**BUỔI 2 TẠO CSDL QUAN HỆ VÀ NGÔN NGỮ TRUY VẤN T-SQL**

# Mục tiêu của buổi học:

Tạo được csdl quan hệ và tìm hiểu về ngôn ngữ truy vấn T-SQL

# Nội dung

* Với phần tạo cơ sở dữ liệu quan hệ ta sẽ đi tìm hiểu về
* Kiểu dữ liệu
* Cách tạo cơ sở dữ liệu quan hệ
* Cách thiết lập các ràng buộc toàn vẹn
* Về Ngôn ngữ truy vấn T-SQL ta sẽ đi tìm hiểu
* Khái niệm biến
* và các lệnh T-SQL

# Kiểu dữ liệu

* Kiểu dữ liệu SQL là một thuộc tính xác định Kiểu dữ liệu của bất kỳ đối tượng nào.
* Mỗi cột, biến và biểu thức có một Kiểu dữ liệu trong SQL.
* SQL Server cung cấp các Kiểu dữ liệu gồm chuỗi kí tự, số và thời gian/ngày tháng được liệt kê như dưới đây:
* Chuỗi: các chuỗi dữ liệu ký tự
* Số: các số nguyên, số có dấu phẩy động, tiền tệ và các kiểu dữ liệu số khác
* Thời gian (ngày/giờ): ngày, giờ hoặc cả ngày và giờ
* Khác: các giá trị chuỗi và nhị phân lớn, XML, dữ liệu hình học, dữ liệu địa lý, dữ liệu phân cấp

# Kiểu chuỗi

Đối với dữ liệu chuỗi ta sử dụng kiểu dữ liệu chuỗi để lưu trữ.

* Khi cần khai báo chuỗi có độ dài cố định (ví dụ mã sinh viên) ta sử dụng char(n) hoặc nchar(n) với n là số ký tự, mặc định là 1. Riêng nchar các bạn dùng được trong trường hợp chuỗi chứa dữ liệu ký tự unicode.
* Còn đối với các dữ liệu có độ dài không cố định (ví dụ họ tên) khi khai báo ta sử dụng kiểu varchar(n) hoặc nvarchar(n), nvarchar(n) cho trường hợp chuỗi chứa dữ liệu ký tự unicode.
* Hệ thống hỗ trợ đa ngôn ngữ ta nên sử dụng nchar và nvarchar.

# Kiểu số

Dữ liệu số được lưu trữ bởi các kiểu dữ liệu số. Có các kiểu dữ liệu số như:

* Bigint và int dành cho khai báo các số nguyên lớn
* Smallint khai báo số nguyên nhỏ và tinyint cho số nguyên rất nhỏ.
* Kiểu luận lý bit dùng khai báo số nguyên với các giá trị là 1 hoặc 0
* Còn decimall [(p[,s])] và numeric: khai báo số thập phân có chĩ rõ độ chính xác p và tỷ lệ sai số s

# Kiểu số dạng tiền tệ

* Đối với dữ liệu tiền tệ còn có các kiểu dữ liệu như: money, smallmoney, float(n) đối với dấu phẩy động chính xác kép hay real dấu phẩy động chính xác đơn.
* Trong một số trường hợp thường sử dụng kiểu chuỗi làm kiểu dữ liệu thay thế khi lưu trữ dữ liệu số như: số điện thoại, số chứng minh, mã ZipCode

# Kiểu thời gian

Có rất nhiều kiểu thời gian, ở đây mình giới thiệu các bạn một số kiểu cơ bản, như:

* Date: chỉ có ngày
* Time: chỉ có giờ
* Datetime2(n) gồm cả ngày giờ
* Hay datetimeoffset(n) gồm cả múi giờ, ngày giờ

# Các kiểu dữ liệu trong csdl QLDA

Trong csdl qlda, ta thấy:

* thuộc tính honv, tenlot, tennv,.. dữ liệu kiểu chuỗi và có độ dài bất kỳ ta có thể sử dụng kiểu varchar, ước độ dài chuỗi tối đa 15 ký tự
* Đối với mã nhân viên là dữ liệu kiểu chuỗi và có độ dài cố định là 9 ký tự nên ta sử dụng kiểu char(9)
* Ngày sinh là dữ liệu thời gian nên ta chọn kiểu date
* Lương nhân viên là dữ liệu kiểu số ta chọn kiểu float
* Đối với mã phòng ban là dữ liệu kiểu số nguyên, ta chọn kiểu int

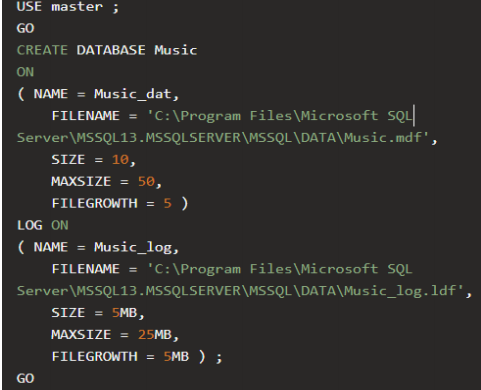
# Tạo csdl

* Một database bao gồm tối thiểu hai file mdf và ldf
* mdf: lưu trữ các đối tượng trong database như table,view, …
* Có thể bổ sung thêm các tập tin lưu trữ khác
* Tổ chức tốt các tập tin lưu trữ giúp tăng tốc độ xử lý
* ldf: lưu trữ quá trình cập nhật/thay đổi dữ liệu
* Hỗ trợ phục hồi dữ liệu
* Hỗ trợ backup/restore dữ liệu
* Các thông số về kích thước file như
* Initial size: kích thước ban đầu
* File growth: tăng trưởng tệp
* Maximum file size: Kích thước tệp tối đa

# Cách tạo csdl quan hệ

Để tạo 1 csdl quan hệ trước tiên ta cần phải tạo database.

