**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ**

Môn học: **Cơ sở dữ liệu**

Đề tài: **Quản lí phân công và theo dõi giảng dạy**

|  |  |
| --- | --- |
| **Thành viên nhóm:** | 1. Đỗ Tiến Đạt  2. Nguyễn Đức Hạnh  3. Lê Minh Hiếu |
| **Lớp :** | Tin Học |
| **Giáo viên hướng dẫn:** | Đỗ Thị Mai Hường |

**Chương I: Mô tả hệ thống**

1. **Khảo sát hệ thống**
2. ***Yêu cầu chức năng****:* sử dụng công cụ SQL Server để quản lý quá trình phân công và theo dõi giảng dạy giảng viên của bộ môn
3. ***Đặc tả yêu cầu****:*

Bộ môn quản lí, theo dõi được quá trình giảng dạy của giảng viên, dựa vào chức danh, chức vụ của giảng viên và quân số của các lớp học phần được bộ môn đảm nhận từ đó phân công cho từng giảng viên cho phù hợp.

Môi trường cài đặt: Hệ quản trị cơ sở dữ liệu: SQL Server Ngôn ngữ lập trình: C#

Kế hoạch cài đặt:

+ Phân chia các công việc trong mục 1 cho từng cá nhân trong nhóm.

+ Lên kế hoạch nội dung công việc cho mỗi cá nhân theo từng tuần.

1. **Phân tích yêu cầu bài toán**
2. **Mục đích đề tài:** Bài toán đặt ra vấn đề lưu trữ và xử lý thông tin về phân công và giảng dạy của các giáo viên
3. **Mô tả nghiệp vụ:**

* Nhiệm vụ giảng dạy của giáo viên do bộ môn đề xuất dựa trên cơ sở về chức danh, nhiệm vụ, học hàm, số tải định mức
* Từ đó đưa lên phòng đào tạo sắp xếp lịch dạy, phòng đào tạo có nhiệm vụ trả về thời khóa biểu giảng dạy cho bộ môn và cho học viên, sinh viên. Bộ môn dựa vào thời khóa biểu để theo dõi giảng dạy

1. **Mô tả dữ liệu phục vụ nghiệp vụ**

* Bộ môn lưu trữ thông tin của bộ môn như Mã BM, tên BM, mã chủ nhiệm bộ môn
* Giảng viên mang thông tin như Mã giảng viên, họ tên, địa chỉ, sđt, chức vụ, … trong mỗi kì giáo viên lại đảm nhiệm những chức vụ,chức danh khác nhau tương ứng với số tải định mức, số tải miễn giảm khác nhau.
* Đối tượng học viên trong bài toán có thể là học viên quân sự hoặc sinh viên dân sự do đó để mang tính tổng quát ta lưu trữ thông tin của các lớp sinh viên dưới dạng lớp học phần . Mỗi lớp học phần sẽ có những thông tin cơ bản như mã lớp, tên lớp , số lượng sinh viên
* Một học phần có nhiều lớp học phần, mỗi lớp học phần sẽ có lịch học nhất định trong mỗi kì học

1. **Mô hình liên kết thực thể**
2. **Cơ sở lý thuyết**
3. Thực thể, tập thực thể, thuộc tính

Mô hình liên kết thực thể dựa trên cơ sở nhận thức của thế giới thực bao gồm tập các đối tượng cơ sở được gọi là các thực thể và một tập các liên kết giữa các đối tượng này. Có 3 ký hiệu cơ bản mà mô hình liên kết thực thể sử dụng là: các tập thực thể, các tập liên kết và các thuộc tính

1. Thực thể, tập thực thể

Một thực thể là một đối tượng cụ thể hay trừu tượng trong thế giới thực mà nó tồn tại và có thể phân biệt đươc với các đối tượng khác

Một nhóm các thực thể giống nhau tạo thành một tập thực thể.

Ví dụ: các tập thực thể: bộ môn, Giảng viên,...

1. Thuôc tính

Thuộc tính là một đặc tính của một tập thực thể. Mỗi thuộc tính thì có một miền giá trị riêng

Ví dụ: Bộ môn gồm các thuộc tính cơ bản sau: Mã bộ môn, tên bộ môn, Mã CNBM.

Các loại thuôc tính:

* Thuộc tính đơn: là các thuộc tính mà không thể phân chia ra được các thành phần nhỏ hơn nữa. ví dụ: mã học viên, điểm trung bình
* Thuộc tính phức hợp: là các thuộc tính có thể phân chia được thành các thành phần nhỏ hơn, biểu diễn các thuộc tính cơ bản hơn với các ý nghĩa đọc lập, ví dụ: họ tên. Việc phân chia mỗi thuộc tính phức hợp tùy thuộc vào hoàn cảnh cụ thể

Các loại giá trị của thuộc tính:

* Đơn trị: các thuộc tính có giá trị duy nhất cho một thực thể: mã sinh viên, điểm trung bình
* Đa trị: các thuộc tính có một tập giá trị cho cùng một thực thể: chức vụ
* Thuộc tính khóa: là thuộc tính mà giá trị của nó khác nhau đối với mỗi thực thể riêng biệt có trong một tập thực thể, cỏ thể dùng để phân biệt các thực thể. Một kiểu thực thể có thể có nhiều hơn một thuộc tính khóa hoặc cũng có thể không có khóa( kiểu thực thể yếu)

1. Các kiểu liên kết, tập liên kết

Một liên kết là một mối liên kết có nghĩa giữa 2 hay nhiều thực thể

Tập liên kết là một tập hợp các lien kết cùng mối liên kết giữa các thực thể tương ứng

Mối liên kết có thể có thuộc tính riêng và được gắn với liên kết

Các kiểu của liên kết:

* 1-1: liên kết của 1 thực thể của một tập thực thể với nhiều nhất 1 thực thể của tập thực thể khác và ngược lại
* 1- n: liên kết 1 thực thể của tập thực thể thứ nhất với nhiều thực thể của tập thực thể thứ 2 và ngược lại, 1 thực thể của tập thực thể thứ 2 chỉ liên kết với 1 thực thể của tập thực thể thứ nhất
* N – n: liên kết 1 thực thể của 1 tập thực thể với nhiều thực thể của tập thực thể khác và ngược lại

1. Các bước xây dựng:

Bước 1: xác định tập thực thể

Bước 2: xác định mối quan hệ

Bước 3: xác định thuộc tính và gắn thuộc tính cho tập thực thể và mối quan hệ

Bước 4: quyết định thuộc tính khóa

Bước 5: quyết định (min, max) cho mối quan hệ

* Quy tắc xây dựng:
* Chính xác
* Tránh trùng lặp
* Dễ hiểu
* Chọn đúng mối quan hệ
* Chọn đúng kiểu thuộc tính

1. **Áp dụng với bài tập lớn**
2. Xác định các thực thể: Giảng viên, lớp học phần, học phần, bộ môn.
3. Xác định các thuộc tính và thuộc tính khóa

