG, MALOP, NGAYDAY, TUTIET, ĐENTIET, BAIDAY, LYTHUYET, GHICHU)

Hãy thực hiện các câu hỏi sau bằng SQL:

a/- Xem lịch báo giảng tuần từ ngày 08/09/2003 đến ngày 14/09/2003 của giáo viên có MAGV (mã giáo viên) là TH3A040. Yêu cầu: MAGV, HOTEN, TENLOP, TENMH, PHONG, NGAYDAY, TUTIET, ĐENTIET, BAIDAY, GHICHU)

b/- Xem lịch báo giảng ngày 08/09/2003 của các giáo viên có mã khoa là CNTT. Yêu cầu: MAGV, HOTEN, TENLOP, TENMH, PHONG, NGAYDAY, TUTIET, ĐENTIET, BAIDAY, GHICHU)

c/- Cho biết số lượng giáo viên (SOLUONGGV) của mỗi khoa, kết quả cần sắp xếp tăng dần theo cột tên khoa. yêu cầu: TENKHOA, SOLUONGGV. SOLUONGGV là thuộc tính tự đặt.

3/- Cho lược đồ CSDL quản lý các kỳ thi nghề như sau:

THISINH(MASV, HOTEN, NGAYSINH, MALOP)

LOP(MALOP, TENLOP, MAKHOA)

KHOA(MAKHOA, TENKHOA, DIENTHOAI)

MONTHI(MAMT, TENMONTHI)

KETQUA(MASV, MAMT, DIEMTHI)

Phân giải thích các thuộc tính: HOTEN (họ tên thí sinh), NGAYSINH (ngày sinh), MALOP (mã lớp), MASV (mã sinh viên), TENLOP(tên lớp), MAKHOA (mã khoa), TENKHOA (tên khoa), DIENTHOAI (số điện thoại khoa), MAMT (mã môn thi), TENMONTHI (tên môn thi), DIEMTHI (điểm thi).

Dựa vào lược đồ CSDL trên, hãy thực hiện các yêu cầu sau bằng ngôn ngữ SQL:

a/- Hãy cho biết số lượng thí sinh của mỗi khoa đăng ký thi giỏi nghề, cần sắp xếp kết quả theo chiều tăng dần của cột TENKHOA.

b/- Lập danh sách những thí sinh đạt danh hiệu giỏi nghề (Thí sinh đạt danh hiệu giỏi nghề nếu thí sinh không có môn thi nào điểm dưới 8).

c/- Lập danh sách những thí sinh nhỏ tuổi nhất có mã khoa là "CNTT" dự thi giỏi nghề.

4/- Cho lược đồ CSDL quản lý nhân viên của một công ty như sau:

Nhanvien(MANV, HOTEN, NU, NGAYSINH, LUONG, MAPB, MACV)

Mỗi nhân viên có một mã nhân viên (MANV) duy nhất, mỗi mã nhân viên xác định họ và tên nhân viên (HOTEN), giới tính (NU), lương (LUONG), mã phòng ban (MAPB), mã chức vụ (MACV).

Phongban(MAPB, TENPB, TRUSO, MANVPHUTRACH, KINHPhi, DOANHThu)

Mỗi phòng ban có tên gọi phòng ban (TENPB), địa điểm đặt trụ sở (TRUSO), mã nhân viên phụ trách (MANVPHUTRACH), kinh phí hoạt động (KINHPhi), và doanh thu (DOANHThu)

Chucvu(MACV, TENCV, LUONGTHAPNHAT, LUONGCAONHAT)

Mỗi chức vụ có tên gọi chức vụ (TENCV), mức lương tối thiểu (LUONGTHAPNHAT), mức lương tối đa (LUONGCAONHAT).

Hãy biểu diễn các câu hỏi sau bằng SQL:

- Lập danh sách gồm các thông tin về các phòng ban trong công ty như: mã số phòng ban, tên phòng ban, địa điểm trụ sở, mã số người phụ trách, kinh phí hoạt động, doanh thu.
- Lập danh sách những nhân viên sinh nhật trong tháng 10.
- Lập danh sách gồm các thông tin mã số nhân viên, họ và tên và lương cả năm của các nhân viên (giả sử rằng lương cả năm = 12 * Lương)
- Lập những phòng ban có kinh phí hoạt động cao nhất.
- Lập danh sách nhân viên của phòng ban có mã số phòng ban là 40.
- Lập danh sách nhân viên của phòng có mã số phòng ban 10,30,50.
- Lập danh sách các nhân viên có lương tháng từ 2.500.000 đến 4.000.000
- Lập danh sách các nhân viên của phòng 10,30,50. Kết quả in ra theo thứ tự tăng dần của mã phòng nếu trùng mã phòng thì sắp xếp giảm dần theo mức lương.
- Lập danh sách các nhân viên phòng 10,30,50 chỉ in ra những người là lãnh đạo của mỗi phòng ban này.

- j. Lập danh sách gồm mã phòng mà người có mức lương cao nhất của phòng lớn hơn hoặc bằng 4.000.000
- k. Lập mã phòng ban, tên phòng ban, họ và tên của lãnh đạo phòng tương ứng.
- l. Lập danh sách những người làm việc cùng phòng với ông Nguyễn Văn Thanh.
- m. Lập biết mã số nhân viên, họ và tên, mức lương của người lãnh đạo ông Nguyễn Văn Thanh.
- n. Lập danh sách nhân viên có mức lương lớn hơn hay bằng mức lương cao nhất của phòng ông Nguyễn Văn Thanh.
- o. Cho biết mã số nhân viên, họ và tên , tổng số nhân viên, mức lương cao nhất, mức lương thấp nhất, mức lương trung bình của từng phòng ban.
- p. Cho biết các nhân viên có mức lương cao nhất của các phòng ban.
- q. Cho biết số lượng nhân viên của mỗi phòng ban.

CHƯƠNG 4 – LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU

(Tổng số: 5 tiết, Lý thuyết: 5 tiết)

§ 1. PHỤ THUỘC HÀM**I. PHỤ THUỘC HÀM - HỆ LUẬT DẪN ARMSTRONG****1. Định nghĩa**

Cho r là một quan hệ được định nghĩa trên lược đồ quan hệ R , X và Y là hai tập con (khác rỗng) các thuộc tính của R . Ta nói X xác định hàm Y (hay Y phụ thuộc hàm vào X), ký hiệu: $X \rightarrow Y$ là một phụ thuộc hàm (PTH) định nghĩa trên R nếu:

$$\forall t_1, t_2 \in r(R): t_1(X) = t_2(X) \Rightarrow t_1(Y) = t_2(Y)$$

Nghĩa là không thể tồn tại hai bộ trong r giống nhau ở các thuộc tính trong tập X mà lại khác nhau ở một hay nhiều thuộc tính nào đó trong tập Y .

Ví dụ: $\text{SINHVIEN}(\text{MaSV}, \text{HotenSV}, \text{Namsinh})$

Có phụ thuộc hàm: $\text{MaSV} \rightarrow \text{HotenSV}$

Qui ước:

- Tập thuộc tính: (X, Y, Z)
- Một thuộc tính: A, B, C
- $XY = X \cup Y$
- $X \rightarrow Y$: X xác định hàm Y hay Y phụ thuộc hàm vào X .
- PTH là phương tiện biểu diễn những ràng buộc dữ liệu, đây là cơ sở để xác định khoá và chuẩn hóa lược đồ CSDL.

2. Cách xác định phụ thuộc hàm cho lược đồ quan hệ

Cách duy nhất để xác định đúng các phụ thuộc thích hợp cho một lược đồ quan hệ là xem xét nội dung tân từ của lược đồ quan hệ đó.

Chẳng hạn với lược đồ CSDL đã cho trong ví dụ trên:

$\text{Sinhvien}(\text{MASV}, \text{HOTENSV}, \text{NU}, \text{NGAYSINH}, \text{NOISINH}, \text{TINH}, \text{MALOP})$

$\text{Lop}(\text{MALOP}, \text{TENLOP}, \text{MAKHOA})$

$\text{Khoa}(\text{MAKHOA}, \text{TENKHOA})$

$\text{Monhoc}(\text{MAMH}, \text{TENMH}, \text{DONVIHT})$

$\text{Giangvien}(\text{MAGV}, \text{HOTENGV}, \text{HOCVI}, \text{CHUYENNGANH}, \text{MAKHOA})$

$\text{Ketqua}(\text{MASV}, \text{MAMH}, \text{LANTHI}, \text{DIEMTHI})$

$\text{Phancong}(\text{MALOP}, \text{MAMH}, \text{MAGV})$

Như vậy, phụ thuộc hàm ứng với từng lược đồ quan hệ được xác định như sau:

$\text{MASV} \rightarrow \text{HOTENSV}, \text{NU}, \text{NGAYSINH}, \text{MALOP}, \text{TINH}$

$\text{MALOP} \rightarrow \text{TENLOP}, \text{MAKHOA}$

$\text{MAKHOA} \rightarrow \text{TENKHOA}$

$\text{MAMH} \rightarrow \text{TENMH}, \text{DONVIHT}$

$\text{MASV}, \text{MAMH}, \text{LANTHI} \rightarrow \text{DIEMTHI}$

3. Một số tính chất của phụ thuộc hàm - Hệ luật dẫn Armstrong

Để có thể xác định được các phụ thuộc hàm khác từ tập phụ thuộc hàm đã có, ta dùng hệ tiên đề Armstrong (1974), gồm các luật sau:

Với $X, Y, Z, W \subseteq R^+$

a. Luật phản xạ

Nếu $X \subseteq R^+$ và $Y \subseteq X$ thì $X \rightarrow Y$

Quy tắc này đưa ra những phụ thuộc hàm hiển nhiên (phụ thuộc hàm tầm thường), đó là những phụ thuộc hàm mà vế trái bao hàm cả vế phải. Những phụ thuộc hàm hiển nhiên đều đúng trong mọi quan hệ.

b. Luật tăng trưởng

Nếu $X \rightarrow Y$ và $Z \subseteq R^+$ thì $XZ \rightarrow YZ$

c. Luật bắc cầu (transitivity)

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

d. Luật hệ quả (Các quy tắc suy rộng)

Luật hợp: Nếu $X \rightarrow Y$ và $X \rightarrow Z$ thì $X \rightarrow YZ$

Luật phân rã: Nếu $X \rightarrow YZ$ thì $X \rightarrow Y$ và $X \rightarrow Z$

Luật bắc cầu giả: Nếu $X \rightarrow Y$ và $WY \rightarrow Z$ thì $XW \rightarrow Z$

II. BAO ĐÓNG

1. Bao đóng của tập phụ thuộc hàm

Bao đóng của tập phụ thuộc hàm F (ký hiệu là F^+) là tập hợp tất cả các phụ thuộc hàm có thể suy ra từ F dựa vào các tiên đề Armstrong. Rõ ràng $F \subseteq F^+$

Ví dụ: Cho lược đồ quan hệ $R(ABCDEFGH)$ và F được cho như sau:

$F = \{B \rightarrow A; DA \rightarrow CE; D \rightarrow H; GH \rightarrow C; AC \rightarrow D\}$

Khi đó $F^+ = \{ B \rightarrow A; DA \rightarrow CE; D \rightarrow H; GH \rightarrow C; AC \rightarrow D;$

$BC \rightarrow AC; BC \rightarrow D; DA \rightarrow AH; DG \rightarrow C; BC \rightarrow AD; \dots \}$

Lưu ý: Nếu mỗi thuộc tính được biểu diễn bằng một ký tự thì danh sách các thuộc tính có hoặc không có dấu phẩy đều được, còn giữa các phụ thuộc hàm phải có dấu chấm phẩy).

Các tính chất của tập F^+ :

- Tính phản xạ: Với mọi tập phụ thuộc hàm F^+ ta luôn có $F \subseteq F^+$
- Tính đơn điệu: Nếu $F \subseteq G$ thì $F^+ \subseteq G^+$
- Tính lũy đẳng: Với mọi tập phụ thuộc hàm F ta luôn luôn có $F^{++} = F^+$

2. Bao đóng của tập thuộc tính

Cho lược đồ quan hệ R . Giả sử F là tập các phụ thuộc hàm trong R , $X \subseteq R^+$.

Bao đóng của tập thuộc tính X đối với F ký hiệu là X^+ (hoặc X_F^+) là tập tất cả các thuộc tính $A \in R^+$ được suy ra từ X dựa vào các phụ thuộc hàm trong F và hệ tiên đề Armstrong, nghĩa là:

$X^+ = \{A: A \in R^+ \text{ và } X \rightarrow A \in F^+\}$

Ví dụ: Cho lược đồ quan hệ $R(ABCDEFGH)$ và tập phụ thuộc hàm F

$$F = \{B \rightarrow A; DA \rightarrow CE; D \rightarrow H; GH \rightarrow C; AC \rightarrow D\}$$

Hãy tính: B^+ ; H^+ ; BC^+

Giải:

Tao có $B^+ = BA$; (do có phụ thuộc hàm $B \rightarrow A$)

$H^+ = H$; (do có phụ thuộc hàm $H \rightarrow H$)

$BC^+ = BCADHEH$; (do có các phụ thuộc hàm $B \rightarrow A$; $AC \rightarrow D$; $DA \rightarrow CE$; $D \rightarrow H$)

Tương tự như tập bao đóng của tập phụ thuộc hàm F^+ , tập bao đóng X^+ cũng chứa các phần tử của tập X , tức là $X \subseteq X^+$.

Các tính chất của bao đóng của tập thuộc tính X^+

Nếu X, Y là các tập con của tập thuộc tính Q thì ta có các tính chất sau đây:

- *Tính phản xạ:* $X \subseteq X^+$
- *Tính đơn điệu:* Nếu $X \subseteq Y$ thì $X^+ \subseteq Y^+$
- *Tính lũy đẳng:* $X^{++} = X^+$
- $(XY)^+ \supseteq X^+Y^+$
- $(X^+Y)^+ = (XY^+)^+ = (X^+Y^+)^+$
- $X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X^+$
- $X \rightarrow Y \Leftrightarrow Y^+ \subseteq X^+$

3. Phương pháp tìm bao đóng

Tính toán bao đóng của tập phụ thuộc hàm F^+ rất tốn kém thời gian vì F^+ có thể rất lớn ngay khi bản thân F nhỏ.

Xét tập hợp: $F = \{A \rightarrow B_1, A \rightarrow B_2, \dots, A \rightarrow B_n\}$

Khi đó F^+ bao gồm tất cả các phụ thuộc hàm dạng $A \rightarrow Y$, trong đó Y là tập con của:

$\{B_1, B_2, \dots, B_n\}$

Bởi vì có đến 2^n tập con như vậy nên danh sách F^+ sẽ rất lớn dù cho n có nhỏ đi nữa.

Nhưng việc tính bao đóng của tập thuộc tính X^+ không tốn kém lắm, thời gian tính toán tỉ lệ thuận với độ dài của tất cả phụ thuộc hàm trong F . Theo kết luận từ các định lý việc xác định $X \rightarrow Y \in F^+$ tương đương với việc tính bao đóng X^+ .

Ví dụ: Cho tập phụ thuộc hàm F :

$AB \rightarrow C \quad D \rightarrow EG \quad C \rightarrow A \quad BE \rightarrow C$

$BC \rightarrow D \quad CG \rightarrow BD \quad ACD \rightarrow B \quad CE \rightarrow AG$

Đặt $X = BD$. Áp dụng giải thuật trên để xác định bao đóng X^+ .

- Đặt $X^{(0)} = X = BD$, $F_0 = F$

- Tính $X^{(1)}$: Ta tìm các phụ thuộc hàm có phía bên trái là B , D , hoặc BD

Ta có: $D \rightarrow EG$, nên ta thêm E và G vào $X^{(0)}$ để có: $X^{(1)} = BDEG$

Tương tự ta có: $X^{(2)} = BCDEG$, $X^{(3)} = ABCDEG = U$. Kết thúc.

Vậy: $(BD)^+ = ABCDEG$

§ 2. PHÉP TÁCH CÁC LƯỢC ĐỒ QUAN HỆ

I. KHÁI NIỆM PHÉP TÁCH

1. Khái niệm

Phép tách một lược đồ quan hệ $R = \{A_1, \dots, A_n\}$ là việc thay thế lược đồ quan hệ R bằng tập lược đồ $\{R_1, \dots, R_k\}$, trong đó $R_1, \dots, R_k$ là các tập con của R , không bắt buộc phân biệt với nhau, thỏa: $R = R_1 \cup \dots \cup R_k$

Mục đích của phép tách là loại bỏ các dị thường dữ liệu, nhằm chuẩn hoá các lược đồ quan hệ.

2. Ví dụ

a. Ví dụ 1

Xét lược đồ quan hệ NHACC(TenNCC, DiaChi, MatHang, DonGia) với các phụ thuộc hàm:

$TenNCC \rightarrow DiaChi$

$TenNCC, MatHang \rightarrow DonGia$

Từ tập phụ thuộc hàm chúng ta thấy rằng có dư thừa dữ liệu trong quan hệ NHACC này. Giả sử một TenNCC cung cấp nhiều mặt hàng thì DiaChi sẽ bị lặp lại nhiều lần cho TenNCC đó. Tuy nhiên, nếu chúng ta thay NHACC bằng hai lược đồ:

$CTY(TenNCC, DiaChi)$

$HANG(TenNCC, MatHang, DonGia)$

Khi đó, có thể loại bỏ được thuộc tính DiaChi để dữ liệu không bị lặp lại nhiều lần, nhưng liệu có giải quyết được mọi vấn đề hay không?

Giả sử quan hệ r là giá trị hiện thời của lược đồ NHACC. Ta ký hiệu r_1 và r_2 là hai thể hiện ứng với lược đồ CTY và HANG. Ta trông đợi rằng r_1 sẽ là phép chiếu của r lên các thuộc tính TenNCC, DiaChi và r_2 sẽ là chiếu của r lên các thuộc tính TenNCC, MatHang, DonGia:

$$r_1 = \pi_{Ten, DiaChi}(r) \ \& \ r_2 = \pi_{Ten, MatHang, DonGia}(r)$$

Bằng cách nào ta biết được r_1 và r_2 chứa thông tin giống như r ? Một hướng trả lời là kiểm tra xem r có thể được phục hồi từ r_1 và r_2 hay không. Ta khẳng định rằng phương pháp duy nhất để phục hồi r là tái tạo nối tự nhiên của r_1 và r_2 .

Nếu ta đặt: $s = r_1 * r_2$

Nếu $r \neq s$ thì từ r_1 và r_2 chúng ta không có cách nào biết được r hay s là quan hệ gốc của lược đồ NHACC.

b. Ví dụ 2:

Cho lược đồ $R = (A, B, C)$ và tập phụ thuộc hàm $F = \{A \rightarrow B\}$. Xét phân rã R thành 2 lược đồ con (A, B) và (B, C) .

Cho quan hệ $r = \{a_1b_1c_1, a_2b_1c_2\}$ là thể hiện của lược đồ R .

r	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₁	c ₂

Khi đó:

$$\pi_{AB}(r) = \{a_1b_1, a_2b_1\}, \pi_{BC}(r) = \{b_1c_1, b_1c_2\}$$

Như vậy:

$$s = \pi_{AB}(r) * \pi_{BC}(r) = \{a_1b_1c_1, a_1b_1c_2, a_2b_1c_1, a_2b_1c_2\} \neq r.$$

Vậy phép tách này không phục hồi đúng thông tin ban đầu.

c. Ví dụ 3

Cho lược đồ: LOPHOC(Lop, Monhoc, Giaovien) với tập phụ thuộc hàm:

{Giaovien $\rightarrow$ Monhoc, Lop;

Monhoc $\rightarrow$ Giaovien}

Xét phép tách thành:

GV(Giaovien, Monhoc)

LOP(Lop, Giaovien)

Phép tách nào bảo tồn thông tin ban đầu:

$$\pi_{\text{Giaovien, Monhoc}}(\text{LOPHOC}) * \pi_{\text{Lop, Giaovien}}(\text{LOPHOC}) = \text{LOPHOC}$$

Tuy nhiên phụ thuộc hàm thứ 2 không còn tồn tại trong các lược đồ con, và như vậy nó cũng không được phục hồi trong phép nối tự nhiên trên.

Vấn đề đặt ra là chúng ta phải phân rã lược đồ sao cho các phép nối tự nhiên của các quan hệ phân rã tương ứng phải thỏa mãn 2 yêu cầu sau:

(1): Phục hồi được thông tin của quan hệ ban đầu.

(2): Phục hồi được các phụ thuộc hàm của các quan hệ ban đầu.

Phép tách thỏa mãn yêu cầu (1) gọi là phép tách bảo tồn thông tin, còn phép tách thỏa mãn yêu cầu (2) là phép tách bảo tồn phụ thuộc hàm. Chúng ta sẽ nghiên cứu các phép tách này trong phần sau:

II. PHÉP TÁCH KẾT NỐI KHÔNG MẤT THÔNG TIN

1. Định nghĩa

Nếu R là một sơ đồ quan hệ tách thành các sơ đồ $R_1, R_2, \dots, R_k$ và F là tập phụ thuộc hàm, ta nói phép tách có kết nối không mất thông tin nếu với mọi quan hệ r của R thỏa F:

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

2. Bổ đề:

Cho R là một sơ đồ quan hệ, $\rho = \{R_1, R_2, \dots, R_k\}$ là một phép tách R, r là một quan hệ trên R. Đặt: $m_\rho(r) = \pi_{R_1}(r) * \pi_{R_2}(r) * \dots * \pi_{R_k}(r)$

Ta có:

(i): $r \subseteq m_\rho(r)$

(ii): Nếu $s = m_\rho(r)$ thì $\pi_{R_i}(s) = \pi_{R_i}(r)$

(iii): $m_\rho(s) = m_\rho(r)$

3. Kiểm tra kết nối không mất thông tin

a. Thuật toán:

Input: Một sơ đồ quan hệ $R = (A_1, \dots, A_n)$, một tập phụ thuộc hàm F và một phép tách

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

Output: Một quyết định xem có là một phép tách với một kết nối không mất thông tin.

Phương pháp:

- Ta xây dựng một bảng k dòng và n cột, cột j tương ứng thuộc tính thứ j và dòng i tương ứng sơ đồ quan hệ R_i . Tại vị trí giao của dòng i và cột j ta đặt ký hiệu a_j nếu A_j có trong R_i , ký hiệu b_{ij} nếu ngược lại.

- Lặp lại việc xét từng phụ thuộc hàm $X \rightarrow Y$ trong F , cho đến khi không còn thay đổi được bảng. Mỗi lần xét $X \rightarrow Y$ ta tìm các dòng mang trị bằng nhau trên tập thuộc tính X . Nếu tìm được hai dòng như vậy, làm cho các ký hiệu trên hai dòng đó giống nhau tại các thuộc tính của Y (nếu một ký hiệu là a_j ta cho ký hiệu thứ hai cũng là a_j ; nếu chúng là b_{ij} và b_{lj} ta cho chúng cùng là b_{ij} hoặc cùng là b_{lj}).

- Nếu sau khi sửa bảng theo cách như trên, ta thấy có dòng nào đó có dạng $a_1, a_2, \dots, a_k$ thì kết nối là không mất thông tin, ngược lại là mất.

b. Ví dụ:

Ta hãy xét phép tách SAIP thành SA và SIP. Các phụ thuộc hàm gồm $S \rightarrow A$ và $SI \rightarrow P$, và bảng ban đầu là:

	S	A	I	P
SA	a1	a2	b13	b14
SIP	a1	b22	a3	a4

Xét $S \rightarrow A$, ta thấy hai dòng có giá trị giống nhau ở cột S, vì vậy giá trị ở cột A cần thay đổi thành a_2 .

	S	A	I	P
SA	a1	a2	b13	b14
SIP	a1	a2	a3	a4

Vì bảng có một dòng chứa toàn a_j nên phép tách này là phép tách không mất thông tin.

c. Định lý 1:

Thuật toán trên xác định đúng nếu phép tách là không mất thông tin.

d. Định lý 2:

* Phát biểu:

Nếu $\rho = (R_1, R_2)$ là một phép tách sơ đồ quan hệ R và F là tập phụ thuộc hàm, thì có một phép tách không mất thông tin thỏa F nếu và chỉ nếu:

$$(R_1 \cap R_2) \rightarrow (R_1 - R_2) \text{ hoặc } (R_1 \cap R_2) \rightarrow (R_2 - R_1)$$

Chú ý: Các PTH này không cần thuộc F , chỉ cần thuộc F^+ .

* Áp dụng:

Trong khi giải thuật trên có thể áp dụng cho một phép tách thành một số bất kỳ các sơ đồ quan hệ, định lý 2 cho một cách kiểm tra đơn giản hơn đối với phép tách R thành hai sơ đồ quan hệ.

* Ví dụ:

Giả thiết $R = ABC$ và $F = \{A \rightarrow B\}$.

Như vậy, phép tách R thành AB và AC là phép tách không mất thông tin vì

$$AB \cap AC = A$$

$$AB - AC = B$$

Và ta có: $A \rightarrow B$

III. PHÉP TÁCH BẢO TỒN PHỤ THUỘC HÀM

Cho lược đồ quan hệ R với tập phụ thuộc hàm F và phép tách $\rho = \{R_1, \dots, R_k\}$. Một tính chất quan trọng nữa của phép tách lược đồ quan hệ là tập phụ thuộc hàm F phải được suy ra từ các chiếu của F lên $R_1, \dots, R_k$.

Ta định nghĩa chiếu của F lên tập thuộc tính Q , ký hiệu $\pi_Q(F)$, như sau:

$$\pi_Q(F) = \{X \rightarrow Y \in F^+ \mid X \cup Y \subset Q\}$$

1. Định nghĩa

Ta nói phép tách $\rho = \{R_1, \dots, R_k\}$ của lược đồ R bảo tồn phụ thuộc hàm (đối với tập phụ thuộc hàm F) nếu hợp các chiếu của F lên $R_1, \dots, R_k$ suy diễn logic tất cả phụ thuộc hàm trong F .

$$F \subset (\pi_{R_1}(F) \cup \pi_{R_2}(F) \cup \dots \cup \pi_{R_k}(F))^+$$

Lý do để phép tách phải bảo tồn phụ thuộc hàm là các phụ thuộc hàm có thể coi là những ràng buộc toàn vẹn của lược đồ R . Nếu các phụ thuộc hàm chiếu lên các lược đồ con không suy diễn F thì sự phân rã có thể vi phạm các phụ thuộc hàm trong F , thậm chí ngay cả khi phân rã bảo tồn thông tin.

Ví dụ:

- Cho sơ đồ quan hệ $R(C, S, Z)$ và tập phụ thuộc hàm F :

$$CS \rightarrow Z$$

$$Z \rightarrow C$$

(1): Phép tách R thành SZ và CZ là phép tách không mất thông tin vì:

$$(SZ \cap CZ) \rightarrow (CZ - SZ) \Leftrightarrow Z \rightarrow C$$

(2): Tuy nhiên, phép chiếu của $F = \{CS \rightarrow Z, Z \rightarrow C\}$ trên SZ chỉ cho các phụ thuộc hàm tầm thường; Trong khi phép chiếu của F trên CZ cho phụ thuộc hàm $Z \rightarrow C$ và các PTH tầm thường và chúng không suy diễn ra được $CS \rightarrow Z$. Do đó phép tách này là không bảo tồn phụ thuộc hàm.

- Chúng ta xét thể hiện cụ thể cho phép tách trên:

R1	S	Z
	12 LTT	02138
	12 LTT	02139

R2	C	Z
	TP CT	02138
	TP CT	02139

S	C	S	Z
	TP CT	12 LTT	02138
	TP CT	12 LTT	02139

Ta thấy, S là kết nối tự nhiên giữa R_1 và R_2 . R_1 thỏa các PTH tầm thường, R_2 thỏa các PTH tầm thường và $Z \rightarrow C$, nhưng bảng S vi phạm PTH $CS \rightarrow Z$.

2. Giải thuật

Input: Lược đồ quan hệ $R = \{A_1, A_2, \dots, A_n\}$, tập phụ thuộc hàm F và phép tách:

$$\rho = \{R_1, \dots, R_k\}$$

Output: Kết luận về tính bảo toàn phụ thuộc hàm của phân rã ρ .

Phương pháp:

Ký hiệu: $G = \bigcup_{i=1}^k R_i(F)$

Ta không tính G mà chỉ kiểm tra xem nó có phủ F hay không. Cho $X \rightarrow Y \in F$, kiểm tra xem bao đóng của X theo G , ký hiệu X_G^+ , có chứa Y hay không.

Thuật toán tính X_G^+ mà không cần có G là xét tác dụng của quá trình tính bao đóng của X đối với các chiều của F lên R_i .

Xuất phát từ X , ta tính X_G^+ như sau:

(1): Đặt $X_0 = X$, $t = 1$;

(2): Tính X_t trên cơ sở X_{t-1} :

(i): Đặt $X_t = X_{t-1}$

(ii): Với mỗi lược đồ con $R_i \in F$, $R_i \not\subset X_t$, ta thực hiện phép toán:

$$X_t = X_t \cup ((X_t \cap R_i)^+ \cap R_i)$$

Trong quá trình tính toán nếu $X_t = R$ thì kết thúc, và kết luận $X_G^+ = R$. Ngược lại sang bước (3).

