

Chương 5

CƠ SỞ DỮ LIỆU

Chương này giới thiệu những kiến thức cơ bản liên quan đến cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu, ngôn ngữ truy vấn dữ liệu SQL. Từ đó, giúp sinh viên hiểu được: cơ sở dữ liệu là gì, sự cần thiết của việc tổ chức dữ liệu dưới dạng cơ sở dữ liệu, phần mềm dùng để tạo lập và quản trị cơ sở dữ liệu, ngôn ngữ truy vấn dữ liệu SQL.

5.1. CƠ SỞ DỮ LIỆU

5.1.1. Khái niệm cơ sở dữ liệu

Cơ sở dữ liệu (CSDL, thuật ngữ tiếng Anh là *database*) là một tập hợp các dữ liệu có liên quan với nhau chứa thông tin về một tổ chức nào đó (như một trường đại học, một ngân hàng, một bệnh viện, một công ty...) được lưu trữ trên các thiết bị nhớ thứ cấp (như băng từ, đĩa từ...) để đáp ứng nhu cầu khai thác thông tin của nhiều người sử dụng với nhiều mục đích khác nhau.

Ví dụ 5.1: Bài toán Quản lý sinh viên đơn giản, ta có thể dùng một cơ sở dữ liệu lưu trữ thông tin về sinh viên và kết quả học tập của họ bao gồm 5 bảng (ví dụ được tạo bởi phần mềm Microsoft Access) như sau:

Bảng KHOA:

MaKhoa	TenKhoa	SoDT
CNTT	Công nghệ thông tin	0462617701
TY	Thú ý	0436782978
KE	Kế toán	
CNSH	Công nghệ sinh học	
NH	Nông học	
CNTP	Công nghệ thực phẩm	
QLDD	Quản lý đất đai	0438674739

Bảng LOP:

MaLop	TenLop	MaK
K56CNSHA	Công nghệ sinh học A K56	CNSH
K57CNSHA	Công nghệ sinh học A K57	CNSH
K57CNTTA	Công nghệ thông tin A K57	CNTT
K57QLTT	Quản lý thông tin K57	CNTT
K58CNTTA	Công nghệ thông tin A K58	CNTT
K58CNTTB	Công nghệ thông tin B K58	CNTT
K58TYA	Thú A K58	TY
K58TYB	Thú Y B K58	TY
K59TYC	Thú Y C- K59	TY

Bảng **SINHVIEN**:

	Masv	Hodem	Ten	Ngaysinh	Gioitinh	Tinh	MaLop
+	561000	Lê Thị Minh	Thùy	04/06/1992	Nữ	Thanh Hóa	K56CNSHA
+	571233	Trần Trung	Hải	04/06/1993	Nam	Nam Định	K57CNSHA
+	573435	Lê Thị	Linh	23/06/1993	Nữ	Hà Nội	K57CNSHA
+	582004	Trần Trung	Kiên	05/07/1994	Nam	Hà Nội	K57CNTTA
+	583456	Lê Thu	Hoài	03/05/1994	Nữ	Hà Nội	K58CNTTA
+	586786	Nguyễn Thị	Huyền	04/06/1994	Nữ	Thanh Hóa	K58CNTTA
*							

Bảng **MONHOC**:

	MaMH	TenMH	DVHT	DieuKien
	TH1009	Tin học đại cương	2	
	TH1010	Tin học ứng dụng	2	Tin học đại cương
	TH2010	Cơ sở dữ liệu	3	Tin học đại cương
	TH2021	Hệ quản trị CSDL	3	Cơ sở dữ liệu
	TH3020	Lập trình mạng	2	Tin học đại cương
*				

Bảng **KETQUA**:

	MaSV	MaMH	Diem
	561000	TH1009	7
	561000	TH2010	3
	561000	TH2021	5
	561000	TH3020	7
	571233	TH1010	6
	573435	TH1009	7
	573435	TH1010	8
	573435	TH2021	8
	573435	TH3020	9
*			

5.1.2. Các mức thể hiện của cơ sở dữ liệu

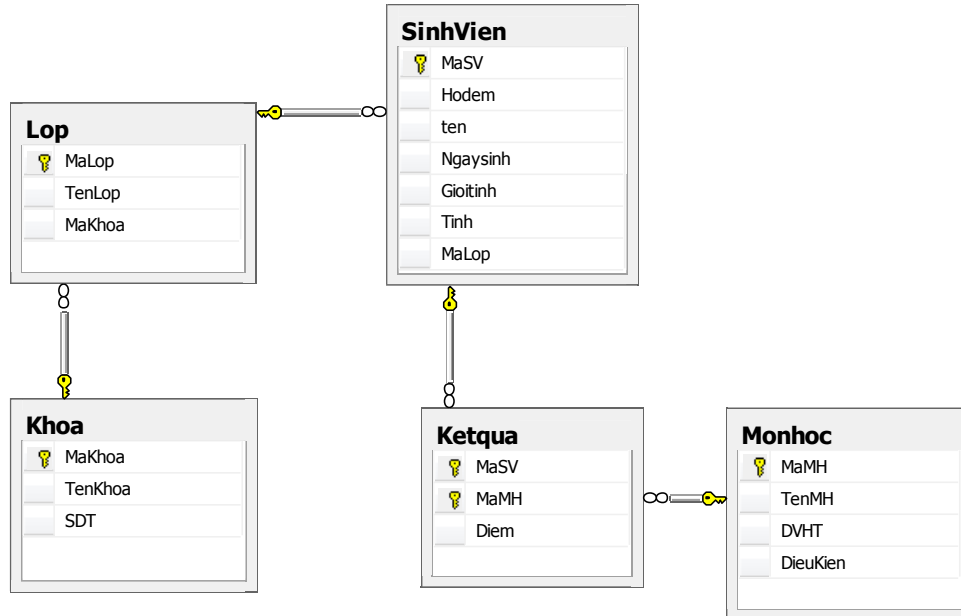
Vì mỗi nhóm người dùng có vai trò, nhu cầu hiểu và khai thác thông tin khác nhau nên để quản lý thông tin một cách hiệu quả, các hệ CSDL phải có các mức thể hiện khác nhau. Có 3 mức thể hiện CSDL là mức vật lý, mức khái niệm, mức khung nhìn.

a. Mức vật lý

Những chuyên gia tin học cần hiểu chi tiết về cách lưu trữ dữ liệu trong bộ nhớ, chẳng hạn: các tệp dữ liệu được lưu trữ trong vùng nhớ nào? mỗi bản ghi chiếm bao nhiêu byte? Mức hiểu biết chi tiết về một CSDL như vậy gọi là mức vật lý của hệ CSDL đó.

b. Mức khái niệm

Nhóm phát triển các ứng dụng không cần hiểu chi tiết ở mức vật lý, nhưng họ cần phải biết những dữ liệu nào được lưu giữ trong CSDL, giữa các dữ liệu có mối quan hệ như thế nào. Mức hiểu CSDL như vậy được gọi là mức khái niệm.

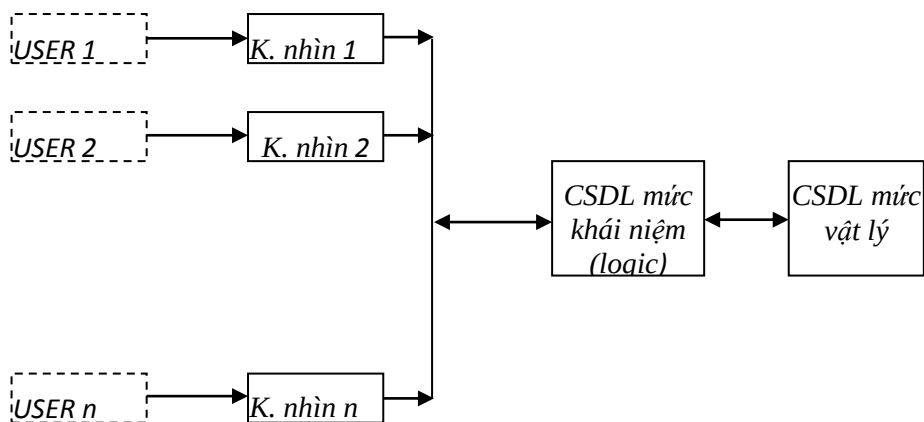


Hình 5.1. Ví dụ về mức khái niệm của CSDL

c. Mức khung nhìn

Mỗi nhóm người dùng chỉ cần biết phần thông tin nào đó của CSDL phù hợp với nghiệp vụ hay mục đích sử dụng của mình. Ví dụ, sinh viên thông qua khung nhìn biết được những thông tin liên quan đến bản thân họ. Người quản trị CSDL cần biết được toàn bộ thông tin về CSDL... Vì vậy, mức khung nhìn là mức hiểu CSDL của người dùng thông qua khung nhìn.

Ba mức hiểu về CSDL như trên chính là ba mức mô tả và làm việc với CSDL, phù hợp với nhu cầu khác nhau của những người liên quan đến CSDL.



Hình 5.2. Ba mức thể hiện của CSDL

5.1.3. Mô hình dữ liệu quan hệ

Mô hình dữ liệu (*data model*) là một tập các khái niệm và kí pháp dùng để mô tả dữ liệu, các mối quan hệ của dữ liệu, các ràng buộc trên dữ liệu của một tổ chức.