* Bộ môn:
  + **Mã BM**: char(10)
  + Tên BM: nvarchar(50)
  + Mã CNBM: char(10)
* Giảng viên:
  + **Mã GV**: char(10)
  + Mã BM: char(10)
  + Họ tên: nvarchar(50)
  + Ngày sinh: date
  + Địa chỉ: nvarchar(50)
  + SĐT: varchar(15)
  + Giới tính: char(10)
  + Chức danh: nvarchar(50)
  + Định mức: int
  + Nhiệm vụ: int
* Lớp học phần:
  + **Mã lớp**: char(10)
  + Mã học phần: char(10)
  + Mã GV: char(10)
  + Số tiết: int
  + Tên lớp: nvachar(50)
  + Quân số: int
  + Bậc học: nvarchar(20)
  + Địa điểm: nvarchar(50)
  + Phòng học: nvarchar(50)
* Học phần:
  + **Mã học phần**: char(10)
  + Tên học phần: nvarchar(50)
  + Số tín chỉ: int
* Buổi học
  + **Mã buổi học**: char(10)
  + Ngày: date
  + Ca: nvarchar(10)
  + Học kì: int
  + Năm học: string
* Tham gia
* **Mã chức vụ**: char(10)
* Mã GV: char(10)
* Học kỳ: int
* Năm học: string
* Thời khóa biểu
* **Mã lớp**: char(10)
* Mã BH char(10)
* Tiết bắt đầu: int
* Tiết kết thúc: int

1. Xác định mối quan hệ

- Một bộ môn có nhiều Giảng viên

- Một Giảng viên có thể dạy nhiều lớp học phần

- Một Giảng viên có thể đảm nhiệm nhiều chức vụ

- Một chức vụ do 1 giảng viên đảm nhiệm

- Một lớp học phần chỉ học 1 học phần

- Một học phần có thể có nhiều lớp học phần

- Một lớp học phần có nhiều buổi học

- Một buổi học có nhiều lớp học phần học

4. Quyết định (min, max) cho mối quan hệ, xây dựng lược đồ ER

|  |
| --- |
| Giảng viên |
| **MÃ GV**  Họ tên  Mã BM  Ngày sinh  Địa chỉ  SĐT  Giới tính  Chức danh  Định Mức  Nhiệm vụ |

|  |
| --- |
| Tham gia |
| **Mã chức vụ**  Mã GV  Học kỳ  Năm học |

|  |
| --- |
| Chức Vụ |
| **Mã chức vụ**  Tên chức vụ  Hệ số |

Có

|  |
| --- |
| Bộ môn |
| **Mã bộ môn**  Tên Bộ môn  Mã CNBM |

|  |
| --- |
| Lớp học phần |
| **Mã lớp**  Tên lớp  Mã học phần  Mã GV  Quân số  Bậc học  Số tiết  Địa điểm  Phòng học |

|  |
| --- |
| Học phần |
| **Mã học phần**  Tên học phần  Số Tín chỉ |

|  |
| --- |
| Thời khóa biểu |
| **Mã lớp**  Mã BH  Tiết bắt đầu  Tiết kết thúc |

Có

|  |
| --- |
| Buổi học |
| **Mã BH**  Ngày  Ca  Học kỳ  NĂm học |

Phân công

Có

Có

Có

1. **Mô hình dữ liệu quan hệ**
2. **Cơ sở lý thuyết**

Mô hình dữ liệu quan hệ bao gồm một hoặc nhiều quan hệ. thực thể và thuộc tính trong mô hình liên kết thực thể trở thành quan hệ. Mối quan hệ sẽ trở thành khóa ngoại

1. Các khái niệm

* Thuộc tính: tương ứng với dặc tính của một đối tượng. Mỗi thuộc tính được xác định trên một miền giá trị nhất định gọi là miền thuộc tính. Miền thuộc tính là tập tất cả các giá trị mà thuộc tính có thể nhận

VD:

* + BOMON(mã BM, tên BM, mã CNBM)
  + Dom(mã BM)={char(10)}
  + Dom(tên BM)={nvarchar(50)}
  + Dom(mã CNBM)={char(10)}
* Khóa:
  + Thuộc tính khóa chính của một quan hệ là một hay một nhóm thuộc tính xác định duy nhất một bộ trong quan hệ. Khóa chính của quan hệ là định danh của tập thực thể tương ứng
  + Thuộc tính khóa ngoài được sử dụng để thiết lập một mối quan hệ. Đó là thuộc tính mô tả của quan hệ này nhưng đồng thời lại là thuộc tính khóa trong quan hệ khác

1. Cách biến đổi mô hình liên kết thực thể sang mô hình quan hệ

* Đối với các thuộc tính đa trị: tách thuộc tính đa trị ra khỏi kiểu thực thể ban đầu, thêm một kiểu thực thể mới chứa thuộc tính đa trị này, xác định lại mối liên kết của kểu thực thể này với kiểu thực thể ban đầu
* Các quy tắc biến đổi
  + Một tập thực thể => một lược đồ quan hệ
  + Mỗi thuộc tính => một thuộc tính
  + Mỗi thuộc tính khóa => khóa chính
  + Mỗi mối liên kết ( áp dụng cho mối liên kết 2 ngôi) => khóa ngoại/ lược đồ quan hệ mới

+ Quy tắc 1: với kiểu liên kết 1:1

Cách 1: chuyển khóa chính của lược đồ quan hệ này thành khóa ngoại của lược đồ quan hệ kia hoặc ngược lại

Cách 2: nhập 2 kiểu thực thể và mối liên kết thành một lược đồ quan hệ, chọn khóa chính cho phù hợp

+ Quy tắc 2: với kiểu liên kết 1:n

Chuyển khóa chính của lược đồ quan hệ bên 1 (cha) sang làm khóa ngoại của lược đồ quan hệ bên n (con)

+ Quy tắc 3: với kiểu liên kết n:n

Chuyển mối liên kết thành một lược dồ quan hệ có thuộc tính là thuộc tính của mối liên kết, them các thuộc tính khóa chính của các lược đồ quan hệ có liên quan, khóa chính của lược đồ quan hệ mới này là các thuộc tính mới thêm vào

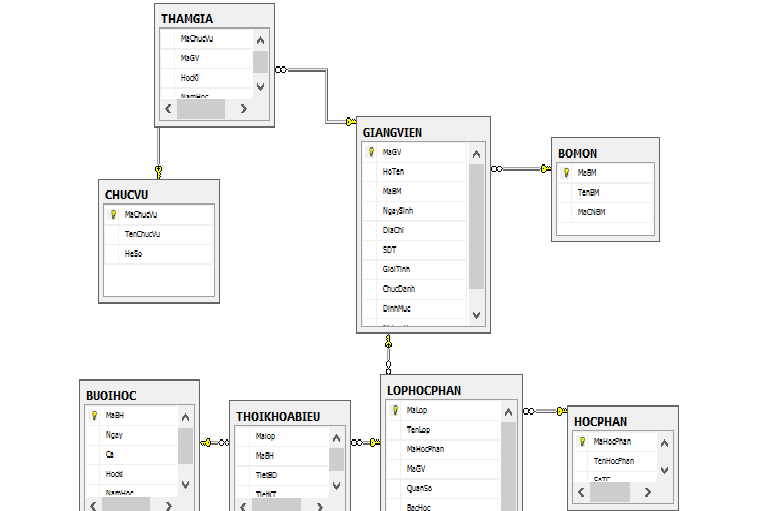
+ Quy tắc 4: xử lý thực thể yếu

Chuyển thành một quan hệ có cùng tên với thực thể yếu và thêm vào thuộc tính khóa của quan hệ liên quan