(3): Nếu $X_t = X_{t-1}$ thì kết thúc, và kết luận $X_G^+ = X_t$.

Ngược lại tăng t lên 1 đơn vị và quay về bước (2).

Nếu Y là tập con của X_G^+ có được từ việc thực hiện các bước trên, thì khi đó $X \rightarrow Y \in G^+$. Nếu mỗi $X \rightarrow Y \in F$ được chứng minh thuộc G^+ bằng cách đó thì phép tách bảo toàn phụ thuộc hàm, ngược lại là không.

3. Ví dụ

Xét lược đồ $R = (A, B, C, D)$ với phép tách $\{AB, BC, CD\}$

Trong đó: $AB = (A, B)$, $BC = (B, C)$, $CD = (C, D)$, và tập phụ thuộc hàm F :

$$\{A \rightarrow B; B \rightarrow C; C \rightarrow D; D \rightarrow A\}$$

Ta thấy trong F^+ mỗi thuộc tính xác định tất cả các thuộc tính khác. Ta có cảm giác rằng khi chiếu F lên AB , BC và CD thì ta bỏ mất phụ thuộc hàm $D \rightarrow A$, nhưng nhận định đó không đúng. Khi tính chiếu của F lên tập thuộc tính, ta thực sự chiếu F^+ lên các lược đồ quan hệ. Như vậy, khi chiếu F lên AB ta không những chỉ có phụ thuộc hàm $A \rightarrow B$, mà còn nhận được phụ thuộc hàm $B \rightarrow A$. Tương tự ta nhận được $C \rightarrow B \in \pi_{BC}(F)$ và $D \rightarrow C \in \pi_{CD}(F)$, và chúng suy diễn logic ra $D \rightarrow A$.

Như vậy, để kết luận phép tách bảo toàn phụ thuộc hàm, ta phải chỉ ra rằng $D \rightarrow A$ suy diễn từ tập $G = \pi_{CD}(F) \cup \pi_{BC}(F) \cup \pi_{AB}(F)$

Áp dụng thuật toán, đặt $X = D$ và tính X_G^+ theo giải thuật trên:

(1): Đặt $X_0 = X = \{D\}$, $t = 1$

(2): Tính X_1 :

$$\text{Đặt } X_1 = X_0 = \{D\}$$

Xét lược đồ con AB . Vì $AB \not\subset X_1$, ta thực hiện phép tính

$$X_1 = \{D\} \cup ((\{D\} \cap \{A, B\})^+ \cap \{A, B\}) = \{D\}$$

Xét lược đồ con BC . Vì $BC \not\subset X_1$, ta thực hiện phép tính

$$X_1 = \{D\} \cup ((\{D\} \cap \{B, C\})^+ \cap \{B, C\}) = \{D\}$$

Xét lược đồ con CD . Vì $CD \subset X_1$, ta thực hiện phép tính

$$X_1 = \{D\} \cup ((\{D\} \cap \{C,D\})^+ \cap \{C,D\}) = \\ = \{D\} \cup (\{A,B,C,D\} \cap \{C,D\}) = \{C,D\}$$

Đến đây ta có $X_1 = \{C,D\} \neq X_0 = \{D\}$. Vậy ta tính tiếp X_2 .

(2): Tính X_2 :

$$\text{Đặt } X_2 = X_1 = \{C,D\}$$

Xét lược đồ con AB. Vì $AB \not\subset X_2$, ta thực hiện phép tính

$$X_1 = \{C,D\} \cup ((\{C,D\} \cap \{A,B\})^+ \cap \{A,B\}) = \{C,D\}$$

Xét lược đồ con BC. Vì $BC \not\subset X_2$, ta thực hiện phép tính

$$X_1 = \{C,D\} \cup ((\{C,D\} \cap \{B,C\})^+ \cap \{B,C\}) = \{B,C,D\}$$

Ta bỏ qua lược đồ CD vì $CD \subset X_2$.

Đến đây ta có $X_2 = \{B,C,D\} \neq X_1 = \{C,D\}$. Vậy ta tính tiếp X_2 :

(3): Tính X_3 :

$$\text{Đặt } X_3 = X_2 = \{B,C,D\}$$

Xét lược đồ con AB. Vì $AB \not\subset X_2$, ta thực hiện phép tính

$$X_1 = \{B,C,D\} \cup ((\{B,C,D\} \cap \{A,B\})^+ \cap \{A,B\}) \\ = \{B,C,D\} \cup (\{A,B,C,D\} \cap \{A,B\}) = \{A,B,C,D\}$$

Ta có $X_3 = R$. Quá trình kết thúc và ta có:

$$X_Q^+ = (D)^+ = R$$

Từ đó suy ra $D \rightarrow A \in G^+$. Các phụ thuộc hàm khác hiển nhiên là thuộc G^+ (thậm chí chúng thuộc G). Ta kết luận phép tách bảo toàn phụ thuộc hàm F .

§ 3. CHUẨN HOÁ LƯỢC ĐỒ QUAN HỆ

I. CÁC DẠNG CHUẨN

1. Dạng chuẩn thứ nhất (1NF)

a. Định nghĩa:

Một sơ đồ quan hệ R được xem là ở dạng chuẩn thứ nhất (1NF - 1 Normal Form) nếu mọi thuộc tính của R đều khác trống, phụ thuộc hàm vào khoá và không thể chia được thành các thành phần nhỏ hơn.

b. Ví dụ

Cho sơ đồ quan hệ:

CUNG_UNG(MaNSX, MaH, SL, VonNSX, TP, Nuoc)

Với các phụ thuộc hàm:

$$\text{MaNSX, MaH} \rightarrow \text{SL} \quad (a)$$

$$\text{MaNSX} \rightarrow \text{VonNSX} \quad (b)$$

$$\text{MaNSX} \rightarrow \text{TP} \quad (c)$$

$$\text{MaNSX} \rightarrow \text{Nuoc} \quad (d)$$

$TP \rightarrow Nuoc$ (e)

Sơ đồ này thoả dạng chuẩn thứ nhất (1NF).

2. Dạng chuẩn thứ hai (2NF)

a. Định nghĩa:

Một sơ đồ quan hệ R được xem là thoả dạng chuẩn 2 nếu nó ở dạng chuẩn 1 và không có phụ thuộc hàm từng phần.

b. Ví dụ 1:

Sơ đồ quan hệ SAIP (khoá SI) với các phụ thuộc hàm: $SI \rightarrow P$ và $S \rightarrow A$ đã vi phạm dạng chuẩn thứ 2. Do khoá là SI, và $S \rightarrow A$ là phụ thuộc hàm từng phần.

c. Ví dụ 2:

Trong ví dụ trên, sơ đồ quan hệ CUNG_UNG không thoả dạng chuẩn thứ 2 vì các phụ thuộc hàm (b), (c) và (d) là các phụ thuộc hàm từng phần.

Ta có thể tách CUNG_UNG thành 2 sơ đồ quan hệ:

$CUNG_UNG2(MaNSX, MaH, SL)$

$NSX(MaNSX, VonNSX, TP, Nuoc)$

Khi đó, cả hai sơ đồ mới này đều thoả 2NF.

3. Dạng chuẩn thứ ba (3NF)

a. Định nghĩa:

Một sơ đồ quan hệ R được xem là thoả dạng chuẩn 3 nếu nó ở dạng chuẩn 2 và không có phụ thuộc hàm truyền (bắc cầu).

b. Ví dụ:

Sơ đồ quan hệ NSX ở ví dụ trên không thoả dạng chuẩn 3 vì có chứa phụ thuộc hàm truyền: $TP \rightarrow Nuoc$

Ta có thể tách NSX thành 2 sơ đồ quan hệ:

$NSX(MaNSX, VonNSX, TP)$

$TP(TP, Nuoc)$

Khi đó, 2 sơ đồ này thoả 3NF.

4. Dạng chuẩn Boyce – Codd (BCNF)

a. Định nghĩa:

Một sơ đồ quan hệ R được gọi là thoả dạng chuẩn Boyce – Codd nếu với mọi phụ thuộc hàm không tầm thường đều có về trái là siêu khoá.

b. Ví dụ 1:

Sơ đồ quan hệ CSZ (khoá là CS và SZ) với các phụ thuộc hàm: $CS \rightarrow Z$ và $Z \rightarrow C$ thoả dạng chuẩn 3 nhưng không thoả Boyce – Codd vì phụ thuộc hàm $Z \rightarrow C$, Z không là siêu khoá.

c. Ví dụ 2:

Xét lược đồ: LOPHOC(Lop, MonHoc, GiaoVien) với 2 phụ thuộc hàm:

$GiaoVien \rightarrow Monhoc$

$Lop, MonHoc \rightarrow GiaoVien$

Lược đồ có 2 khoá:

$K_1 = \text{Lop, MonHoc}$ và

$K_2 = \text{Lop, GiaoVien}$

Tất cả các thuộc tính đều là thuộc tính khoá, do đó lược đồ ở dạng chuẩn 3. Tuy nhiên lược đồ không ở dạng chuẩn Boyce – Codd vì phụ thuộc hàm: $\text{GiaoVien} \rightarrow \text{MonHoc}$ không thỏa yêu cầu về trái phải là siêu khoá.

II. KẾT HỢP CHUẨN HOÁ VỚI CÁC PHÉP TÁCH LƯỢC ĐỒ QUAN HỆ

1. Dùng phép tách bảo tồn phụ thuộc hàm để đưa về dạng chuẩn 3

a. Giải thuật:

- *Input*: Lược đồ quan hệ R, tập các phụ thuộc hàm F; không làm mất tính tổng quát giả sử rằng nó là phủ tối thiểu.

- *Output*: Một phép tách ρ trên R có bảo toàn phụ thuộc hàm, sao cho mọi sơ đồ quan hệ đều ở dạng chuẩn ba và thỏa chiều F trên sơ đồ quan hệ đó.

- *Phương pháp*:

+ Nếu có một hoặc một số thuộc tính nào đó của R không có mặt trong tất cả các phụ thuộc hàm của F thì về nguyên tắc bản thân chúng sẽ tạo thành một sơ đồ quan hệ và ta sẽ xoá chúng ra khỏi R.

+ Nếu có một phụ thuộc hàm của F chứa hết tất cả các thuộc tính của R thì kết quả cần tìm chính là R.

+ Ngược lại thì kết quả sẽ chứa sơ đồ quan hệ XA cho từng phụ thuộc hàm $X \rightarrow A$ trong F. Tuy nhiên, nếu $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$ trong F, ta có thể dùng sơ đồ $X \rightarrow A_1A_2\dots A_n$ thay vì n sơ đồ $X \rightarrow A_i$ ($1 \leq i \leq n$). Thực tế cách thay thế này được ưa thích hơn.

b. Ví dụ:

Cho lược đồ quan hệ: HOC(Mon, GV, TG, Phong, SV, KQ) và tập phụ thuộc hàm:

$\text{Mon} \rightarrow \text{GV}; \quad \text{TG, SV} \rightarrow \text{Phong};$

$\text{Mon, SV} \rightarrow \text{KQ}; \quad \text{TG, GV} \rightarrow \text{Phong}$

$\text{TG, Phong} \rightarrow \text{Mon};$

Theo giải thuật ta tách HOC thành các sơ đồ quan hệ:

H1(Mon, GV) H4(Mon, SV, KQ)

H2(TG, Phong, Mon) H5(TG, SV, Phong)

H3(TG, GV, Phong)

c. Định lý:

Giải thuật tạo ra một phép tách bảo toàn phụ thuộc hàm để đưa về dạng chuẩn 3.

2. Dùng phép tách kết nối không mất thông tin và phép tách bảo tồn phụ thuộc hàm để đưa về dạng chuẩn 3

a. Giải thuật:

- *Input*: Một phép tách ρ trên R có bảo toàn phụ thuộc hàm sao cho mọi sơ đồ quan hệ đều ở dạng chuẩn thứ ba và thỏa chiều của F trên sơ đồ quan hệ đó.

- *Output*: Một phép tách σ trên R với kết nối không mất thông tin và bảo toàn phụ thuộc hàm sao cho mọi quan hệ đều ở dạng chuẩn thứ ba và thỏa chiều của F trên sơ đồ quan hệ đó.

- *Phương pháp*: Nếu ρ không có lược đồ con chứa khoá của R và X là khoá nào đó của R thì ta chỉ đơn giản là thêm vào σ một sơ đồ quan hệ chứa X: $\sigma = \rho \cup \{X\}$

b. *Ví dụ*:

Xét lại ví dụ trên, ta có kết quả:

H1(<u>Mon</u> , GV)	H4(<u>Mon</u> , <u>SV</u> , KQ)
H2(<u>TG</u> , <u>Phong</u> , Mon)	H5(<u>TG</u> , <u>SV</u> , Phong)
H3(<u>TG</u> , <u>GV</u> , Phong)	H6(<u>TG</u> , <u>SV</u>)

Ta thấy H3 đã chứa TG, SV nên kết quả là:

H1(<u>Mon</u> , GV)	H4(<u>Mon</u> , <u>SV</u> , KQ)
H2(<u>TG</u> , <u>Phong</u> , Mon)	H5(<u>TG</u> , <u>SV</u> , Phong)
H3(<u>TG</u> , <u>GV</u> , Phong)	

c. *Định lý*:

Gọi ρ là phép tách đưa R về dạng chuẩn thứ ba bảo toàn phụ thuộc hàm, và gọi X là một khoá của R. Ta có $\sigma = \rho \cup \{X\}$ là một phép tách R thành các sơ đồ quan hệ ở dạng chuẩn thứ ba, phép tách này bảo toàn phụ thuộc hàm và có kết nối không mất thông tin.

PHẦN II – HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MICROSOFT ACCESS

(Tổng số: 45 tiết, Lý thuyết: 20, Thực hành: 25 tiết)

**CHƯƠNG 5 – TỔNG QUAN HỆ QUẢN TRỊ CSDL
MICROSOFT ACCESS**

(Tổng số: 2 tiết, Lý thuyết: 1 tiết, Thực hành: 1 tiết)

§ 1. TỔNG QUAN HỆ QUẢN TRỊ CSDL MICROSOFT ACCESS**1. Giới thiệu**

Là một hệ quản trị CSDL quan hệ (Relation Database Management System) chạy trên môi trường Windows 98/Me/2000/XP hoặc Windows 7 trở lên do hãng phần mềm Microsoft sản xuất và phát triển trong thời gian từ 1992-2001.

Đây là phần mềm chuyên dùng trong quản lý, là hệ thống các chương trình hỗ trợ các tác vụ quản lý, khai thác dữ liệu theo mô hình CSDL quan hệ thực thể kết hợp.

Microsoft Access là một hệ quản trị CSDL tương tác người sử dụng chạy trong môi trường Windows. Microsoft Access cho chúng ta một công cụ hiệu lực và đầy sức mạnh trong công tác tổ chức, tìm kiếm và biểu diễn thông tin.

Microsoft Access cho ta các khả năng thao tác dữ liệu, khả năng liên kết và công cụ truy vấn mạnh mẽ giúp quá trình tìm kiếm thông tin nhanh. Người sử dụng có thể chỉ dùng một truy vấn để làm việc với các dạng CSDL khác nhau. Ngoài ra, có thể thay đổi truy vấn bất kỳ lúc nào và xem nhiều cách hiển thị dữ liệu khác nhau chỉ cần động tác nhấp chuột.

Microsoft Access và khả năng kết xuất dữ liệu cho phép người sử dụng thiết kế những biểu mẫu và báo cáo phức tạp đáp ứng đầy đủ các yêu cầu quản lý, có thể vận động dữ liệu và kết hợp các biểu mẫu và báo cáo trong một tài liệu và trình bày kết quả theo dạng thức chuyên nghiệp.

Microsoft Access là một công cụ đầy năng lực để nâng cao hiệu suất công việc. Bằng cách dùng các Wizard của MS Access và các lệnh có sẵn (macro) ta có thể dễ dàng tự động hóa công việc mà không cần lập trình. Đối với những nhu cầu quản lý cao, Access đưa ra ngôn ngữ lập trình Access Basic (Visual Basic For application) một ngôn ngữ lập trình mạnh trên CSDL.

2. Các đặc điểm của Access

Trong MS Access 2010, dữ liệu được lưu trữ trong một tập tin duy nhất có phần mở rộng dạng .ACCDB chứa tất cả các đối tượng của một ứng dụng quản lý.

Công cụ thông minh Wizard: chỉ dẫn người sử dụng thao tác từng bước, cho phép thiết kế các đối tượng trong CSDL Access một cách nhanh chóng.

Công cụ truy vấn bằng lưới QBE (Query by Example): công cụ này sẽ hỗ trợ người sử dụng thực hiện các câu truy vấn SQL (Structure Query Language) qua thao tác mà không cần quan tâm đến cú pháp lệnh trong ngôn ngữ con truy vấn này.

Cơ chế tự động kiểm tra: kiểm tra khóa chính (Primary key), các ràng buộc về khóa, miền giá trị... của dữ liệu trong bảng và giữa các bảng liên kết một cách chặt chẽ.

Khả năng trao đổi dữ liệu: trao đổi dữ liệu qua lại giữa các ứng dụng khác như Word, Excel...

Ứng dụng có thể sử dụng trên môi trường mạng máy tính: cho phép nhiều người sử dụng và có chế độ bảo mật CSDL tốt.

Kiểu dữ liệu OLE (Object Linking and Embedding) cho phép người sử dụng có thể nhúng vào bên trong tập tin CSDL Access các đối tượng khác như tập tin văn bản Word, bảng tính Excel, hình ảnh (.bmp, .gif...), âm thanh (.wav...)...

§ 2. QUY TRÌNH CƠ BẢN THIẾT KẾ MỘT CSDL

1. Thiết kế cơ sở dữ liệu

Một CSDL được thiết kế tốt cho phép người sử dụng truy cập nhanh chóng đến những thông tin cần tham khảo, giúp tiết kiệm được thời gian truy xuất thông tin. Một CSDL thiết kế tốt giúp người sử dụng rút ra được những kết quả nhanh chóng và chính xác hơn.

Để thiết kế một CSDL tốt chúng ta phải hiểu cách mà một hệ quản trị CSDL quản trị các CSDL như thế nào. Microsoft Access hay bất kỳ một hệ quản trị CSDL nào có thể cung cấp các thông tin cho chúng ta một cách chính xác và hiệu quả nếu chúng được cung cấp đầy đủ mọi dữ kiện về nhiều đối tượng khác nhau lưu trữ trong các bảng dữ liệu. Ví dụ ta cần một bảng để chứa thông tin về lý lịch của cán bộ, một bảng khác để chứa các đề tài nghiên cứu khoa học của các cán bộ...

Khi bắt tay thiết kế CSDL, chúng ta phải xác định và phân tích các thông tin muốn lưu trữ thành các đối tượng riêng rẽ, sau đó báo cho hệ quản trị CSDL biết các đối tượng đó liên quan với nhau như thế nào. Dựa vào các quan hệ đó mà hệ quản trị CSDL có thể liên kết các đối tượng và rút ra các số liệu tổng hợp cần thiết.

2. Các bước thiết kế cơ sở dữ liệu

Bước 1: Xác định mục tiêu khai thác CSDL của chúng ta. Điều này quyết định các loại sự kiện chúng ta sẽ đưa vào MS Access.

Bước 2: Xác định các bảng dữ liệu cần thiết. Mỗi đối tượng thông tin sẽ hình thành một bảng trong CSDL của chúng ta.

Bước 3: Sau khi đã xác định xong các bảng cần thiết, tiếp đến ta phải chỉ rõ thông tin nào cần quản lý trong mỗi bảng, đó là xác định các trường. Mỗi loại thông tin trong bảng gọi là trường. Mọi mẫu in trong cùng một bảng đều có chung cấu trúc các trường. Ví dụ: Trong lý lịch khoa học cán bộ, những trường (thông tin) cần quản lý là: “HỌ VÀ TÊN”, “CHUYÊN MÔN”, “HỌC VỊ”, “HỌC HÀM”...

Bước 4: Xác định các mối quan hệ giữa các bảng. Nhìn vào mỗi bảng dữ liệu và xem xét dữ liệu trong bảng này liên hệ thế nào với dữ liệu trong bảng khác. Thêm trường hoặc tạo bảng mới để làm rõ mối quan hệ này. Đây là vấn đề hết sức quan trọng, tạo được quan hệ tốt sẽ giúp chúng ta nhanh chóng truy tìm và kết xuất dữ liệu.

Bước 5: Tinh chế, hiệu chỉnh lại thiết kế. Phân tích lại thiết kế ban đầu để tìm lỗi, tạo bảng dữ liệu và nhập vào vài bản ghi, thử xem CSDL đó phản ánh thế nào với những yêu cầu truy xuất của chúng ta, có rút được kết quả đúng từ những bảng dữ liệu đó không. Thực hiện các chỉnh sửa thiết kế nếu thấy cần thiết.

§ 3. LÀM VIỆC VỚI MÔI TRƯỜNG ACCESS

I. LÀM QUEN VỚI ACCESS

1. Khởi động chương trình Access 2010

Chọn menu Start/All Programs/Microsoft Office/Microsoft Office Access 2010

2. Làm quen với môi trường Access

Sau khi khởi động Access, màn hình làm việc của Access sẽ xuất hiện. Các thành phần của cửa sổ này gồm:

Thanh menu: Gồm các Tab chứa các chức năng riêng biệt. Khi chọn một trong các tab này sẽ hiện ra các biểu tượng. Mỗi biểu tượng đại diện cho một hành động nào đó mà khi người sử dụng chọn thì nó sẽ thực hiện. Các công cụ này sẽ được thay đổi tùy vào đối tượng làm việc hiện hành của Access.

Tab Home: chứa các tập hợp lệnh mà ta sẽ sử dụng để thay đổi các khung xem, thêm, định dạng, lọc và tìm kiếm các bản ghi trong CSDL.

Tab Create: chứa các công cụ cần thiết để thiết kế và thêm các đối tượng của Access như Table, Query, Form, Report, Macro...

Tab External Data: cung cấp các công cụ để nhập và xuất dữ liệu giữa tập tin Access và các tập tin Excel, text, XML v.v. Ta cũng sẽ tìm thấy các công cụ để lưu file dưới dạng định dạng PDF và XPS, thu thập thông tin qua Email và đồng bộ hóa thông tin với các danh sách Office SharePoint.

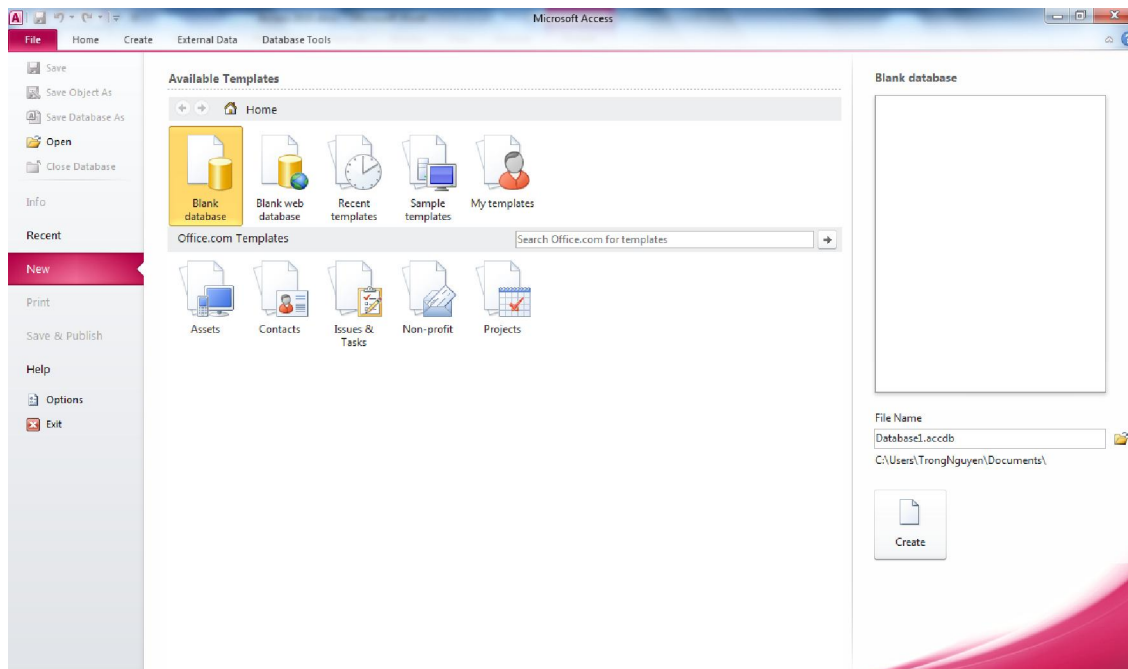
Tab Database Tools: chứa các công cụ cao cấp hơn để làm việc với Macro, hiển thị các mối quan hệ và đối tượng phụ thuộc, phân tích, quản lý và đồng bộ hóa thông tin.

Thanh trạng thái (Status bar): thể hiện tình trạng hiện hành của các phím đặc biệt (NumLock, CapsLock...), các thông báo, hướng dẫn của ứng dụng trong quá trình sử dụng Access

II. TẠO CSDL MỚI VÀ LÀM VIỆC VỚI CSDL ĐÃ CÓ

1. Tạo mới một tập tin CSDL

- ✓ Sau khi khởi động Access 2010 ta chọn tiếp mục File/New
- ✓ Chọn tiếp mục “Blank database”



Hình: Tạo mới một tập tin CSDL

- ✓ Mục File name: Đặt tên cho tập tin CSDL (tập tin có phần mở rộng là **accdb**)

- ✓ Chọn Create để tạo tập tin CSDL

2. Mở CSDL đã có

- ✓ Chọn vào menu tab File, chọn Open (hoặc nhấn Ctrl + N)
- ✓ Chọn tập tin CSDL cần mở, chọn Open.

3. Các đối tượng trong tập tin CSDL MS Access

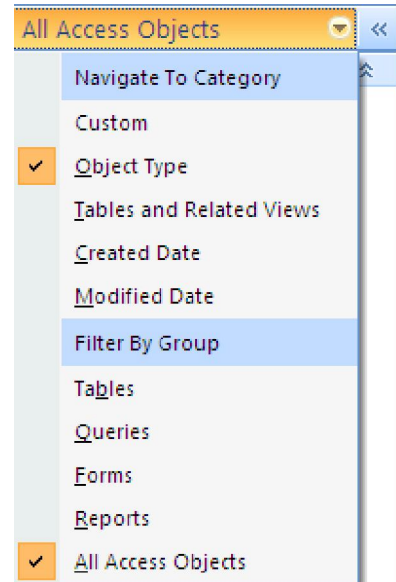
Để quản lý các đối tượng trong tập tin CSDL Access, ta chọn mục “All Access Objects” bên trái phân giao diện MS Access.

Tables (Bảng): Dùng để thiết kế cấu trúc các bảng dữ liệu và toàn bộ dữ liệu sẽ chứa đựng trong Table. Một CSDL thường gồm nhiều bảng có quan hệ với nhau.

Queries (Truy vấn): Dùng để thiết kế truy vấn nhằm tổng hợp, sắp xếp, tìm kiếm dữ liệu, cập nhật dữ liệu cho bảng. Khi thực hiện truy vấn sẽ nhận được một tập hợp kết quả trên màn hình dưới dạng bảng.

Forms (Mẫu biểu): Dùng để thiết kế các biểu mẫu nhằm hai mục đích chính:

- ✓ Nhập dữ liệu cho các bảng (có thể nhập trực tiếp trong Tables)
- ✓ Tạo giao diện quản lý chương trình, tạo giao diện chương trình dưới dạng một bảng nút lệnh hoặc một hệ thống menu.



Reports: Dùng để thiết kế các bản báo cáo phục vụ công việc in ấn. Nó có các khả năng:

- ✓ In dữ liệu dưới dạng bảng
- ✓ Sắp xếp dữ liệu trước khi in
- ✓ Sắp xếp dữ liệu theo nhóm. Cho phép thực hiện các phép toán để nhận dữ liệu tổng hợp trên mỗi nhóm
- ✓ In dữ liệu của nhiều bảng có quan hệ trên một báo cáo
- ✓ Có thể định dạng font chữ, cỡ chữ, màu chữ của báo cáo

Macros: Một tập hợp các lệnh nhằm tự động hóa các thao tác thường ngày thay vì phải lặp lại một cách nhàm chán, tốn thời gian thao tác. Khi chạy một macro, Access tự động thực hiện hàng loạt các thao tác đã được ghi lại trong macro.

4. Chuẩn bị môi trường làm việc

Thực hiện phần này để nhập số liệu kiểu ngày theo dạng dd/mm/yyyy (ngày/tháng/năm) theo kiểu Việt Nam và định dạng kiểu biểu diễn của dữ liệu số.

Trong Windows 7, chọn Start / Control Panel / Region and Language. Xuất hiện hộp thoại Region and Language.

Trong cửa sổ Region and Language, chọn mục Additional settings. Xuất hiện hộp thoại Customize Format.