Hiện nay, có khá nhiều mô hình dữ liệu như:

- Mô hình dữ liệu quan hệ (Relational Data Model).
- Mô hình dữ liệu phân cấp (Hierarchical Data Model).
- Mô hình dữ liệu mạng (Network Data Model).
- Mô hình dữ liệu thực thể - liên kết (Entity-Relationship Data Model).
- Mô hình dữ liệu hướng đối tượng (Object Data Model).

Trong các mô hình trên thì mô hình dữ liệu quan hệ được sử dụng khá phổ biến. Mô hình này được đề xuất bởi E. F. Codd vào những năm 1970 - 1972. Nó cung cấp một cấu trúc dữ liệu đơn giản đó là quan hệ (bảng).

Cơ sở dữ liệu được xây dựng trên mô hình dữ liệu quan hệ được gọi là CSDL quan hệ. Một CSDL quan hệ thông thường chứa nhiều bảng. Mỗi bảng chứa dữ liệu của một tập thực thể, bao gồm các hàng và các cột. Mỗi hàng là một bản ghi (Record), mỗi cột là một trường (Field).

a. Một số khái niệm cơ bản trong mô hình dữ liệu quan hệ

- * *Quan hệ*: Dữ liệu lưu trữ trong CSDL được tổ chức thành bảng 2 chiều. Mỗi bảng 2 chiều được gọi là một quan hệ.

Dưới đây là ví dụ của một quan hệ:

SINHVIEN	MaSV	HoDem	Ten	NgaySinh	GioiTinh	Tinh
	521234	Lê Thị	Lan	02/04/90	Nữ	Hà Nội
	521235	Nguyễn Văn	Nam	23/06/90	Nam	Thanh Hóa
	521235	Lê Văn	Hùng	03/05/91	Nam	Hà Nội

Hình 5.3. Ví dụ về quan hệ SINHVIEN

- * *Lược đồ (schema)*

Tên của một quan hệ và tập các thuộc tính của nó được gọi là một lược đồ đối với quan hệ đó. Ta biểu diễn lược đồ cho một quan hệ bởi **Tên** của quan hệ và theo sau là danh sách các thuộc tính của nó. Vậy lược đồ của quan hệ SINHVIEN trong hình 5.3 là:

SINHVIEN(MaSV, HoDem, Ten, NgaySinh, GioiTinh, Tinh)

Lược đồ cơ sở dữ liệu quan hệ (relational database schema) là tập các lược đồ quan hệ của bài toán.

Ví dụ 5.2: Bài toán quản lý sinh viên trong ví dụ 5.1 có lược đồ CSDL bao gồm 5 lược đồ quan hệ sau:

KHOA(MaKhoa, TenKhoa, SoDT)

LOP(MaLop, TenLop, MaKhoa)

SINHVIEN(MaSV, HoDem, Ten, NgaySinh, GioiTinh, Tinh, MaLop)

MONHOC(MaMH, TenMH, DVHT, Dieukien)

KETQUA(MaSV, MaMH, Diem)

* *Bộ (tuple)*

Bộ là dòng của một quan hệ, trừ dòng tiêu đề (tên của các thuộc tính). Bộ còn có cách gọi khác là bản ghi (record). Trong một quan hệ các bộ không được trùng nhau.

* *Miền (domain)*

Miền là tập các giá trị mà thuộc tính có thể nhận.

Ví dụ, miền của thuộc tính Gioitinh (giới tính) trong ví dụ 5.2 gồm hai giá trị {Nam, Nữ}.

* *Khóa (key, còn gọi là khóa chính)*

Khóa của một quan hệ là một hoặc nhiều thuộc tính tối thiểu để xác định tính duy nhất của mỗi bộ trong quan hệ đó

Ví dụ 5.3: Trong quan hệ SINHVIEN ở trên, dễ hiểu là MaSV của mỗi sinh viên là duy nhất, không thể có 2 mã sinh viên trùng nhau. Vậy MaSV được thiết lập là khóa.

Ví dụ 5.4: Trong quan hệ KETQUA(MaSV, MaMH, Diem) ở trên, vì một sinh viên có thể học nhiều môn học nên để ghi điểm các môn của sinh viên đó vào bảng thì các bộ có cùng MaSV, khác nhau MaMH. Tương tự, một môn học có thể được học bởi nhiều sinh viên, nên các bộ có thể trùng MaMH, khác nhau MaSV (xem bảng KETQUA ở trên). Nhưng cặp MaSV, MaMH không thể trùng nhau để xác định duy nhất một điểm môn học của sinh viên.

Chú ý: Một quan hệ có thể có nhiều khóa, khi đó mỗi một khóa được gọi là một khóa dự tuyển. Thông thường có một khóa dự tuyển được chỉ định làm khóa chính. Việc lựa chọn một khóa dự tuyển làm khóa chính là tùy ý, nhưng nên chọn khóa dự tuyển đặc trưng cho bộ và chỉ gồm một thuộc tính hoặc có ít thuộc tính nhất làm khóa chính.

Ví dụ 5.5: Lược đồ quan hệ KHOA(MaKhoa, TenKhoa, SoDT) có hai khóa ứng cử là: $K_1 = \{MaKhoa\}$, $K_2 = \{TenKhoa\}$, tuy nhiên ta chọn MaKhoa làm khóa chính vì nó đặc trưng cho Khoa hơn và các giá trị của thuộc tính này ngắn, không có dấu và không có khoảng trống.

* *Khóa ngoại (foreign key)*

Khóa ngoại (khóa ngoài) của một lược đồ quan hệ là một tập gồm một hay nhiều thuộc tính không phải là khóa chính của lược đồ quan hệ này nhưng lại là khóa chính của một lược đồ quan hệ khác.

Khoá ngoại dùng để biểu thị liên kết giữa quan hệ này và quan hệ khác trong mô hình quan hệ.

Ví dụ 5.6: Xét lược đồ CSDL trong ví dụ 5.2. Ta thấy, trong lược đồ quan hệ LOP có MaKhoa là khoá ngoại (vì nó là khoá chính trong lược đồ quan hệ KHOA nhưng không phải là khoá chính của lược đồ quan hệ LOP). Khi đã tạo mối quan hệ (Relationship) hợp lý giữa trường MaKhoa trong bảng KHOA với trường MaKhoa trong bảng LOP thì ta có thể truy vấn các thông tin liên quan giữa 2 quan hệ này như lớp này thuộc khoa nào, có tên khoa là gì. Những nội dung như này sẽ được minh họa rõ ràng hơn ở phần sau và khi thực hành trên máy tính.

Một ví dụ tương tự trong lược đồ quan hệ SINHVIEN có MaLop là khoá ngoại (vì nó là khoá chính trong lược đồ quan hệ LOP nhưng không phải là khoá chính của lược đồ quan hệ SINHVIEN).

Lưu ý là khoá ngoại không xác định tính duy nhất của bộ dữ liệu như khoá chính. Với ví dụ này:

- Trong quan hệ LOP, thuộc tính MaKhoa của các bộ có thể trùng nhau (tương đương một khoa có nhiều lớp); nhưng trong quan hệ KHOA thì thuộc tính MaKhoa của mỗi bộ phải là duy nhất.
- Trong quan hệ SINHVIEN, thuộc tính MaLop của các bộ có thể trùng nhau (tương đương một lớp có nhiều sinh viên); nhưng trong quan hệ LOP thì thuộc tính MaLop của mỗi bộ phải là duy nhất.

5.1.4. Hệ cơ sở dữ liệu

Hệ CSDL (Database system) là một hệ thống gồm 4 thành phần:

- Cơ sở dữ liệu.
- Người sử dụng cơ sở dữ liệu: Là bất cứ một người nào đó có thể truy nhập (hợp pháp) vào CSDL, có nghĩa là bao gồm tất cả những người sử dụng cuối, những người viết chương trình ứng dụng và những người điều khiển toàn bộ hệ thống (còn gọi là người quản trị CSDL).
- Hệ quản trị CSDL (Phần 5.2).
- Phần cứng: Bao gồm các thiết bị nhớ thứ cấp được sử dụng để lưu trữ CSDL.

5.1.5. Lợi ích của hệ cơ sở dữ liệu

Trước khi các hệ CSDL ra đời (khoảng đầu những năm 60) là giai đoạn tiền xử lý cơ sở dữ liệu: Dữ liệu được tổ chức và xử lý bởi các tệp ghi trên các băng từ. Các ngôn ngữ lập trình như COBOL, BASIC được sử dụng để lập trình xử lý dữ liệu. Mỗi chương trình ứng dụng đều có một tệp dữ liệu tương ứng và mỗi khi chương trình ứng dụng cần được sửa đổi hoặc mở rộng thì tệp dữ liệu tương ứng cũng phải thay đổi theo. Cách tổ chức lưu trữ như vậy sẽ bị *đur thừa dữ liệu, dữ liệu không nhất quán, khó khăn trong việc truy cập và chia sẻ dữ liệu, dữ liệu không được bảo mật cao...* Việc sử dụng hệ CSDL để lưu trữ dữ liệu theo lý thuyết cơ sở dữ liệu sẽ khắc phục được những hạn chế của cách lưu trữ trên, cụ thể 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, dư thừa thông tin và lãng phí bộ nhớ.