1. **Áp dụng với bài tập lớn**

Chuyển từ mô hình ER sang mô hình quan hệ

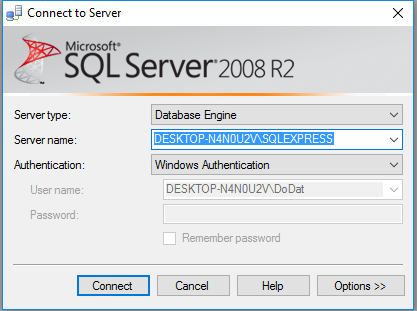
* Bước 1: Chuyển các tập thực thể thành quan hệ
  + BOMON (**Mã BM**, Tên BM, Mã CNBM)
  + GIANGVIEN (**Mã GV**, Tên GV, Ngày sinh, Địa Chỉ, SĐT, Giới tính, Học hàm, Học vị, Chức vụ)
  + LOPHOCPHAN (**Mã lớp**, Tên lớp, Quân số, Bậc học, Phòng học, Địa điểm).
  + HOCPHAN (**Mã học phần**, Tên Học phần, Số tín chỉ)
  + BUOIHOC (**Mã buổi học**, Ca, Ngày, học kì, năm học)
* Bước 2: Chuyển các mối kết hợp
  + Mối kết hợp BOMON-GIANGVIEN: Khóa chính của BOMON sang làm khóa ngoại của GIANGVIEN
  + Mối kết hợp GIANGVIEN-LOPHOCPHAN: Khóa chính của GIANGVIEN sang làm khóa ngoại của LOPHOCPHAN : Mã GV1, Mã GV2.
  + Mối kết hợp LOPHOCPHAN-HOCPHAN: Chuyển khóa chính HOCPHAN sang làm khóa ngoại của LOPHOCPHAN
  + Mối kết hợp LOPHOCPHAN-BUOIHOC: Tạo một quan hệ mới THOIKHOABIEU(Mã Buổi học, Mã lớp,Tiết bắt đầu, Tiết kết thúc)
* Bước 3: Mô hình quan hệ
  + BOMON (**Mã BM**,Tên BM, Mã CNBM)
  + GIANGVIEN (**Mã GV**, Mã BM, Tên GV, Ngày sinh, Địa Chỉ, SĐT, Giới tính, Chức danh, Định mức, Nhiệm vụ)
  + CHUCVU (**Mã chức vụ**, Mã GV, Tên chức vụ, Hệ số, Mức giảm, Học kì, Năm học)
  + LOPHOCPHAN (**Mã lớp**, Mã học phần, Mã GV, Tên lớp, Quân số, Bậc học, Số tiết).
  + HOCPHAN (**Mã học phần**, Tên Học phần, Số tín chỉ)
  + BUOIHOC (**Mã buổi học**, Ca, Ngày, Địa điểm, Học kì, Năm học)
  + THOIKHOABIEU(Mã Buổi học, Mã lớp, Tiết bắt đầu, Tiết kết thúc)



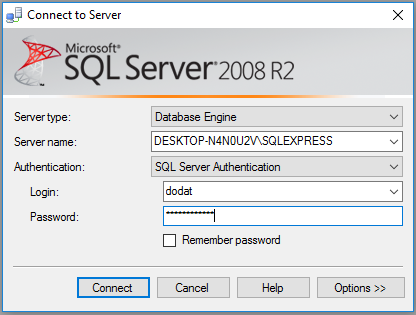
**Chương II. Hệ quản trị CSDL**

1. **Người dùng**

* Xác thực thông qua hệ điều hành: với kiểu xác thực này, SQL server dựa vào windows để cấp bảo mật đăng nhập. Khi dùng đăng nhập vào windows, số định danh tài khoản người dùng được kiểm tra sự hợp lệ. SQL server tích hợp quá trình bảo mật đăng nhập của nó với quá trình bảo mật đăng nhập của windows để cung cấp những dịch vụ này. Khi dùng xác thực bằng hệ điều hành thì không cần thêm chế độ xác thực nào nữa để truy cập SQL server

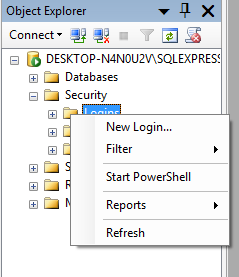


* Chế độ xác thực bằng SQL server: SQL server xác thực đăng nhập bằng cách kiểm tra tài khoản đăng nhập có được thiết lập để truy cập hay không. SQL server thực hiện xác thực tài khoản này bằng cách so sánh tên tài khoản và mật khẩu do người dùng cung cấp để thực hiện kết nối với thông tin tài khoản được lưu trong CSDL. Nếu tài khoản đăng nhập chưa được thiết lập hoặc người dùng nhập sai thì SQL server sẽ từ chối kết nối

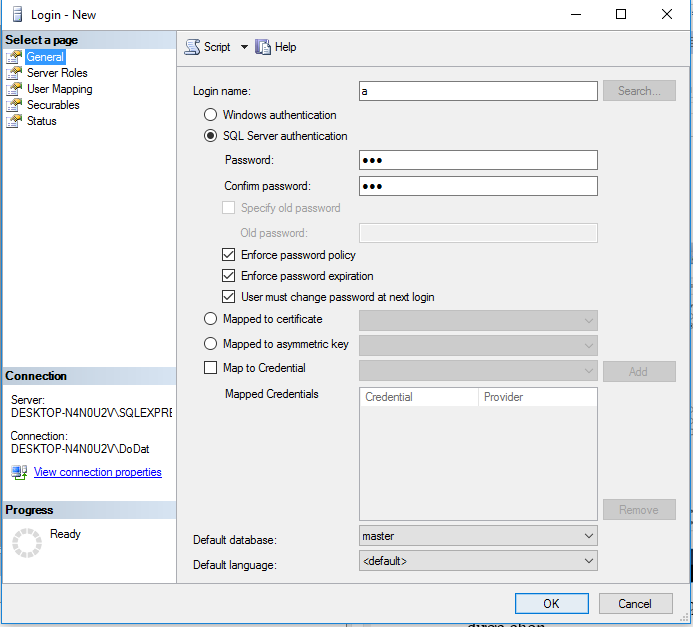


Cách lập tài khoản trong SQL server:

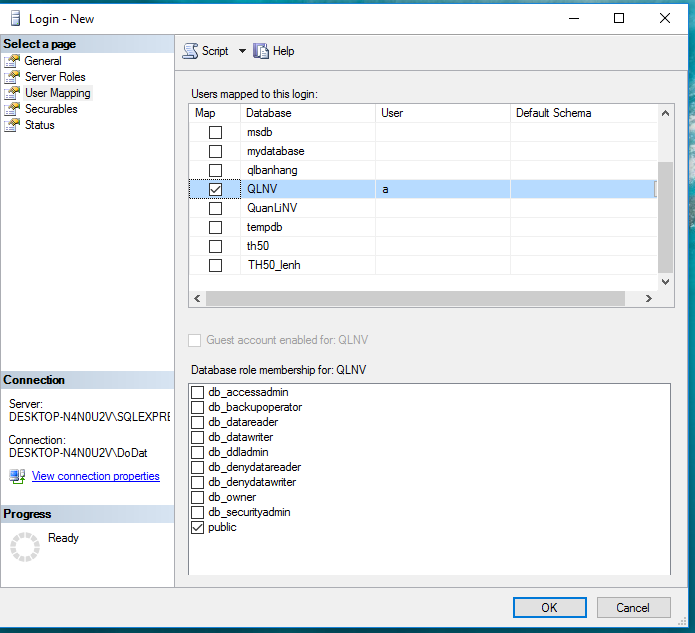
* Chọn mục Security ->logins. Nhấn chuột phải vào new login…



* Sau khi chọn sẽ hiện lên cửa sổ “login new” , chọn general->nhập tên đăng nhập-> chọn sql server authentication->nhập mật khẩu->ok