Chọn thẻ Number, xem và hiệu chỉnh các mục sau:

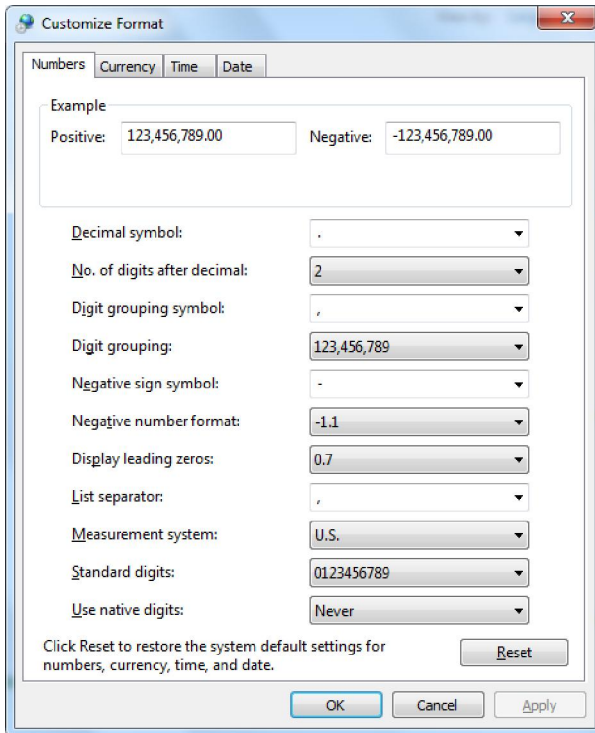
- ✓ Decimal symbol: Ký hiệu phân cách phần nguyên và phần thập phân.
- ✓ No. of digits after decimal: Số ký số ở phần thập phân.

- ✓ Digit grouping symbol: Ký hiệu phân cách phần ngàn, triệu,...
- ✓ List separator: Dấu phân cách dùng phân cách các đối số trong hàm Access (dùng trong cả Excel và Word,...)
- ✓ Measurement system: Chọn đơn vị đo lường US (inch) hoặc Metric (Centimeter).

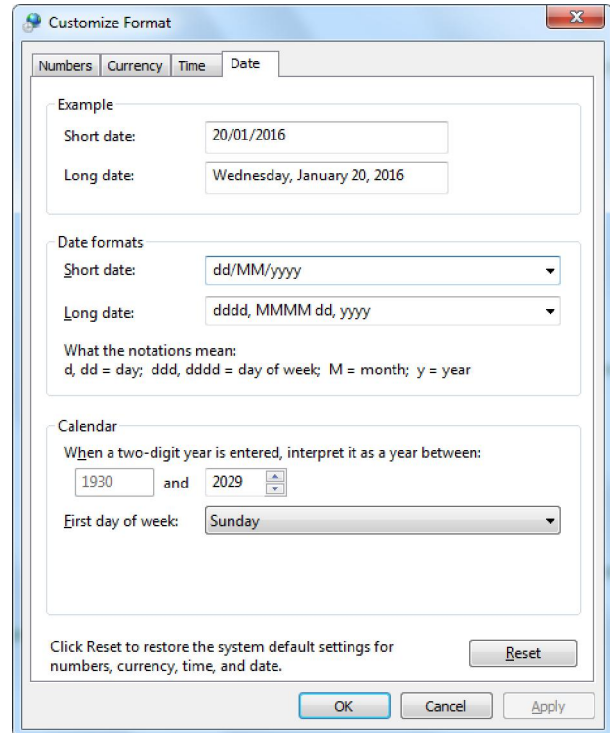
Chọn thẻ Date, xem và hiệu chỉnh mục sau:

- ✓ Short date: nhập vào cú pháp dd/mm/yy (day: ngày; month: tháng; year: năm)

Chọn Apply / OK để chấp nhận việc định dạng.



Hình: Thẻ hộp thoại Numbers



Hình: Thẻ hộp thoại Date

CHƯƠNG 6 – TẠO LẬP CƠ SỞ DỮ LIỆU

(Tổng số: 9 tiết, Lý thuyết: 4 tiết, Thực hành: 5 tiết)

§ 1. TẠO LẬP BẢNG VÀ HIỆU CHỈNH DỮ LIỆU

Bảng là đối tượng chủ yếu chứa các thông tin cần quản lý, có thể đó chỉ là một vài địa chỉ đơn giản hay cả vài chục nghìn bản ghi chứa đựng thông tin liên quan đến các hoạt động sản xuất kinh doanh của một công ty xuất nhập khẩu nào đó. Trước khi ta muốn làm việc với bất kỳ một CSDL nào thì ta phải có thông tin để quản lý, các thông tin đó nằm trong các bảng, nó là cơ sở để cho người sử dụng tạo các đối tượng khác trong CSDL như truy vấn, biểu mẫu, báo cáo...

I. TẠO LẬP BẢNG DỮ LIỆU

1. Khái niệm về bảng

Bảng là nơi trực tiếp chứa dữ liệu về một đối tượng thông tin nào đó như SINHVIEN (thông tin đối tượng sinh viên), HOADON (thông tin hóa đơn)... gồm có nhiều cột và dòng.

Mỗi cột / trường (field) chỉ cho phép chứa duy nhất một loại dữ liệu, ví dụ như dữ liệu dạng số, văn bản, ngày tháng, logic... Dòng / mẫu tin / bản ghi (record) là một thể hiện của dữ liệu trên 1 dòng của bảng.

Trong một CSDL có thể chứa nhiều bảng, thường mỗi bảng lưu trữ nhiều thông tin (dữ liệu) về một đối tượng thông tin nào đó, mỗi một thông tin đều có những kiểu đặc trưng riêng, mà với Access nó sẽ cụ thể thành những kiểu dữ liệu của các trường.

2. Một số quy định về bảng

Số ký tự tối đa trong tên đặt cho bảng	64
Số ký tự tối đa trong tên đặt cho trường (cột)	64
Số trường tối đa trong một bảng	255
Số bảng mở tối đa cùng lúc	1024
Kích thước tối đa của bảng	1 GB
Số ký tự tối đa trong trường kiểu Text	255
Số ký tự tối đa trong trường kiểu Memo	65535
Số chỉ mục (Index) tối đa trong một bảng	32
Số ký tự tối đa trong một thông báo kiểm tra dữ liệu nhập (Validation Text)	255
Số ký tự tối đa trong phần mô tả (Description) bảng hay cột	255

Trong MS Access có hai cách để tạo bảng: dùng Table Wizard hoặc dùng Design View. Khi dùng Table Wizard, các trường ở đây MS Access tự động đặt tên và không có bàn tay can thiệp của người sử dụng. Ở đây, ta sẽ đưa ra cách tạo mới bảng hoàn toàn do người sử dụng.

Ghi chú: Kích cỡ tối đa của một tập tin CSDL Access là 2 Gigabytes.

3. Thiết kế bảng mới bằng Design View

Các bước để tạo một bảng mới bằng màn hình thiết kế Design View

Bước 1: Trong cửa sổ của CSDL đang thiết kế, chọn menu tab Create / Table Design. Xuất hiện cửa sổ thiết kế cấu trúc bảng:

Hình: Cửa sổ thiết kế cấu trúc bảng (Design View)

Bước 2: Nhập các thông tin cần thiết trong cửa sổ thiết kế cấu trúc bảng:

- **Field Name:** đặt tên trường (cột) với các quy ước đặt tên như sau:

Phải bắt đầu bằng ký tự chữ cái (A-Z, a-z) hoặc số (0-9). Chiều dài tối đa 64 ký tự.

Tên trường có thể có khoảng trắng. Tên trường không được chứa dấu chấm câu (.), dấu chấm than (!), dấu ngoặc vuông [].

Thường đặt tên ngắn gọn, không chứa khoảng trắng và gọi nhớ. Tên trường có thể đặt dài để dễ mô tả được thông tin quản lý, nhưng sẽ khó khăn hơn khi ta dùng câu lệnh truy vấn SQL và lập trình Access. Do đó khi đặt tên trường ta nên đặt ngắn gọn, dễ gọi nhớ, gõ không dấu và không nên chứa ký tự khoảng trắng.

- **Data Type:** chọn kiểu dữ liệu của trường (sẽ giới thiệu phần sau)
- **Description:** gõ chú thích ý nghĩa của cột, nội dung này sẽ hiện ở thanh trạng thái (Status bar) để hướng dẫn nhập liệu khi đang ở chế độ Datasheet View.
- **Field Properties:** xác định các thuộc tính của trường (sẽ giới thiệu phần sau)

Bước 3: Xác định khoá chính (Primary Key) của bảng (sẽ giới thiệu phần sau)

Bước 4: Lưu cấu trúc bảng và đặt tên bảng.

- Chọn biểu tượng hình ổ đĩa A trên thanh công cụ hoặc chọn menu tab File / Save hoặc nhấn tổ hợp phím Ctrl + S.
- Xuất hiện thông báo nhập tên bảng cần lưu. Nhập tên bảng cần lưu, chọn OK.

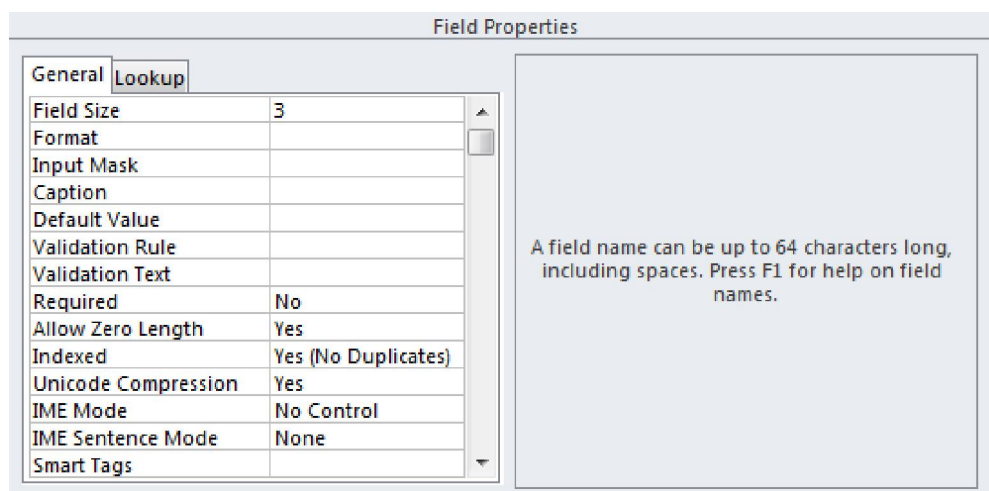
4. Các kiểu dữ liệu của trường

Microsoft Access cung cấp các kiểu dữ liệu sau:

Data Type	Giải thích
Text	Kiểu ký tự: tối đa 255 ký tự
Memo	Kiểu ký ức: tối đa 64000 ký tự, thường được dùng để lưu trữ những văn bản nội dung dài chẳng hạn như chú thích, tiểu sử cá nhân...
Number	Kiểu số: gồm các kiểu số có miền giá trị khác nhau
Date/Time	Kiểu ngày tháng: dùng lưu trữ dữ liệu dạng ngày tháng
Currency	Kiểu tiền tệ: đây là một dạng kiểu Number nhưng có ký hiệu tiền tệ.
Yes/No	Kiểu Logic: lưu dữ liệu chỉ trong 2 trạng thái.
AutoNumber	Kiểu số tự động: dữ liệu trong cột có kiểu này sẽ được tự động thêm vào các giá trị tuần tự hay ngẫu nhiên mà người dùng không cần nhập vào.
OLE Object	Liên kết đối tượng: lưu âm thanh, hình ảnh tối đa 1GB bộ nhớ.
Hyperlink	Kiểu siêu liên kết: lưu đường dẫn tới một máy khác trên mạng hay địa chỉ liên kết tới một trang Web, địa chỉ Email, tối đa 2048 ký tự.
Lookup wizard	Tham chiếu tới danh sách các giá trị của một cột nào đó ở bảng khác.

II. CÁC THUỘC TÍNH CỦA TRƯỜNG

Đặt thuộc tính là một phần không kém quan trọng, nó quyết định đến dữ liệu thực sự lưu giữa trong bảng, kiểm tra độ chính xác dữ liệu khi nhập vào, định dạng dữ liệu nhập vào ... Mỗi một kiểu dữ liệu sẽ có các thuộc tính và các đặc trưng và khác nhau.



Hình: Đặt thuộc tính của trường

1. General - Các thuộc tính chung

a. *Field Size*: Quy định độ rộng của dữ liệu trong cột.

Đối với dữ liệu kiểu **Text**, ta chọn độ rộng từ 0 đến 255. Ví dụ: ta quy định là 10 thì chỉ được nhập tối đa 10 kí tự.

Đối với dữ liệu kiểu **Number**, ta chọn các kiểu Field Size với miền giá trị sau:

Field Size	Miền giá trị	Số Byte
Byte	Số nguyên (0 → 255)	1
Integer	Số nguyên (-32,768 → 32,767)	2
Long Integer	Số nguyên (-2,147,483,648 → 2,147,483,647)	4
Decimal	Số nguyên ($-10^{28}-1 \rightarrow 10^{28}-1$)	2
Single	Số thực (đơn) ($-3.4 \cdot 10^{38} \rightarrow 3.4 \cdot 10^{38}$)	4
Double	Số thực (kép) ($-1.797 \cdot 10^{308} \rightarrow 1.797 \cdot 10^{308}$)	8

b. **Format**: Định dạng các thể hiện của dữ liệu trong cột. Với từng kiểu dữ liệu ta có các kiểu định dạng khác nhau.

Đối với dạng **Text**:

Ký tự	Tác dụng
@	Dữ liệu bắt buộc nhập
>	Dữ liệu trong cột tự động chuyển thành chữ hoa
<	Dữ liệu trong cột tự động chuyển thành chữ thường

Đối với dữ liệu dạng **Date/Time**, ta chọn các dạng có sẵn:

Format	Short Date
Input Mask	General Date
Caption	Long Date
Default Value	Medium Date
Validation Rule	Short Date
Validation Text	Long Time
Required	Medium Time
Indexed	Short Time

Hoặc ta dùng các kí tự d, m, y để thiết lập như sau: (với d: day, m: month, y:year)

- ✓ dd/mm/yyyy, ví dụ: 19/12/2010
- ✓ dd-mmm-yyyy, ví dụ: 19-Dec-2010
- ✓ mm/dd/yy, ví dụ: 12/19/07

Đối với dữ liệu dạng **Number**, ta dùng các định dạng có sẵn. Giả sử có giá trị số 3456.789

Format	Standard
Decimal Places	General Number
Input Mask	Currency
Caption	Euro
Default Value	Fixed
Validation Rule	Standard
Validation Text	Percent
Required	Scientific

Đối với dạng dữ liệu **Yes/No**: có các dạng Yes/No, True/False, On/Off. Nếu không khai báo Format thì sẽ hiện các giá trị 1 (Yes), 0 (No)

- c. *Decimal Places*: Quy định số chữ số thập phân (chỉ có trong kiểu dữ liệu Number, Currency).
- d. *Caption*: Chuỗi tiêu đề cột khi bảng ở chế độ Datasheet (chế độ cập nhật dữ liệu), nếu bỏ qua thì Access tự động lấy tên trường làm tiêu đề cột.
- e. *Default Value*: Xác định giá trị mặc định (định trước)
- f. *Validation Rule*: Quy tắc kiểm tra dữ liệu hợp lệ khi nhập liệu cho bảng.

Ví dụ: với trường kiểu dữ liệu Number, nếu tại dòng thuộc tính này nhập:

>0 thì chỉ các giá trị lớn hơn 0 mới được phép nhập vào trường.

>=100000 and <=1000000 thì chỉ các giá trị trong khoảng 100000 đến 1000000 mới được nhập vào trường.

- g. *Validation Text*: Chuỗi thông báo lỗi sẽ xuất hiện nếu nhập dữ liệu vào bảng sai với quy tắc đã định nghĩa trong Validation Rule.
- h. *Required*: Có bắt buộc phải nhập dữ liệu không?
 - ✓ Yes: tương ứng với việc buộc nhập dữ liệu
 - ✓ No: không buộc nhập dữ liệu tại cột đó.
- i. *Allow Zero Length*: Áp dụng cho Text, Memo, Hyperlink cho phép nhập hay không nhập dữ liệu là giá trị rỗng.
- j. *Input Mask*: Mặt nạ định dạng dữ liệu dùng để bắt buộc người sử dụng khi nhập dữ liệu vào bảng phải tuân theo đúng định dạng đó.

Nhóm ký tự định nghĩa Input Mask:

Kí tự mặt nạ	Ý nghĩa
0	Ký số 0-9 (bắt buộc nhập)
9	Ký số 0-9 hay khoảng trắng (không bắt nhập)
#	Ký số 0-9, khoảng trắng cho phép nhập dấu.
L	Kí tự A-Z (bắt buộc nhập)
?	Kí tự A-Z (không bắt nhập)
A	Ký tự và ký số (bắt buộc)
a	Ký tự và ký số (không bắt buộc)
&	Bất kì một ký tự nào hoặc khoảng trắng (bắt buộc)
C	Bất kì một ký tự nào hoặc khoảng trắng (không bắt buộc)
. , ; - /	Các dấu phân cách: số lẻ, phần ngàn, ngày, giờ.
Password	Những kí tự đánh vào sẽ hiển thị dấu * (dạng mật khẩu)

Ví dụ sử dụng Input Mask:

- Giá trị trường Mã sinh viên được quy định bắt buộc nhập 1 ký tự chữ cái đầu tiên và 2 chữ số theo sau, ta thiết lập thuộc tính Input Mask: L00
- Giá trị trong cột ngày sinh có dạng 19/12/1978, ta thiết lập Input Mask: 00/00/0000.

2. Lookup - Tham chiếu dữ liệu

Display Control: chọn cách hiển thị

- ✓ Text Box: hộp văn bản
- ✓ Combo Box: hộp liệt kê thả
- ✓ List Box: hộp liệt kê danh sách

Nếu Display Control đặt là Text box thì sẽ không có các tùy chọn tiếp theo, nếu Display Control đặt là List Box / Combo Box sẽ có thêm các phần tùy chọn như sau:

- ✓ Row Source Type: kiểu nguồn dữ liệu
 - Table/Query: lấy dữ liệu có sẵn trong Table hoặc Query.
 - Value List: tự đánh dữ liệu cho danh sách.
 - Field List: danh sách là tên các trường trong Table hoặc query.
- ✓ Row Source: chọn nguồn giá trị trong danh sách (có thể chọn tên của Table/Query, hoặc tự liệt kê giá trị...)
- ✓ Bound Column: thứ tự của cột được lấy làm giá trị hiển thị.
- ✓ Column Count: số lượng cột hiển thị trong danh sách.

VD: Trong bảng SINHVIEN, trường MAKHOA có giá trị lấy từ bảng KHOA như sau:

SINHVIEN								
	MASV	HOSV	TENS	PHAI	NGAYSINH	NOISINH	MAKHOA	HOCBON
+	A01	Nguyễn Thu	Hải	☐	25/02/1980	Sài Gòn	AV	100,000
+	A02	Trần Văn	Chính	☒	24/12/1982	Sài Gòn	AV	100,000
+	A03	Lê Thu Bạch	Yến	☐	21/02/1982	Hà Nội	LS	140,000
+	A04	Trần Anh	Tuấn	☒	08/12/1984	Long An	SH	80,000
+	A05	Trần Thanh	Triều	☒	01/02/1980	Hà Nội	TH	80,000
+	B01	Trần Thanh	Mai	☐	20/12/1981	Bến Tre	TR	200,000
+	B02	Trần Thị Thu	Thúy	☐	13/02/1982	Sài Gòn	VL	30,000
+	B03	Trần Thị	Thanh	☐	31/12/1982	Sài Gòn	TH	50,000

Việc thiết kế dữ liệu nhập cho trường MAKHOA sử dụng thẻ Lookup như sau:

Field Properties	
General	Lookup
Display Control	Combo Box
Row Source Type	Table/Query
Row Source	KHOA
Bound Column	1
Column Count	2
Column Heads	No
Column Widths	
List Rows	16
List Width	Auto
Limit To List	No
Allow Multiple Values	No
Allow Value List Edits	No
List Items Edit Form	
Show Only Row Source Values	No

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

III. ĐẶT KHOÁ CHÍNH

1. Khái niệm khóa chính

Sức mạnh của một Hệ Quản trị CSDL như Microsoft Access, là khả năng mau chóng truy tìm và rút dữ liệu từ nhiều bảng khác nhau trong CSDL. Để hệ thống có thể làm được điều này một cách hiệu quả, mỗi bảng trong CSDL cần có một trường hoặc một nhóm các trường có thể xác định duy nhất một bản ghi trong số rất nhiều bản ghi đang có trong bảng. Đây thường là một mã nhận diện như Mã nhân viên hay Số Bảo Danh của học sinh. Theo thuật ngữ CSDL trường này được gọi là khóa chính (primary key) của bảng. MS Access dùng trường khóa chính để kết nối dữ liệu nhanh chóng từ nhiều bảng và xuất ra kết quả yêu cầu.

Nếu trong bảng chúng ta đã có một trường sao cho ứng với mỗi trị thuộc trường đó chúng ta xác định duy nhất một bản ghi của bảng, chúng ta có thể dùng trường đó làm trường khóa của bảng. Từ đó cho ta thấy rằng tất cả các trị trong trường khóa chính phải khác nhau. Chẳng hạn không dùng tên người làm trường khóa vì tên trường là không duy nhất.

Nếu không tìm được mã nhận diện cho bảng nào đó, chúng ta có thể dùng một trường kiểu **AutoNumber** (ví dụ Số thứ tự) để làm trường khóa chính.

Khi chọn trường làm khóa chính chúng ta lưu ý mấy điểm sau:

- ✓ MS Access không chấp nhận các giá trị trùng nhau hay giá trị rỗng (null) trong trường khóa chính.
- ✓ Chúng ta sẽ dùng các giá trị trong trường khóa chính để truy xuất các bản ghi trong CSDL, do đó các giá trị trong trường này không nên quá dài vì khó nhớ và khó nhập.
- ✓ Kích thước của khóa chính ảnh hưởng đến tốc độ truy xuất CSDL. Để đạt hiệu quả tối ưu, dùng kích thước nhỏ nhất để xác định mọi giá trị cần đưa vào trường.

2. Mục đích

- ✓ Access tự động tạo chỉ mục (Index) trên khoá nhằm tăng tốc độ truy tìm thông tin và sắp xếp dữ liệu.
- ✓ Khi tìm kiếm dữ liệu, các bản ghi sẽ được đưa ra theo thứ tự của khóa chính.
- ✓ Access dùng khoá để đảm bảo tính toàn vẹn dữ liệu giữa các bảng, có nghĩa là khi xoá, sửa dữ liệu trong bảng chính thì các bảng quan hệ tự động xoá, sửa theo.

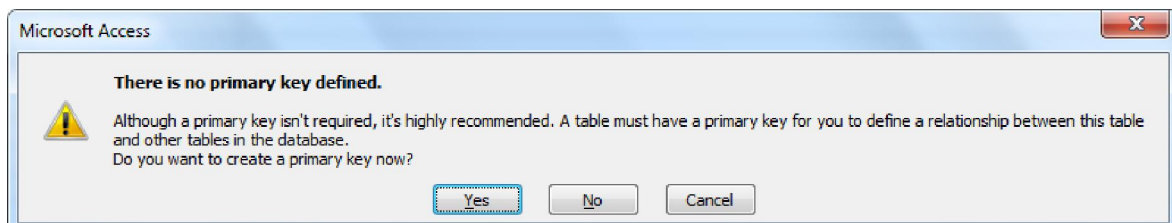
3. Cách đặt khoá

- ✓ Tại giao diện thiết kế cấu trúc bảng (Design View), chọn trường làm khoá chính (có thể chọn nhiều trường bằng cách kết hợp phím Ctrl để chọn các trường không liên tục)
- ✓ Chọn biểu tượng Primary Key (hình chìa khóa màu vàng) trên menu tab Design.

Chú ý:

Không phải mọi trường đều có thể làm khóa chính, mà chỉ có các trường có các kiểu dữ liệu không phải là **Memo** và **OLE Object**, **Hyper Link**.

Những bảng không thiết lập trường khoá chính, trong quá trình lưu lại cấu trúc bảng, Access sẽ xuất hiện một bảng thông báo:



Nếu chọn **Yes** thì Access sẽ tự tạo một trường ID dạng số tự động tăng (AutoNumber) làm trường khoá.

Nếu chọn **No** thì không tạo khoá và sẽ lưu đúng cấu trúc mà ta thiết kế. Nên chọn No vì mỗi bảng có một khóa riêng do người dùng tạo ra.

4. Sửa khoá

- ✓ Chọn lại trường làm khoá đã cài đặt trước.
- ✓ Chọn lại biểu tượng Primary Key để gỡ bỏ thuộc tính khóa chính.

§ 2. TẠO CÁC QUAN HỆ GIỮA CÁC BẢNG

I. THIẾT LẬP QUAN HỆ GIỮA CÁC BẢNG

1. Mục đích

Trong một CSDL thường có rất nhiều vấn đề cần quản lý nên được bóc tách thành nhiều bảng nhỏ. Vì vậy, để quản lý được dữ liệu thì bắt buộc các bảng phải có quan hệ mật thiết với nhau thông qua các trường khoá chính mà ta đã tạo ở trên.

2. Các loại quan hệ trong cơ sở dữ liệu ACCESS

a. Quan hệ một - một (1-1)

Trong quan hệ một - một, mỗi bản ghi trong bảng A có tương ứng với một bản ghi duy nhất trong bảng B và ngược lại.

Ví dụ: Cho 2 bảng dữ liệu

Bảng **SINHVIENTN**(MaSV, Ten, NgaySinh, GioiTinh) và bảng **DIEMTNTN**(MaSV, DiemTB)

Ten	NgaySinh	GioiTinh	MaSV		MaSV	DiemTB
An	20/10/77	Yes	A001	↔	A001	9
Bình	21/07/80	No	A002	↔	A002	7
Thủy	02/12/77	Yes	A003	↔	A003	9
Lan	03/04/80	No	A004	↔	A004	4
Hồng	12/11/77	No	A005	↔	A005	5

Bảng **SINHVIENTN** và **DIEMTNTN** có mối quan hệ 1-1 dựa trên trường **MaSV**. Mỗi sinh viên có một điểm trung bình (**DiemTB**) duy nhất trong bảng **DIEMTNTN**.

b. Quan hệ một - nhiều (1-∞)

Là mối quan hệ phổ biến nhất trong CSDL. Bảng A có quan hệ một - nhiều với bảng B nếu một bản ghi trong bảng A sẽ có thể có nhiều bản ghi tương ứng trong bảng B (hoặc không có bản ghi tương ứng nào trong bảng B), nhưng ngược lại một bản ghi trong bảng B có duy nhất một bản ghi tương ứng trong bảng A.

Ví dụ: Trong một khoa của một trường học nào đó có nhiều sinh viên, nhưng một sinh viên thuộc một khoa nhất định. Ta có 2 bảng dữ liệu như sau:

Bảng **KHOA**(**MaKhoa**, TenKhoa, SoDT)

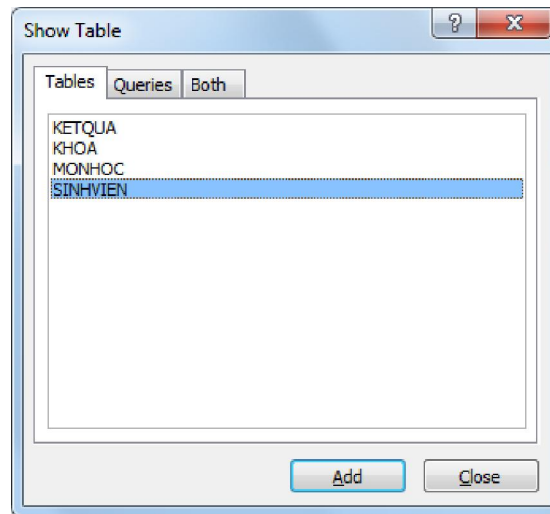
Bảng SINHVIEN(MaKhoa, MaSV, TenSV, Lop)

TenKhoa	SoDT	MaKhoa	MaKhoa	MaSV	TenSV	Lop
CNTT	826767	01	01	A01	Thanh	K23
TOÁN	878787	02	01	A02	Tùng	K24
LÝ	868785	03	02	A03	Thủy	K25
			02	A04	Hùng	K26
			03	A05	Lan	K25
			03	A06	Hương	K26

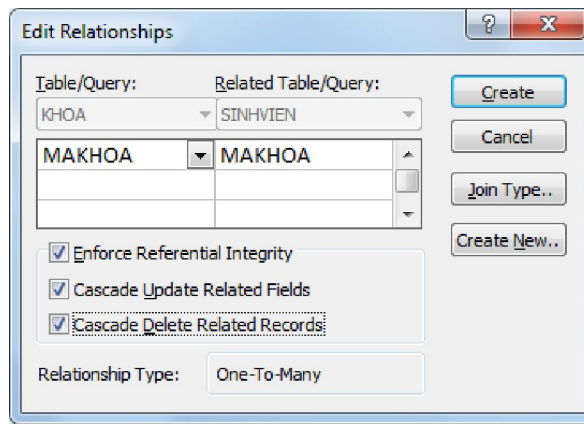
Bảng KHOA và bảng SINHVIEN có mối quan hệ 1-∞ dựa trên trường Makhoa.

