**BUỔI 2 TẠO CSDL QUAN HỆ**

# Giới thiệu bài học

* Trước đây, các trường đại học thường lưu trữ thông tin sinh viên bằng hồ sơ giấy. Sau đó, cất giữ hồ sơ trong kho. Khi cần tìm kiếm hoặc thêm/xóa/sửa thông tin nào đó, họ phải mất rất nhiều thời gian để lục tìm lại hồ sơ. Chưa kể, một số trường hợp sau khi thêm hoặc sửa thông tin, hồ sơ sẽ không còn hợp lệ.
* Ngày nay, bất kì tổ chức lớn nào cũng cần xây dựng một hệ thống CSDL, mọi thứ trong CSDL này sẽ được quy ra thành nhiều bảng, có mối quan hệ với nhau. Bạn chỉ cần gõ một câu lệnh SQL ngắn là đã có thể trích xuất được thông tin bạn cần. Việc thêm/xóa/sửa cũng được thực hiện một cách dễ dàng, nhanh chóng.
* Các thao tác truy vấn cũng được sử dụng rất nhiều trong các hệ thống phần mềm hoặc website, chẳng hạn khi các bạn đăng nhập vào facebook thì hệ thống sẽ thực hiện truy vấn dữ liệu để kiểm tra tính hợp lệ của tài khoản đăng nhập.
* CSDL giúp các công việc quản lý hiệu quả, truy vấn dữ liệu nhanh và giúp bảo mật thông tin cao.

Trong bài hôm nay mình sẽ giới thiệu các bạn cách để tạo 1 csdl quan hệ.

# Mục tiêu của buổi học:

* Mục tiêu của buổi học hôm nay chúng ta sẽ
* Tìm hiểu về các kiểu dữ liệu
* Cách tạo csdl quan hệ bao gồm các thao tác tạo database, tạo table và thiết lập các ràng buộc toàn vẹn khóa chính, khóa ngoại

# SLIDE 5 Kiểu dữ liệu

* Kiểu dữ liệu SQL là một thuộc tính xác định Kiểu dữ liệu của bất kỳ đối tượng nào, vd biến, cột, biểu thức...
* SQL Server cung cấp các Kiểu dữ liệu gồm
* Chuỗi: chứa 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
* Và một số kiểu dữ liệu Khác như: các giá trị chuỗi và nhị phân lớn, XML, dữ liệu hình học, dữ liệu địa lý, dữ liệu phân cấp,…

# SLIDE 6, 7 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.

# SLIDE 8 Kiểu số

Dữ liệu số được lưu trữ bởi các kiểu dữ liệu số dùng trong tính toán. SQL Server 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

# SLIDE 9,10 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.
* Với các trường hợp đặc biệt như: số điện thoại, số chứng minh, mã ZipCode ta dùng kiểu dữ liệu CHUỖI để lưu dữ liệu số

# SILDE 11 Kiểu thời gian

Với dữ liệu thời gian SQL server 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, có chứa dấu tiếng việt nên ta có thể sử dụng kiểu nvarchar, 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

# Database

* 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
* Khi ta tổ chức tốt các tập tin lưu trữ sẽ giúp tăng nhanh 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
* Đối với kích thước file bao gồm các thông số như:
* Initial size: kích thước ban đầu
* File growth: kích thước tăng
* Maximum file size: Kích thước tệp tối đa