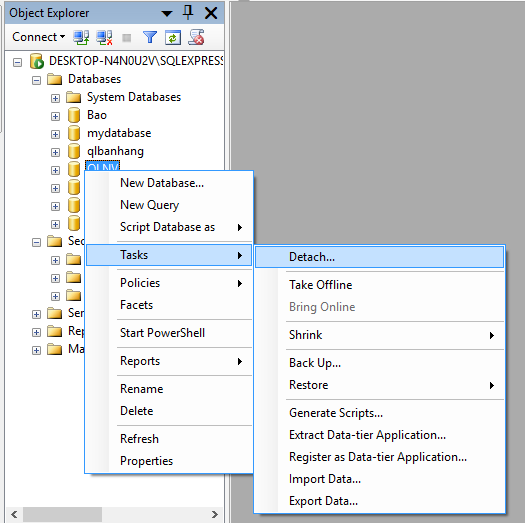


* Chọn user mapping để cấp quyền truy cập đến những CSDL được chọn

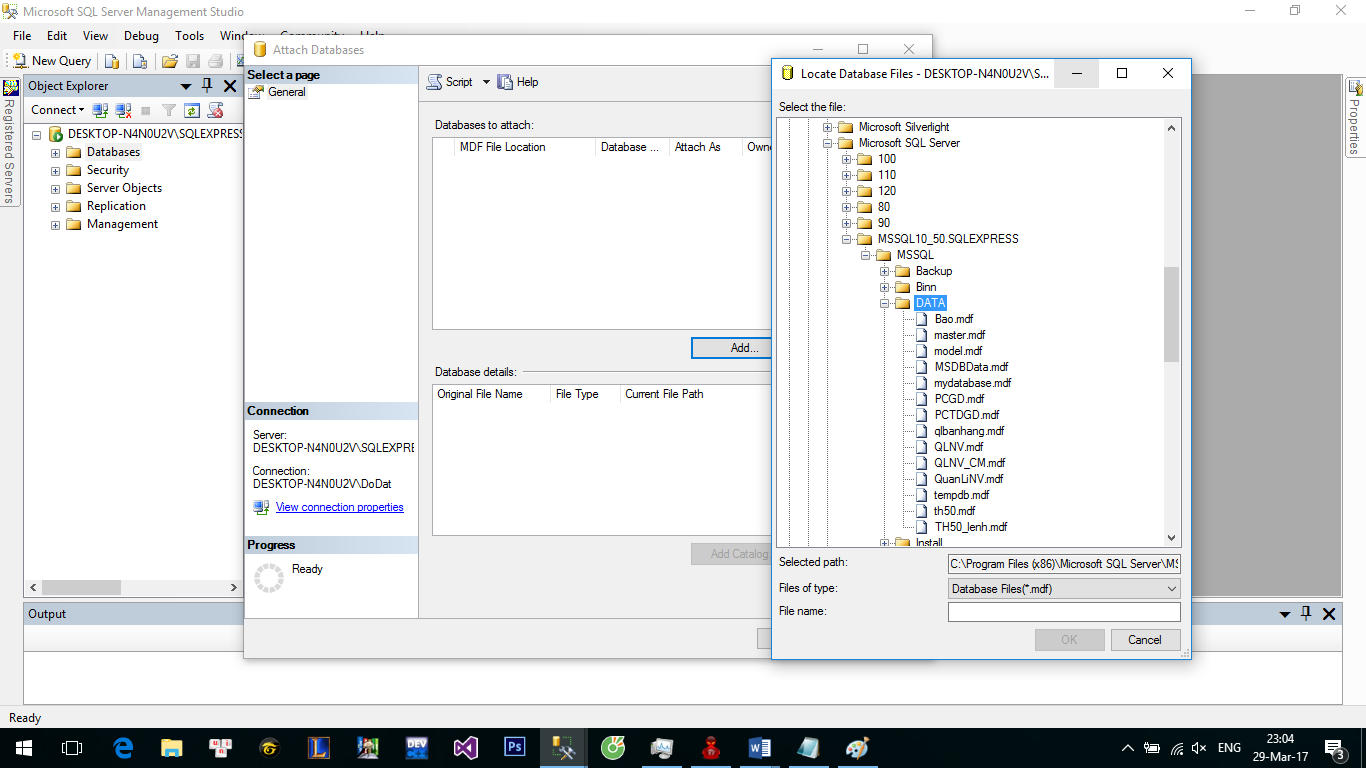
****

1. **Sao lưu và phục hồi dữ liệu**
2. Detach/ Attach

* Detach:
* Chọn database: chọn tasks -> detach
* Việc detach CSDL có thể chưa sẵn sang nếu như có ít nhất một user khác đang kết nối đến CSDL này.

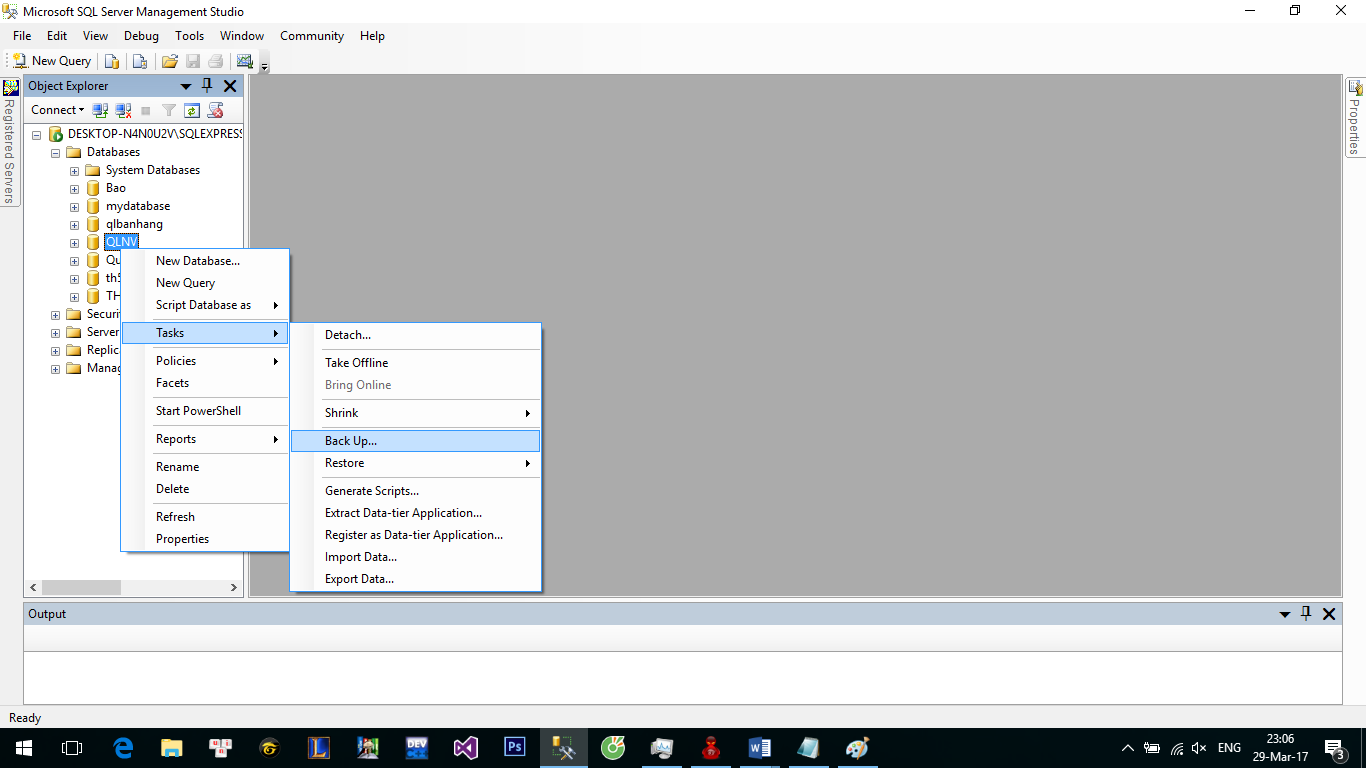
****

* Attach:
* Trong cửa sổ Object browser của database chọn attach.
* Chương trình mở của sổ attach. Chọn nút add và mở thư mục chứa file dữ liệu (.mdf)->ok