3. Thiết lập mối quan hệ giữa các bảng dữ liệu (Relationships)

- ✓ Chọn menu tab Database Tools, chọn Relationships
- ✓ Thêm bảng vào cửa sổ quan hệ từ hộp thoại Show Table: Chọn bảng tham gia tạo quan hệ, chọn nút Add. Chọn nút OK.



- ✓ Đưa trỏ chuột vào trường khoá sau đó giữ chuột trái, kéo và thả vào trường tương ứng của bảng quan hệ (hai trường quan hệ phải giống nhau)
- ✓ Đánh dấu vào mục **Enforce Referential Integrity** để mở hai thuộc tính đảm bảo tính toàn vẹn dữ liệu:
 - **Cascade Update Related Field:** nếu tùy chọn này được chọn thì khi sửa dữ liệu trong trường khoá chính của bảng chính (bảng 1) thì dữ liệu tương ứng trong các bảng quan hệ (bảng ∞) sẽ tự sửa theo.
 - **Cascade Delete Related record:** nếu tùy chọn này được chọn thì khi xóa một bản ghi trong bảng chính, thì các bản ghi chứa dữ liệu tương ứng liên quan trong trường khóa ngoại của các bảng quan hệ sẽ tự xóa theo.



✓ Chọn **Create**.

Chú ý:

Nếu muốn đảm bảo tính toàn vẹn dữ liệu thì ít nhất phải có một trường trong đường quan hệ là khoá chính.

Quan hệ có tính tham chiếu toàn vẹn sẽ đảm bảo các vấn đề sau:

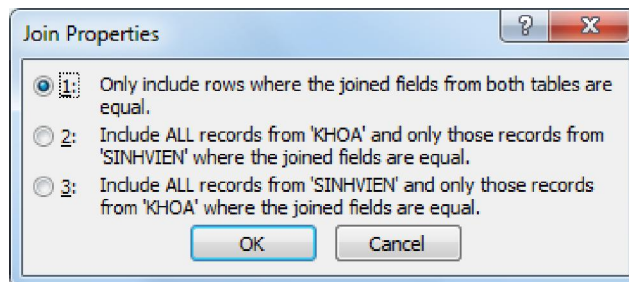
- Khi nhập dữ liệu cho trường tham gia trong bảng quan hệ (bảng ∞) thì phải tồn tại bên bảng chính (bảng 1).
- Trường hợp vi phạm các quy tắc trên thì sẽ nhận được thông báo lỗi.

4. Kiểu kết nối (Join type)

Trong quá trình thiết lập quan hệ giữa các bảng, nếu không chọn nút Create, chọn nút Join type để chọn kiểu liên kết.

Mục 1: Liên kết nội (Inner join)

Mục 2 và mục 3 là liên kết ngoại (Outer join)



5. Chỉnh sửa đường quan hệ

- ✓ Chỉnh sửa đường quan hệ: ta nhấp chuột phải tại đường quan hệ, chọn Edit Relationships và thực hiện các chỉnh sửa mong muốn. Chọn OK.
- ✓ Xoá đường quan hệ: ta nhấp chuột phải tại đường quan hệ cần xoá, chọn Delete.

II. NHẬP DỮ LIỆU VÀO BẢNG

1. Nhập dữ liệu vào bảng

Để nhập dữ liệu vào bảng, ta nhấp đúp chuột vào bảng cần nhập. Ngoài ra, ta có thể nhấp chuột phải tại bảng cần nhập, chọn lệnh Open.

Lưu ý: Ngoài cách nhập trực tiếp bằng bảng, ta có thể tạo các mẫu biểu (Form) để nhập dữ liệu cho đơn giản và dễ nhập hơn (cách tạo các mẫu biểu để nhập dữ liệu sẽ được hướng dẫn cụ thể trong phần làm việc trên Form).

2. Kiểm tra tính hợp lệ của dữ liệu khi nhập

Ví dụ: Khi nhập dữ liệu ngày tháng cho CSDL quản lý hoá đơn bán hàng với hai trường ngày lập hoá đơn (**NgàyLapHD**) và ngày nhận hàng (**NgàyNhan**) trong bảng **HoaDon**, muốn kiểm tra điều kiện ngày nhận hàng $\geq$ ngày lập hoá đơn, ta làm như sau:

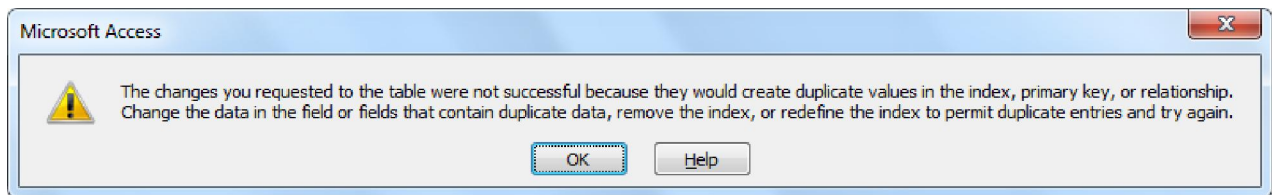
- ✓ Mở bảng **HoaDon** ở chế độ thiết kế cấu trúc bảng (Design View)
- ✓ Nhấp chuột phải tại phần nhập tên trường, chọn **Properties**
- ✓ Tại dòng thuộc tính **Validation Rule** gõ điều kiện: **[NgàyLapHD]<=[NgàyNhan]**

III. CÁC THÔNG BÁO LỖI KHI NHẬP DỮ LIỆU

Trong khi cập nhật dữ liệu vào bảng chúng ta có thể sẽ gặp một số thông báo lỗi của Microsoft Access. Để có thể hiểu và thấy được lỗi sai của mình và để khắc phục sau đây là một số thông báo lỗi thường gặp:

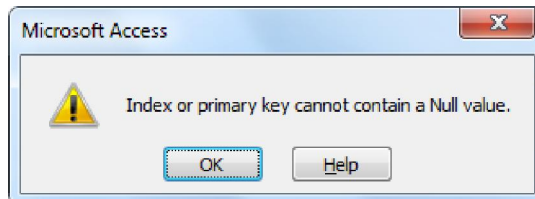
1. Trùng khóa chính (Primary key)

Xảy ra khi thêm hoặc sửa giá trị của cột có chứa khóa chính mà giá trị nhập vào đã có.



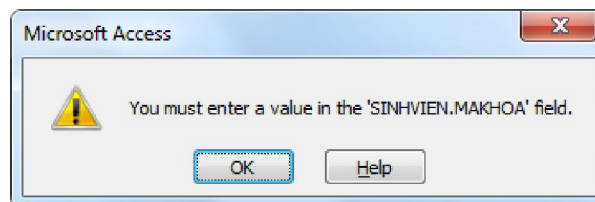
2. Khóa chính không được rỗng (Null)

Xảy ra khi ta không nhập dữ liệu cho cột là khóa chính.



3. Dữ liệu bắt buộc nhập (Required)

Xảy ra khi ta chọn thuộc tính Required của một trường nào đó mà ta không nhập dữ liệu cho trường này.

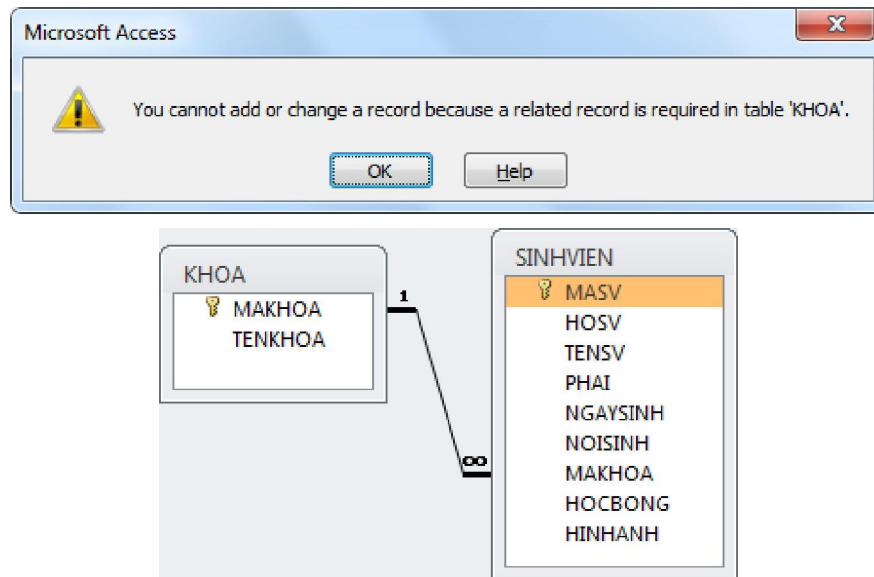


4. Khóa ngoại không tồn tại (Foreign key)

Xảy ra khi ta nhập giá trị cho khóa ngoại mà giá trị này không có trong bảng chứa khóa chính.

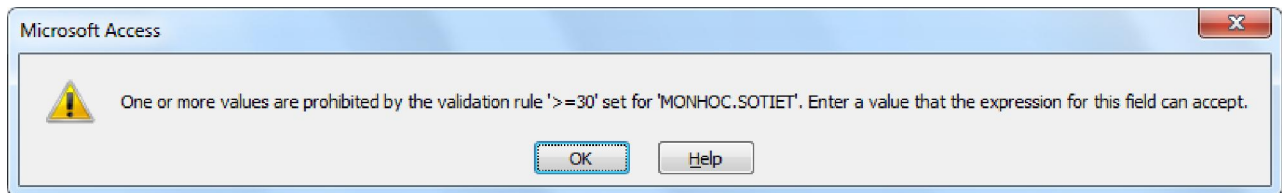
Ví dụ: Cho CSDL quản lý sinh viên, giữa 2 bảng SINHVIEN và KHOA có mối liên kết khóa chính - khóa ngoại thông qua thuộc tính MAKHOA. Khi ta nhập dữ liệu cho một sinh

viên trong bảng SINHVIEN trước mà chưa nhập dữ liệu danh sách các khoa trong bảng KHOA, lúc đó ta sẽ nhận được thông báo lỗi như sau:



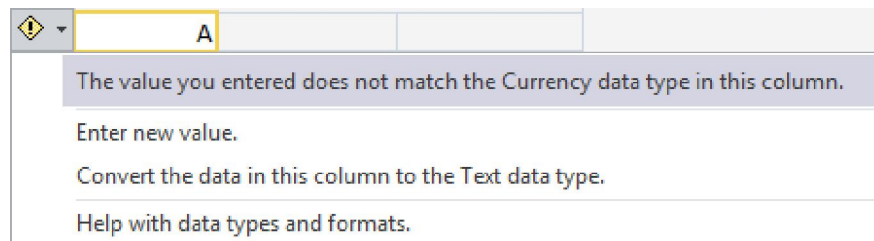
5. Vi phạm quy tắc kiểm tra dữ liệu (Validation rule)

Xảy ra khi ta nhập sai quy tắc kiểm tra dữ liệu.



6. Sai kiểu dữ liệu (Field type)

Xảy ra khi ta nhập sai kiểu dữ liệu.



CHƯƠNG 7 – TRUY VẤN

(Tổng số: 8 tiết, Lý thuyết: 4 tiết, Thực hành: 4 tiết)

§ 1. QUY TRÌNH TẠO MỘT TRUY VẤN

Sức mạnh thực sự của CSDL là khả năng tìm đúng và đầy đủ thông tin mà chúng ta cần biết, trình bày dữ liệu sắp xếp theo ý muốn. Để đáp ứng yêu cầu trên, Access cung cấp một công cụ truy vấn cho phép đặt câu hỏi với dữ liệu đang chứa bên trong các bảng trong CSDL.

I. KHÁI NIỆM TRUY VẤN**1. Khái niệm truy vấn**

Truy vấn là một công cụ cho phép đặt câu hỏi với dữ liệu trong bảng dữ liệu trong CSDL.

Loại truy vấn thông dụng nhất là truy vấn chọn (Select Query). Với kiểu truy vấn này chúng ta có thể xem xét dữ liệu trong các bảng, thực hiện phân tích và chỉnh sửa trên dữ liệu đó, có thể xem thông tin từ 1 bảng hoặc có thể thêm nhiều trường từ nhiều bảng khác nhau.

2. Các loại truy vấn trong Access

Select Query: Truy vấn lựa chọn

Action Query: Truy vấn hành động, bao gồm:

- ✓ Truy vấn tạo bảng (Make table Query)
- ✓ Truy vấn nối (Append Query)
- ✓ Truy vấn cập nhật (Update Query)
- ✓ Truy vấn xóa dữ liệu (Delete Query)

SQL Query: Truy vấn được viết bởi ngôn ngữ SQL.

Crosstab Query: Truy vấn chéo (Thể hiện dòng và cột)

3. Các chế độ hiển thị truy vấn*a. Cửa sổ thiết kế truy vấn (Design View)*

Trong chế độ này, người sử dụng có thể tạo, sửa chữa một truy vấn nào đó. Màn hình truy vấn chứa 2 phần: phần thứ nhất chứa các bảng (hoặc truy vấn) tham gia truy vấn, phần thứ hai gọi là vùng lưới QBE (Query By Example).

b. Cửa sổ câu lệnh SQL (SQL View).

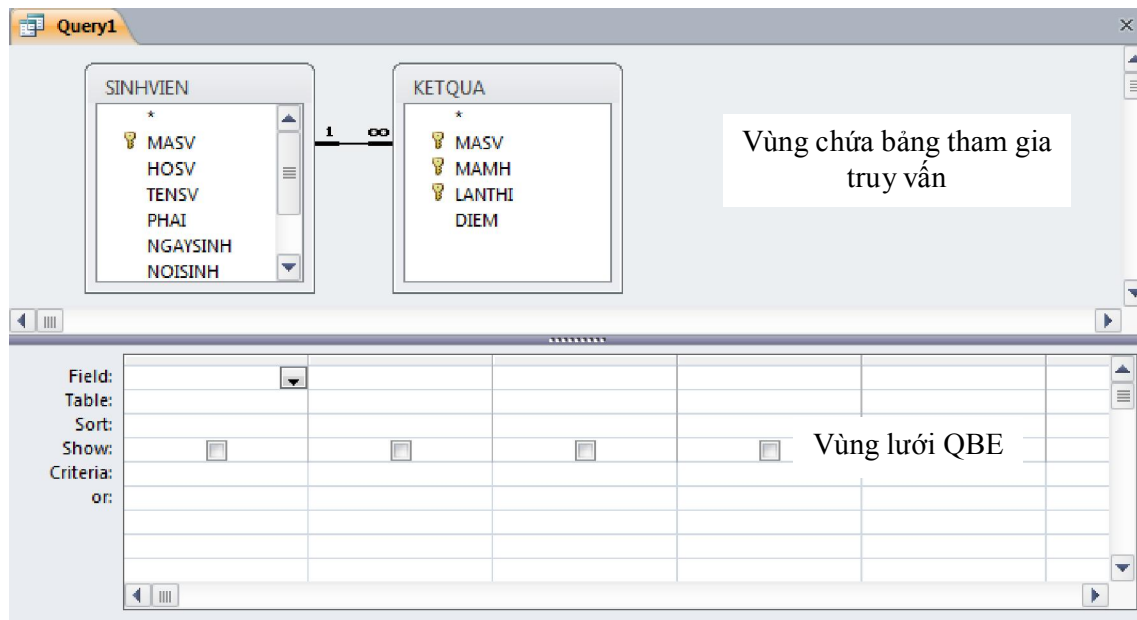
Sử dụng chế độ này để xem mã câu lệnh SQL của truy vấn đang tạo.

II. TẠO TRUY VẤN BẰNG QBE**1. Các thành phần trong màn hình truy vấn**

Vùng chứa bảng tham gia truy vấn: chứa các bảng hoặc truy vấn khác (đã tạo trước đó) làm nguồn dữ liệu cho truy vấn.

Vùng lưới QBE: nơi chứa các cột mà truy vấn muốn hiển thị thông tin, trong đó:

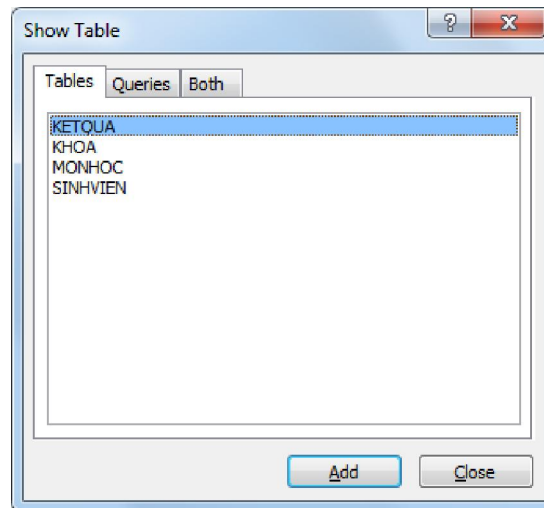
- ✓ Field: chứa cột, biểu thức, hàm trong truy vấn.
- ✓ Table: tên bảng tương ứng của cột.
- ✓ Sort: sắp xếp dữ liệu.
- ✓ Show: hiện hoặc ẩn cột trong kết quả truy vấn.
- ✓ Criteria: chứa giá trị, biểu thức dùng làm điều kiện lọc trong truy vấn.



Các thành phần trong màn hình truy vấn

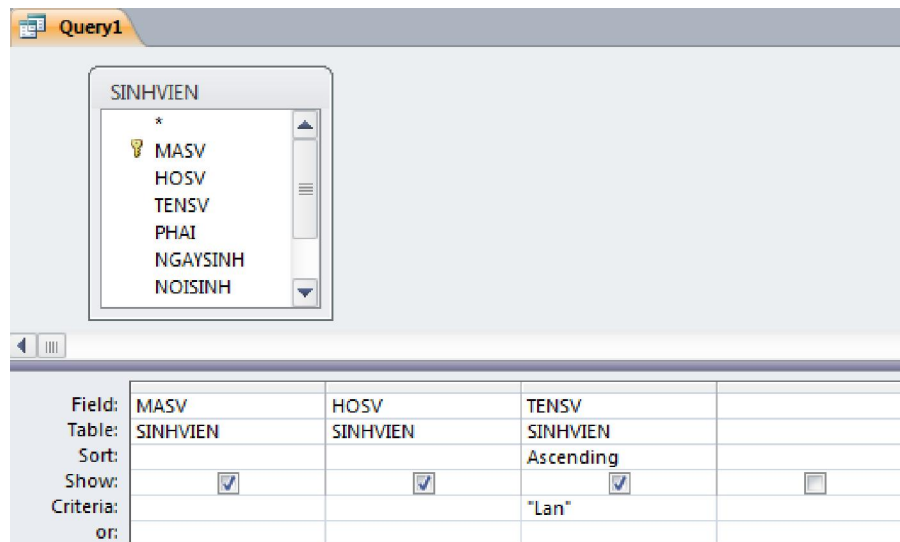
2. Các bước thực hiện tạo truy vấn

- Chọn menu tab Create, chọn Query Design.
- Trong hộp thoại Show Table hiện ra, lần lượt chọn các bảng tham gia truy vấn và chọn nút Add. Chọn Close hoặc đóng hộp thoại (X) để hoàn tất việc chọn bảng.



- Kéo các trường từ bảng cần hiển thị trong kết quả truy vấn xuống dòng **Field**.
- Đưa điều kiện chọn lọc vào dòng **Criteria**. Nếu không có điều kiện chọn lọc thì toàn bộ dữ liệu sẽ được hiển thị.
- Chọn trường sắp xếp dữ liệu trong dòng **Sort** nếu cần. (Accending: sắp xếp tăng dần, Descending: sắp xếp giảm dần)
- Ký tự * dùng để đại diện cho tất cả các cột có trong bảng.
- Tạo các trường mới từ các trường đã có trong bảng nếu cần. Trường mới có thể là hàm hoặc biểu thức tính toán tổng, hiệu, tích, ghép của nhiều trường. Lưu ý: tên trường mới tạo tại dòng Field và phân cách với tổng, hiệu, tích... bằng dấu hai chấm “:”
- Đóng và lưu truy vấn.

Ví dụ: Truy vấn lập danh sách gồm Mã, họ và tên các sinh viên có tên là “Lan”. Danh sách trong kết quả truy vấn được sắp xếp tăng dần theo tên.



Chú ý:

+ Kết quả của truy vấn là một bảng dạng Datasheet, luôn cập nhật theo dữ liệu mới nhất đưa vào bảng.

+ Giá trị điều kiện nếu là dữ liệu dạng:

Số: nhập giá trị số trực tiếp

Ký tự: nhập giá trị ký tự đặt giữa cặp nháy kép “ ”

Tên trường: đặt giữa cặp móc vuông []

Ngày tháng năm: nhập giá trị đặt giữa cặp dấu # #

Cách thi hành truy vấn:

Trong cửa sổ đang thiết kế (lưới QBE), chọn nút lệnh **Run** (biểu tượng dấu chấm than màu đỏ) trên menu tab Design.

3. Các phép toán và hàm thường sử dụng trong truy vấn

a. Các phép toán thường sử dụng

Phép toán số học : +, -, *, /

Phép so sánh: =, >, >=, <, <=, <> (khác)

Phép toán logic: AND, OR, IS NULL (rỗng =0), IS NOT NULL (khác 0)

LIKE: dùng để chọn ra các dữ liệu giống với mẫu dữ liệu được nêu ra. Trong đó:

- Dùng ký tự * để thay cho một nhóm các ký tự không biết.
- Dùng ký tự ? để thay cho một ký tự không biết.

BETWEEN ... AND ...: phép toán lấy giá trị nằm trong khoảng. Phép toán BETWEEN ... AND ... có thể được thay thế bằng >=.... AND <=....

Ví dụ: Muốn lấy các giá trị trong khoảng từ 20 đến 100 thì đưa vào dòng Criteria điều kiện sau: BETWEEN 20 AND 100 hoặc >= 20 AND <=1000

Phép toán IN: dùng để lấy giá trị trong một tập hợp nào đó.

Phép & dùng để nối nhiều chuỗi ký tự thành một. Ví dụ: HOSV & " " & TENS

b. Một số hàm thường dùng

Hàm xử lý dữ liệu chuỗi (Text)

- ✓ Hàm LEFT(<Chuỗi>, n): Trích bên trái chuỗi n ký tự.
- ✓ Hàm RIGHT(<Chuỗi>, n): Trích bên phải chuỗi n ký tự.
- ✓ Hàm MID(<Chuỗi>, i, n): Trích n ký tự nằm giữa một chuỗi bắt đầu từ vị trí thứ i.

Hàm xử lý dữ liệu số (Number)

- ✓ Hàm ROUND(<Số>, n): Làm tròn số thập phân với phần thập phân có n số.

Hàm về ngày tháng

- ✓ DAY([Tên trường]): Trả về giá trị ngày của chuỗi dữ liệu chứa ngày/tháng/năm.
- ✓ MONTH([Tên trường]): Trả về giá trị tháng của chuỗi dữ liệu ngày/tháng/năm.
- ✓ YEAR([Tên trường]): Trả về giá trị năm của chuỗi dữ liệu chứa ngày/tháng/năm.
- ✓ DATE(): Trả về giá trị ngày/tháng/năm hiện tại (theo đồng hồ hệ thống).
- ✓ TIME(): Trả về giá trị giờ:phút:giây hiện tại (theo đồng hồ hệ thống).
- ✓ NOW(): Trả về giá trị ngày, giờ hiện tại (theo đồng hồ hệ thống).

Hàm điều kiện IIF

Cú pháp: **IIF**(Điều kiện, Giá trị 1, Giá trị 2)

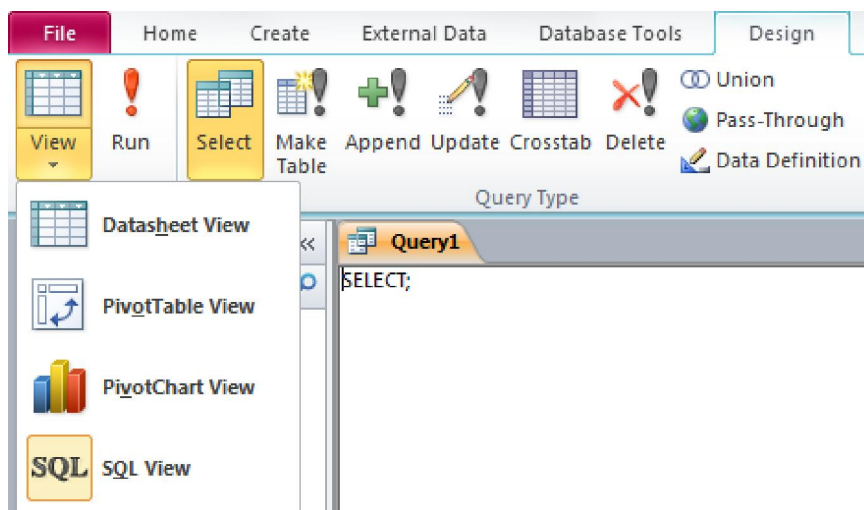
Đây là hàm điều kiện loại trừ, dùng tìm kiếm các bản ghi thoả mãn điều kiện thì nhận giá trị 1, các bản ghi không thoả mãn điều kiện thì nhận giá trị 2 tại vị trí đặt hàm.

Lưu ý: Có thể sử dụng lồng nhiều hàm IIF trong câu lệnh giống như trong MS Excel.

III. TẠO TRUY VẤN BẰNG CÂU LỆNH SQL

Các bước thực hiện viết và thực thi câu lệnh truy vấn bằng ngôn ngữ SQL:

- Chọn menu tab Create, chọn Query Design.
- Trong hộp thoại Show Table hiện ra, chọn Close để đóng hộp thoại (hoặc có thể chọn bảng tham gia truy vấn nếu đã xác định được nguồn dữ liệu truy vấn).
- Trên menu tab Design, chọn View \ SQL View



- Viết nội dung câu lệnh SQL trong khung chứa từ khóa SELECT;
- Chọn nút lệnh **Run** (biểu tượng dấu chấm than màu đỏ) để thực thi câu lệnh.

§ 2. TRUY VẤN HÀNH ĐỘNG

I. NHÓM TRUY VẤN HÀNH ĐỘNG

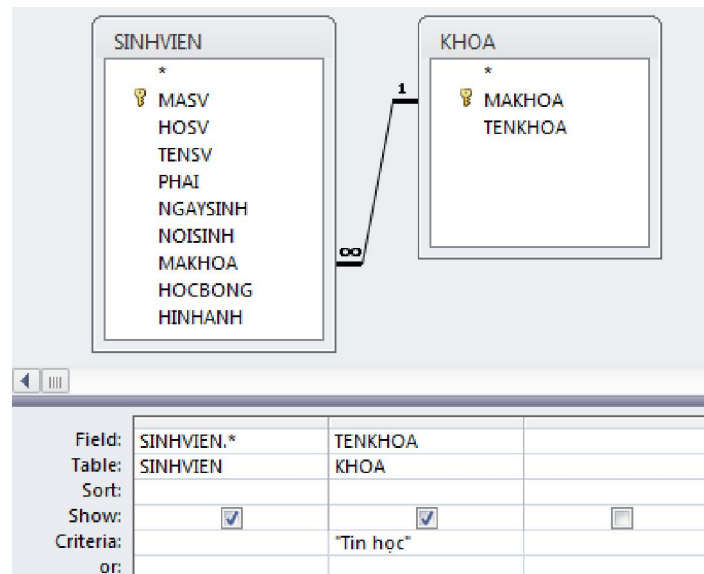
1. Truy vấn tạo bảng (Make Table Query)

Truy vấn tạo bảng là loại truy vấn dùng để tạo bảng dữ liệu mới dựa vào các bảng đã có trong CSDL thỏa mãn điều kiện cho trước. Bảng mới có thể tạo trên CSDL hiện tại hoặc trên CSDL khác. Các bước thực hiện:

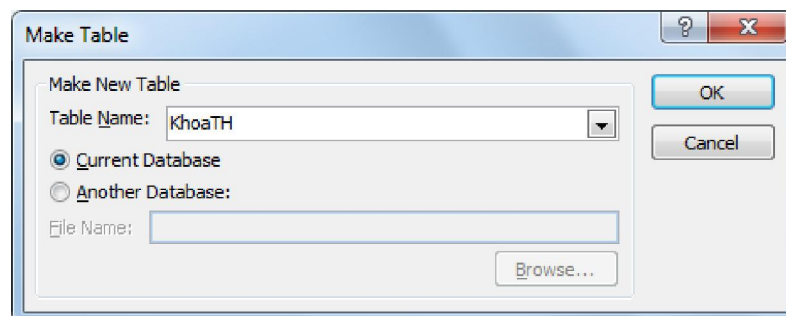
- Chọn menu tab Create, chọn Query Design.
- Trong hộp thoại Show Table hiện ra, chọn bảng tham gia truy vấn.
- Trong phần **Query Type** trên thanh công cụ Design, chọn **Make Table**.
- Gõ tên cho bảng sẽ được tạo mới trong hộp Table Name. Check vào tùy chọn:
 - ✓ Current Database: tạo bảng trong CSDL hiện hành.
 - ✓ Another Database: tạo bảng trong một CSDL khác.
- Chọn các trường cần có trong bảng mới từ bảng nguồn trong dòng **Field**.
- Đặt điều kiện vào dòng **Criteria**.
- Chọn lệnh Run để chạy truy vấn.