# Tạo csdl quan hệ

* Câu hỏi đặt ra tại sao chúng ta phải tạo database. Như bình thường nếu chúng ta chỉ truy vấn, đóng, mở file vậy việc gì phải tạo database?
* Các bạn có thể hình dung việc chúng ta tạo database giống như chúng ta đi tạo file vậy đó, database bản chất giống như 1 file để lưu trữ dữ liệu của các bạn.
* Bản chất database không phải lưu 1 file mà nó sẽ lưu ra 3 file, nhưng cái đó chúng ta chưa cần quan tâm, chúng ta chỉ cần tập trung vào ngôn ngữ truy vấn của nó thôi.
* Và sql giúp cho việc chúng ta quản lý dữ liệu được dễ dàng hơn.
* Để Tạo Database các bạn có thể sử dụng công cụ hoặc dùng câu query đều được
* C1: Trước tiên ta sẽ tạo database bằng công cụ
  + - Ta vào thư mục Database, hiện tại chúng ta đang có rất nhiều database ở đây,
    - Phải chuột tại mục Databases/ Chọn New Database …
    - Nhập tên database. VD ta sẽ tạo database QLDA\_GUI
    - Ở đây có một số thuộc tính Intialsize, Maxsize thì các bạn có thể điều chỉnh tăng giảm tùy các bạn
    - Mở rộng giao diện để chọn lại nơi lưu database, thường sẽ lưu ở thư mục mặc định như thế này
    - và nhấn OK Và như vậy nó sẽ tạo 1 database mới ở đây cho các bạn.

Nếu bạn nào tạo xong mà chưa thấy xuất hiện database thì có thể nhấp phải vào thư mục database, ta chọn refresh

* C2 ta tạo database bằng lệnh **CREATE DATABASE Têndatabase**
  + - Các bạn lưu ý: sau khi quét lệnh, Các bạn phải chuột tại thư mục database và chọn refesh lại thì database vừa tạo mới xuất hiện. Nhiều bạn quét lệnh xong k thấy xuất hiện database nên quét lệnh lần 2 thì sẽ bị báo lỗi.
    - Vì tên database là duy nhất nên nếu các bạn quét lệnh 2 lần tức là tạo 2 database cùng tên, lúc này sẽ bị báo lỗi.
    - Ví dụ mình tạo database QLDA\_Q bằng lệnh

CREATE DATABASE QLDA\_Q

GO

USE QLDA\_Q

GO

Mỗi lệnh mình sẽ dùng cặp GO để bọc vào

# 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 SSMS
* Lệnh GO được sử dụng để nhóm các lệnh SQL thành các lô được gửi đến máy chủ cùng nhau.
* Mục đích sử dụng chủ yếu để các lệnh trong file sql có thể  thực hiện tuần tự, thực hiện xong lệnh trước mới thực hiện lệnh sau (thay vì thực hiện đồng thời các lệnh cùng lúc)
* 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 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) hoặc phải chuột vào thuộc tính chọn Set primary key.
    - Chọn save. Nhớ refresh lại database tên bảng mới xuất hiện.
    - Để thay đổi tên cột, ta vào cửa sổ Column Properties ở góc dưới bên phải, nhập tên mới cho cột
* C2 ta tạo table bằng lệnh CREATE TABLE
  + - **CREATE TABLE TableName (column1\_name datatype, column2\_name datatype,…)**
    - VD: Tạo bảng phòng ban, có 2 cột mã phòng và tên phòng, 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 (**

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

**TenPhg nvarchar(50) NOT NULL default N’Tên phòng’**

**)**

# Chỉnh sửa TABLE với lệnh ALTER

CREATE TABLE KHACHHANG (

Makh int IDENTITY(1,1) PRIMARY KEY,

Tenkh nvarchar(50) DEFAULT N'Tên Khách Hàng' NOT NULL

)

--Thêm, sửa, xóa các cột trong bảng

--Thêm cột email cho bảng khách hàng

ALTER TABLE KHACHHANG ADD email varchar(30)

--Sửa kiểu dữ liệu của cột email trong bảng kh

ALTER TABLE KHACHHANG ALTER COLUMN email char(20)

--Xóa cột email trong bảng khách hàng

ALTER TABLE KHACHHANG DROP COLUMN email

--Xóa bảng khách hàng

DROP TABLE KHACHHANG

# Thiết lập khóa ngoại

* Khóa chính đặc điểm của nó là UNIQUE duy nhất và NOT NULL , khóa chính giúp cho khóa ngoại tham chiếu đến được từ bảng khác qua bảng mình.
* Nhưng vấn đề đặt ra ta thiết lập khóa ngoại để làm gì?
* VD ta có 2 bảng nhân viên và phòng ban: NhanVien(MaNV, HoNV, TenLot)