Nhiều chương trình ứng dụng khác nhau cùng xử lý trên các dữ liệu như nhau nhưng lại không dùng chung dữ liệu, dẫn đến sự dư thừa đáng kể về dữ liệu.

- *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 các bản ghi bị lặp lại nhiều lần thì khi thực hiện việc cập nhật, sửa đổi, bổ sung 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. Nếu một thuộc tính của một đối tượng chỉ được lưu trữ một lần trong một CSDL thì sẽ đảm bảo được tính toàn vẹn và nhất quán của dữ liệu.
- *Có thể triển khai đồng thời nhiều ứng dụng trên cùng một CSDL:* Điều này có nghĩa là trên cùng một CSDL có thể triển khai đồng thời nhiều ứng dụng khác nhau tại các thiết bị đầu cuối khác nhau.
- *Thống nhất các tiêu chuẩn, thủ tục và các biện pháp bảo vệ, an toàn dữ liệu:* Các CSDL sẽ được quản lý tập trung bởi một người hay một nhóm người 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. Nhờ đó công việc bảo trì dữ liệu trở nên dễ dàng. 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, có thể cho phép nhiều người truy nhập đồng thời mà vẫn đảm bảo tính đúng đắn của dữ liệu. Người quản trị CSDL có thể cho phép mỗi người dùng của hệ CSDL chỉ được phép truy cập một phần CSDL, điều đó cũng là một biện pháp giữ cho dữ liệu trong CSDL được an toàn. Ví dụ, trong hệ thống quản lý học tập theo tín chỉ trên mạng, các sinh viên của trường chỉ nhìn thấy một phần CSDL chứa thông tin về sinh viên đó thông qua tài khoản họ được cấp chứ không nhìn thấy các thông tin của sinh viên khác.

Như vậy, việc tổ chức lưu trữ dữ liệu trong CSDL giúp người dùng quản lý dữ liệu tốt hơn và thông qua các hệ quản trị CSDL ta có thể thực hiện các nhiệm vụ quan trọng như: tổng hợp, sắp xếp, tìm kiếm, thêm, xóa, sửa... và khai thác dữ liệu.

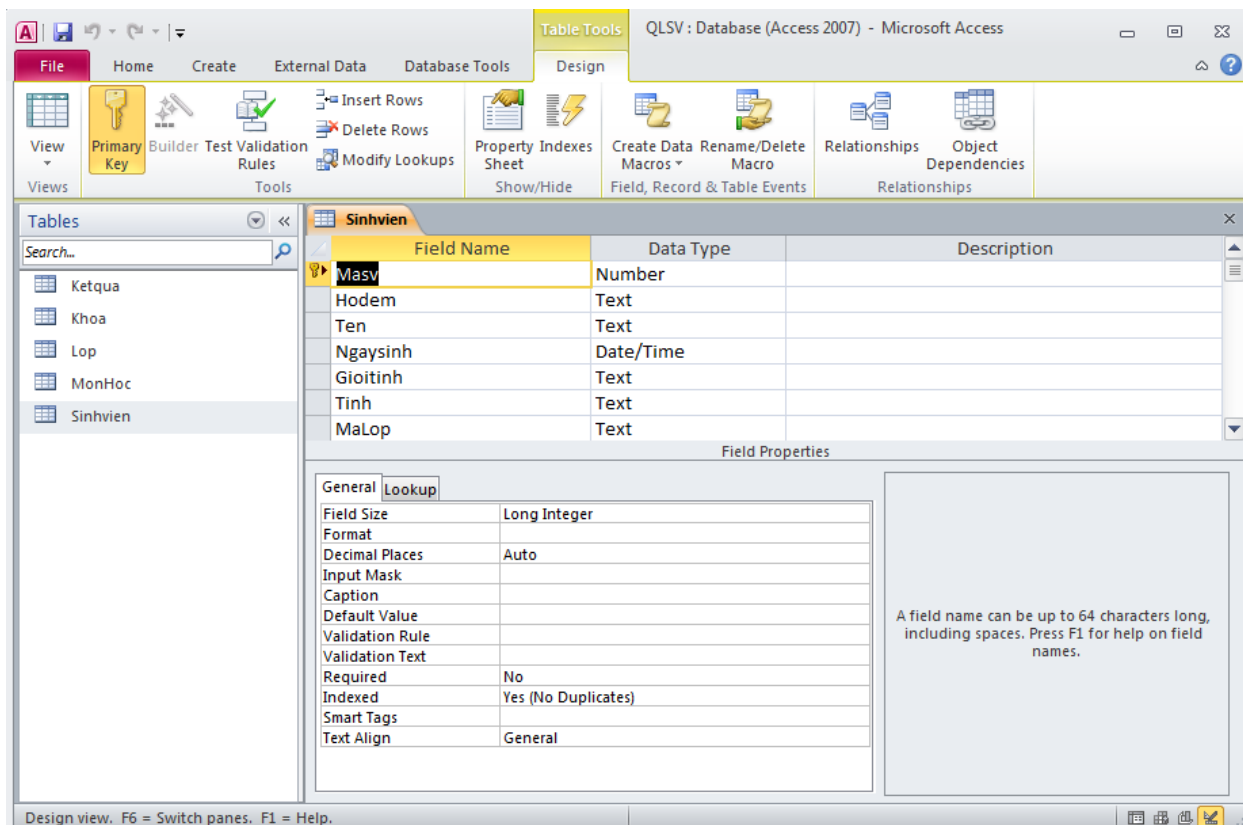
5.2. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

5.2.1. Khái niệm

Hệ quản trị cơ sở dữ liệu (HQTCSDL, thuật ngữ tiếng Anh là Database Management System - DBMS) là phần mềm được thiết kế để thuận lợi việc tạo lập, lưu trữ và khai thác thông tin của CSDL.

Như vậy, HQTCSDL cung cấp một môi trường thuận lợi, đơn giản và hiệu quả để người sử dụng có thể tạo lập, lưu trữ và thao tác trên CSDL mà không cần quan tâm nhiều đến thuật toán chi tiết và cách biểu diễn dữ liệu trong bộ nhớ.

Dưới đây là giao diện của việc dùng HQTCSDL Microsoft Access để tạo lập cơ sở dữ liệu:



Hình 5.4. Giao diện thiết kế các bảng trong HQTCSDL Microsoft Access

5.2.2. Phân loại hệ quản trị cơ sở dữ liệu

a. Phân loại

Các hệ quản trị cơ sở dữ liệu được phân thành 3 loại: XML DBMS, ODBMS và RDBMS.

- ✓ *XML DBMS*: Phù hợp cho dữ liệu đã được định dạng XML (eXtensible Markup Language).
- ✓ *ODBMS (object database management system)*: Phù hợp cho mô hình CSDL hướng đối tượng.
- ✓ *RDBMS (relational database management system)*: Phù hợp cho mô hình cơ sở dữ liệu quan hệ. Ngày nay, các HQTCSDL này đều có tính năng thao tác trên dữ liệu XML và các lớp đối tượng.

Thông thường, mỗi một HQTCSDL được xây dựng để phục vụ cho một mô hình dữ liệu nhất định, nhưng cũng có một số HQTCSDL có thể phục vụ cho nhiều mô hình dữ liệu khác nhau.

b. Một số hệ quản trị cơ sở dữ liệu phổ biến

Ngày nay, có một số HQTCSDL phổ biến như: Microsoft Access, Microsoft SQL Server, MySQL, SQLite, Oracle.

- DB2 là sản phẩm của IBM thuộc dòng HQTCSDL quan hệ, phiên bản đầu tiên ra đời năm 1982. Đây là HQTCSDL được dùng cho tất cả các máy tính, từ máy tính cá nhân đến những dòng máy tính lớn.
- Microsoft SQL Server và Microsoft Access là sản phẩm của Microsoft, thuộc loại HQTCSDL quan hệ.
- MySQL là HQTCSDL nguồn mở đa luồng, đa người dùng, được hãng MySQL sản xuất. MySQL rất phổ biến với những ứng dụng web và có thể làm việc với các CSDL trên nền Linux/Mac/Windows.
- SQLite là một hệ quản trị cơ sở dữ liệu có đặc điểm của gọn, nhẹ, đơn giản. Chương trình gồm 1 file duy nhất có dung lượng chưa đến 500kB, không cần cài đặt mà có thể sử dụng ngay. CSDL được lưu ở một file duy nhất, chạy tốt trên platform Mac OS-X, Android, iOS.

5.2.3. Chức năng cơ bản của hệ quản trị cơ sở dữ liệu

Một hệ quản trị cơ sở dữ liệu có các chức năng cơ bản sau đây:

- * Cung cấp môi trường tạo lập CSDL
- * Cung cấp môi trường cập nhật và khai thác dữ liệu
 - *Cập nhật:* Thêm, xóa, sửa dữ liệu.
 - *Khai thác:* Sắp xếp, tìm kiếm, kết xuất báo cáo...
- * Cung cấp công cụ kiểm soát, điều khiển CSDL
 - Phát hiện và ngăn chặn sự truy cập không được phép.
 - Duy trì tính nhất quán của dữ liệu.
 - Tổ chức và điều khiển các truy nhập đồng thời.
 - Khôi phục CSDL khi có sự cố ở phần cứng hay phần mềm.
 - Quản lý các mô tả dữ liệu.