Ví dụ: Tạo bảng tên KhoaTH chứa danh sách các sinh viên thuộc khoa có tên “Tin học”.

- Tạo truy vấn chọn theo yêu cầu:



- Trong phần **Query Type** trên thanh công cụ Design, chọn **Make Table**. Đặt tên cho bảng mới:



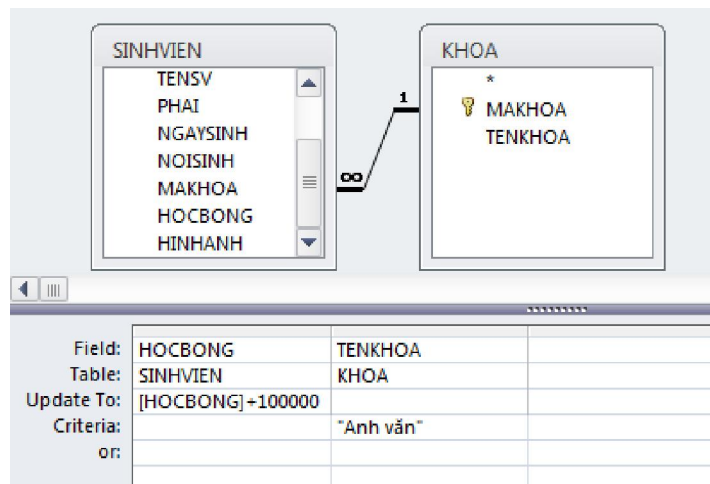
2. Truy vấn cập nhật (Update Query)

Truy vấn cập nhật là truy vấn dùng để sửa chữa hoặc thay thế một loạt bản ghi thỏa mãn yêu cầu bằng một giá trị mới. Các bước thực hiện:

- Chọn menu tab Create, chọn Query Design.
- Trong hộp thoại Show Table hiện ra, chọn bảng tham gia truy vấn.
- Trong phần **Query Type** trên thanh công cụ Design, chọn **Update**.
- Kéo trường cần sửa chữa hoặc tính toán xuống dòng **Field**.
- Nhập giá trị sẽ gán cho các bản ghi trong trường vào dòng **Update to** (có thể nhập trực tiếp hoặc bấm chuột phải vào ô Update to, chọn Build và xây dựng biểu thức)
- Đưa điều kiện vào dòng **Criteria** (nếu không có điều kiện thì tất cả các bản ghi của trường sẽ được thay thế bằng giá trị mới).
- Chọn lệnh Run để chạy truy vấn. Khi xuất hiện bảng thông báo, nếu đồng ý thay thế thì chọn Yes, ngược lại chọn No.

Lưu ý: Kết quả của truy vấn sẽ được cập nhật trực tiếp vào các bản ghi tương ứng của trường cần sửa chữa hoặc tính toán.

Ví dụ: Cập nhật học bổng tăng thêm 100000 cho các sinh viên thuộc khoa có tên “Anh văn”.

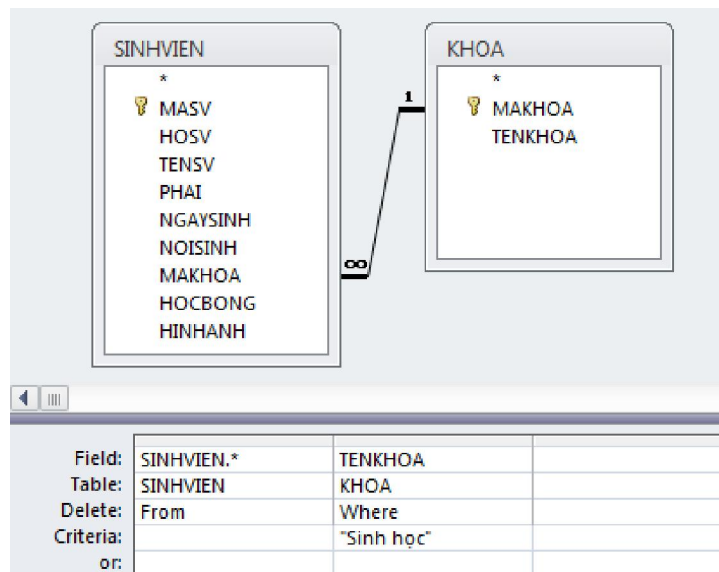


3. Truy vấn xóa (Delete Query)

Truy vấn xóa là loại truy vấn dùng xóa các bản ghi thỏa mãn điều kiện cho trước. Các bước thực hiện:

- Chọn menu tab Create, chọn Query Design.
- Trong hộp thoại Show Table hiện ra, chọn bảng tham gia truy vấn.
- Trong phần **Query Type** trên thanh công cụ Design, chọn **Delete**.
- Chọn bảng chứa các bản ghi (dòng) cần xóa trong dòng **Table**, lưu ý chọn “From” trong dòng **Delete**.
- Chọn trường chứa điều kiện xóa tại dòng **Field**. Nhập điều kiện xóa vào dòng **Criteria**
- Chọn lệnh Run để chạy truy vấn.

Ví dụ: Xóa danh sách các sinh viên thuộc khoa có tên “Sinh học”.



II. CÁC LOẠI TRUY VẤN KHÁC

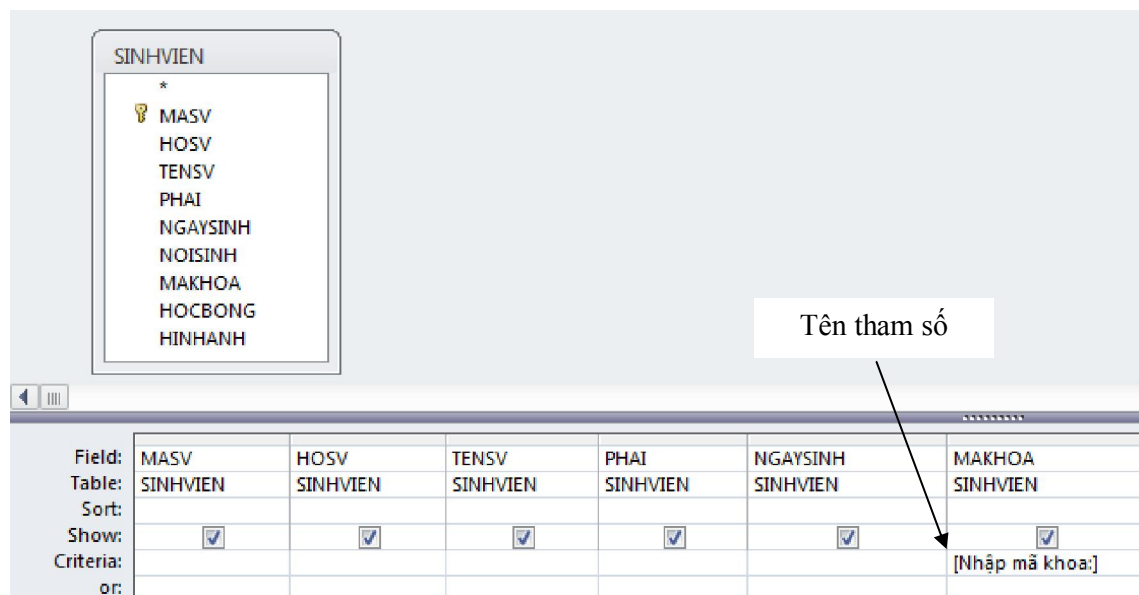
1. Truy vấn truyền tham số

Đây là loại truy vấn dùng tìm các bản ghi thỏa mãn điều kiện chưa được xác định trước mà sẽ tự nhập giá trị điều kiện vào mỗi khi truy vấn được thực hiện. Đây là loại truy vấn rất tiện lợi cho việc tìm kiếm thông tin.

Các bước thực hiện giống như với Select Query nhưng điều kiện đưa vào dòng Criteria được đưa vào cặp dấu ngoặc vuông [] và thường là một dòng thông báo để nhập điều kiện.

Ví dụ: Tạo truy vấn hiển thị danh sách sinh viên thuộc một khoa nào đó với điều kiện Mã khoa (MAKHOA) do người dùng nhập vào từ bàn phím sau mỗi lần thực hiện truy vấn.

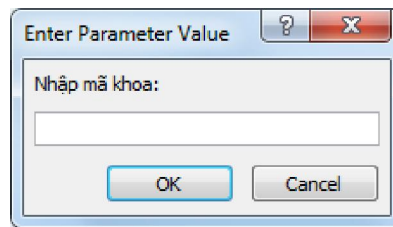
- Tạo truy vấn chọn. Đặt tên cho tham số vào dòng điều kiện (Criteria) của trường MAKHOA và được đặt trong cặp ngoặc vuông [].



- Nhấp phải chuột tại màn hình thiết kế truy vấn, chọn Parameters. Nhập vào tên của tham số và chọn kiểu dữ liệu tương ứng. Chọn OK.

Parameter	Data Type
Nhập mã khoa:	Text
	Yes/No
	Byte
	Integer
	Long Integer
	Currency
	Single
	Double
	Date/Time
	Binary
	Text
	OLE Object
	Memo
	Replication ID
	Decimal
	Value

- Khi thực hiện truy vấn này sẽ xuất hiện bảng thông báo. Người dùng nhập vào giá trị Mã khoa cần xem và chọn OK.



2. Truy vấn gom nhóm dữ liệu

Loại truy vấn này tiến hành phân nhóm các bản ghi sau đó thực hiện phép tính trên từng nhóm. Các bước thực hiện truy vấn:

- Chọn menu tab Create, chọn Query Design.
- Trong hộp thoại Show Table hiện ra, chọn bảng tham gia truy vấn.
- Nhấp chuột phải tại vùng lưới QBE, chọn **Totals**.
- Sử dụng các hàm tính toán tại dòng Total để thực hiện các phép tính cho các trường trong truy vấn:

Group By: Gom các bản ghi có giá trị giống nhau tại một số trường được chỉ định thành 1 nhóm.

Sum: Tính tổng các giá trị trong một nhóm bản ghi tại trường được chỉ định.

First: Tìm giá trị đầu tiên trong nhóm bản ghi tại trường được chỉ định.

Last: Tìm giá trị cuối cùng.

Max: Tìm giá trị lớn nhất.

Min: Tìm giá trị nhỏ nhất.

Avg: Tính trung bình cộng các giá trị.

Count: Đếm tổng số bản ghi trong nhóm.

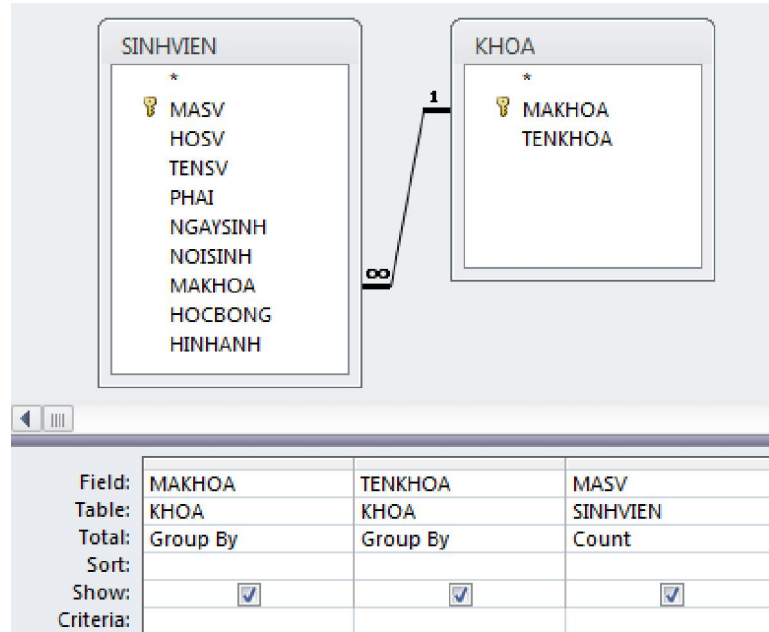
Expression: Thiết lập biểu thức tính toán từ các trường khác trong bảng.

Where: Điều kiện lọc dữ liệu trước khi gom nhóm và không hiển thị trong kết quả.

Ví dụ: Lập thống kê số sinh viên theo danh sách từng khoa. Kết quả truy vấn bao gồm: Mã khoa, tên khoa, số lượng sinh viên.

MAKHOA	TENKHOA	CountOfMASV
AV	Anh văn	2
LS	Lịch sử	1
TH	Tin học	4
VL	Vật lý	1

Màn hình thiết kế truy vấn:



Kết quả truy vấn trước khi tổng hợp dữ liệu:

	MaKhoa	TenKhoa	MaSV	
Group by	AV	Anh văn	A01	Count
	AV	Anh văn	A03	
Group by	LS	Lịch sử	A04	Count
Group by	TH	Tin học	A02	Count
	TH	Tin học	B01	
	TH	Tin học	B02	
	TH	Tin học	B03	

Sau khi tổng hợp dữ liệu, ta có kết quả:

MaKhoa	TenKhoa	CountOfMaSV
AV	Anh văn	2
LS	Lịch sử	1
TH	Tin học	4

3. Truy vấn chéo (Crosstab Query)

Truy vấn chéo là một truy vấn dùng để tổng hợp dữ liệu dưới dạng bảng tính hai chiều, trong đó tiêu đề của dòng và cột của bảng là các giá trị được kết nhóm từ các trường trong bảng dữ liệu, phần thân của bảng tính là dữ liệu được thống kê bởi các hàm: Sum, Count, Avg, Min, Max, và các chức năng khác.

Ví dụ: Lập bảng thống kê điểm thi cao nhất ở mỗi môn học của từng sinh viên.

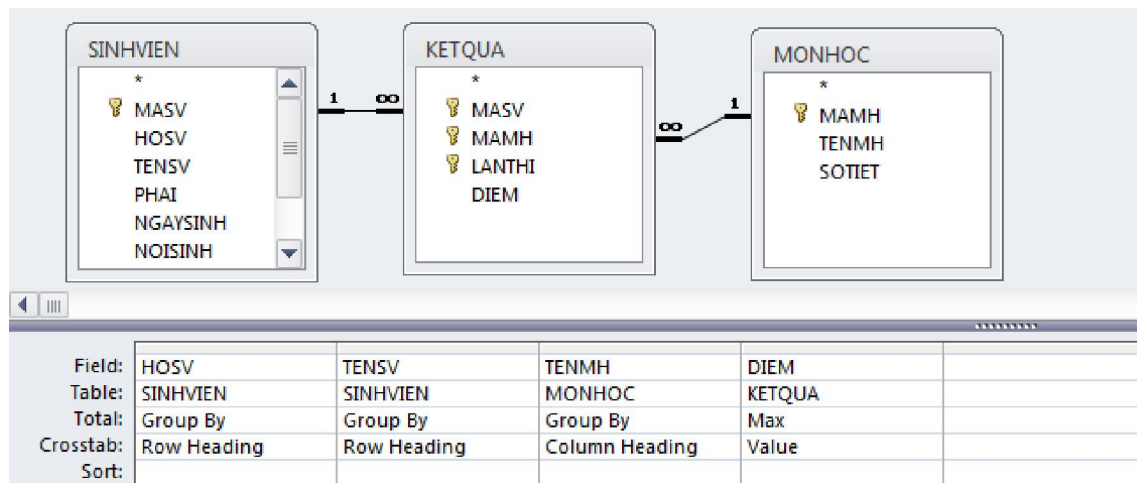
Khi tạo truy vấn loại này cần tối thiểu 3 cột hiển thị dữ liệu:

- Cột hiển thị dữ liệu dòng (Row Heading): Họ và tên sinh viên (HOSV, TENSX)
- Cột hiển thị dữ liệu cột (Column Heading): Tên môn học (TENMH)
- Cột thống kê dữ liệu (Value): Điểm thi lớn nhất.

Row Heading		Column Heading				
HOSV	TENSX	Đồ họa	Nhập môn máy	Trí tuệ nhân tạo	Truyền tin	Văn phạm
Lê Thu Bạch	Yến	10		5		
Nguyễn Thu	Hải		10	5	7	9
Trần Anh	Tuấn	6				
Trần Thanh	Mai	8				
Trần Thị	Thanh			10	9	
Trần Thị Thu	Thúy	8			6	
Trần Văn	Chính		5	7	6	

Các bước tạo truy vấn theo ví dụ trên:

- Chọn menu tab Create, chọn Query Design.
- Trong hộp thoại Show Table hiện ra, chọn bảng tham gia truy vấn.
- Trong phần **Query Type** trên thanh công cụ Design, chọn **Crosstab**.
- Chọn tên trường làm **Row Heading**. Trong ví dụ này ta chọn 2 trường HOSV và TENSX thuộc bảng SINHVIEN. Tại phần Total ta chọn Group By, và tại phần Crosstab ta chọn Row Heading.
- Chọn tên trường làm **Column Heading**. Trong ví dụ này ta chọn trường TENMH thuộc bảng MONHOC. Tại phần Total ta chọn Group By, và tại phần Crosstab ta chọn Column Heading.
- Chọn tên trường cần thống kê và hàm dùng để tính toán tương ứng. Trong ví dụ này ta chọn trường DIEM thuộc bảng KETQUA. Tại phần Total ta chọn hàm Max, và tại phần Crosstab ta chọn **Value**.

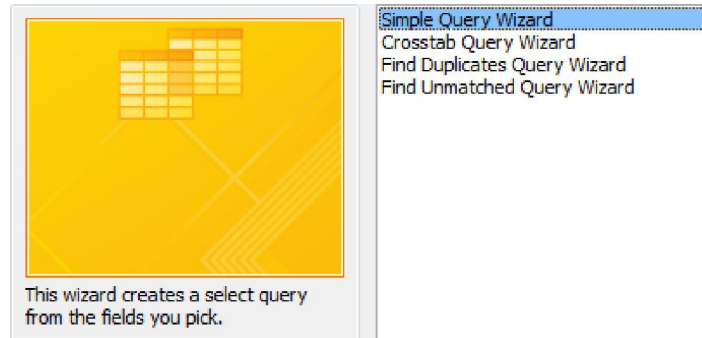


4. Query Wizard

Ngoài các loại truy vấn nêu trên, ta có thể sử dụng công cụ Query Wizard để tạo một số truy vấn có chức năng đặc thù riêng biệt như:

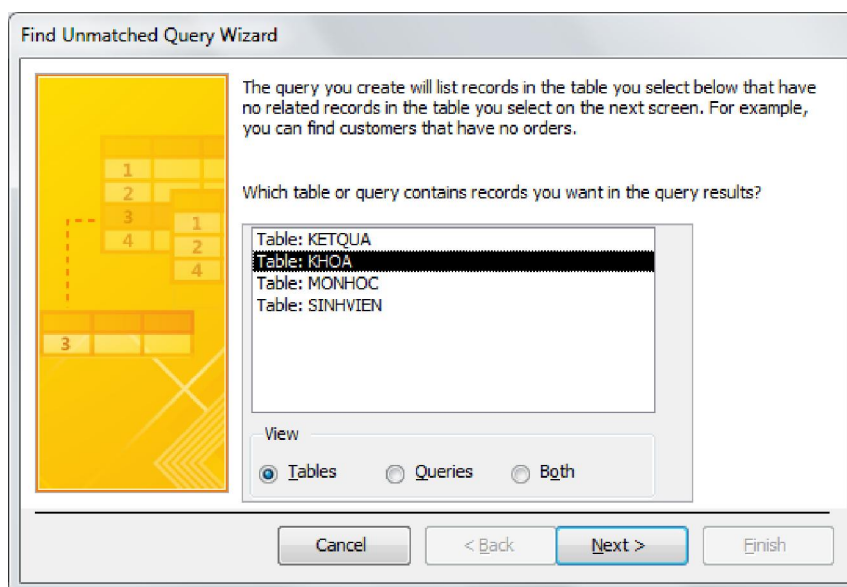
- Tìm nhanh các bản ghi trùng lặp trong bảng theo một hoặc nhiều trường nào đó bằng truy vấn **Find Duplicate Query Wizard**.
- Tìm các bản ghi có ở bảng này mà không có trong bảng so sánh với nó bằng truy vấn **Find Unmatched Query Wizard**.

Ví dụ: Trong bảng danh sách các khoa, ta có thể tìm các khoa mà chưa có sinh viên nào theo học. Khi đó, ta cần so sánh bảng danh sách khoa với bảng danh sách sinh viên. Các bước thực hiện như sau:

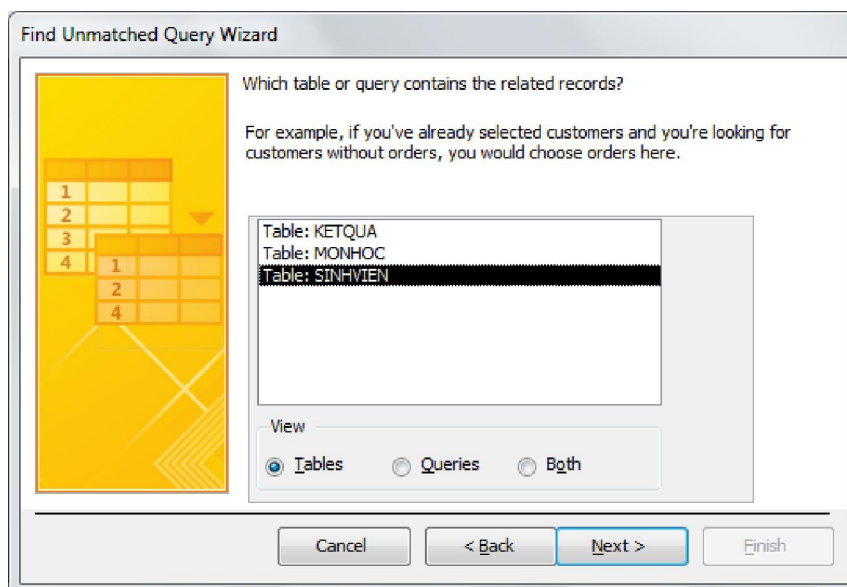


B1: Chọn menu tab Create \ Query Wizard. Trong hộp thoại New Query hiện ra, chọn **Find Unmatched Query Wizard**.

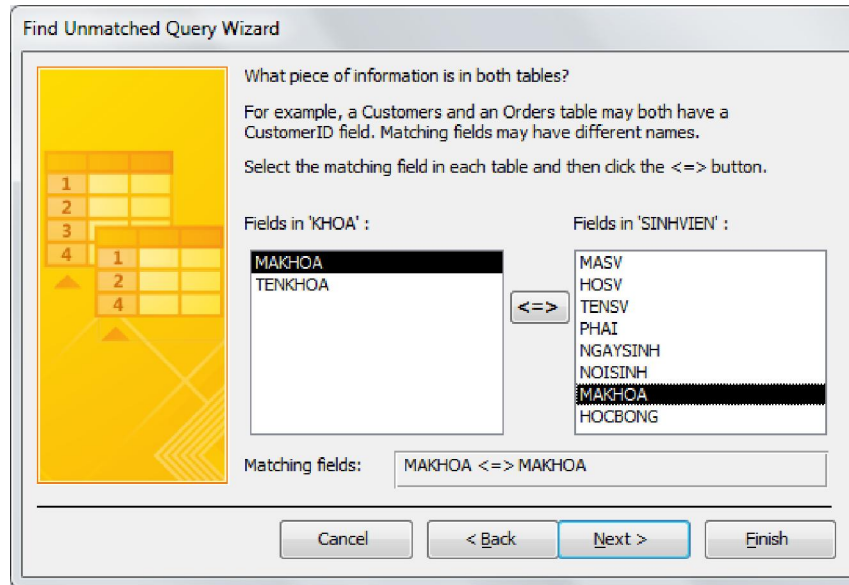
B2: Trong ví dụ tìm các khoa mà chưa có sinh viên nào theo học, xét thấy có liên hệ đối với liên kết dữ liệu giữa bảng KHOA và bảng SINHVIEN (quan hệ 1-∞). Do đó ở bước này ta chọn bảng chính (bảng KHOA) trước.



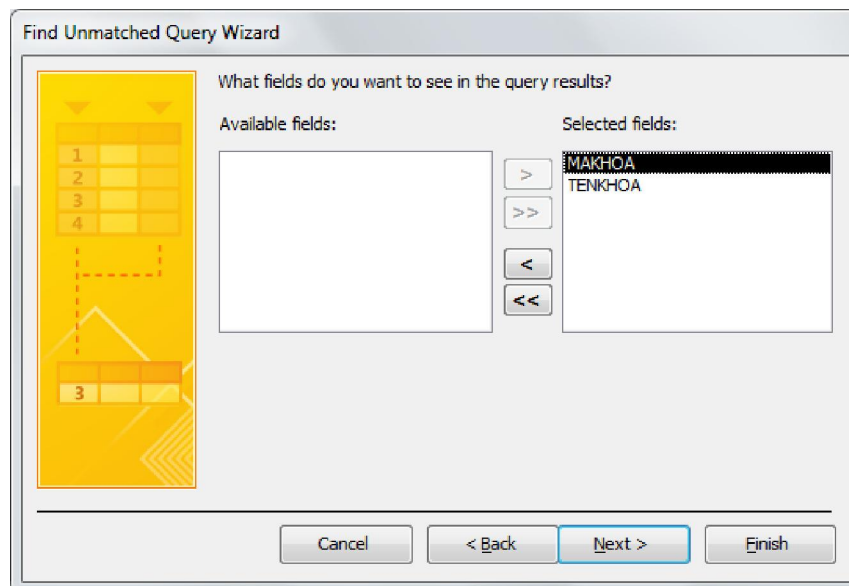
B3: Chọn bảng quan hệ (bảng ∞).



B4: Xác định trường liên kết giữa 2 bảng.



B5: Chọn trường hiển thị kết quả truy vấn.



B6: Chọn Next \ Finish để kết thúc. Kết quả truy vấn sẽ là danh sách gồm Mã khoa, Tên khoa của các khoa mà chưa có sinh viên nào theo học.

KHOA Without Matching SINHVIEN	
MAKHOA	TENKHOA
SH	Sinh học
TR	Triết

CHƯƠNG 8 – BIỂU MẪU

(Tổng số: 10 tiết, Lý thuyết: 5 tiết, Thực hành: 5 tiết)

§ 1. KHÁI NIỆM

Từ trước đến nay chúng ta vẫn làm việc một cách đơn điệu với các bảng, truy vấn với cách trình bày dữ liệu hiệu quả nhưng không đẹp mắt. Với biểu mẫu (form) trong Access sẽ giúp chúng ta khắc phục điều này. Biểu mẫu trong Access rất linh động, chúng ta có thể dùng biểu mẫu để nhập, xem, hiệu chỉnh dữ liệu.

1. Khái niệm về biểu mẫu

Nếu chúng ta đã quen điền các tờ biểu, mẫu trong cuộc sống hàng ngày thì chúng ta có thể hình dung một biểu mẫu trong Access cũng vậy. Một biểu mẫu trong Access định nghĩa một tập dữ liệu chúng ta muốn lấy và từ đó đưa vào CSDL. Cũng vậy biểu mẫu cũng có thể dùng để xem xét dữ liệu hay in ra máy in.

Trong môi trường của Hệ quản trị CSDL Access chúng ta có thể thiết kế các biểu mẫu có hình thức trình bày đẹp, dễ sử dụng và thể hiện đúng các thông tin cần thiết. Chúng ta có thể đưa vào biểu mẫu các đối tượng như văn bản, hình ảnh, đường vẽ kết hợp với các màu sắc sao cho biểu mẫu của chúng ta đạt được nội dung và hình thức trình bày ưng ý nhất. Hình thức và cách bố trí các đối tượng ra sao trên biểu mẫu hoàn toàn tùy thuộc vào khả năng thẩm mỹ và ng khiếu trình bày của chúng ta.

2. Tác dụng của biểu mẫu

Biểu mẫu cung cấp một khả năng thuận lợi để hiển thị dữ liệu. Chúng ta có thể xem mọi thông tin của một bản ghi thay vì ở chế độ Datasheet nhằm chán trước đây bằng chế độ Form View của biểu mẫu theo một phương cách tiên tiến hơn.

Sử dụng biểu mẫu tăng khả năng nhập dữ liệu, tiết kiệm thời gian và ngăn ngừa các lỗi do đánh sai. Chẳng hạn thay vì gõ vào các giá trị của tất cả các trường chúng ta có thể tạo những danh sách (gọi là combo box) để chọn trên biểu mẫu. Đây là phương cách áp dụng rất hiệu quả để tránh đánh sai dữ liệu.