PhongBan(MaPhg,TenPhg)

**CREATE TABLE NHANVIEN (**

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

**Honv nvarchar(50) NOT NULL,**

**TenLot nvarchar(50) NOT NULL,**

**)**

* Vậy làm sao biết nhân viên này thuộc phòng ban nào? Hỏi như vậy thôi thì nhìn vô các bạn đã biết câu trả lời sao rồi đúng không?
* Vậy để biết nhân viên này thuộc phòng ban nào có phải là chúng ta cần phải thêm 1 trường phòng vô trong bảng Nhân viên đúng không? Tên cột thêm vào không không nhất thiết giống tên cột khóa chính mã phòng trong bảng phòng ban, tuy nhiên bắt buộc phải cùng kiểu dữ liệu.
* VD ta thêm dữ liệu: nv1 ở phòng 4, tuy nhiên trong bảng phòng ban chỉ có phòng 1,2,3 lúc này dẫn đến dữ liệu bị sai. Mà tính chất của csdl là đúng chứ không sai. Cho nên ta cần tạo khóa ngoại cho nó, cột phòng ở bảng nhân viên sẽ tham chiếu đến dữ liệu khóa chính (mã phòng) ở bảng phòng ban mà cùng kiểu dữ liệu với nó.
* Tập hợp các các giá trị của khóa ngoại sẽ luôn nằm trong tập hợp các giá trị của khóa chính đảm bảo được sự ràng buộc toàn vẹn dữ liệu, không có trường hợp tham chiếu đến dữ liệu không tồn tại.
* Tức là nếu ta chỉ có phòng ban 1 2 3, thì khóa ngoại phòng trong bảng nhân viên cũng chỉ có thể là 1, 2, 3, không thể có giá trị khác.
* Csdl giống như 1 nền móng, nếu csdl không tốt, bị lỗi thì đưa vào ứng dụng chạy hư liền. Có ràng buộc, có quan hệ, các khóa ràng buộc lẫn nhau, các bảng có quan hệ với nhau để đảm bảo toàn vẹn dữ liệu.Nên mới phải xảy ra chuyện rườm rà là khóa chính, khóa ngoại như vậy.
* Tóm lại điều kiện để tạo khóa ngoại
* Cần tham chiếu tới khóa chính bảng khác
* Cùng kiểu dữ liệu, cùng số lượng trường có sắp xếp. Tức là nếu khóa chính có 2 trường thì khóa ngoại cũng có tối đa 2 trường
* Các bạn có thể thiết lập khóa ngoại ngay khi tạo bảng hoặc sau khi tạo bảng.
* C1: Dùng công cụ
  + - Hiển thị lược đồ csdl
    - Bảng nhân viên sẽ tham chiếu đến bảng phòng ban thông qua mã phòng.
    - Ta bấm chuột vào khóa chính mã phòng ở bảng PHONGBAN kéo đến khóa ngoại Phg trong bản NHANVIEN
    - Nó tạo ra một mối quan hệ relationship tên là FK\_DEAN\_phongban, các bạn có thể đổi tên khác nếu muốn. Ta thấy bảng cha là PHONGBAN khóa chính là Maphong, bảng con là NHANVIEN khóa ngoại là Phong
    - Chọn OK OK
    - Như vậy đã có 1 sự kết nối tạo ra giữa 2 bảng NHANVIEN và PHONGBAN
    - Các bạn có thể phải chuột bên ngoài lược đồ, chọn Arrange table để nó tự động sắp xếp các bảng thay vì tự động kéo các bảng theo ý của mình
* C2: Dùng lệnh SQL
  + - VD: ThiẾt lập khóa ngoại cho bản DEAN
    - **Tạo khóa ngoại ngay khi tạo bảng:**