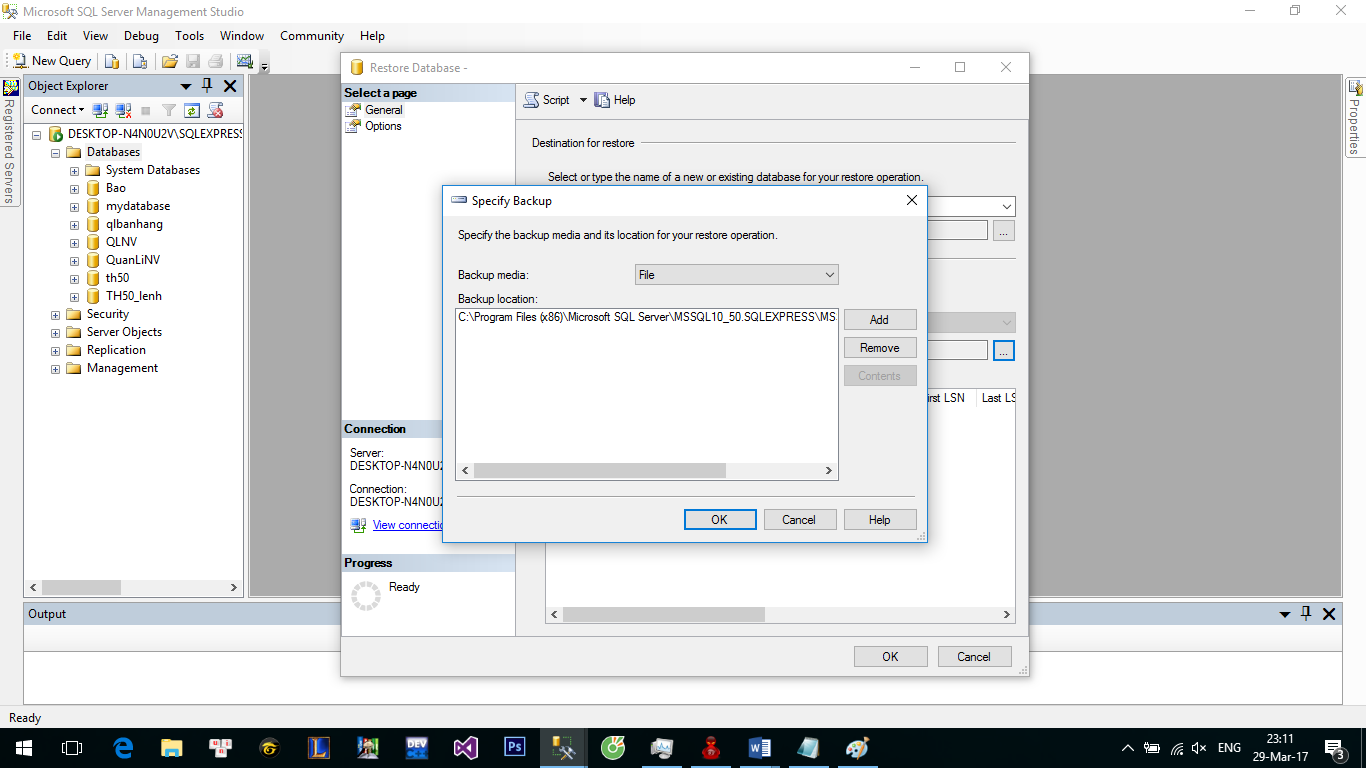


1. Backup/ restore

* Backup : chọn database, nhấn chuột phải->chọn task->backup



* Restore:
* Trong cửa sổ Object browser của database chọn restore-> database…
* Chương trình mở của sổ restore database. Chọn from device->add dữ liệu



**Chương III. Đại số quan hệ**

1. **Cơ sở lý thuyết**
2. **Phép toán tập hợp**

**Phép hợp**: Hợp của hai quan hệ R và S, được ký hiệu là R ∪ S, cho kết quả là một quan hệ chứa tất cả các bộ có trong R hoặc ở trong S hoặc ở trong cả hai. Các bộ trùng lặp bị loại bỏ.

Ví dụ: đưa ra mã giảng viên là nữ hoặc dạy lớp học phần có quân số trên 50

π<Mã\_GV>σ<giớitinh=nữ>(GIẢNGVIÊN)v π <Mã\_GV>σ<quânsố>50>(Lớp học phần)

**Phép giao**: Giao của hai quan hệ R và S , được ký hiệu là R ∩ S , cho kết quả là một quan hệ chứa tất các các bộ có trong cả hai quan hệ R và S.

Ví dụ: đưa ra mã giảng viên là nữ và dạy lớp học phần có quân số trên 50

π<Mã\_GV>σ<giowistinh=nữ>(GIẢNGVIÊN)^ π <Mã\_GV>σ<quânsố>50>(Lớp học phần)

**Phép trừ 2 quan hệ (Minus):** Hiệu của 2 quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S với các bộ giá trị là các bộ giá trị của R sau khi đã loại bỏ đi các bộ có mặt trong quan hệ S.

Ví dụ: đưa ra mã giảng viên dạy lớp có quân số >50 nhưng không là tiến sĩ

π<Mã\_GV>σ<chức danh!=tiến sĩ>(GIẢNGVIÊN)^ π <Mã\_GV>σ<Quân số>50>(Lớp học phần)

**Phép chọn (SELECT)**

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

Phép chọn được ký hiệu là

σ< điều kiện chọn>( R)

trong đó ký hiệu σ được dùng để ký hiêu phép chọn, còn điều kiện chọn là một biểu thức lôgic được chỉ ra trên các thuộc tính của R. Chú ý rằng R nói chung là một biểu thức đại số quan hệ. Kết quả của một biểu thức đại số quan hệ là một quan hệ. Biểu thức đơn giản nhất chính là tên của một quan hệ của một cơ sở dữ liệu. Quan hệ kết quả của phép chọn có cùng thuộc tính như R.

Ví dụ đưa ra lớp học phần có số tiết trên 45

σ< số tiết > 45>( Lớp học phần)

Biểu thức logic chỉ ra trong <điều kiện chọn> được tạo nên từ một số hạng mục có dạng :

<tên thuộc tính> <phép so sánh> <giá trị hằng>

hoặc <tên thuộc tính> <phép so sánh> <tên thuộc tính>

trong đó <tên thuộc tính> là tên của một thuộc tính trong R, <phép so sánh> là một trong các phép toán so sánh {<, <=, =, >=, >, ≠} còn <giá trị hằng> là một giá trị hằng từ miền giá trị của thuộc tính. Các hạng mục có thể được nối với nhau bằng các phép toán lô gic AND, OR, NOT để tạo ra một điều kiện chọn chung

# Phép chiếu (PROJECT)

Nếu ta coi một quan hệ như một bảng thì phép chọn chọn 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 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. Phép chiếu được ký hiệu là:

π<danh sách các thuộc tính>( R)

trong đó π là ký hiệu dùng để biểu diễn phép chiếu và <danh sách các thuộc tính> là một danh sách con các thuộc tính của quan hệ R. Nói chung R là một biểu thức đại số quan hệ. Trường hợp đơn giản nhất nó là tên của một quan hệ của cơ sở dữ liệu. Kết quả của phép chiếu là một quan hệ chỉ có các thuộc tính nằm trong <danh sách các thuộc tính> và có cùng thứ tự như thứ tự của chúng có trong danh sách. Như vậy, cấp của quan hệ kết quả là số các thuộc tính có trong <danh sách các thuộc tính>.