Nói chung các HQTCSDL đều có các chức năng trên, nhưng các HQTCSDL khác nhau có chất lượng và khả năng khác nhau khi đáp ứng các nhu cầu thực tế. Các HQTCSDL luôn phát triển theo hướng đáp ứng đòi hỏi ngày càng cao của người dùng, bởi vậy các chức năng của chúng ngày càng được mở rộng.

5.3. NGÔN NGỮ TRUY VẤN SQL

SQL (*Structured Query Language* – ngôn ngữ truy vấn có cấu trúc) là ngôn ngữ truy vấn dựa trên đại số quan hệ được xác nhận là rất mạnh, phổ dụng và dễ sử dụng. SQL được sử dụng hầu hết trong các thao tác: truy vấn, thêm, xóa, sửa trên các hệ quản trị CSDL quan hệ. SQL được thiết lập như một ngôn ngữ chuẩn đối với cơ sở dữ liệu.

Ngôn ngữ SQL gồm các thành phần:

- Ngôn ngữ định nghĩa dữ liệu (*Data Definition Language - DDL*): Cung cấp các câu lệnh cho phép định nghĩa các lược đồ quan hệ, các ràng buộc toàn vẹn mà dữ liệu được lưu trữ trong CSDL phải thỏa mãn, cho phép xóa, sửa cấu trúc các quan hệ.

- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML): Đây là nhóm câu lệnh cho phép thao tác trên các dữ liệu của quan hệ. Nó dùng để tìm kiếm, trích rút, tổng hợp dữ liệu từ các quan hệ, đồng thời cho phép thêm, xóa, sửa dữ liệu trong các quan hệ

- Nhóm ngôn ngữ kiểm soát dữ liệu (Data Control Language - DCL): Bao gồm các câu lệnh đảm bảo tính an toàn và toàn vẹn dữ liệu, cấp phát quyền truy cập vào dữ liệu.

Trong nội dung giáo trình Tin học đại cương này chủ yếu giới thiệu nhóm ngôn ngữ thao tác dữ liệu cơ bản.

Nhóm ngôn ngữ SQL thao tác dữ liệu bao gồm các câu lệnh cho phép thao tác trên dữ liệu của CSDL. Nó dùng để thực hiện các truy vấn như: tìm kiếm, thêm, xóa, sửa các bản ghi.

- SELECT – trích dữ liệu từ một cơ sở dữ liệu
- INSERT – chèn dữ liệu mới vào trong một cơ sở dữ liệu
- DELETE – xóa dữ liệu từ một cơ sở dữ liệu
- UPDATE – cập nhật dữ liệu trong một cơ sở dữ liệu

5.3.1. Câu lệnh truy vấn dữ liệu

Loại câu lệnh này cho phép ta tìm kiếm, trích rút dữ liệu từ cơ sở dữ liệu. Kết quả của câu lệnh được hiển thị dưới dạng bảng 2 chiều.

Cú pháp:

```
SELECT [DISTINCT] <danh sách các cột>| *| <biểu thức>
FROM <bảng 1> [,bảng 2...]
[WHERE <điều kiện>]
[GROUP BY <danh sách tên cột> [HAVING <biểu thức điều kiện>]]
[ORDER BY <danh sách tên cột>|<biểu thức> [ASC|DESC]]
```

Giải thích:

- DISTINCT là từ khoá để được một danh sách không có các bản ghi trùng nhau.
- Các thành phần mà người dùng phải điền cụ thể vào khi viết lệnh được viết trong cặp < >.
- Các thành phần tùy chọn (những thành phần có thể có hoặc không) được viết trong cặp [].
- Danh sách các cột là tên các cột cần truy vấn trong các bảng (chú ý nếu tên cột xuất hiện trên nhiều bảng thì phải chỉ rõ cột đó được tham chiếu qua bảng nào bằng cú pháp: **Tenbảng.Tencot**)
- SELECT *: dùng để hiển thị tất cả các cột trong bảng.
- Bảng 1, bảng 2... là tên các bảng hoặc tên các khung nhìn dùng để truy vấn dữ liệu.

- Mệnh đề WHERE dùng để chỉ định một tiêu chuẩn chọn các bản ghi.
- <điều kiện> là một biểu thức logic có kết quả trả về là TRUE hoặc FALSE.
- GROUP BY <tên các cột> dùng để nhóm kết quả hiển thị theo từng loại giá trị của cột.
- Having là tiêu chuẩn chọn trên từng nhóm, được đặt sau GROUP BY.
- ORDER BY <danh sách tên cột> dùng để sắp xếp kết quả vừa chọn ở trên theo cột nào (ASC là tăng, DESC là giảm).

Biểu thức <điều kiện> có các toán tử sau có thể được dùng:

Bảng 5.1. Các toán tử trong SQL

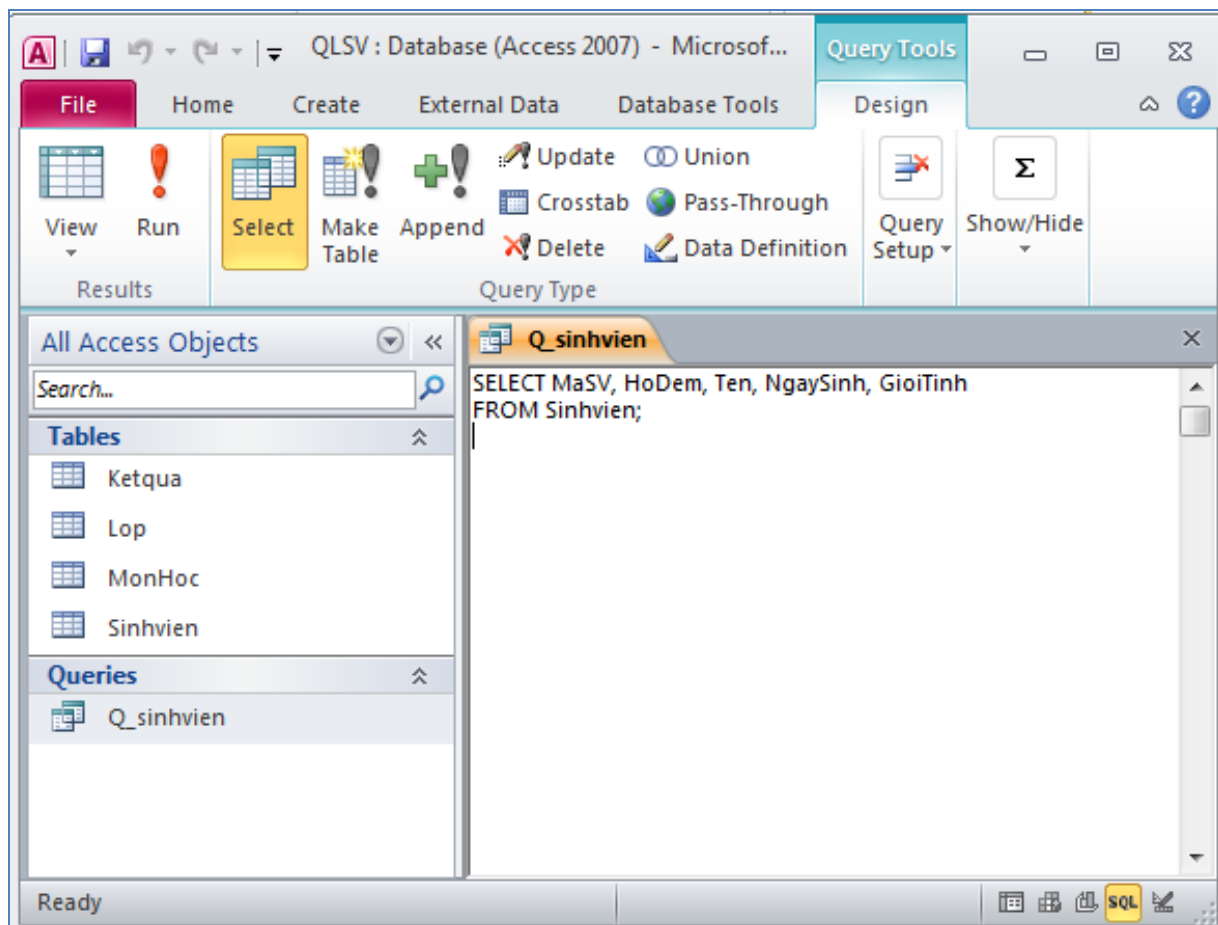
Phép toán	Giải thích
=	Bằng
<>	Khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
[NOT] Like	Phép toán tìm một mẫu kí tự
AND	Và
OR	Hoặc
BETWEEN <giá trị 1> AND <giá trị 2>	Chọn tất cả các trị trong khoảng giới hạn giữa hai giá trị. Các trị này có thể là các số, chuỗi kí tự, hay ngày tháng.
<biểu thức> [NOT] IN (danh sách câu truy vấn)	Kiểm tra giá trị của biểu thức có trong tập danh sách các giá trị không
[NOT] EXISTS (<câu truy vấn>)	Kết quả trả về TRUE nếu câu truy vấn khác rỗng.
<biểu thức><phép so sánh> SOME ALL ANY <câu truy vấn>	Kết quả trả TRUE nếu phép so sánh thỏa mãn.

