**Kiểu dữ liệu là gì?**

KIỂU DỮ LIỆU – DATA TYPE là một quy trình về cấu trúc, miền giá trị của dữ liệu có thể nhập vào và tập các phép toán / toán tử có thể tác động lên miền giá trị đó.

Kiểu dữ liệu trong SQL là:

- Dạng dữ liệu chỉ định mà mỗi Column (Trường thuộc tính) có thể lưu trữ để thể hiện dữ liệu thuộc tính của nó.

- Quyết định kích thước mà dữ liệu trong Column đó sử dụng trong vùng nhớ.

- Dựa trên cơ sở các Kiểu dữ liệu do chuẩn ANSI/ISO cung cấp.

- Trong đó, 16/143 kiểu được sử dụng thông dụng trong các Hệ quản trị cơ sở dữ liệu thương mại hiện nay.

**Tại sao phải dùng các kiểu dữ liệu riêng?**

Tại sao không thể dùng chung một kiểu dữ liệu cho tất cả các Column (Trường thuộc tính) trong một Table hoặc một Database?

🡪 Nếu việc lưu trữ các dữ liệu cùng một kiểu, bạn không thể phân loại đâu là ngày tháng, đâu là chuỗi, đâu là số…. Vì vậy việc thực hiện các toán tử, tìm kiếm cũng trở nên rất khó khăn trong quá trình truy vấn dữ liệu.

Mặt khác, đặt ra một bài toán thực tế: Khi bạn thiết kế một CSDL thương mại, việc cài đặt Database lưu trữ rất quan trọng. Ta có một ví dụ nhỏ như sau, giả sử:

Trong một Table, cứ 1 record (bản ghi) tương ứng 1 byte bộ nhớ.

Một ngày bạn lưu 1.000.000 records sẽ chiếm 1.000.000 byte.

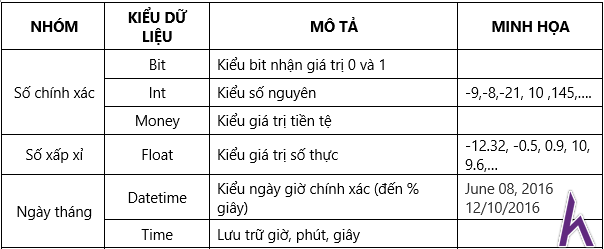
Nếu lưu một năm sẽ là 365.000.000 byte.

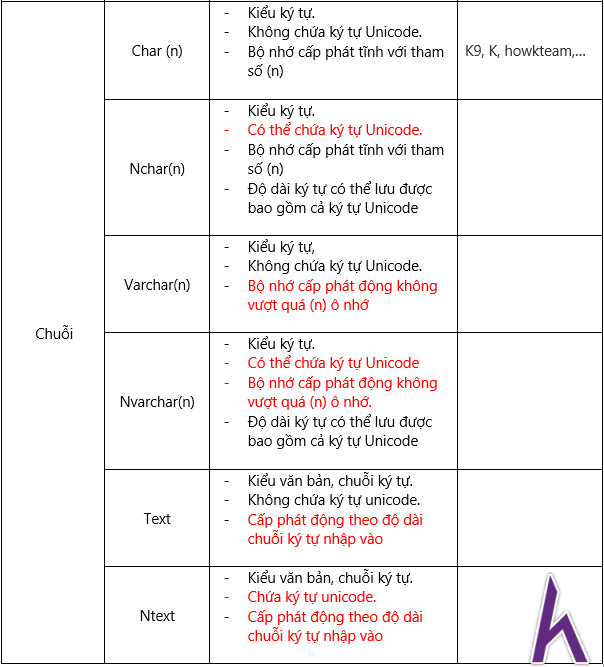
Vậy nếu lưu trữ thừa 1byte/ ngày, bạn sẽ gây ra thất thoát dung lượng đến 365.000.000 byte /năm. Ngược lại, việc thiếu dung lượng gây ra trì trệ hệ thống như lỗi thiếu bộ nhớ, không đủ dung lượng lưu trữ. Đó chỉ là một bài toán minh họa nhỏ, còn thực tế thường gấp nhiều lần như vậy.