Nếu <danh sách các thuộc tính> chỉ bao gồm các thuộc tính không phải thuộc tính khoá của R thì quan hệ kết quả có thể có những bộ trùng nhau. Phép chiếu 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.

Ví dụ: đưa ra mã giảng viên. họ tên, địa chỉ của giảng viên

π < Mã\_GV, HọTên, địa chỉ>(GIẢNG VIÊN)

cho kết quả là một quan hệ có các thuộc tính Mã\_GV, HọTên, địa chỉ

**Tích Đề-các của 2 quan hệ (Cartesian):**

Tích Đề-các của 2 quan hệ R và S là một quan hệ Q có số ngôi bằng tổng số ngôi của R và S, với các bộ giá trị gồm 2 phần: phần bên trái là một bộ giá trị của R và phần bên phải là một bộ giá trị của S. Như vậy, nếu R có n1 bộ giá trị và S có n2 bộ giá trị, thì Q sẽ có n1 x n2 bộ giá trị.

Ví dụ: đưa ra họ tên Giảng viên dạy lớp học phần

π<Họ tên>σ<GIẢNG VIÊN.MaGV=LỚP HỌC PHẦN.MaGV>(GIẢNGVIÊN x LỚP HỌC PHẦN)

**Phép nối**

Với θ là phép so sánh ≠, =, <, >, ≤• Cho r và s là hai quan hệ tương ứng trên các lược đồ rời nhau R và S.  
• Phép kết nối của các quan hệ r và s, kí hiệu: r |><|  
trên R ∪ S gồm những bộ thuộc tính Decac của r và s sao cho thành  
phần thứ i của quan hệ r có liên hệ s.  
• Vậy kết nối θ : r |><|iθj s là chọn trong r thứ i, j của các quan hệ r, s tương ứng thỏa mãn i r |><|iθj s ={t ∈ r × s: t(θ)}  
Ví dụ: đưa ra mã pb, tên pb, tên dự án mà phòng ban phụ trách

π < Mãpb, ten pb, ten da>(PHONGBAN |><|DUAN)

**Phép chia**

Cho LĐQH R(A1, A2, …, An), S là lược đồ con của R. Giả sử r,s  
là các quan hệ trên R, S tương ứng.  
• Phép chia của quan hệ r cho quan hệ s kí hiệu: r lược đồ R-S gồm các phần tử r sao cho mọi phần tử u ∈ s và ghép t với u ta được phần tử thuộc r:  
• Vậy r **÷**s = {t: ∀u ∈s & <t,u> ∈ r }  
• R **÷S=**T | TxS ∈ R

Ví dụ:

Cho biết mã nhân viên tham gia tất cả các đề án  
– Quan hệ: PHANCONG, DUAN  
– Thuộc tính: MANV  
B1: DA ← πMADA(DEAN)  
  
B2: NV\_DUAN ← πMANV, MADA(PHANCONG)  
  
B3: MA\_NV ← πMANV(NV\_DUAN÷DA)

Phép nối ngoài

Ví dụ:  
Cho biết họ tên nhân viên và tên phòng ban mà họ phụ trách nếu có  
– Quan hệ: NHANVIEN, PHONGBAN  
– Thuộc tinh: TENNV, TENPH  
R1 ← NHANVIEN ]><|MANV=TRPHGPHONGBAN  
KQ ← π < HONV,TENNV, TENPHG (R1)>

1. **Áp dụng bài tập lớn**

* Đưa ra thông tin giảng viên: Tên giảng viên, chức danh, chức vụ, số tiết định mức, miễn giảm, nhiệm vụ

select g.HoTen,g.ChucDanh,c.TenChucVu,g.DinhMuc, c.MucGiam,g.NhiemVu

from GIANGVIEN g,CHUCVU c

where g.MaGV=c.MaGV

* Đưa ra tổng số tiết dạy thực tế, tổng quân số của từng giảng viên

select g.MaGV,g.hoten, SUM(l.sotiet) as TongST, SUM(l.quanso) as TongQS

from GIANGVIEN g,LOPHOCPHAN l

where g.MaGV=l.MaGV

group by g.MaGV,g.HoTen;

* Theo dõi thời khóa biểu ngày, ca , tiết bắt đầu, tiết kết thúc, mã lớp

select b.Ngay, b.Ca , t.TietBD, t.TietKT, l.MaLop

from LOPHOCPHAN l,BUOI1HOC b, THOIKHOABIEU t

where l.MaLop=t.Malop and b.MaBH=t.MaBH

1. **Khai thác với TSQL**

**1. Xử lý theo lô**

Quá trình trong đó một tập lệnh được xử lý cùng lúc được gọi là xử lý theo lô

Ví dụ:

Use QlGD

Select \* from Hocphan, LopHocphan

Update sotc

set sotc=3

where sotiet=45

Go

Mục đích: hay dùng phân tách nhóm để thực hiện độc lập

Ví dụ:

CREATE DaTaBASE QlGD

USE QlGD

CREATE TABLE bomon

(

Mabm char(10),

Tenbm nvarchar(50)

)

SELECT \* FROM bomon

--Sẽ bị báo lỗi

--Sửa lại

CREATE DaTaBASE QlGD

GO

USE QlGD

CREATE TABLE bomon

(

Mabm char(10),

Tenbm nvarchar(50)

)

GO

SELECT \* FROM bomon

GO

Chú thích trong một lô xử lý:

SQL Server hỗ trợ hai hình thức chú thích:

1) --(hai ghạch ngang)

Ví dụ:

USE QlGD

GO

-- Đây là chú thích.

2) /\* ... \*/ (cặp dấu ghạch chéo và dấu sao)

Ví dụ:

SELECT \* FROM GIANGVIEN /\*Đây là chú thích\*/

Chú ý:

Đối với các lệnh CREATE DEFAULT, CREATE PROCEDURE,

CREATE RULE, CREATE TRIGGER, CREATE VIEW không được phép kết hợp với lệnh khác trong cùng một lô.

* **Biến**
* **Khai báo biến**

**DECLARE**  @Tên\_biến Kiểu\_dữ\_liệu[,…]

Kiểu dữ liệu text, ntext hoặc image không được chấp nhận khi khai báo biến.