**CREATE TABLE NHAN\_VIEN (**

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

**TenNV nvarchar(50) NOT NULL default N’Tên NV’**

**Phong int ,**

**FOREIGN KEY(Phong) REFERENCES PHONGBAN(MaPhong)**

**)**

* Khóa ngoại là Phong, REFERENCES tham chiếu đến bảng phòng ban thông qua khóa chính MaPhong
* Nếu tham chiếu đến 2 trường thì đặt 2 trường vô tương tự khóa chính thôi
  + - **Tạo khóa ngoại ngoài bảng:**

**ALTER TABLE DEAN ADD FOREIGN KEY(MaPhong) REFERENCES PHONG\_BAN(MaPhong)**

* Chúng ta có thể đặt tên cho khóa ngoại để sau này có thể hủy khóa ngoại dể dàng.

**ALTER TABLE DEAN ADD CONSTRAINT FK\_DEAN FOREIGN KEY(MaPhong) REFERENCES PHONG\_BAN(MaPhong)**

* Hủy khóa ngoại:

**ALTER TABLE DEAN DROP CONSTRAINT FK\_DEAN**

* Để xem khóa chính, khóa ngoại mới tao, ta hiển thị lược đồ csdl để xem.
* Ta thấy các cặp, các mũi tên. Nhân viên tham chiếu đến phòng ban, các bạn nhìn đường đi cũa mũi tên sẽ biết ngay, mũi tên nằm ờ bảng nào, bảng đó nắm khóa chính, dấu vô cực đứng nằm ở đâu thì bảng đó nắm khóa ngoại.
* Bảng nhân viên tham chiếu đến bảng phòng ban thông qua khóa chính của bảng phòng ban là mã phòng ban. Nhìn vào diagram chúng ta sẽ nắm được thông tin.

# Nhập dữ liệu

Bậy giờ mình sẽ thử insert, update dữ liệu để xem tác dụng như nào nhé.

* Để tiến hành nhập liệu cho bảng ta có thể
* **C1: Dùng công cụ**
  + - Chọn tên bảng cần thêm dữ liệu, 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,...)

Ta thêm dữ liệu phong1,phong2 vào bảng phòng ban.

Sau đó thêm dữ liệu nv1 vào bảng nhân viên nhưng mã phòng ta chọn là phong3 k có trong danh sách trong bảng phòng ban, ta sẽ chạy thử xem như thế nào.

Nó sẽ báo lỗi ngay, nó sẽ báo lỗi k thể nào insert vào được bởi vì khóa ngoại k đúng

Bây giờ mình sẽ thử đưa vào phòng 1, chạy ok ngay.

Tạo thêm 1 lệnh thêm dữ liệu nv1 ở phong2 trong bảng nhân viên, lúc này bị báo lỗi nhưng k phải lỗi do khóa ngoại mà là do khóa chính vì nv1 đã tồn tại rồi.

Như vậy các bạn có thể thấy sự ràng buộc dữ liệu rất là chắc chắn, đó là mô hình quan hệ

# BÀI LAB2

* Hôm nay các bạn sẽ làm bài 1 trong file LAB2. Các bạn sẽ tiến hành xây dựng csdl QLDA như mô tả bên dưới.
* Chúng ta sẽ tiến hành tạo database QLDA, tạo các bảng với các cột của bảng như trong lược đồ csdl đã cho. ( kiểu dữ liệu của các cột như trong bảng mô tả các thuộc tính) bằng lệnh query
* Lưu ý, tạo bảng cha (bảng chứa khóa chính) trước, tạo bảng con (bảng chứa khóa ngoại) sau
* Sau đó dựa vào lược đồ csdl các bạn sẽ thiết lập khóa ngoại của các bảng bằng lệnh query.
* Các bạn tiến hành nhập liệu cho bảng để chúng ta có được dữ liệu thực hiện truy vấn ở bài 2. Lưu ý nhập liệu cho bảng cha trước, bảng con sau. Khi chúng ta muốn xóa dữ liệu thì làm ngược lại xóa bảng con trước, cha sau.

--làm bài lab2