Ngày nay, dung lượng máy tính thường khá lớn, việc thất thoát dung lượng lưu trữ nên trên có thể tạm chấp nhận. Nhưng còn đối với lập trình di động thì sao? Theo xu hướng sử dụng thiết bị hiện đại, bạn thấy rõ rằng việc tiết kiệm dung lượng là rất quan trọng. Các thiết bị nhỏ không thể lưu trữ dung lượng quá lớn, vì vậy nếu xảy ra thất thoát dữ liệu sẽ gây trì trệ ảnh hưởng đến hệ thống thiết bị.

**Danh sách các kiểu dữ liệu**

Theo chuẩn ANSI/ISO SQL, các kiểu dữ liệu trong SQL được quy định và phân loại thành các nhóm chính như sau:





**Các kiểu dữ liệu hay sử dụng:**

--int: kiểu số nguyên, tập hợp N. Vd: -9, -8, 7, 5

--float: kiểu số thực, tập hợp R. Vd: -0.6, -0.1, 9.6

--char: kiểu ký tự. Bộ nhớ cấp phát cứng. Không viết tiếng Việt được. Vd: A, B, C, D.   
char(10) 🡪 10 ô nhớ này không ai được động. Luôn giữ 10 ô nhớ

--varchar Kiểu ký tự. Bộ nhớ cấp phát động. Không viết tiếng Việt được Vd: A, a, B, b, f. varchar(10) 🡪 10 ô nhớ này chỉ được lấy khi có dữ liệu nằm bên trong  
‘Kteam’ 🡪 chỉ dùng 5 ô nhớ

--nchar Kiểu ký tự. Viết tiếng Việt được, giống char

--nvarchar Kiểu ký tự. Viết tiếng Việt được. Bộ nhớ cấp phát động

--date Lưu trữ ngày, tháng, năm, giờ

--time Lưu trữ giờ, phút, giây

--bit Lưu giá trị 0 và 1

--text Lưu văn bản lớn

--ntext Lưu văn bản lớn có tiếng Việt

Thêm n ở đầu là có dùng unicode

**Một số điều cần lưu ý**

**Cấp phát động và cấp phát tĩnh**

Ở kiểu dữ liệu char(n), varchar(n) ta có thể thấy sự khác biệt giữa hai kiểu dữ liệu này là khả năng cấp phát bộ nhớ. Vậy chúng khác nhau như thế nào?

Ví dụ sau sẽ giúp bạn dễ hiểu hơn khái niệm này

USE SQLDBUI

GO

CREATE TABLE TEST

(

--Cấp phát bộ nhớ tĩnh cho column MASV

MASV CHAR(10),

--Cấp phát bộ nhớ động cho column MALop

MALop VARCHAR(50)

)

Với đoạn code trên, khi bạn nhập dữ liệu “Kteam” cho 2 Column trên

Ở column MASV sẽ CẤP PHÁT TĨNH và luôn lưu trữ trên 10 ô nhớ trong bộ nhớ, với 5 ký tự nhập vào và 5 ký tự rỗng.

Ở column MALop sẽ CẤP PHÁT ĐỘNG theo số ký tự nhập vào. Ở đây, sẽ là 5 ký tự tương ứng “Kteam” không có ký tự rỗng theo sau.

Việc sử dụng varchar(n) giúp việc lưu trữ trở nên linh hoạt hơn và tiết kiệm bộ nhớ hơn. Tuy nhiên trong thực tế, có lúc bạn cần đảm bảo số lượng ký tự lưu trữ có độ dài giống nhau, chính xác ở mọi dữ liệu trong một trường thược tính, vì vậy cần thiết sử dụng char(n) trong trường hợp này.