* **Tạo Database ta có 2 cách**
* C1:Tạo bằng giao diện bởi công cụ sql server management studio
  + - Phải chuột tại mục Databases
    - Chọn New Database …
    - Nhập tên CSDL
    - Mở rộng giao diện để chọn lại nơi lưu database nếu muốn
    - và nhấn OK
* C2 ta tạo bằng lệnh CREATE DATABASE
* Trong sql server, để tạo dựng cái gì đó chúng ta sẽ sử dụng từ khóa CREATE
* Và công cụ SSMS của mình có cài bộ tool đó là sql prombt để giúp mình code nhanh hơn.(nhắc lệnh)
  + - CREATE DATABASE <Tên CSDL>
    - Lệnh USE <Tên CSDL>



# Tạo Table

* **Để tạo bảng ta cũng có 2 cách**
* C1:Tạo bằng giao diện bởi công cụ sql server management studio
  + - Chọn Database muốn tạo Table
    - Nhấn nút phải chuột tại mục Tables
    - Chọn New/ Table …
    - Tạo các cột (nhập tên cột, chọn kiểu dữ liệu, ràng buộc có cho phép thuộc tính null hay không?)
    - Xác định các cột làm khóa chính (Primary Key) bằng cách chọn thuộc tính, sau đó chọn nút có chứa icon chiếc khóa. (Các bạn vào Views/Toolbars/TableDesigner để hiển thị tab TableDesigner để thuận tiện cho việc đặt khóa chính)
    - Để thay đổi tên cột, ta vào cửa sổ Column Properties ở góc dưới bên phải, nhập tên mới cho cột
    - Chọn save. Nhớ refresh lại database tên bảng mới xuất hiện
* C2 ta tạo bằng lệnh CREATE TABLE
  + - **CREATE TABLE TableName (column1\_name datatype, column2\_name datatype,…)**
    - VD: Tạo bảng phòng ban, có 2 cột mã phòng làm khóa chính và tăng tự động sau mỗi lần thêm, bước nhảy 1, tên phòng kiểu chuỗi có chứa ký tự unicode không được phép null
    - Với MySQL thì chúng ta có thuộc tính [auto\_increment](https://freetuts.net/tim-hieu-auto-increment-trong-mysql-318.html)dùng để xác định một column nào đó tăng tự động, còn SQL Server thì chúng ta có thuộc tính IDENTITY với cú pháp là:
    - **IDENTITY [(giá trị record đầu tiên, bước nhảy sau mỗi lần thêm)]**
    - Vậy ta có câu lệnh:

**CREATE TABLE PHONG\_BAN (**

**MaPhong int IDENTITY (1,1) PRIMARY KEY,**

**TenPhong nvarchar(50) NOT NULL**

**)**

# Chỉnh sửa TABLE với lệnh ALTER

* **Thêm cột mới**

**ALTER TABLE TableName ADD Column\_Name DataType**

VD: ALTER TABLE customers ADD email varchar(255)

* **Xóa cột**

**ALTER TABLE TableName DROP Column\_Name**

VD: ALTER TABLE customers DROP email

* **Đổi kiểu dữ liệu cho thuộc tính trong bảng**

**ALTER TABLE TableName ALTER COLUMN Column\_Name DataType**

VD: ALTER TABLE Persons ALTER COLUMN DayOfBirth year

* **Xóa bảng**

**DROP TABLE TableName**

# Thiết lập khóa ngoại

* **C1: Dùng giao diện**
  + - Chọn Table, nhấn nút phải chuột
    - Chọn menu Script Table As /Insert To / New Query Editor Window (công cụ hiển thị cho chúng ta các lệnh sql để thêm dữ liệu mà k cần phải nhập tay)
* **C2: Dùng lệnh SQL**

**VD:**

**CREATE TABLE NhanVien (**

**MaNV ,**

**column2\_name datatype FOREIGN KEY REFERENCES TableName2 (,…)**

# Nhập dữ liệu

* **C1: Dùng giao diện**
  + - Chọn Table, nhấn nút phải chuột
    - Chọn menu Script Table As /Insert To / New Query Editor Window (công cụ hiển thị cho chúng ta các lệnh sql để thêm dữ liệu mà k cần phải nhập tay)
* **C2: Dùng lệnh SQL**

**INSERT INTO** TableName (Column1, Column2,..)

**VALUES** (Column1 DataType, Column2 DataType,...)

# Lệnh GO

* Lệnh GO không phải là câu lệnh Transact-SQL, mà là một lệnh đặc biệt được nhận biết bởi một số tiện ích MS bao gồm trình soạn thảo mã SQL Server Management Studio.
* Lệnh GO được sử dụng để nhóm các lệnh SQL thành các lô được gửi đến máy chủ cùng nhau.
* VD:

Select \* from employees;

GO -- GO 1

update employees set empID=21 where empCode=123;

GO -- GO 2

* Trong ví dụ trên, các câu lệnh trước GO 1 sẽ chuyển đến sql sever trong một đợt và sau đó bất kỳ câu lệnh nào khác trước GO 2 sẽ chuyển đến máy chủ sql trong một đợt khác. Vì vậy, như chúng ta thấy nó đã tách các lô. Mục đích cuối cùng để các lệnh thực hiện tuần tự thay vì thực hiện đồng thời.

# Tạo Database Diagram

* Để khai báo các ràng buộc khóa ngoại
* Thao tác
  + - Chọn Database muốn tạo Diagram
    - Nhấn nút phải chuột tại mục Database Diagrams
    - Đưa các Table vào Diagram
    - Kéo và thả các cột từ bảng cha đến bảng con
    - Lưu Diagram