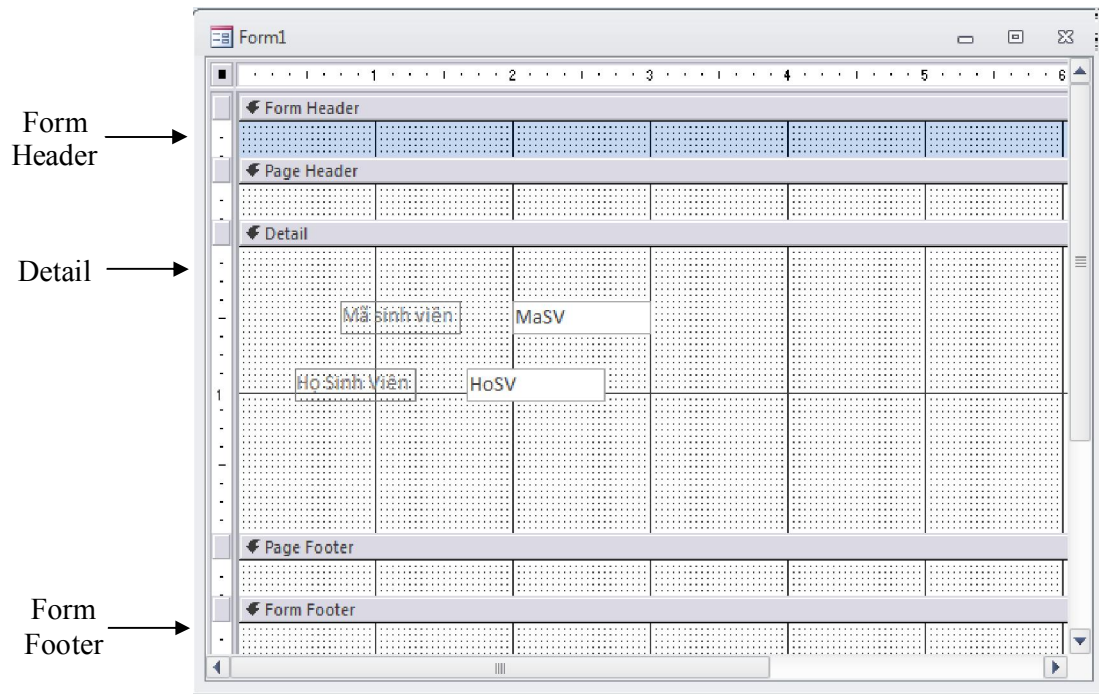
Biểu mẫu cung cấp một hình thức trình bày hết sức tiện nghi để xem, nhập và hiệu chỉnh các bản ghi trong CSDL. Access cung cấp các công cụ hỗ trợ rất đặc lực cho chúng ta trong việc thiết kế những biểu mẫu vừa dễ sử dụng mà lại có thể tận dụng được các khả năng như:

- Thể hiện dữ liệu đẹp, trình bày lôi cuốn với các kiểu font chữ và hiệu ứng đồ họa.
- Quen thuộc với người sử dụng vì nó giống các biểu mẫu trên giấy thông thường.
- Có thể tính toán được.
- Có thể hiển thị dữ liệu từ nhiều bảng (hoặc truy vấn).
- Tự động hóa một số thao tác phải làm thường xuyên.

3. Các thành phần của biểu mẫu

- Form Header: tiêu đề đầu của Form. Thường chứa nội dung tiêu đề, các Label.
- Detail: phần chính của Form, hiển thị chi tiết dữ liệu của biểu mẫu (thường chứa các Label, Text Box).
- Form Footer: tiêu đề cuối của Form. (Thường chứa các nút lệnh hay các ô dữ liệu tính toán thống kê)
- Page Header: tiêu đề đầu của trang khi in Form.
- Page Footer: tiêu đề cuối của trang khi in Form.

Để bật hoặc tắt thanh form Header/Footer, nhấp chuột phải trên thanh Header/Footer của Form, chọn hoặc bỏ chọn lệnh Form Header/Footer.



Hình: Các thành phần của biểu mẫu

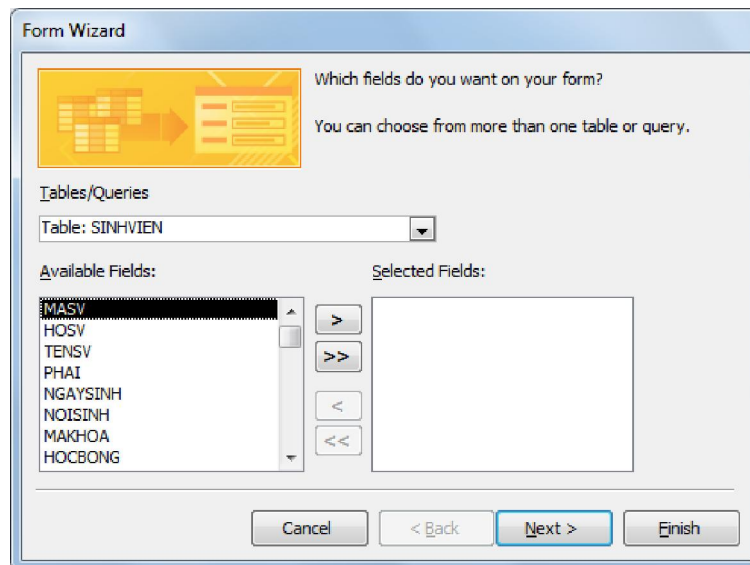
§ 2. THIẾT KẾ

1. Tạo form bằng chức năng Form Wizard

Chức năng Form Wizard là công cụ tạo form qua các bước trung gian với một loạt các câu hỏi gợi ý để giúp ta chọn lựa và xây dựng một form phù hợp nhất. Cách tạo:

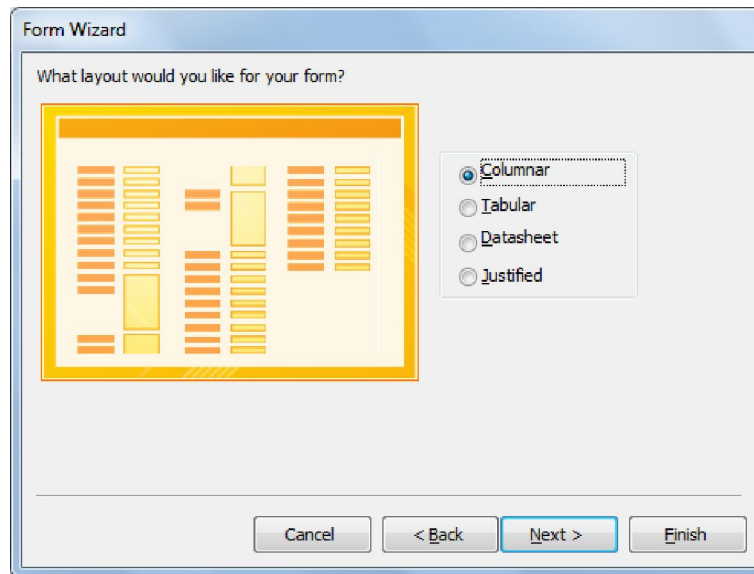
Tại giao diện Access, chọn tab Create, chọn Form Wizard trong nhóm lệnh Forms.

Xuất hiện cửa sổ Form Wizard, chọn Table hoặc Query làm dữ liệu nguồn cho form.

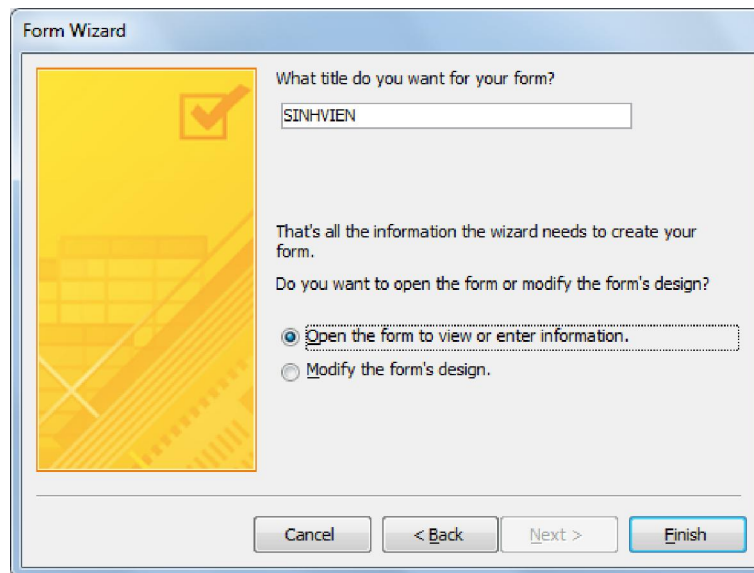


Chọn các trường hiển thị trên Form trong khung Available Fields, chọn nút > để chọn một trường, hoặc nút >> để chọn tất cả các trường trong Table/Query dữ liệu nguồn, có thể chọn nhiều trường trên nhiều bảng. Chọn Next.

Chọn dạng form, bao gồm các dạng: Columnar, Tabular, Datasheet. Chọn Next.



Nhập tiêu đề cho form. Chọn Finish.



2. Tạo Form bằng Form Design

Chọn menu tab Create, chọn nút Design View trong nhóm lệnh Forms. Ta thấy xuất hiện một form trống ở chế độ Design view.

Trên menu tab Design, chọn Property Sheet (hoặc nhấn phím F4). Trong hộp thoại Property Sheet hiện ra, chọn tab Data, tại mục Record Resource, chọn nguồn cho biểu mẫu (bảng hoặc truy vấn).

Property Sheet	
Selection type: Form	
Form	
Format Data Event Other All	
Record Source	Form
Recordset Type	Dynaset
Fetch Defaults	Yes
Filter	
Filter On Load	No
Order By	
Order By On Load	Yes
Wait for Post Processing	No
Data Entry	No
Allow Additions	Yes
Allow Deletions	Yes
Allow Edits	Yes
Allow Filters	Yes
Record Locks	No Locks

Trên menu tab Design, chọn Add Existing Fields. Xuất hiện hộp thoại Field List, lần lượt kéo thả các cột cần hiển thị đưa vào form tại những vị trí thích hợp.

Form1

Form Header

Form1

Detail

MASV

HOSV

Form Footer

Field List

Show only fields in the current record source

Fields available for this view:

- SINHVIEN
 - MASV
 - HOSV
 - TENSV
 - PHAI
 - NGAYSINH
 - NOISINH

Fields available in related tables:

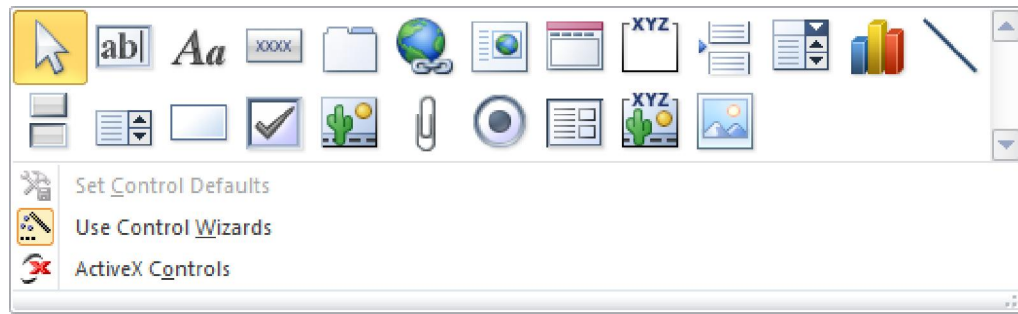
- KETQUA
- KHOA

Lưu Form: Nhấn Ctrl + S, nhập tên cần lưu cho form. Chọn OK.

3. Form Control

Để chèn các điều khiển (Control) vào biểu mẫu, chúng ta thực hiện:

- Mở biểu mẫu ở chế độ thiết kế (Design View).
- Tại menu tab Design, chọn điều khiển cần chèn vào biểu mẫu trong nhóm Control.
- Đưa con trỏ chuột vào vùng thiết kế tại vị trí thích hợp, kéo chuột để tạo điều khiển với kích thước vừa ý và thả chuột.



Các loại Form Control:

Control	Tên	Ý nghĩa
	Text Box	Có 2 loại Textbox: - Bound control: chứa nội dung của trường. - Unbound control: không có dữ liệu nguồn, thường dùng để nhập công thức tính toán.
	Label	Dùng để tạo nhãn, ghi chú ký tự hoặc tạo hyperlink
	Button	Nút lệnh dùng để thực hiện lệnh khi nhấp chuột
	Tab	Hiển thị dữ liệu thành từng nhóm trên nhiều tab khác nhau.
	Hyperlink	Tạo hyperlink
	Web Browser	Cửa sổ trình duyệt
	Navigation	Tab dùng để hiển thị Form hoặc Report trong CSDL.
	Option group	Nhóm các tùy chọn
	Page Break	Ngắt trang
	Combo Box	Là một Drop-down menu cho phép chọn một tùy chọn trong danh sách hoặc nhập thêm tùy chọn mới
	Chart	Tạo một đồ thị bằng wizard
	Line	Vẽ đường thẳng
	Toggle button	Nút có hai trạng thái: On/Off
	List box	Là một Drop-down menu cho phép chọn một tùy chọn trong danh sách nhưng không được nhập thêm giá trị mới

	Rectangle	Vẽ hình chữ nhật
	Check box	Hộp chọn, có hai trạng thái: check và uncheck
	Unbound object frame	Nhúng các đối tượng từ các phần mềm khác như: graph, picture... (không được lưu trữ trong các trường)
	Attachment	Sử dụng cho những trường có kiểu Attachment
	Option button	Là một thành phần của option group
	Subform / Subreport	Dùng để tạo SubForm hoặc SubReport
	Bound object frame	Nhúng các đối tượng từ các phần mềm khác như: graph, picture... (được lưu trữ trong các trường)
	Image	Hình ảnh loại Bitmap (*.bmp)

4. Các điều khiển thường dùng và cách sử dụng

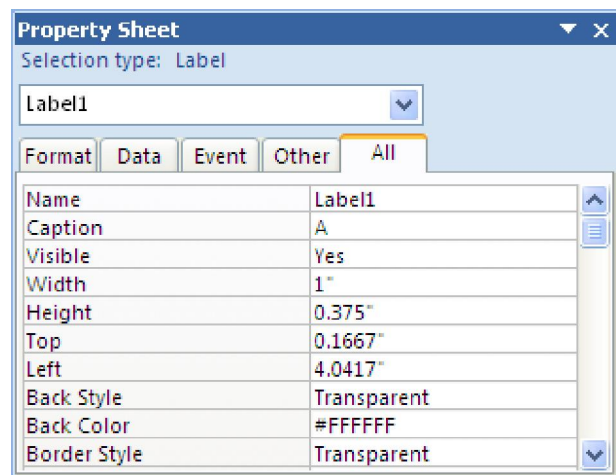
4.1. Nhãn (Label)

- Nhấp chuột chọn nút Label trong nhóm công cụ Controls trên menu tab Design.
- Đưa con trỏ chuột đến vị trí cần tạo nhãn, nhấn giữ nút trái và kéo để thực hiện chèn nhãn lên form.
- Nhập văn bản theo yêu cầu, nhấn Enter.

Các thuộc tính cơ bản của Label:

Nhấp phải chuột tại đối tượng, chọn Properties, xuất hiện hộp thoại Property Sheet. Chọn các thuộc tính:

- Name: Tên của đối tượng.
- Caption: Dòng ký tự xuất hiện trên nhãn.
- Visible: Cho phép hiện nhãn (Yes), hay không hiện (No).
- Display when
 - + Always: Luôn luôn hiện
 - + Print Only: Chỉ hiện khi in ra giấy.
 - + Screen Only: Chỉ hiện khi xem Form.



4.2. Hộp văn bản (Text Box)

Việc chèn Text Box vào form tương tự như chèn Label. Có 3 loại Text Box:

- Unbound: không gắn kết (không có nguồn dữ liệu khai báo ở Control Source)

- Bound: bị gắn kết (có nguồn dữ liệu khai báo ở Control Source)
- Calculated Control: giá trị của nó được tính toán qua biểu thức.

Để gắn kết dữ liệu Text Box vào cột của bảng, ta thực hiện:

- Nhấp chuột phải Text Box cần gắn kết dữ liệu, chọn lệnh Properties.
- Trong hộp thoại Property Sheet, chọn tab Data, chọn cột dữ liệu cần gắn kết trong thuộc tính Control Source.

Để thiết lập biểu thức tính toán, ta nhập theo cú pháp sau ở thuộc tính Control Source:

= Biểu thức

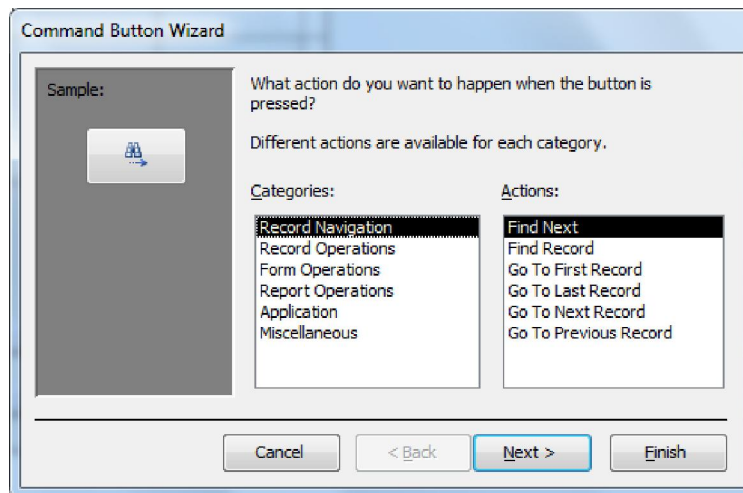
Ví dụ: Ta có hai cột dữ liệu trong bảng đã được kéo vào Form là Đơn giá, Số lượng ứng với hai Text Box có tên **TxtDonGia** và **TxtSoLuong**. Để tính Thành tiền bằng một Text Box ta phải nhập công thức sau trong thuộc tính Control Source:

= [TxtDonGia]*[TxtSoLuong]

4.3. Nút lệnh (Button)

Nút lệnh là điều khiển thường dùng, bên trong nó chứa các hành động. Ta có thể tạo Button bằng trình thông minh theo các bước:

- Tại menu tab Design, trong nhóm lệnh Controls chọn biểu tượng Use Control Wizard.
- Chọn biểu tượng Button trong nhóm lệnh Controls.
- Vẽ điều khiển tại vị trí thích hợp trên form. Khi thả chuột sẽ xuất hiện hộp thoại Command Button Wizard:



- Làm theo hướng dẫn theo trình tự. Trong đó:
 - + Categories: liệt kê nhóm hành động.
 - + Actions: liệt kê các hành động theo nhóm được chọn.

Dưới đây là bảng liệt kê các hành động trong các mục khác:

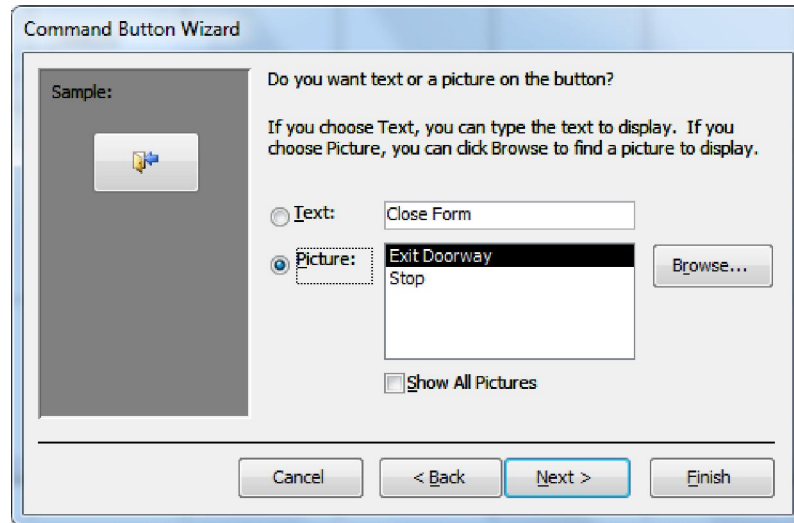
Nhóm hành động (Categories)	Hành động (Action)
Record Navigation: Di chuyển con trỏ tới bản ghi, tìm kiếm các bản ghi	Find Record: tìm kiếm các bản ghi thỏa điều kiện. Find Next: tìm bản ghi kế tiếp thỏa điều kiện.

	Go to First Record: di chuyển tới bản ghi đầu tiên. Go to Last Record: di chuyển tới bản ghi cuối cùng. Go to Next Record: di chuyển tới bản ghi kế tiếp. Go to Previous Record: di chuyển tới bản ghi trước đó.
Record Operations: Các hành động xử lý bản ghi	Add New Record: thêm bản ghi mới. Delete Record: xóa bản ghi hiện hành. Save Record: lưu bản ghi hiện hành. Duplicate Record: sao chép giá trị bản ghi hiện hành sang một dòng mới. Print Record: in bản ghi hiện hành. Undo Record: phục hồi giá trị trước khi sửa bản ghi hiện hành.
Form Operations: Các hành động xử lý Form	Edit Form Filter: sửa bộ lọc dữ liệu. Apply Form Filter: lọc dữ liệu cho Form. Close Form: đóng Form hiện hành. Open Form: mở một Form khác. Print Form: in Form bất kỳ. Print Current Form: in Form hiện hành. Refresh Form Data: cập nhật lại dữ liệu trên Form.
Report Operations: Các hành động xử lý Report	Preview Report: xem trước Report. Print Report: in nội dung Report ra máy in. Mail Report: gửi report sang người sử dụng khác bằng Email. Send Report to File: in nội dung Report ra tập tin.
Application: Các hành động trên ứng dụng	Quit Application: thoát chương trình Access.
Miscellaneous: Các hành động khác	Auto Dialer: tự động gọi. Print Table: in dữ liệu một Table được chỉ định. Run Macro: thực hiện một Macro được chỉ định. Run Query: thực hiện một Query được chỉ định.

Ví dụ: Muốn tạo nút dùng để đóng Form hiện hành, ta chọn loại hành động Form Operations trong khung Categories rồi chọn hành động Close Form trong khung Action.

- Nhấn Next để sang bước tiếp theo. Ta có các tùy chọn sau:
 - ✓ Text: nhập chuỗi kí tự hiển thị trên nút lệnh.
 - ✓ Picture: chọn biểu tượng (hình ảnh) hiển thị trên nút lệnh

- ✓ Show All Pictures: liệt kê các biểu tượng có sẵn trong Access
- ✓ Browse: chọn biểu tượng từ hình ảnh khác.



- Nhấn Next để sang bước tiếp theo. Nhập vào tên cho nút lệnh, chọn Finish để kết thúc.

4.4. Nhóm chức năng lựa chọn (Option Group)

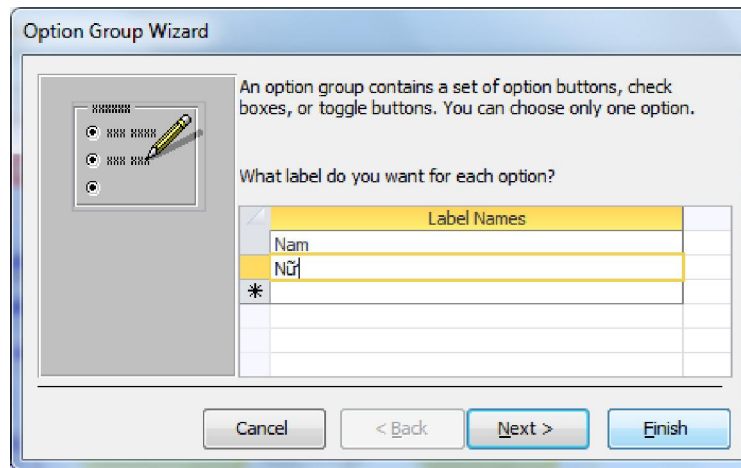
Nhóm chức năng lựa chọn trên biểu mẫu sẽ giúp ta xác định chọn một trong các giá trị cho một trường dữ liệu nào đó, như ví dụ chọn giá trị Nam/Nữ trong biểu mẫu dưới đây:

DANH SÁCH SINH VIÊN

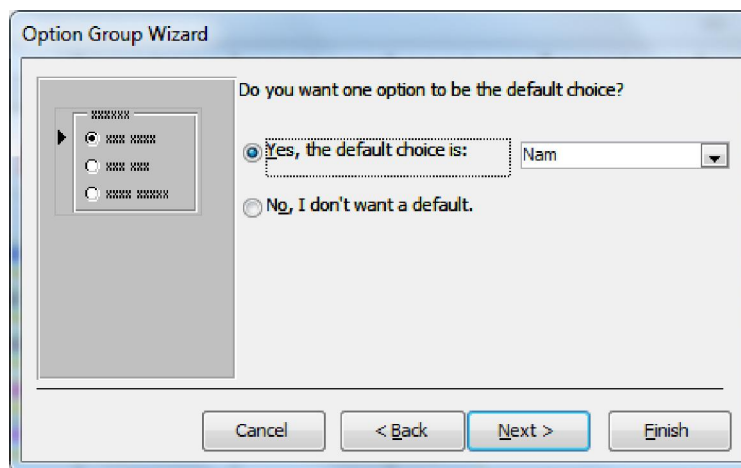
Mã sinh viên	A01	Phái	☐ Nam ☒ Nữ	Option Group
Họ sinh viên	Nguyễn Thu			
Tên sinh viên	Hải			
Nơi sinh:	Sài Gòn	Mã khoa:	AV	
Ngày sinh:	25/02/1980	Học bổng	100000	

Các bước thực hiện:

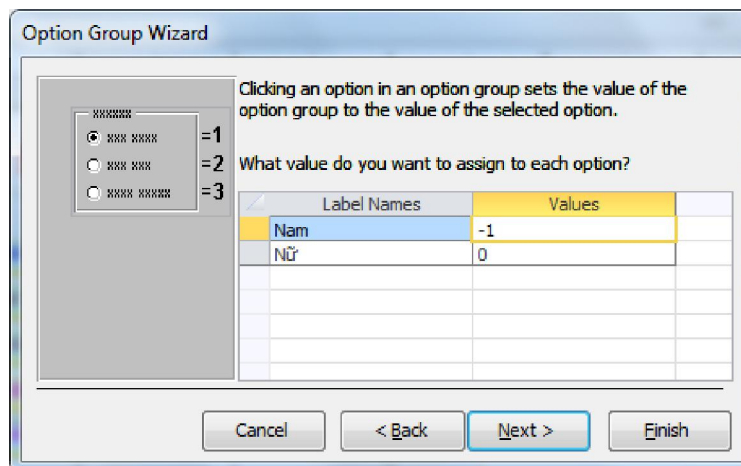
- Tại menu tab Design, trong nhóm lệnh Controls chọn biểu tượng Use Control Wizard.
- Chọn biểu tượng Option Group trong nhóm lệnh Controls.
- Vẽ điều khiển tại vị trí thích hợp trên form.
- Xuất hiện hộp thoại Option Group Wizard. Nhập vào chuỗi chứa tên chức năng chọn lựa cần tạo. Mỗi chức năng được nhập trên một dòng. Chọn Next.



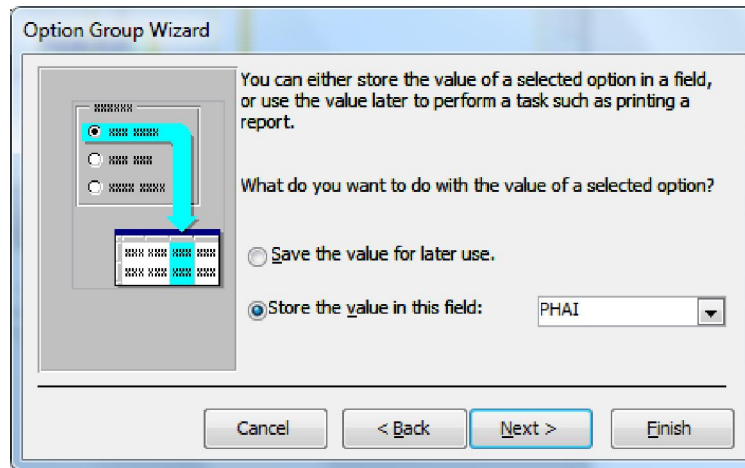
- Chọn giá trị mặc định cho nhóm (là một trong các chức năng của nhóm) khi hiển thị.



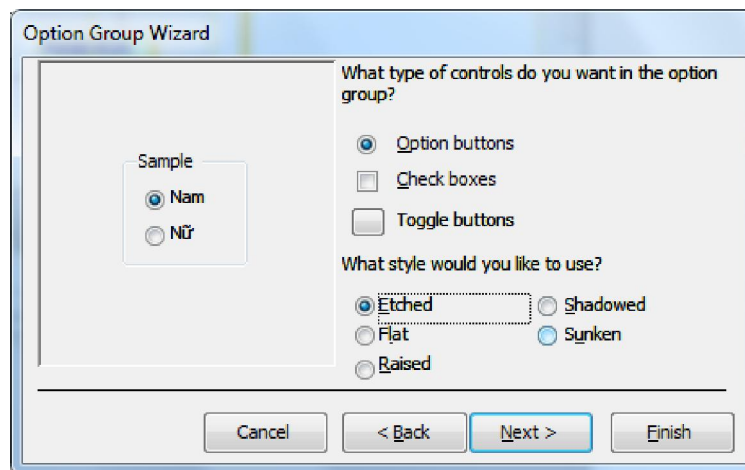
- Chọn giá trị tương ứng cho từng chức năng trong nhóm. Thường thì chúng ta chọn theo mặc định của Access (mỗi giá trị gán cho chức năng tương ứng là duy nhất).