Ví dụ 5.7: Sử dụng CSDL trong ví dụ 5.1 để thực hiện các yêu cầu sau:

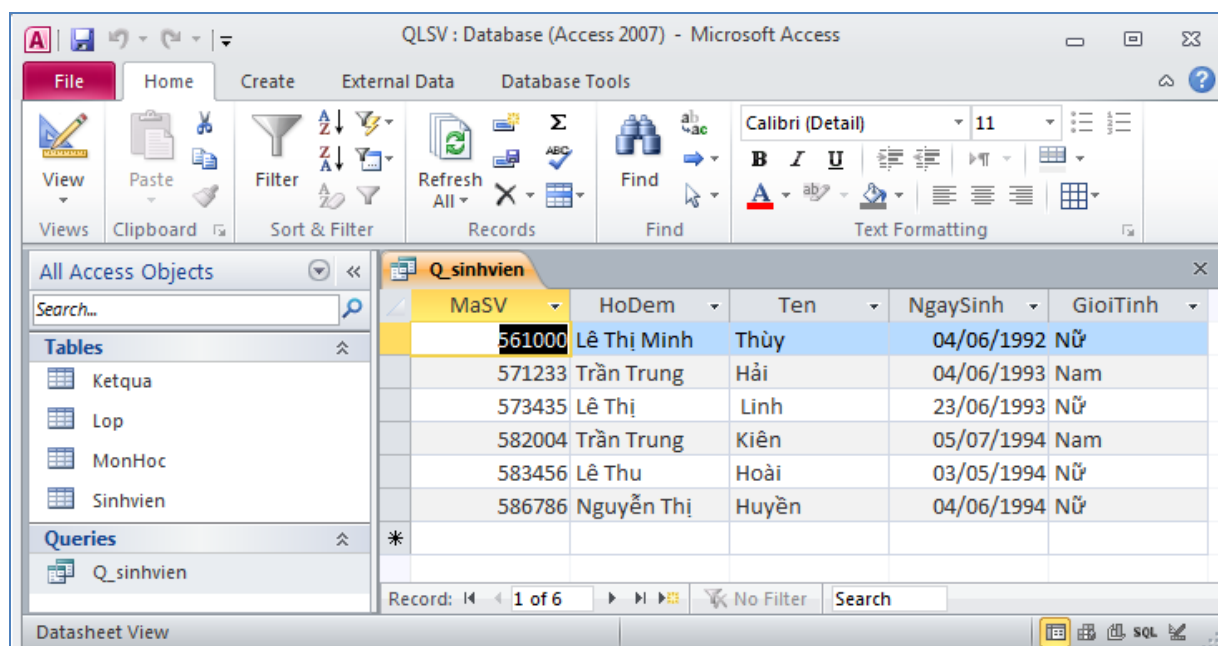
Câu lệnh SQL hiển thị thông tin về tất cả sinh viên, thông tin hiển thị cần: mã sinh viên, họ tên, ngày sinh, giới tính:

```
SELECT MaSV, HoDem, Ten, NgaySinh, GioiTinh
FROM Sinhvien;
```

Để chạy thử câu lệnh SQL, ta mở CSDL của bài toán trên đã được cài đặt trong một HQTCSDL, sau đó gõ câu lệnh SQL và chạy thử, màn hình sẽ hiển thị kết quả truy vấn.



Hình 5.5. Giao diện soạn thảo câu lệnh SQL trong HQTCSDL Microsoft Access



Hình 5.6. Kết quả thực hiện câu lệnh SQL hình 5.5 trong HQTCSDL Microsoft Access

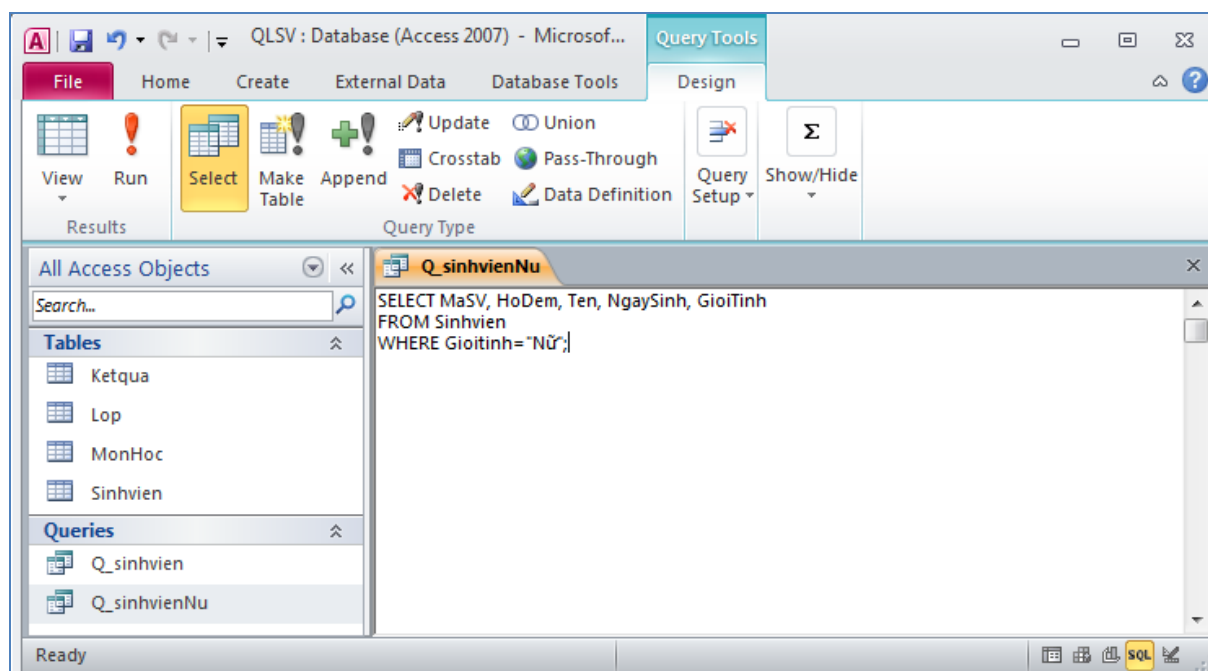
Câu lệnh SQL hiển thị thông tin về các sinh viên nữ, thông tin hiển thị cần: mã sinh viên, họ tên, ngày sinh, giới tính.

```
SELECT MaSV, HoDem, Ten, NgaySinh, GioiTinh
```

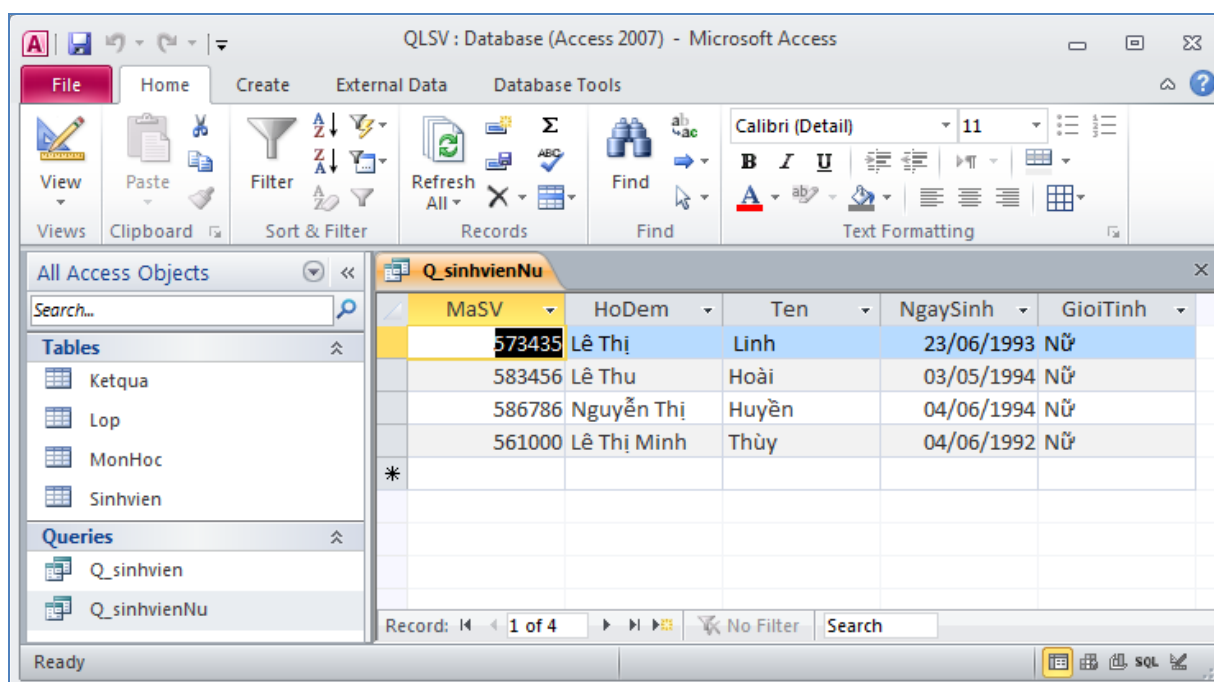
```
FROM Sinhvien
```

```
WHERE Gioitinh="Nữ";
```