Ví dụ: Để khai báo các biến lưu trữ giá trị số tiết nhiều nhất,ngày bắt đầu và ngày kết thúc. Sử dụng lệnh DECLARE như sau:

DECLARE @sotietmax int, @tietbatdau int, @tietkt int

* **Gán giá trị cho biến**

Dùng từ khóa SET hay SELECT để gán giá trị cho biến

Cú pháp:

SET @<tên biến cục bộ> = <giá trị>

Hoặc là:

SELECT @<Tên biến cục bộ> = <giá trị>

Ví dụ: gán giá trị cho số tiết lớn nhất là120

DECLARE @sotietmax int

SET @sotietmax= 1200000

Ví dụ:

Tính số tiết học phần lớn nhất của lớp tin k50

DECLARE @sotietmax int

SELECT @sotietmax=max(sotiet)

FROM Lophocphan

WHERE tenlop=’tinhoc50’

* **Xem giá trị hiện hành của biến**

PRINT @Tên\_biến | Biểu\_thức\_chuỗi

Ví dụ: Để xem số tiết lớn nhất, số tiết nhỏ nhất, tổng số tiết của lớp tin học k50. Sử dụng lệnh SELECT và lệnh Print như sau:

DECLARE @sotietmax int ,@sotietmin int,@sotietsum int

SELECT @sotietmax=MAX(sotiet),

@sotietmin= MIN(sotiet),

@sotietsum=Sum(sotiet)

FROM Lophocphan

WHERE tenlop=’tinhoc50’

Print số tiết lớn nhất là ' + convert( varchar(10),@sotietmax)

Print số tiết nhỏ nhất là ' + convert(varchar(10),@sotietmin)

Print ‘Tổng số tiết là ' + cast(@sotietsum as varchar)

* **Các loại biến**
* Biến toàn cục
* Biến cục bộ

**Biến toàn cục**

Biến toàn cục trong SQL Server bắt đầu bằng 2 ký tự @. Ta có thể truy xuất giá trị của các biến này bằng truy vấn SELECT đơn giản

Danh sách các biến toàn cục:

Các biến Ý nghĩa

@@CONNECTIONS Số các kết nối đên máy chủ từ lần khởi động

cuối.

@@CPU\_BUSY Số milliseconds (một phần ngìn giây) hệ

thống đã xử lý từ khi SQL Server được khởi

động

@@CURSOR\_ROWS Số bản ghi trong cursor mở gần nhất.

@@DATEFIRST Giá trị hiện tại của tham số trong lệnh SET

DATEFIRSTquyết định ngày đầu tiên của

tuần.

@@ERROR Mã lỗi của lỗi xảy ra gần nhất

@@FETCH\_STATUS 0 nếu trạng thái lần truy xuất cuối thành

công. -1 nếu có lỗi

@@IDENTITY Giá trị identity gần nhất được sinh ra @@LANGUAGE Tên của ngôn ngữ đang được sử dụng. @@MAX\_CONNECTIONS Số kết nối tối đa có thể. @@ROWCOUNT Số bản ghi bị tác động bởi câu lệnh SQL

gần nhất.

@@SERVERNAME Tên của máy chủ

@@SERVICENAME Tên của dịch vụ SQL trên máy chủ @@TIMETICKS Số milliseconds trong một tick trên máy chủ @@TRANSCOUNT Số giao dịch đang hoạt động trên kết nối

hiện tại

@@VERSION Thông tin về phiên bản của SQL Server

* **Các lệnh điều khiển**

- BEGIN … END

BEGIN...END : Một tập lệnh SQL được thực thi sẽ được đặt trong BEGIN..END.

Cú pháp: BEGIN

<lệnh> | <đoạn lệnh>

END

- IF … ELSE

IF...ELSE: Chúng ta có thể thực thi các tập lệnh SQL khác nhau dựa vào các điều kiện khác nhau.

Cú pháp:

IF <điều kiện>

< lệnh sql1> | <tập lệnh1>

[ ELSE <

< lệnh sql2>|< tập lệnh2> ]

* WHILE

WHILE: Có thể thực thi một lệnh SQL hay một tập lệnh dựa vào điều kiện nào đó. Các câu lệnh được thực thi nhiều lần khi nào điều kiện vẫn còn đúng.

Cú pháp:

WHILE <Điều kiện>

BEGIN

Các\_lệnh\_lặp

END

* BREAK và CONTINUE

Chúng ta có thể dùng từ khóa CONTINUE và BREAK trong vòng lặp while để điều khiển phần thực thi của các câu lệnh.

**Sơ đồ đầy đủ:**

WHILE Biểu\_thức\_logic

BEGIN

Các\_lệnh\_nhóm\_lặp1

[ IF Biểu\_thức\_lặp\_tiếp

CONTINUE ]

[ IF Biểu\_thức\_thoát

BREAK ]

Các\_lệnh\_nhóm\_lặp2

END

Các\_lệnh\_khác

Ví dụ:

USE qlGD

GO

WHILE (SELECT AGV(luong) FROM Giangvien) >5000000

BEGIN

UPDATE detai

SET luong = luong + 500000

SELECT MAX(luong)

FROM Giangvien

IF (SELECT MAX(luong) FROM Giangvien) <5000000

BREAK

ELSE

CONTINUE

END

* GOTO

GOTO: Có thể thay đổi dòng thực thi của chương trình đến một điểm (còn gọi là nhãn).

Các lệnh sau từ khóa GOTO sẽ được bỏ qua và tiến trình thực thi tiếp tục ở vị trí nhãn chỉ ra trong mệnh đề GOTO.

Cú pháp:

GOTO <nhãn>

* RETURN

RETURN: Ta có thể dùng RETURN bất cứ lúc nào để thoát khỏi một đoạn lệnh hay một thủ tục. Các lệnh sau từ khóa RETURN sẽ không được thực thi.

Cú pháp:

RETURN [số nguyên]

* CASE

Cú pháp 1:

CASE <input\_expression>

WHEN when\_expression THEN result\_expression

[WHEN …]

[ ELSE else\_result\_expression ]

END

Cú pháp 2:

CASE

WHEN Boolean\_expression THEN result\_expression

[WHEN ... ]

[ ELSE else\_result\_expression]

END

1. **TRIGGER**

Trigger là một dạng đặc biệt của thủ tục lưu trữ , được thực thi một cách tự động khi có sự thay đổi dữ liệu ( do tác động của các câu lệnh insert, update, delete lên 1 bảng nào đó).

1. Các đặc điểm **TRIGGER**

- Trigger chỉ thực hiện thông qua các sự kiện tự động

- Trigger sử dụng được với khung nhìn

- Khi trigger thực hiện với 2 thủ tục insert và delete thì dữ liệu khi thay đổi sẽ được chuyển sang các bảng **inserted** và **deleted.** Khi câu lệnh update được thực hiện, các dòng dữ liệu cũ chịu sự tác động của câu lệnh được sao chép vào **deleted** và các dòng dữ liệu sau khi sửa được sao chép vào **inserted**.

- Trigger được chia thành 2 loại : **INSTEAD OF** và **AFTER**

INSTEAD OF là loại trigger mà hoạt động của sự kiện gọi trigger sẽ bị bỏ qua và thay vào đó là các lệnh trong trigger được thực hiện.

AFTER là loại ngầm định sẽ thực hiện các lệnh bên trong sau khi thực hiện lệnh kích hoạt trigger

2. Các trường hợp sử dụng trigger

- Tính toàn vẹn

- Khi CSDL chưa được chuẩn hóa thì có thể xảy ra dữ liệu thừa=> cần có trigger thực hiện chuẩn hóa