- Chọn tên trường dùng để lưu trữ giá trị của chức năng được chọn.



- Chọn kiểu hiển thị của chức năng cho nhóm.



- Nhập tiêu đề cho nhóm chức năng đang tạo, chọn Finish để kết thúc.

4.5. Hộp kéo thả (Combo Box)

Combo Box trên biểu mẫu sẽ giúp ta xác định chọn một trong danh sách các giá trị cho một trường dữ liệu nào đó, như ví dụ chọn giá trị Mã khoa trong biểu mẫu dưới đây:

DANH SÁCH SINH VIÊN

Mã sinh viên	A01	Phái	☐ Nam
Họ sinh viên	Nguyễn Thu	☒ Nữ	
Tên sinh viên	Hải		
Nơi sinh:	Sài Gòn	Mã khoa	
Ngày sinh:	25/02/1980	Học bổng	

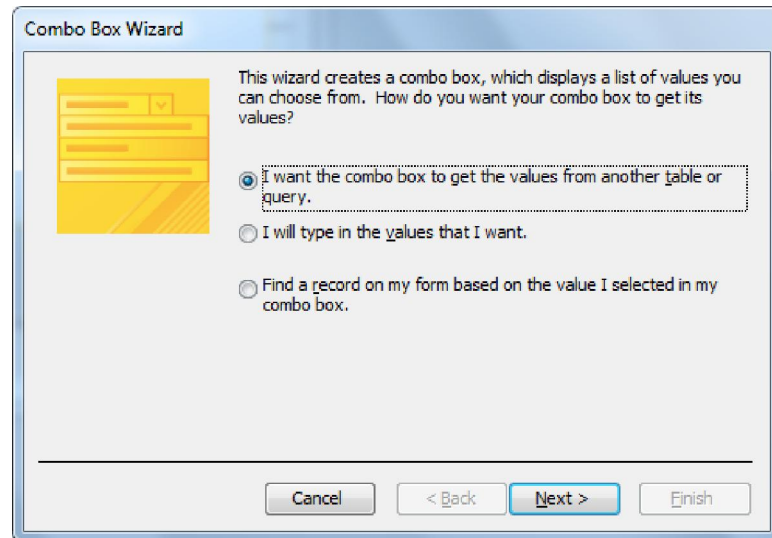
Combo Box

AV	Anh văn
LS	Lịch sử
SH	Sinh học
TH	Tin học
TR	Triết
VL	Vật lý

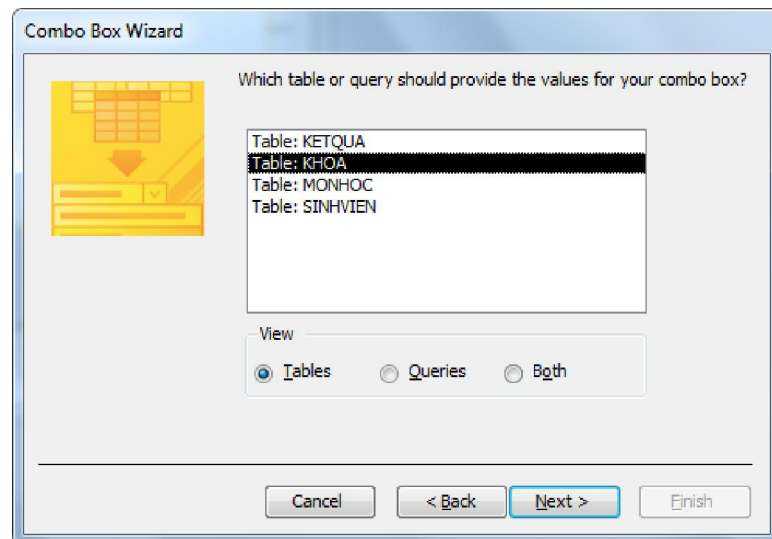
Các bước thực hiện:

- Tại menu tab Design, trong nhóm lệnh Controls chọn biểu tượng Use Control Wizard.
- Chọn biểu tượng Combo Box trong nhóm lệnh Controls.

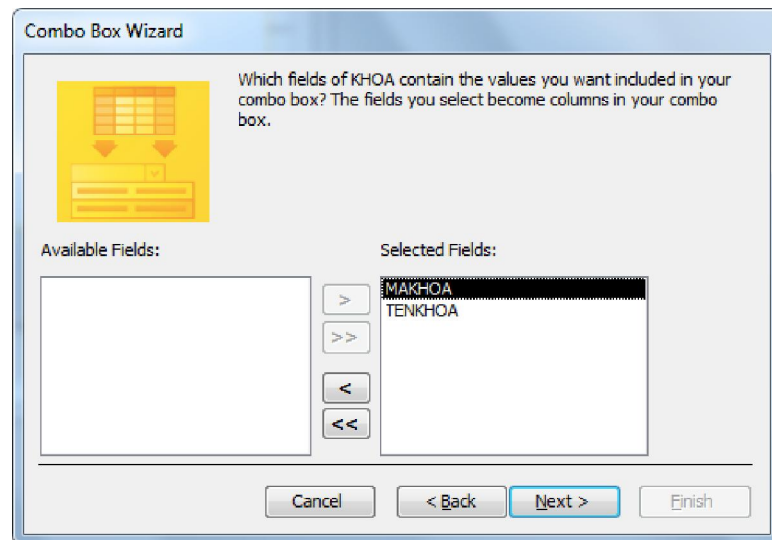
- Vẽ điều khiển tại vị trí thích hợp trên form.
- Xuất hiện hộp thoại Combo Box Wizard. Chọn loại dữ liệu cho combo box, gồm:
 - + I want the Combo box to lookup the valuse in a table or query: lấy giá trị cho Combo Box từ các cột của bảng hoặc truy vấn bất kỳ.
 - + I will type in the value that I want: tự nhập giá trị vào cho Combo Box.
 - + Find a record on my form...: tìm các bản ghi trên Form trùng với giá trị chọn trong Combo Box.



- Chọn tên bảng hoặc truy vấn làm nguồn cho combo Box.



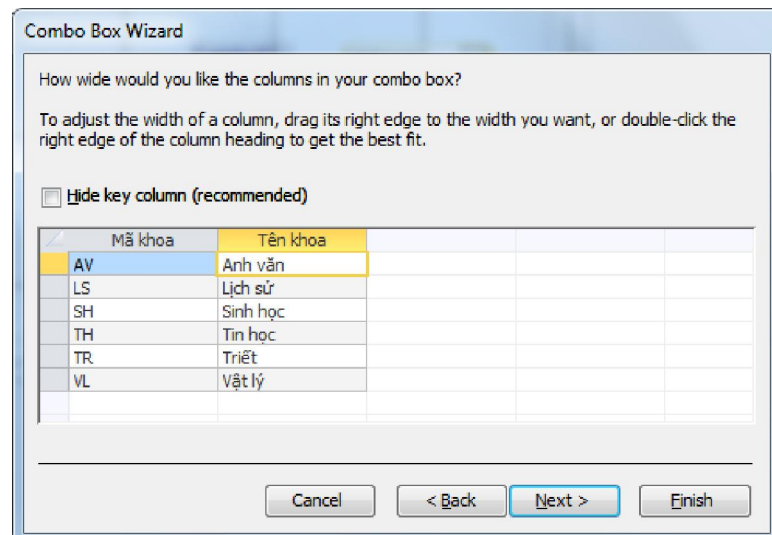
- Chọn các cột trong bảng hoặc truy vấn muốn hiển thị trong Combo Box (chọn các nút >, hoặc >>)



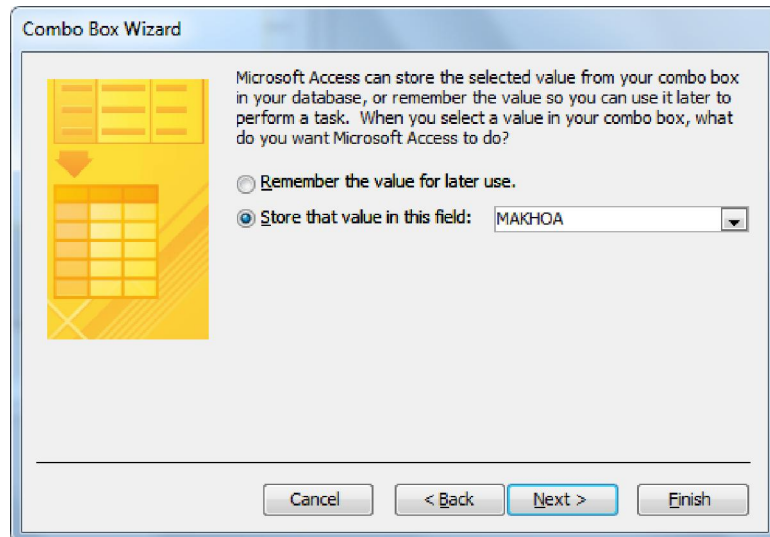
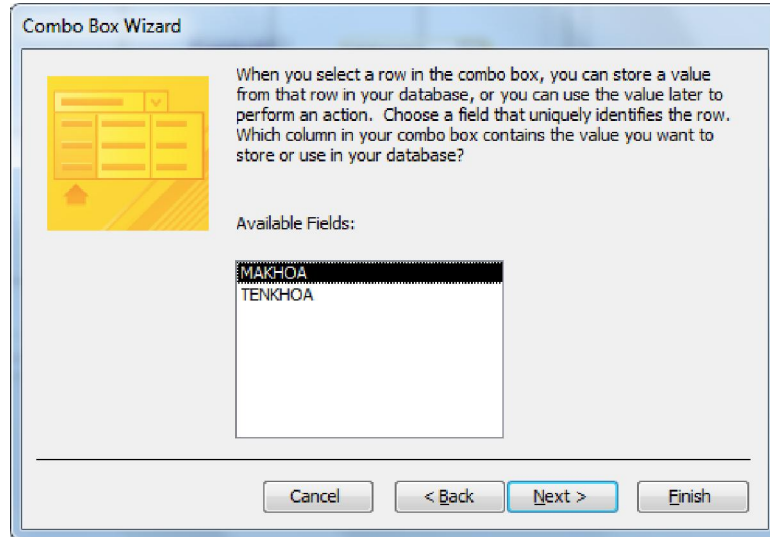
- Chọn cột dữ liệu làm khóa sắp xếp cho Combo box.



- Thay đổi độ rộng của các cột hiển thị trong combo box bằng cách đưa chuột tới đường biên của cột cần thay đổi, xuất hiện mũi tên hai đầu và kéo chuột theo chiều mũi tên.



- Chọn tên trường nhận giá trị sau khi chọn.



- Thực hiện các bước tiếp theo theo sự hướng dẫn của Access. Nhập tiêu đề cho hộp kéo xuống. Chọn Finish để kết thúc.

CHƯƠNG 9 – BIỂU BÁO CÁO

(Tổng số: 9 tiết, Lý thuyết: 4 tiết, Thực hành: 5 tiết)

§ 1. KHÁI NIỆM

Báo cáo (Report) là phương thức hữu hiệu giúp người sử dụng trình bày dữ liệu dưới dạng đầy đủ và dễ hiểu, nhanh chóng, đẹp mắt để khi in ấn. Người sử dụng có thể tích hợp trong báo cáo các dạng thức trình bày dữ liệu khác nhau như hình ảnh, biểu đồ, văn bản... Báo cáo được xây dựng trên một nguồn dữ liệu đó là bảng hoặc truy vấn, một câu lệnh SQL hoặc một dạng biểu mẫu nào đó.

1. Các dạng mẫu của báo cáo

Báo cáo dạng cột (Columnar): báo cáo dạng này sẽ được trình bày theo dạng một cột và kèm theo phần nhãn của mỗi cột dữ liệu bên trái, mỗi dòng tương ứng với một trường dữ liệu

DANH SÁCH SINH VIÊN

Mã sinh viên	A01
Họ sinh viên	Nguyễn Thu
Tên sinh viên	Hải
Phái	☒
Ngày sinh	25/02/1980
Mã khoa	AV

Mã sinh viên	A02
Họ sinh viên	Trần Văn
Tên sinh viên	Chính
Phái	☒
Ngày sinh	24/12/1982
Mã khoa	TH

Báo cáo dạng hàng (Tabular): Báo cáo sẽ trình bày dữ liệu theo dạng bảng bao gồm nhiều hàng và nhiều cột.

DANH SÁCH SINH VIÊN

Mã sinh viên	Họ sinh viên	Tên sinh viên	Phái	Ngày sinh	Mã khoa
A01	Nguyễn Thu	Hải	☒	25/02/1980	AV
A02	Trần Văn	Chính	☒	24/12/1982	TH
A03	Lê Thu Bạch	Yến	☐	21/02/1982	AV
A04	Trần Anh	Tuấn	☒	08/12/1984	LS
A05	Trần Thanh	Triều	☒	01/02/1980	VL
B01	Trần Thanh	Mai	☐	20/12/1981	TH
B02	Trần Thị Thu	Thủy	☐	13/02/1982	TH
B03	Trần Thị	Thanh	☐	31/12/1982	TH

Báo cáo tổng hợp (Group/Total): Báo cáo dạng này sẽ tổ chức dữ liệu thành các nhóm, mỗi nhóm sẽ trình bày dữ liệu theo dạng Tabular. Người sử dụng có thể nhóm dữ liệu theo cấp và có thể tính toán giá trị tổng cho mỗi nhóm và một giá trị tính tổng cho toàn bộ các nhóm. Ví dụ: Báo cáo thu nhập của cán bộ theo từng phòng ban (phân nhóm theo phòng ban).

Báo cáo danh sách sinh viên theo Khoa

Mã khoa	Mã sinh viên	Họ sinh viên	Tên sinh viên	Phái	Ngày sinh	Nơi sinh	Học bổng
AV							
	A01	Nguyễn Thu	Hải	☒	25/02/1980	Đồng Nai	210,000
	A03	Lê Thu Bạch	Yến	☐	21/02/1982	Lâm Đồng	210,000
Tổng học bổng							420,000
LS							
	A04	Trần Anh	Tuấn	☒	08/12/1984	Long An	210,000
Tổng học bổng							210,000

2. Các chế độ hiển thị của báo cáo

Báo cáo có thể được trình bày theo 3 chế độ sau:

Report Design: Chế độ thiết kế báo cáo.

Layout Preview: Chế độ trình bày dữ liệu trong báo cáo.

Print Preview: Chế độ xem hình thức báo cáo trước khi in ấn.

§ 2. TẠO VÀ HIỆU CHỈNH BÁO CÁO

I. THIẾT KẾ BÁO CÁO

1. Các thành phần chính trong báo cáo

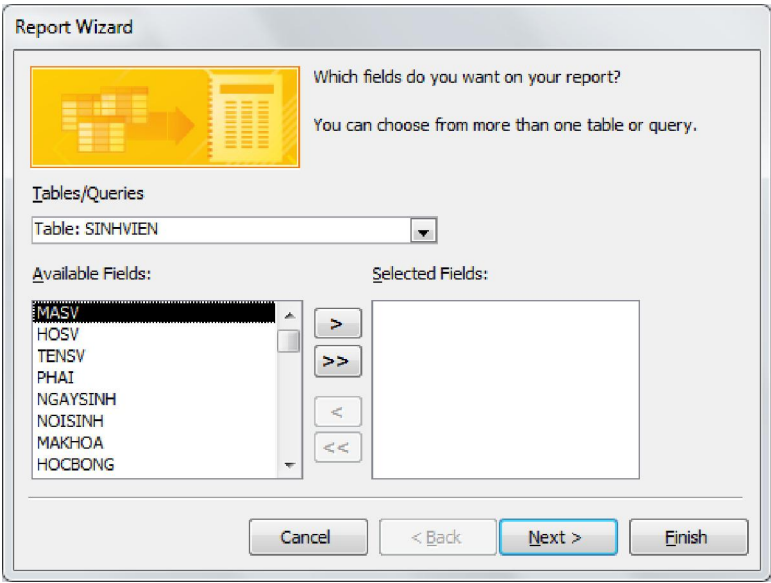
Trong báo cáo thường có các thành phần sau:

- Detail: phần chi tiết, hiển thị phần nội dung chính của báo cáo
- Page Header: tiêu đề đầu trang
- Page Footer: tiêu đề cuối trang như số trang, ngày, giờ,...
- Group Header: tiêu đề đầu nhóm, chứa tiêu đề của nhóm.
- Group Footer: tiêu đề cuối nhóm, chứa các công thức tính toán theo từng nhóm riêng.
- Report Header: tiêu đề đầu báo cáo.
- Report Footer: tiêu đề cuối báo cáo.

2. Thiết kế báo cáo bằng Report Wizard

Ví dụ: Thiết kế báo cáo danh sách sinh viên theo từng khoa, tính tổng học bổng sinh viên trong từng khoa.

- Chọn tab Create / Report Wizard
- Chọn nguồn dữ liệu cho báo cáo (Table hoặc Query).



Report Wizard

Which fields do you want on your report?
You can choose from more than one table or query.

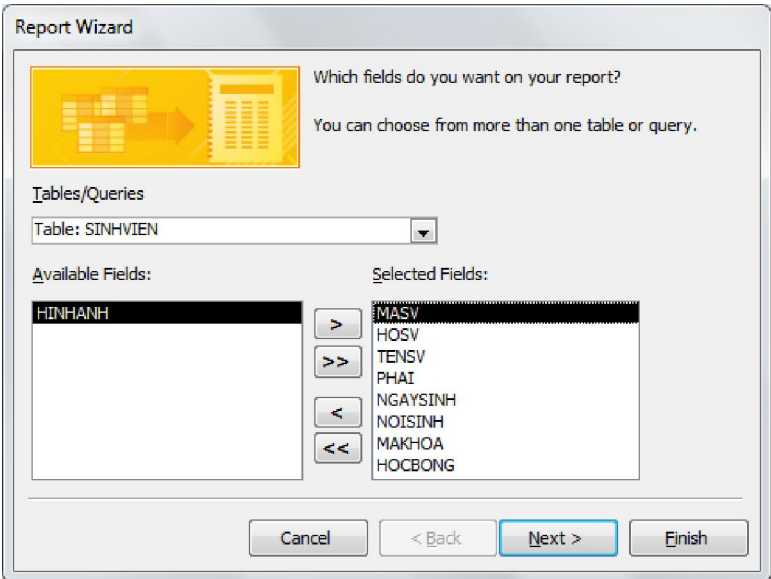
Tables/Queries
Table: SINHVIEN

Available Fields:
MASV
HOSV
TENSU
PHAI
NGAYSINH
NOISINH
MAKHOA
HOCBONG

Selected Fields:

Cancel < Back Next > Finish

- Chọn tên các trường muốn hiển thị trong báo cáo.



Report Wizard

Which fields do you want on your report?
You can choose from more than one table or query.

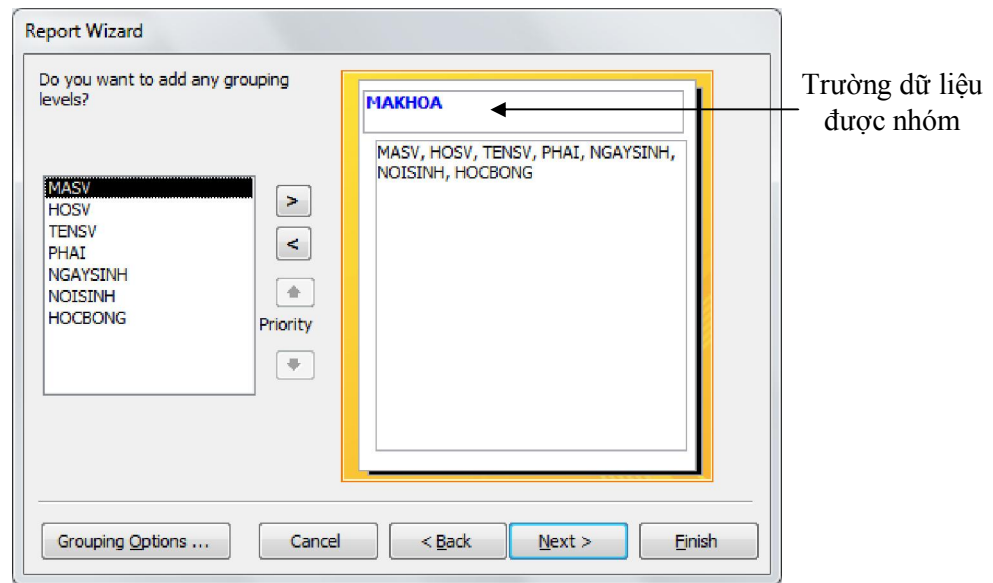
Tables/Queries
Table: SINHVIEN

Available Fields:
HINHANH

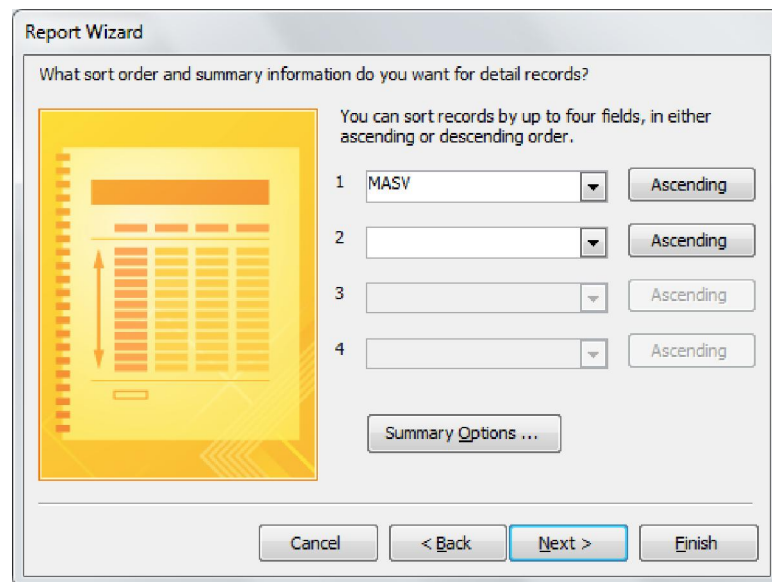
Selected Fields:
MASV
HOSV
TENSU
PHAI
NGAYSINH
NOISINH
MAKHOA
HOCBONG

Cancel < Back Next > Finish

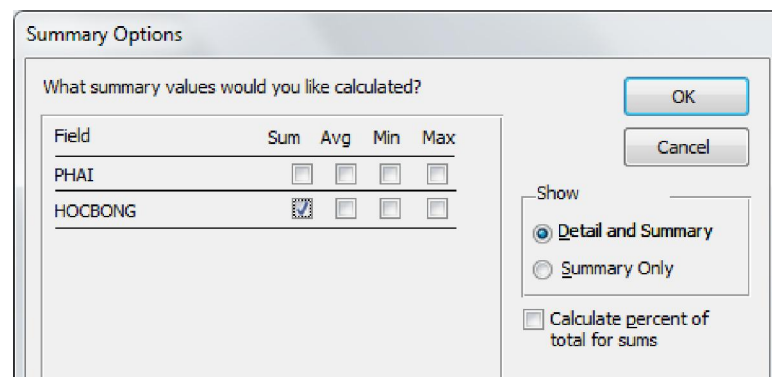
- Chọn tên trường muốn nhóm dữ liệu (grouping levels), có thể chọn nhiều trường nếu muốn nhóm nhiều cấp. Theo như ví dụ trên là báo cáo danh sách sinh viên theo từng khoa, nên trường để phân nhóm dữ liệu là Mã khoa (MAKHOA).



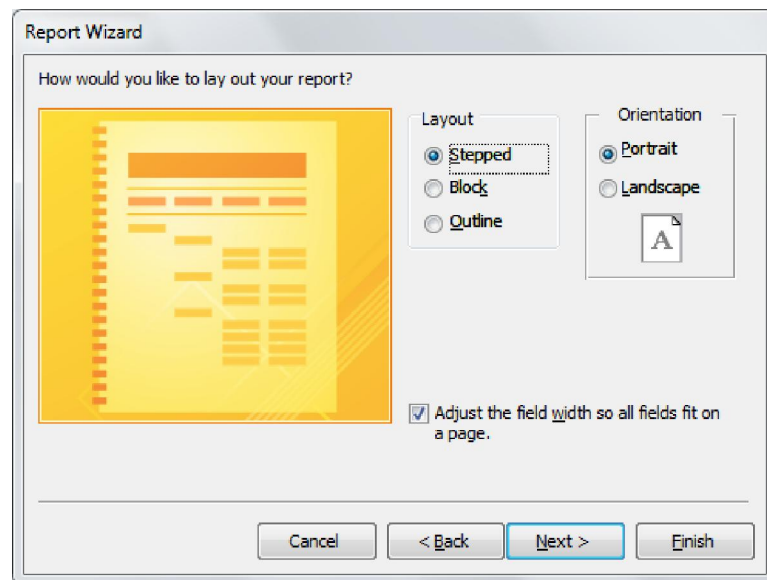
- Chọn tên trường dùng làm tiêu chí sắp xếp dữ liệu trong mỗi nhóm (nếu cần).



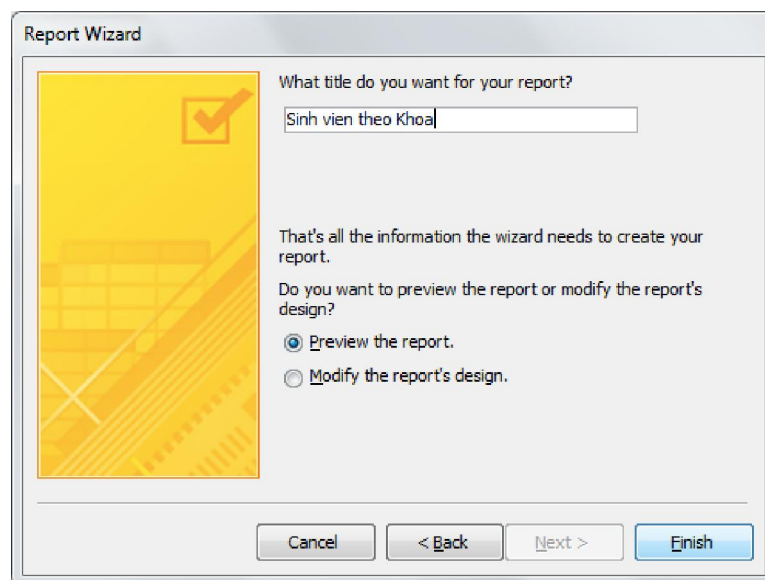
- Chọn Summary Options nếu muốn tổng hợp số liệu theo nhóm. Theo như ví dụ trên là báo cáo bao gồm tổng hợp số liệu học bổng sinh viên trong từng khoa nên ta chọn phép toán Sum đối với trường học bổng (HOCBONG). Chọn OK.



- Chọn hướng giấy in (Orientation) và chế độ nhìn (Layout) cho Report. Chọn chế độ Layout cho ví dụ này là Stepped.



- Nhập vào tiêu đề cho báo cáo. Chọn Finish để kết thúc.



- Kết quả:

Báo cáo danh sách sinh viên theo Khoa							
Mã khoa	Mã sinh viên	Họ sinh viên	Tên sinh viên	Phái	Ngày sinh	Nơi sinh	Học bổng
AV							
	A01	Nguyễn Thu	Hải	☒	25/02/1980	Đồng Nai	210,000
	A03	Lê Thu Bạch	Yến	☐	21/02/1982	Lâm Đồng	210,000
Tổng học bổng							420,000
LS							
	A04	Trần Anh	Tuấn	☒	08/12/1984	Long An	210,000
	Tổng học bổng						210,000

3. Chỉnh sửa báo cáo

Nhấp chuột phải tại Report cần chỉnh sửa, chọn Design View.

Report Header							
Báo cáo danh sách sinh viên theo Khoa							
Page Header							
Mã khoa	Mã sinh viên	Họ sinh viên	Tên sinh viên	Phái	Ngày sinh	Nơi sinh	Học bổng
MAKHOA Header							
MAKHOA							
Detail							
MASV		HOSV	TENSV	☒	NGAYSINH	NOISINH	HOCBONG
MAKHOA Footer							
					Tổng học bổng	=Sum([HOCBONG])	
Page Footer							
Report Footer							

Ta có giao diện thiết kế báo cáo theo ví dụ gồm các thành phần sau:

- Report header: tiêu đề đầu báo cáo.
- Page header: tiêu đề đầu trang, nếu báo cáo chia thành nhiều trang thì đây là nội dung sẽ được lặp lại đầu mỗi trang.
- MAKHOA Header (Group header): tên trường phân nhóm dữ liệu, tiêu đề đầu nhóm.
- Detail: phần chi tiết, hiển thị phần nội dung chính của báo cáo.
- MAKHOA Footer (Group Footer): tiêu đề cuối nhóm, chứa các công thức tính toán theo từng nhóm riêng.
- Page Footer: tiêu đề cuối trang, thường được ghi số trang, ngày, giờ,... Nếu báo cáo chia thành nhiều trang thì đây là nội dung sẽ được lặp lại cuối mỗi trang.
- Report Footer: tiêu đề cuối báo cáo.

4. Tính toán trên nhóm

Một số hàm thường sử dụng trong thiết kế báo cáo:

[Page]: chèn số trang hiện hành của báo cáo.