Chạy thử câu lệnh SQL trong HQTCSDL Microsoft Access:



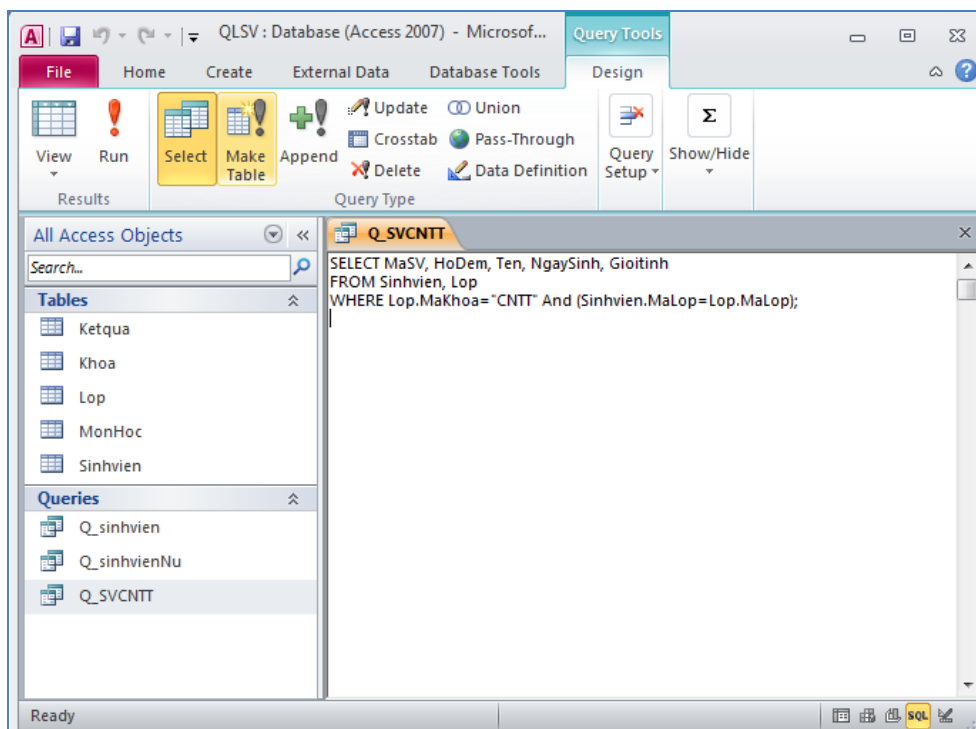
Hình 5.7. Giao diện soạn thảo câu lệnh SQL trong HQTCSDL Microsoft Access



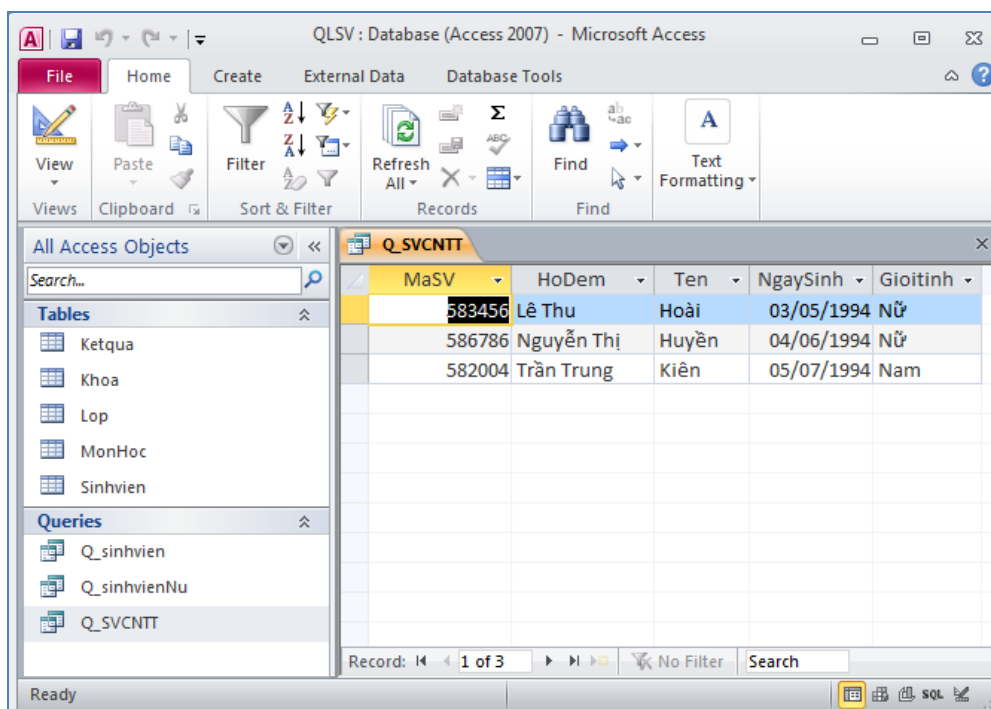
Hình 5.8. Kết quả thực hiện câu lệnh SQL hình 5.7 trong HQTCSDL Microsoft Access

Câu lệnh SQL hiển thị thông tin về các sinh viên khoa CNTT (có MaKhoa="CNTT").
Thông tin hiển thị cần: mã sinh viên, họ tên, ngày sinh, giới tính:

```
SELECT MaSV, HoDem, Ten, NgaySinh, GioiTinh
FROM Sinhvien, Lop
WHERE Lop.MaKhoa="CNTT" AND (Sinhvien.MaLop=Lop.MaLop);
```



Hình 5.9. Giao diện soạn thảo câu lệnh SQL trong HQTCSDL Microsoft Access



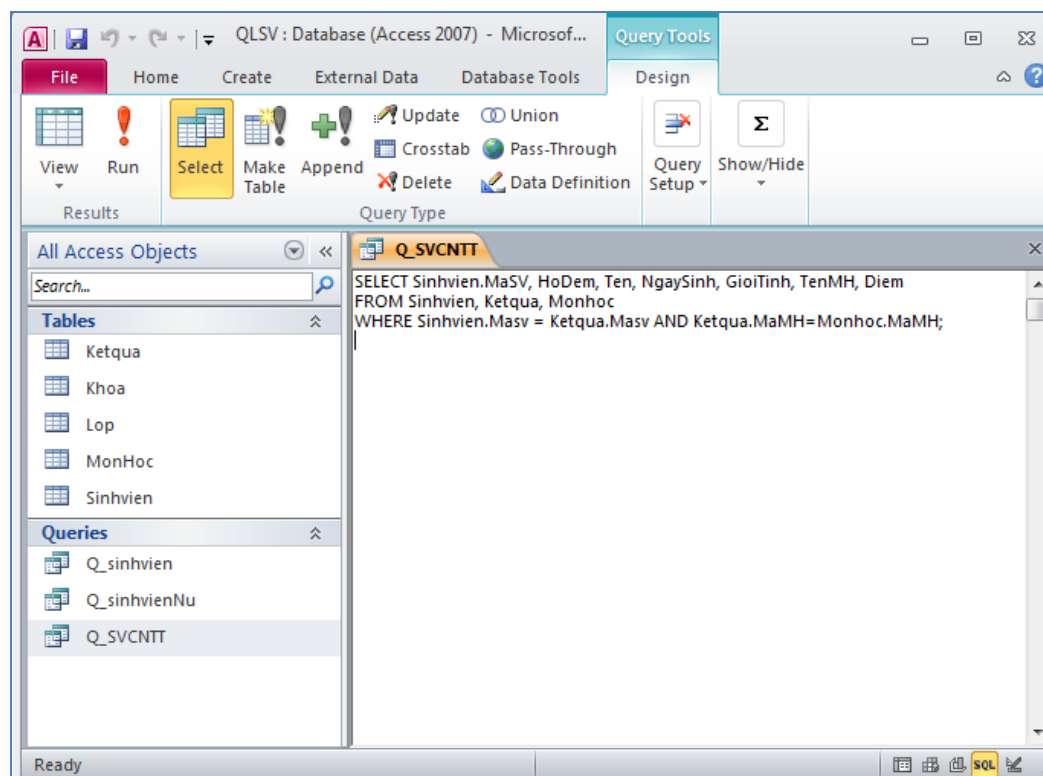
Hình 5.10. Kết quả thực hiện câu lệnh SQL hình 5.9 trong HQTCSDL Microsoft Access

Câu lệnh SQL hiển thị thông tin về các sinh viên với các kết quả học tập của họ. Thông tin hiển thị cần: mã sinh viên, họ tên, ngày sinh, giới tính, tên môn học, điểm.