- Khi xảy ra thay đổi dây chuyền dữ liệu => dùng trigger

3. Khả năng sau của trigger

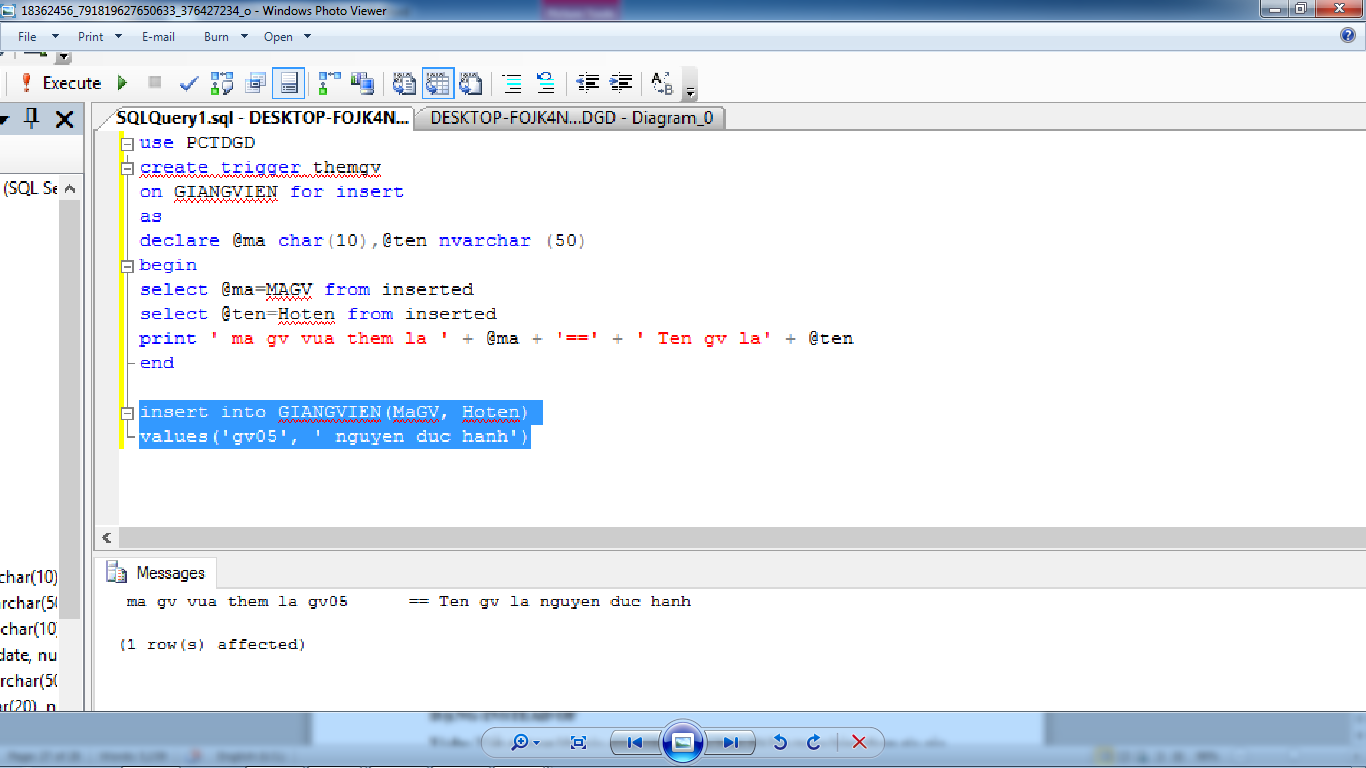
- Một trigger có thể nhận biết thay đổi một hành động làm thay đổi trái phép CSDL.

- Thông qua trigger có thể tạo được các ràng buộc phưc tạp hơn giữa các bảng.

4. Định nghĩa trigger

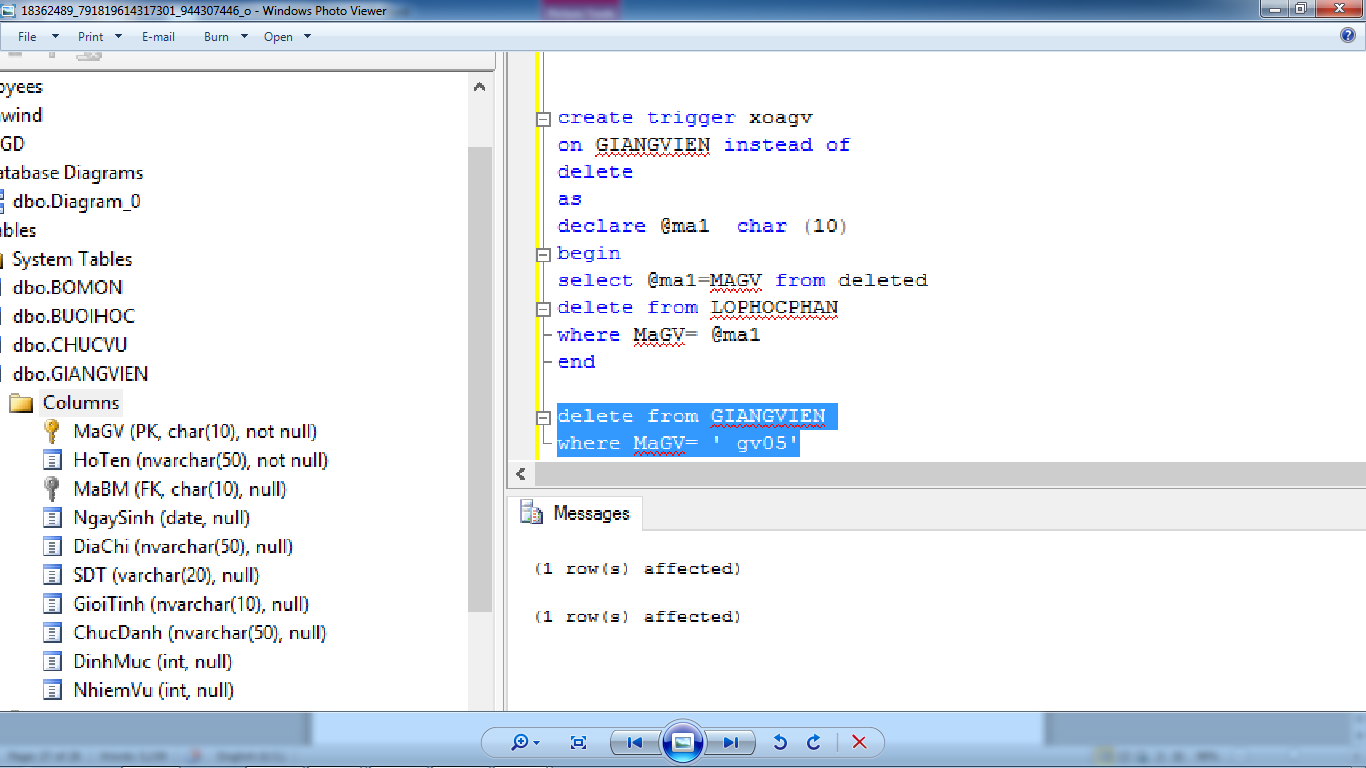
CREATE TRIGGER tên\_triggerON tên\_bảngFOR {[INSERT][,][UPDATE][,][DELETE]}AS[IF UPDATE(tên\_cột)[AND UPDATE(tên\_cột)|OR UPDATE(tên\_cột)]...]các\_câu\_lệnh\_của\_trigger

Ví dụ : Thêm một giáo viên, trigger thực hiện thông báo magv và họ tên



**DẠNG INSTEAD OF**

**Ví dụ:** Viết strigger khi xóa giáo viên thì xóa luôn dữ liệu trong bảng tham gia của gv đó



**Kích hoạt trigger dựa trên sự thay đổi dữ liệu trên cột**

Thay vì kích hoạt trigger khi thay đổi dữ liệu trên một bảng ta có thể kích hoạt trigger khi thay đổi dữ liệu trên một cột.

Ta sử dụng **IF UPDATE,** if updateko sử dụng đc cho sự liện **DELETE.**

Ví dụ tạo trigger không cho phép cập nhật dữ liệu cột magv trên bảng gv

create trigger chan

on GIANGVIEN

for update

as

if update(MaGV)

begin

print 'Ban khong the cap nhat cot magv'

rollback tran

end

update GIANGVIEN

set GIANGVIEN.MaGV='gv1000'

where GIANGVIEN.hoten=N'Lê Minh Hiếu'

Kết quả