[Pages]: chèn tổng số trang của báo cáo.

Date(): chèn ngày tháng năm hiện hành của hệ thống.

Now(): chèn ngày tháng năm, giờ hiện hành của hệ thống.

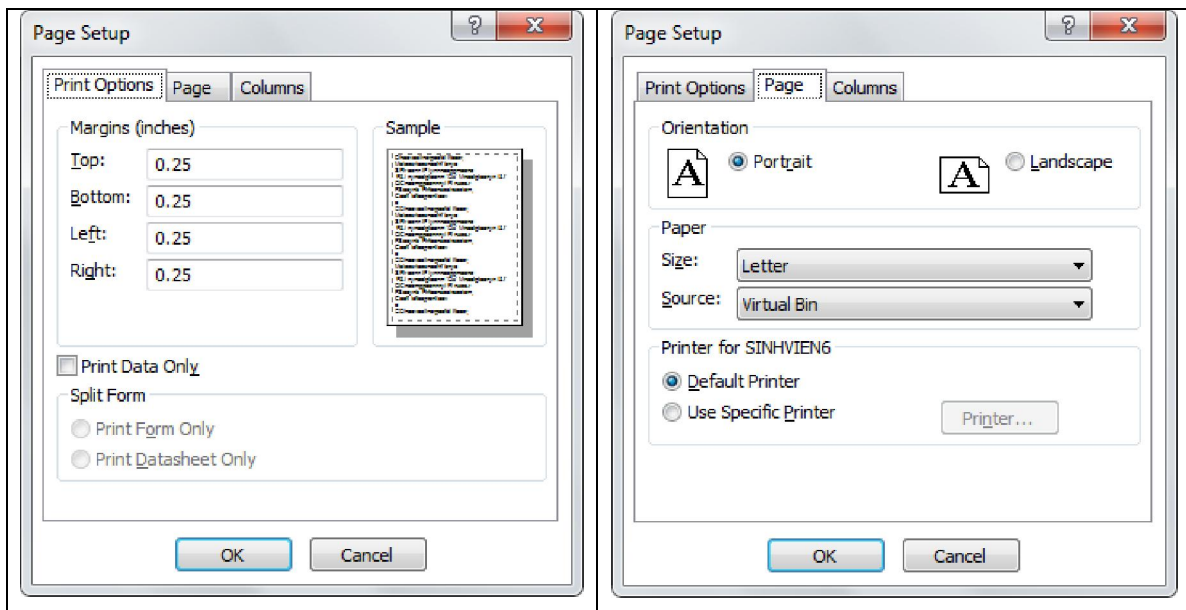
Các hàm thống kê, tổng cộng dữ liệu trên nhóm: Count, Sum, Min, Max, Avg

III. XUẤT DỮ LIỆU

1. Thiết lập trang in

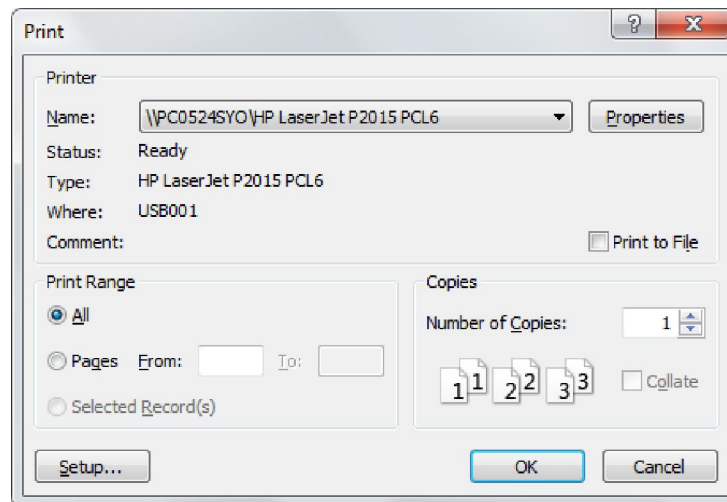
- Nhấp chuột phải tại Report cần định dạng trang in, chọn Design View.
- Chọn menu tab Page Setup, chọn biểu tượng Page Setup.

Chọn tab Print Options để canh lề cho trang in:	Chọn tab Page để chọn khổ giấy (size) và hướng giấy:
--	---



2. In dữ liệu

- Chọn Report cần in, chọn menu File / Print / Print.
- Xuất hiện hộp thoại Print như sau:



- Name: Chọn máy in.
 - Print range: All - In tất cả; Page from - chỉ định các trang cần in.
- Chọn OK để in.

BÀI TẬP THỰC HÀNH

Bài tập cơ sở dữ liệu Quản lý sinh viên:

A. Tạo bảng dữ liệu (Table Design)

Tạo cấu trúc các bảng như mô tả với các trường được gạch chân là khóa chính của bảng.

Bảng KHOA

Field Name	Data Type	Field Size	Format	Caption
<u>MAKHOA</u>	Text	2		Mã khoa
TENKHOA	Text	30		Tên khoa

Bảng MONHOC

Field Name	Data Type	Field Size	Format	Caption
<u>MAMH</u>	Text	2		Mã môn học
TENMH	Text	25		Tên môn học
SOTC	Number	Byte		Số tín chỉ

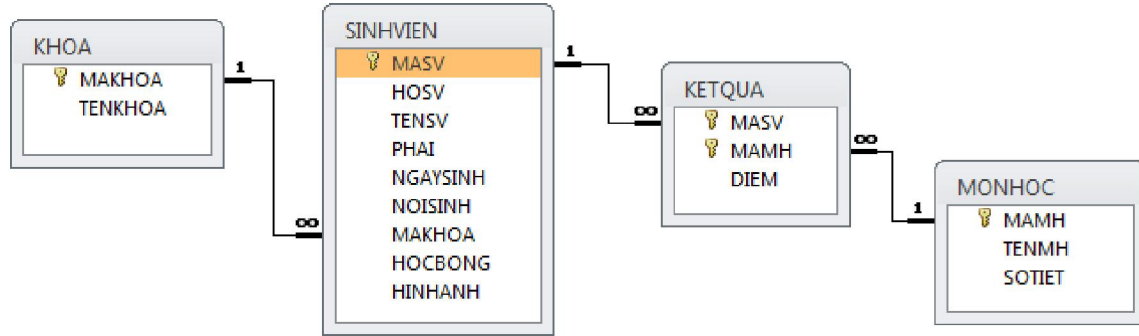
Bảng SINHVIEN

Field Name	Data Type	Field Size	Format	Caption	Description
<u>MASV</u>	Text	3		Mã sinh viên	
HOSV	Text	15		Họ sinh viên	
TENSV	Text	7		Tên sinh viên	
PHAI	Yes/No			Phái	Phái:Yes (Nam); No (Nữ)
NGAYSINH	Date/Time		Short Date	Ngày sinh	
NOISINH	Text	15		Nơi sinh	
MAKHOA	Text	2		Mã khoa	
HOCBONG	Number	Double		Học bổng	

Bảng KETQUA

Field Name	Data Type	Field Size	Format	Caption
<u>MASV</u>	Text	3		Mã sinh viên
<u>MAMH</u>	Text	2		Mã môn học
DIEM	Number	Single		Điểm

B. Thiết lập mối quan hệ (Relationships)



C. Ràng buộc dữ liệu (Input Mask / Validation)

- Mã sinh viên bắt buộc có 03 ký tự, ký tự đầu là chữ cái, 2 ký tự sau là chữ số.
- Mã khoa bắt buộc có 02 ký tự chữ cái. Trường dữ liệu Mã khoa của sinh viên không được để trống.
- Ngày sinh không được phép nhập sau ngày hiện tại.
- Học bổng không được phép nhập số âm. Giá trị mặc định của Học bổng là 0.
- Điểm thi phải là số trong phạm vi từ 0 đến 10.
- Số tín chỉ của môn học phải từ 2 trở lên.

D. Nhập dữ liệu cho các bảng

Bảng SINHVIEN

MASV	HOSV	TEN SV	PHAI	NGAY SINH	NOISINH	MA KHOA	HOC BONG
A01	Nguyễn Thu	Hải	No	23/02/1995	Đồng Nai	AV	100000
A02	Trần Văn	Chính	Yes	24/12/1997	Đồng Nai	TH	100000
A03	Lê Thu Bạch	Yến	No	21/02/1996	Lâm Đồng	AV	140000
A04	Trần Anh	Tuấn	Yes	08/12/1998	Long An	LS	80000
A05	Trần Thanh	Triều	Yes	01/02/1995	Lâm Đồng	VL	80000
B01	Trần Thanh	Mai	No	20/12/1996	Bến Tre	TH	200000
B02	Trần Thị Thu	Thủy	No	13/02/1997	Bình Thuận	TH	30000
B03	Trần Thị	Thanh	No	31/12/1996	Bình Thuận	TH	50000

Bảng KETQUA

MASV	MAMH	DIEM
A01	01	10
A01	02	4

Bảng MONHOC

MAMH	TENMH	SOTC
01	Nhập môn máy tính	3
02	Trí tuệ nhân tạo	4

A01	05	9
A01	06	3
A02	01	5
A03	02	5
A03	04	10
A03	06	1
A04	02	4
A04	04	6
B01	01	0
B01	04	8
B02	03	6
B02	04	8
B03	02	10
B03	03	9

03	Truyền tin	4
04	Đồ họa	5
05	Văn phạm	4
06	Đàm thoại	2
07	Vật lý nguyên tử	3

Bảng KHOA

MAKHOA	TENKHOA
AV	Anh văn
LS	Lịch sử
TH	Tin học
TR	Triết
VL	Vật lý
SH	Sinh học

E. Sắp xếp, trích lọc dữ liệu trong bảng (Sort / Filter)

1. Trong bảng SINHVIEN, sắp xếp dữ liệu tăng dần theo Mã khoa, nếu cùng khoa thì sắp tăng dần theo Tên sinh viên.
2. Trong bảng SINHVIEN, lọc ra các sinh viên thuộc khoa có mã “AV”.
3. Trong bảng SINHVIEN, lọc ra các sinh viên không thuộc khoa có mã “TH”.
4. Trong bảng SINHVIEN, lọc ra các sinh viên thuộc khoa có mã “TH” và có Học bổng lớn hơn hoặc bằng 100,000.
5. Trong bảng SINHVIEN, lọc ra các sinh viên có tên bắt đầu bằng chữ “T”.
6. Trong bảng SINHVIEN, lọc ra các sinh viên có ký tự thứ 2 của phần tên là “H”.
7. Trong bảng SINHVIEN, lọc ra các sinh viên mà trong phần tên có chứa chữ “T”.
8. Trong bảng SINHVIEN, lọc ra các sinh viên có ngày sinh trong tháng 12.

F. Truy vấn (Query)

Truy vấn lựa chọn bằng công cụ QBE

1. Hiện thị tất cả các sinh viên theo thứ tự tăng dần của Tên, bao gồm các thông tin: Mã sinh viên, Họ tên sinh viên, Ngày sinh, Nơi sinh, Học bổng.
2. Liệt kê danh sách sinh viên, trong tên có chứa chữ N.

3. Lập danh sách các sinh viên có ký tự đầu của phần tên nằm trong khoảng từ A đến M. Danh sách gồm các thông tin: Mã sinh viên, Họ và tên sinh viên, Ngày sinh, Nơi sinh. Gợi ý: Left([TENS],1) Between A And M
4. Lập danh sách các sinh viên của khoa “Anh văn” và khoa “Tin học”, gồm các thông tin: Mã sinh viên, Tên khoa, Giới.
5. Liệt kê các nam sinh viên của khoa có tên “Tin học”, gồm các thông tin: Mã sinh viên, Họ tên sinh viên, Giới, Tên khoa.
6. Cho biết những sinh viên có ngày sinh từ ngày 01/01/1996 đến ngày 01/06/1996, gồm các thông tin: Mã sinh viên, Họ và tên sinh viên, Ngày sinh, Nơi sinh.
7. Lập danh sách các sinh viên sinh năm 1997. Danh sách gồm các thông tin: Mã sinh viên, Họ và tên sinh viên, Ngày sinh.
8. Lập danh sách các sinh viên có ngày tháng năm sinh từ tháng 1 đến tháng 6. Danh sách gồm các thông tin: Mã sinh viên, Họ và tên sinh viên, Ngày sinh. Gợi ý: Month([NGAYSINH]) Between 1 And 6
9. Lập danh sách sinh viên gồm các thông tin: Mã sinh viên, Họ và tên, Giới, Tuổi.
10. Lập danh sách gồm: Mã sinh viên, Họ và tên, Ngày sinh của các sinh viên có sinh nhật trong ngày hôm nay (ngày hiện hành của hệ thống).
11. Lập bảng điểm môn “Trí tuệ nhân tạo” của sinh viên khoa “Tin học”, gồm các thông tin: Họ và tên, Ngày sinh, Điểm. Danh sách được sắp xếp tăng dần theo Tên sinh viên.
12. Lập bảng điểm theo mẫu sau: Mã sinh viên, Họ và tên, Tên môn học, Điểm, Kết quả. Biết rằng Kết quả là “Đậu” nếu điểm từ 5 trở lên, ngược lại Kết quả là “Rớt”.

Truy vấn truyền tham số

1. Trong bảng SINHVIEN, xem danh sách sinh viên theo Mã khoa bằng truyền tham số.
2. Trong bảng SINHVIEN, xem danh sách sinh viên theo Tên sinh viên bằng truyền tham số. Tìm chính xác và tìm gần đúng theo Tên sinh viên.
3. Trong bảng SINHVIEN, xem danh sách sinh viên theo Ngày sinh (hoặc theo tháng, năm sinh).
4. Trong bảng SINHVIEN, xem danh sách thông tin các sinh viên theo Ngày sinh trong khoảng giá trị truyền tham số. (Xem danh sách sinh viên có ngày sinh từ [Ngày 1] đến [Ngày 2], với [Ngày 1] và [Ngày 2] được nhập từ bàn phím).
5. Xem danh sách sinh viên theo Mã sinh viên bằng truyền tham số, danh sách gồm các thông tin: Họ và tên, Giới, Ngày sinh, Nơi sinh, Tên khoa, Học bổng.
6. Xem bảng điểm sinh viên theo Mã sinh viên bằng truyền tham số, danh sách gồm các thông tin: Mã sinh viên, Họ và tên, Tên môn học, Điểm thi, Kết quả (Đậu/Rớt).

Truy vấn cập nhật (Update Query)

1. Tạo Query tăng học bổng 10% cho tất cả sinh viên.
2. Tạo Query tăng học bổng 10% cho các sinh viên thuộc khoa có tên là “Tin học”.
3. Tạo Query tăng số tín chỉ thêm 1 cho mỗi môn học.
4. Tạo Query tăng thêm 1 điểm thi cho sinh viên thi môn có tên “Văn phạm”.
5. Tạo Query tăng thêm 1 điểm thi cho sinh viên khoa “Tin học” thi môn tên “Đồ họa”.
6. Tạo Query tăng học bổng thêm 100000 cho các sinh viên đạt điểm 10 môn học có tên “Nhập môn máy tính”.

Truy vấn tính toán thống kê dữ liệu (Total / Group by)

1. Cho biết điểm thi cao nhất của môn học “Nhập môn máy tính”.
2. Cho biết trung bình điểm thi theo từng môn, gồm các thông tin: Mã môn, Tên môn, Trung bình điểm thi.
3. Danh sách số môn thi của từng sinh viên, gồm các thông tin: Họ tên sinh viên, Tên khoa, Tổng số môn thi.
4. Cho biết tổng số sinh viên mỗi khoa, gồm các thông tin: Tên khoa, Tổng số sinh viên.
5. Cho biết điểm cao nhất của mỗi sinh viên, gồm các thông tin: Họ tên, Điểm cao nhất.
6. Cho biết tổng số sinh viên nam của từng Khoa là bao nhiêu, thông tin gồm: MaKhoa, TenKhoa, TongSVNam.
7. Cho biết tổng số sinh viên nữ của từng Khoa là bao nhiêu, thông tin gồm: MaKhoa, TenKhoa, TongSVNu.
8. Cho biết độ tuổi trung bình của sinh viên trong từng Khoa là bao nhiêu thông tin gồm: MaKhoa, TenKhoa, TuoiTB. Gợi ý: Avg(Year(Date()))-Year([NGAYSINH]))
9. Hiện thị bảng điểm tổng hợp của sinh viên gồm thông tin: MaSV, HoTenSV, DiemTB, TongSoMonThi. Trong đó DiemTB là điểm trung bình của các Môn thi, TongSoMonThi là tổng số các môn thi của các sinh viên. Điểm TB lấy 2 số lẻ.
10. Cho biết Tổng Sinh viên của từng Môn học thông tin gồm: TenMH, TongSoSV.

Truy vấn sử dụng Query Wizard

1. Lập danh sách gồm Mã sinh viên, Họ và tên các sinh viên có cùng Nơi sinh với nhau.
2. Lập danh sách gồm Mã sinh viên, Họ và tên các sinh viên có cùng Tên với nhau.
3. Lập danh sách gồm Mã sinh viên, Họ và tên các sinh viên chưa thi môn học nào.
4. Lập danh sách gồm Mã khoa, Tên khoa các khoa mà chưa có sinh viên nào theo học.
5. Lập danh sách gồm Mã môn, Tên môn các môn học mà chưa có sinh viên nào thi.

Truy vấn tổng hợp / Truy vấn chéo (Crosstab Query)

1. Danh sách điểm các môn của từng sinh viên:

Họ và tên	Đàm thoại	Đồ họa	Nhập môn máy tính	Trí tuệ nhân tạo	Truyền tin	Văn phạm
Lê Thu Bạch Yến	1	10		5		
Nguyễn Thu Hải	3		10	4		9
Trần Anh Tuấn		6		4		
Trần Thanh Mai		8	0			
Trần Thị Thanh				10	9	
Trần Thị Thu Thủy		8			6	
Trần Văn Chính			5			

2. Điểm môn và trung bình điểm các môn. Làm tròn Điểm trung bình lấy 1 số lẻ.

Họ và tên	Điểm Trung bình	Đàm thoại	Đồ họa	Nhập môn máy tính	Trí tuệ nhân tạo	Truyền tin
Lê Thu Bạch Yến	5.33333333333333	1	10		5	
Nguyễn Thu Hải	6.5	3		10	4	
Trần Anh Tuấn	5		6		4	
Trần Thanh Mai	4		8	0		
Trần Thị Thanh	9.5				10	9
Trần Thị Thu Thủy	7		8			6
Trần Văn Chính	5			5		

3. Thống kê điểm các môn, số môn đậu, số môn rớt của sinh viên

Gọi ý: Số môn đậu: $\text{Sum}(\text{If}([\text{DIEM}] \geq 5, 1, 0))$

Họ và tên	Điểm Trung bình	Số môn đậu	Số môn thi lại	Đàm thoại	Đồ họa	Nhập môn máy tính
Lê Thu Bạch Yến	5.33333333333333	2	1	1	10	
Nguyễn Thu Hải	6.5	2	2	3		10
Trần Anh Tuấn	5	1	1		6	
Trần Thanh Mai	4	1	1		8	0
Trần Thị Thanh	9.5	2	0			
Trần Thị Thu Thủy	7	2	0		8	
Trần Văn Chính	5	1	0			5

G. Biểu mẫu (Form)

1. Thiết kế Form xem điểm sinh viên:

Yêu cầu:

- Tạo các nút di chuyển đến bản ghi tiếp theo, đến bản ghi trước đó.
- Tạo nút thoát Form.
- Tạo công thức xét kết quả, nếu điểm từ 5 trở lên thì kết quả là “Đậu”, ngược lại thì ghi kết quả là “Thi lại”.

2. Thiết kế Form nhập thông tin một sinh viên:

Yêu cầu: Tạo nút thoát Form, các nút di chuyển đến bản ghi tiếp theo, bản ghi trước đó...

3. Thiết kế Form xem điểm sinh viên theo dạng Form chính phụ:

QUẢN LÝ ĐIỂM SINH VIÊN

Mã sinh viên:

Họ tên sinh viên:

Mã khoa:

Điểm TB tích lũy:

Kết quả xếp loại:

Mã môn học	Tên môn học	Số tín chỉ	Điểm
01	Nhập môn máy tính	3	10
02	Trí tuệ nhân tạo	4	4
03	Truyền tin	4	7
05	Văn phạm	5	9

Yêu cầu: - Tạo nút thoát Form, các nút di chuyển đến bản ghi tiếp theo, bản ghi trước đó.

- Tạo công thức tính Điểm trung bình tích lũy các môn của mỗi sinh viên.

$ĐTB = (Điểm\ môn\ 1 * TC\ môn\ 1 + Điểm\ môn\ 2 * TC\ môn\ 2 + ...) / Tổng\ số\ tín\ chỉ\ tích\ lũy$

- Tạo công thức xét kết quả xếp loại dựa theo Điểm trung bình, nếu ĐTB từ 8 trở lên thì kết quả là “Giỏi”, ĐTB từ 7 trở lên thì kết quả là “Khá”, ĐTB từ 5 trở lên thì kết quả là “Trung bình”, ngược lại (ĐTB < 5) thì kết quả là “Yếu”.

4. Thiết kế Form xem danh sách sinh viên theo dạng Form chính phụ:

DANH SÁCH SINH VIÊN

Mã khoa:

Tên khoa:

Tổng số sinh viên:

Tổng học bổng:

Danh sách chi tiết:

	Mã sinh viên	Họ sinh viên	Tên sinh viên	Phái	Ngày sinh	Nơi sinh	Học bổng
▶	A01	Nguyễn Thu	Hải	☒	23/02/1980	Sài Gòn	100,000
	A03	Lê Thu Bạch	Yến	☐	21/02/1982	Hà Nội	140,000
*				☐			, 0

Yêu cầu: - Tạo các nút di chuyển đến bản ghi tiếp theo, đến bản ghi trước đó.

- Tạo nút thoát Form.

- Tạo công thức tính tổng số sinh viên, tổng học bổng của từng khoa.

H. Báo cáo (Report)

1. Thiết kế báo cáo danh sách sinh viên theo từng khoa:

Yêu cầu: Tạo công thức tính tổng số sinh viên, tổng học bổng của từng khoa.

BÁO CÁO DANH SÁCH SINH VIÊN THEO KHOA

Mã khoa: AV

Tổng số SV:

Tên khoa: Anh văn

Tổng học bổng:

TT	Mã SV	Họ tên sinh viên	Phái	Ngày sinh	Nơi sinh	Học bổng
1	A03	Lê Thu Bạch Yến	☐	21/02/1982	Lâm Đồng	210,000
2	A01	Nguyễn Thu Hải	☒	25/02/1980	Đồng Nai	210,000

Mã khoa: LS

Tổng số SV:

Tên khoa: Lịch sử

Tổng học bổng:

TT	Mã SV	Họ tên sinh viên	Phái	Ngày sinh	Nơi sinh	Học bổng
1	A04	Trần Anh Tuấn	☒	08/12/1984	Long An	210,000

2. Thiết kế báo cáo bảng điểm kết quả học tập của sinh viên:

- Yêu cầu:**
- Truyền tham số theo Mã sinh viên. Cần tạo truy vấn truyền tham số trước.
 - Tạo công thức tính Điểm trung bình các môn thi của sinh viên.
 - Tạo công thức xét kết quả xếp loại (Giỏi/Khá/TB/Yếu) dựa theo Điểm TB.

BẢNG ĐIỂM SINH VIÊN

Mã sinh viên: A01 Điểm TB:

Họ sinh viên: Nguyễn Thu Xếp loại:

Tên sinh viên: Hải

Mã môn học	Tên môn học	Điểm
01	Nhập môn máy tính	10
02	Trí tuệ nhân tạo	4
03	Truyền tin	7
05	Văn phạm	9

MỤC LỤC

PHẦN I – NHẬP MÔN CƠ SỞ DỮ LIỆU QUAN HỆ	1
CHƯƠNG 1 – TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU	1
§ 1. NHỮNG KHÁI NIỆM CƠ BẢN VỀ CƠ SỞ DỮ LIỆU	1
I. SỰ CẦN THIẾT CỦA CƠ SỞ DỮ LIỆU	1
II. NHỮNG KHÁI NIỆM CƠ BẢN VỀ CƠ SỞ DỮ LIỆU.....	2
§ 2. KIẾN THỨC TỔNG QUÁT CỦA HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU.....	6
I. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU – HỆ CƠ SỞ DỮ LIỆU	6
II. MÔ HÌNH CƠ SỞ DỮ LIỆU	8
CHƯƠNG 2 – MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ	12
§ 1. NHỮNG KHÁI NIỆM CƠ BẢN	12
I. MÔ HÌNH QUAN HỆ LÀ GÌ?.....	12
II. NHỮNG KHÁI NIỆM CƠ BẢN CỦA MÔ HÌNH QUAN HỆ	13
§ 2. KHÁI NIỆM VỀ KHOÁ	17
I. SIÊU KHÓA – KHÓA.....	17
II. KHÓA DỰ TUYÊN – KHÓA CHÍNH – KHÓA NGOẠI.....	18
CHƯƠNG 3 – NGÔN NGỮ THAO TÁC DỮ LIỆU	20
§ 1. ĐẠI SỐ QUAN HỆ.....	20
I. GIỚI THIỆU	20
II. CÁC PHÉP TOÁN ĐẠI SỐ QUAN HỆ	20
III. CÁC PHÉP TOÁN TẬP HỢP	23
§ 2. NGÔN NGỮ SQL.....	28
I. GIỚI THIỆU	28
II. CÁC LỆNH TRUY VẤN LỰA CHỌN DỮ LIỆU	28
III. CÁC LỆNH TRUY VẤN CẬP NHẬT – XÓA DỮ LIỆU	34
BÀI TẬP NGÔN NGỮ SQL	35
CHƯƠNG 4 – LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU.....	38
§ 1. PHỤ THUỘC HÀM.....	38
I. PHỤ THUỘC HÀM - HỆ LUẬT DẪN ARMSTRONG.....	38
II. BAO ĐÓNG	39
§ 2. PHÉP TÁCH CÁC LƯỢC ĐỒ QUAN HỆ.....	41
I. KHÁI NIỆM PHÉP TÁCH	41
II. PHÉP TÁCH KẾT NỐI KHÔNG MẤT THÔNG TIN.....	42
III. PHÉP TÁCH BẢO TỒN PHỤ THUỘC HÀM	44
§ 3. CHUẨN HOÁ LƯỢC ĐỒ QUAN HỆ	46
I. CÁC DẠNG CHUẨN.....	46

II. KẾT HỢP CHUẨN HOÁ VỚI CÁC PHÉP TÁCH LƯỢC ĐỒ QUAN HỆ	48
PHẦN II – HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MICROSOFT ACCESS	50
CHƯƠNG 5 – TỔNG QUAN HỆ QUẢN TRỊ CSDL MICROSOFT ACCESS.....	50
§ 1. TỔNG QUAN HỆ QUẢN TRỊ CSDL MICROSOFT ACCESS	50
§ 2. QUY TRÌNH CƠ BẢN THIẾT KẾ MỘT CSDL.....	51
§ 3. LÀM VIỆC VỚI MÔI TRƯỜNG ACCESS	51
I. LÀM QUEN VỚI ACCESS.....	51
II. TẠO CSDL MỚI VÀ LÀM VIỆC VỚI CSDL ĐÃ CÓ.....	52
CHƯƠNG 6 – TẠO LẬP CƠ SỞ DỮ LIỆU	55
§ 1. TẠO LẬP BẢNG VÀ HIỆU CHỈNH DỮ LIỆU.....	55
I. TẠO LẬP BẢNG DỮ LIỆU	55
II. CÁC THUỘC TÍNH CỦA TRƯỜNG.....	57
III. ĐẶT KHOÁ CHÍNH.....	61
§ 2. TẠO CÁC QUAN HỆ GIỮA CÁC BẢNG	62
I. THIẾT LẬP QUAN HỆ GIỮA CÁC BẢNG	62
II. NHẬP DỮ LIỆU VÀO BẢNG	64
III. CÁC THÔNG BÁO LỖI KHI NHẬP DỮ LIỆU	65
CHƯƠNG 7 – TRUY VẤN.....	67
§ 1. QUY TRÌNH TẠO MỘT TRUY VẤN.....	67
I. KHÁI NIỆM TRUY VẤN	67
II. TẠO TRUY VẤN BẰNG QBE	67
III. TẠO TRUY VẤN BẰNG CÂU LỆNH SQL.....	70
§ 2. TRUY VẤN HÀNH ĐỘNG	71
I. NHÓM TRUY VẤN HÀNH ĐỘNG.....	71
II. CÁC LOẠI TRUY VẤN KHÁC	73
CHƯƠNG 8 – BIỂU MẪU	80
§ 1. KHÁI NIỆM	80
§ 2. THIẾT KẾ.....	81
CHƯƠNG 9 – BIỂU BÁO CÁO.....	94
§ 1. KHÁI NIỆM	94
§ 2. TẠO VÀ HIỆU CHỈNH BÁO CÁO	95
I. THIẾT KẾ BÁO CÁO	95
III. XUẤT DỮ LIỆU.....	99
BÀI TẬP THỰC HÀNH.....	101