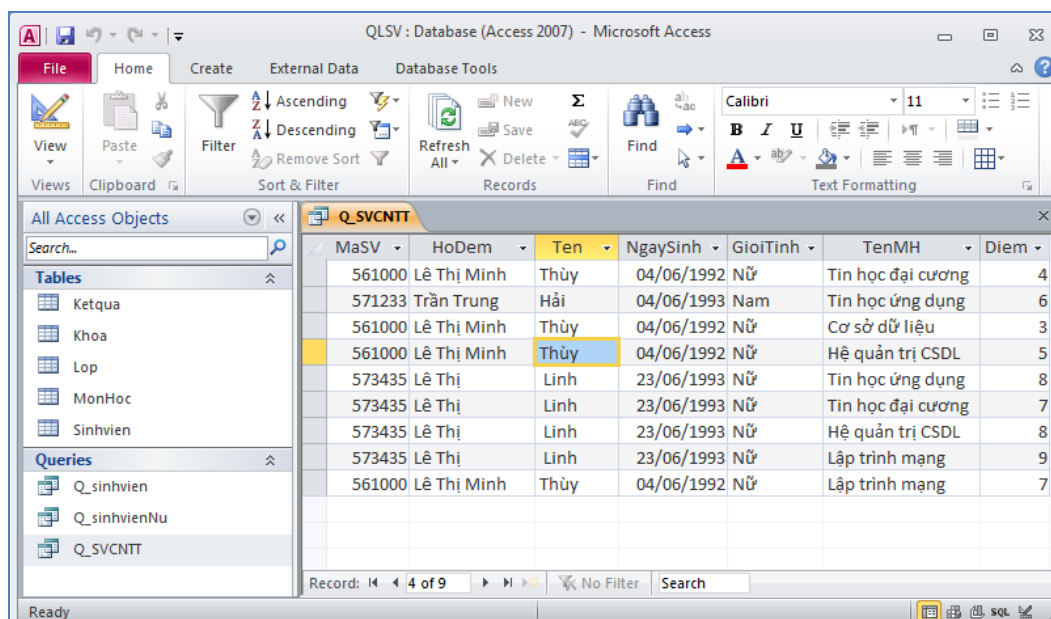
```
SELECT Sinhvien.MaSV, HoDem, Ten, NgaySinh, GioiTinh, TenMH, Diem
FROM Sinhvien, Ketqua, Monhoc
```

```
WHERE Sinhvien.MaSV = Ketqua.MaSV AND Ketqua.MaMH=Monhoc.MaMH;
```

Chạy câu lệnh SQL trong HQTCSDL Microsoft Access:



Hình 5.11. Giao diện soạn thảo câu lệnh SQL trong HQTCSDL Microsoft Access



Hình 5.12. Kết quả thực hiện câu lệnh SQL hình 5.11 trong HQTCSDL Microsoft Access

Câu lệnh SQL hiển thị thông tin về các sinh viên đạt điểm A học phần Tin học đại cương (MaMH="TH01009"). Thông tin hiển thị cần (mã sinh viên, họ tên, ngày sinh, tên môn học, điểm) và được sắp xếp theo vần alphabet của tên và họ (nếu trùng tên thì sắp xếp theo họ đệm):

```
SELECT Sinhvien.MaSV, HoDem, Ten, NgaySinh, TenMH, Diem
FROM Sinhvien, Ketqua, Monhoc
WHERE Sinhvien.MaSV=Ketqua.MaSV AND Ketqua.MaMH="TH1009" AND
Ketqua.MaMH=Monhoc.MaMH AND Diem>=8.5
ORDER BY Ten, Hodem;
```

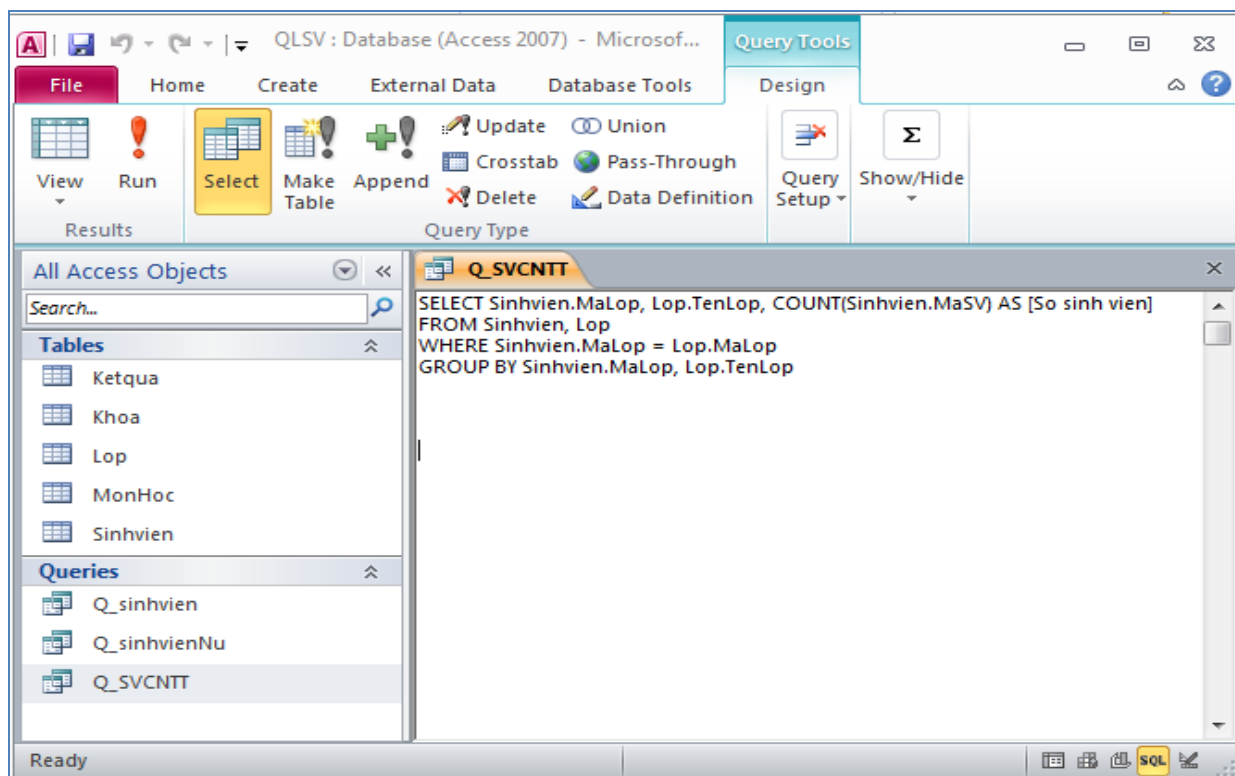
* Toán tử **GROUP BY**

Có thể phân hoạch các bộ của một quan hệ thành các nhóm tách biệt nhau và áp dụng các phép toán gộp cho các nhóm. Trong câu lệnh SELECT – FROM – WHERE, mệnh đề GROUP BY nhóm lại bởi một danh sách các thuộc tính của quan hệ cần nhóm.

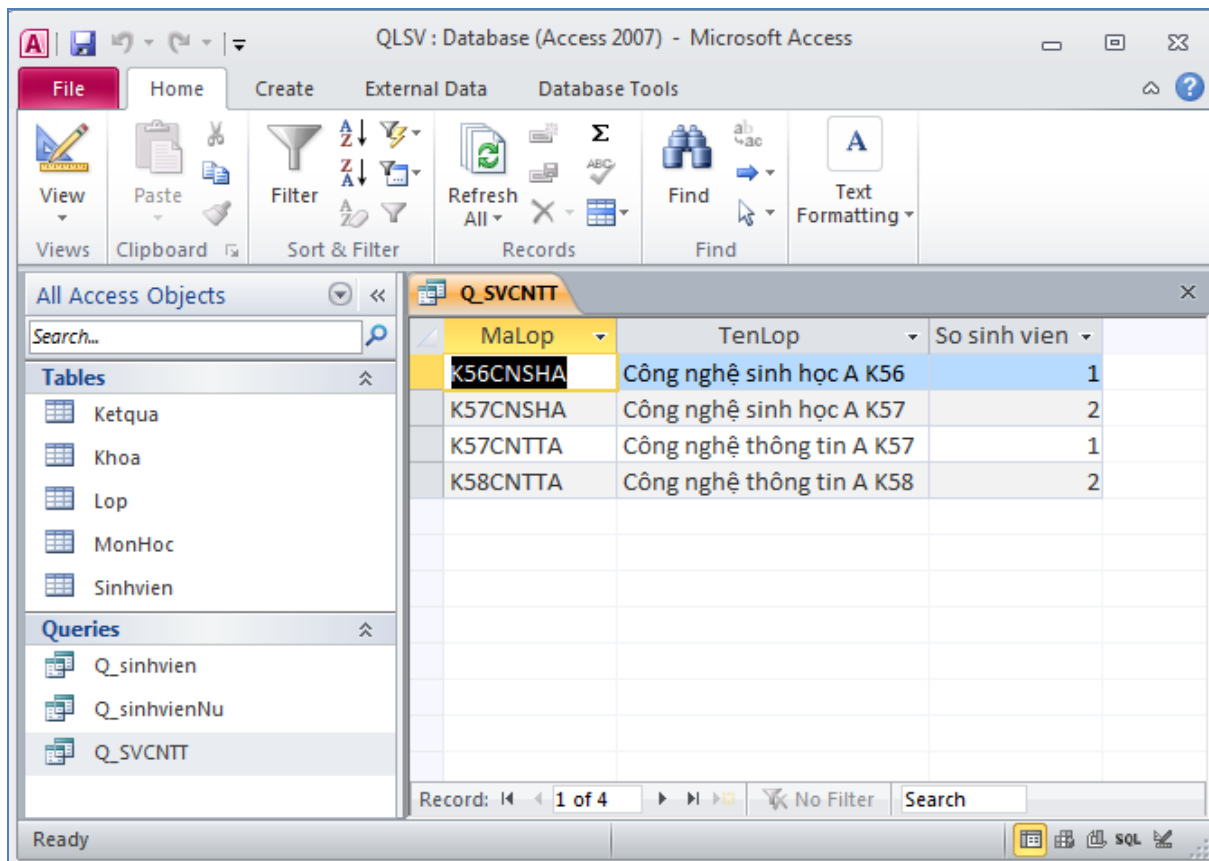
Ví dụ 5.8: In ra danh sách các lớp và số sinh viên trong mỗi lớp:

```
SELECT Sinhvien.MaLop, Lop.TenLop, COUNT(Sinhvien.MaSV) AS [So sinh vien]
FROM Sinhvien, Lop
WHERE Sinhvien.MaLop = Lop.MaLop
GROUP BY Sinhvien.MaLop, Lop.TenLop
```

Chạy câu lệnh SQL trong HQTCSDL Microsoft Access:



Hình 5.13. Giao diện soạn thảo câu lệnh SQL trong HQTCSDL Microsoft Access



Hình 5.14. Kết quả thực hiện câu lệnh SQL hình 5.13 trong HQTCSĐL Microsoft Access

* Toán tử **GROUP BY ... HAVING ...**

Phân hoạch các bộ của một quan hệ thành các nhóm tách biệt nhau và áp dụng các phép toán gộp cho các nhóm. Trong câu lệnh SELECT – FROM – WHERE, mệnh đề GROUP BY nhóm lại bởi một danh sách các thuộc tính của quan hệ cần nhóm và thỏa mãn một điều kiện nhóm HAVING:

GROUP BY $A_1, A_2, \dots, A_k$

HAVING E

Phân hoạch quan hệ thành các nhóm sao cho hai bộ cùng trong một nhóm khi và chỉ khi chúng giống nhau ở mọi thuộc tính $A_1, A_2, \dots, A_k$. Để cho kết quả của câu vấn tin có nghĩa, các thuộc tính $A_1, A_2, \dots, A_k$ cũng phải xuất hiện trong mệnh đề SELECT mặc dù chúng có thể có những bí danh để in ra nếu cần.

Ví dụ 5.9: In ra danh sách các lớp có số sinh viên ≥ 2 . Thông tin hiển thị cần: Mã lớp, Tên lớp, Số sinh viên.

```
SELECT Sinhvien.MaLop, Lop.TenLop, COUNT(Sinhvien.MaSV) AS 'So sinh vien'
FROM Sinhvien, Lop
WHERE Sinhvien.MaLop = Lop.MaLop
GROUP BY Sinhvien.MaLop, Lop.TenLop
HAVING COUNT(Sinhvien.MaSV) >= 2;
```

- * Toán tử LIKE: dùng chỉ định việc tìm gần đúng một chuỗi ký tự trong một cột.

Cú pháp:

SELECT <tên cột> FROM <tên bảng> WHERE <tên cột> LIKE <“chuỗi ký tự”>;

Một dấu "%" có thể dùng như ký tự đại diện cho một số ký tự

Ví dụ 5.10: Hiển thị thông tin về những sinh viên có tên bắt đầu bằng chữ “N”.

SELECT * FROM Sinhvien WHERE Ten LIKE “N%”;

- * Toán tử BETWEEN <giá trị 1> AND <giá trị 2>: chọn tất cả các trị trong khoảng giới hạn giữa hai giá trị. Các giá trị này có thể là các số, chuỗi ký tự, hay ngày tháng.

Ví dụ 5.11: Hiển thị thông tin về những sinh viên có ngày sinh trong khoảng 01/01/93 đến 31/12/94. Lưu ý là cần chuyển chuỗi ngày tháng trong câu lệnh thành dạng tháng trước ngày sau.

SELECT * FROM Sinhvien

WHERE Ngaysinh BETWEEN #01/01/93# AND #12/31/94#;

- * Từ khóa DISTINCT dùng để trả về chỉ các giá trị khác biệt (distinct).

Ví dụ 5.12: Hiển thị tên các tỉnh của sinh viên.

SELECT DISTINCT Tinh

FROM Sinhvien;

❖ Truy vấn trên nhiều bảng dùng kết nối Join

- * Kết nối bằng trên các thuộc tính cùng tên

Cú pháp:

```
SELECT <danh sách các cột>
FROM Bảng1 INNER JOIN Bảng2 ON <Bảng1.khóa chính = Bảng2.khóa ngoại>
[WHERE <điều kiện>];
```

Ví dụ 5.13: Hiển thị thông tin về các sinh viên cùng với tên lớp của họ.

SELECT Sinhvien.*, Lop.TenLop

FROM Sinhvien INNER JOIN Lop ON Sinhvien. MaLop=Lop.MaLop;

- * Kết nối ngoài trên các thuộc tính cùng tên

Cú pháp:

```
SELECT <danh sách các cột>
FROM Bảng1 LEFT|RIGHT JOIN Bảng2 ON <Bảng1.khóa chính= Bảng2.khóa ngoại>
[WHERE <điều kiện>];
```

- LEFT JOIN trả về tất cả các hàng từ bảng thứ nhất, cho dù nó không được so trùng trong bảng thứ hai. Nếu các hàng trong bảng R1 không so trùng trong bảng R2, những hàng này **cũng được** liệt kê.
- RIGHT JOIN trả về tất cả các hàng từ bảng thứ hai, cho dù nó không được so trùng trong bảng thứ nhất. Nếu có bất kỳ hàng nào trong bảng R1 không được so trùng trong bảng R2, các hàng này **cũng được** liệt kê.

Ví dụ 5.14: Hiển thị thông tin về các lớp của các khoa, kể cả những khoa chưa có lớp trong bảng lớp. Thông tin cần hiển thị gồm: mã khoa, tên khoa, tên lớp.

```
SELECT Khoa.MaKhoa, TenKhoa, TenLop
FROM Khoa LEFT JOIN Lop ON Khoa.MaKhoa=Lop.MaK;
```

❖ Câu lệnh truy vấn lồng

Cú pháp:

SELECT <danh sách các cột>	}	<i>Câu truy vấn cha</i>
FROM <danh sách các bảng>		
WHERE <so sánh tập hợp> (		
SELECT <danh sách các cột>	}	<i>Câu truy vấn con</i>
FROM <danh sách các bảng>		
WHERE <điều kiện>);		

Trong đó, phép so sánh tập hợp thường đi cùng với một số toán tử: IN, NOT IN, ALL, ANY, SOME, EXISTS, NOT EXISTS.

Ví dụ 5.15: Hiển thị thông tin về những SV đã có kết quả điểm ít nhất một học phần:

```
SELECT *
FROM Sinhvien
WHERE MaSV IN (
    SELECT MaSV
    FROM Ketqua);
```

Ví dụ 5.16: Hiển thị thông tin về những sinh viên đã đăng kí học và không phải học lại học phần nào.

```
SELECT *
FROM Sinhvien, Ketqua
WHERE (Sinhvien.MaSV=Ketqua.MaSV) AND Sinhvien.MaSV NOT IN
    (SELECT MaSV
    FROM Ketqua
    WHERE Diem<4);
```

5.3.2. Câu lệnh cập nhật dữ liệu

Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng.

Cú pháp:

```
UPDATE <tên bảng> SET
    <tên cột 1> = <giá trị| biểu thức mới>
    [, <tên cột 2> = <giá trị| biểu thức mới>...]
[WHERE <điều kiện>];
```

Ví dụ 5.17: Sửa tỉnh của sinh viên có mã sinh viên là 531236 từ Nam Định về Hà Nội.

```
UPDATE Sinhvien SET Tinh="Hà Nội"
```

```
WHERE MaSV="531236";
```

Lưu ý:

- Những dòng thỏa mãn điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới.
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ được cập nhật.
- Lệnh UPDATE có thể gây ra vi phạm ràng buộc tham chiếu như sau:
 - Không cho sửa.
 - Sửa luôn những dòng có giá trị đang tham chiếu đến (ví dụ với HQTCSDL Microsoft Access, trong Relationship nếu ta chọn Cascade Update Related Fields).

5.3.3. Thêm dữ liệu

Khi muốn thêm các dòng mới vào một bảng ta sử dụng cú pháp sau:

Cú pháp 1: Thêm 1 dòng mới vào bảng với các giá trị cụ thể:

```
INSERT INTO <tên bảng> [<danh sách các thuộc tính>]
VALUES (danh sách các giá trị);
```

Ví dụ 5.18: Thêm sinh viên có MaSV="536780", Hodem="Lê Thị", Ten="Hà", Ngaysinh="#25/5/90#", Gioitinh="Nữ", Tinh="Hà Nội", MaLop="K52THA" vào bảng Sinhvien.

```
INSERT INTO Sinhvien
```

```
VALUES ("536780", "Lê Thị", "Hà", #25/5/90#, "Nữ", "Hà Nội", "K52THA");
```

Chú ý:

- Thứ tự các giá trị chèn vào phải trùng với thứ tự các cột trong bảng cần chèn.
- Có thể chèn giá trị NULL ở những thuộc tính không là khóa chính.