Lệnh xóa 1 cột trong bảng DELETE FROM NHANVIEN WHERE MANV='A01'

# MỤC TIÊU

Mục tiêu của buổi học hôm nay chúng ta sẽ đi tìm hiểu về các hàm xử lý chuỗi và các hàm ngày tháng năm

# SLIDE 21 CÁC HÀM XỬ LÝ CHUỖI

* Hàm xử lý chuỗi trong SQL được sử dụng để thao tác với chuỗi. Các bạn lưu ý rằng các hàm này chỉ áp dụng được cho câu lệnh SELECT.
* Sau đây là một số hàm xử lý chuỗi quan trọng trong SQL.
* Hàm **LEN(‘String’):** trả về độ dài (hay số lượng ký tự) trong chuỗi, bao gồm cả các khoảng trắng đầu chuỗi

VD: SELECT LEN(' FPT POLY ')=> 9

* Hàm **LTRIM(‘String’):**trả về chuỗi mới đã xóa các ký tự khoảng trắng đầu chuỗi

VD: SELECT LTRIM(' FPT ')=> 'FPT '

* Hàm **RTRIM(‘String’):**trả về chuỗi mới đã xóa các ký tự khoảng trắng cuối chuỗi

VD: SELECT RTRIM(' FPT ') => ' FPT'

* Hàm **LEFT(‘String’,n):** dùng để lấy n ký tự đầu tiên bên trái của 1 chuỗi. Hàm LEFT gồm 2 đối số, đối số 1 là chuỗi đầu vào string, đối số 2 là chiều dài n chuỗi con

VD: SELECT LEFT(' FPT POLY ',4)=> ' FPT'

* Hàm **RIGHT(‘String’,n):** dùng để lấy n ký tự đầu tiên bên phải của 1 chuỗi, hàm RIGHT cũng gồm 2 đối số, đối số 1 là chuỗi đầu vào string, đối số 2 là chiều dài n chuỗi con

VD: SELECT RIGHT(' FPT POLY ',4)=> 'OLY '

# SLIDE 22 CÁC VD HÀM XỬ LÝ CHUỖI

* Chuỗi SQL Server gồm 10 ký tự bao gồm cả khoảng trắng giữa 2 từ nên LEN('SQL Server')=10
* Ở VD 2: trước chuỗi SQL Server ta thấy có 2 ký tự khoảng trắng nên LEN(' SQL Server ')=12. Ta không quan tâm đến khoảng trắng ở cuối chuỗi.
* LEFT('SQL Server', 3)thì từ trái qua ta lấy 3 ký tự nên KQ là SQL
* LTRIM(' SQL Server ') ta xóa bỏ các khoảng trắng ở đầu nên kq trả về chuỗi 'SQL Server '
* Còn RTRIM(' SQL Server ') ta xóa bỏ các khoảng trắng ở cuối nên kq trả về chuỗi ' SQL Server '
* Với lệnh LTRIM(RTRIM(' SQL Server '))ta thực hiện theo thứ tự từ trong ra ngoài, hàm RTRIM xóa bỏ các khoảng trắng ở cuối, hàm LTRIM xóa bỏ các khoảng trắng ở đầu, kq trả vể chuỗi con đã được xóa bỏ các khoảng trắng ở cả 2 đầu.

# SLIDE 23: HÀM CHARINDEX

* Hàm charindex() trả về vị trí xuất hiện của chuỗi con trong chuỗi cha [,tùy chọn vị trí bắt đầu xét]. Nếu không chỉ rõ tham số thứ 3 thì mặc định tính từ vị trí đầu chuỗi.
* Cú pháp CHARINDEX(‘CHUOICON’,’CHUOICHA)

VD: SELECT CHARINDEX('Se','SQL Server') ta tìm vị trí của chuỗi con Se trong chuỗi cha SQL Server. Ở đây ta không chỉ rõ tham số thứ 3 là vị trí bắt đầu xét nên ta sẽ tính từ vị trí đầu tiên từ trái qua. Thì rõ ràng ta thấy chuỗi Se xuất hiện ở vị trí thứ 5, nên kq trả về 5

# SLIDE 24: HÀM SUBSTRING

* Hàm SUBSTRING() để lấy chuỗi con trong một chuỗi với cú pháp:

SUBSTRING(‘String’,vị trí lấy,độ dài chuỗi con)

VD: Ta có đoạn lệnh sau: Ta khai báo biến @fullname kiểu varchar kích thước 25 ký tự

DECLARE @FullName VarChar(25) -- gán giá trị cho biến @fullname là 'www.poly.edu.vn'

SET @FullName = 'www.poly.edu.vn'

SELECT SUBSTRING(@FullName, 5, 4)

* Ta dùng hàm SUBSTRING lấy ra chuỗi con từ vị trí 5 trong chuỗi cha, ta lấy 4 ký tự, nên kq trả về chuỗi poly

# SLIDE 25: HÀM REPLACE

* Hàm REPLACE(search,find,replace): trả về một chuỗi mới mà chuỗi find đã được thay thế bởi chuỗi replace trong chuỗi search.

VD: SELECT REPLACE ('0973-456-226','-','.'): vd này trong chuỗi '0973-456-226'ta đã thay thế ký tự - bởi ký tự . nên kq trả vể chuỗi '0973.456.226'

# SLIDE 26: CÁC HÀM XỬ LÝ CHUỖI KHÁC

* Ngoài ra còn có các hàm xử lý chuỗi khác như:
* REVERSE(String ) để đảo ngược chuỗi.

VD: SELECT REVERSE ('ABCDE') sẽ trả về EDCBA

* Hay hàm SELECT LOWER ('ABCDE') để trả về chuỗi gồm các ký tự thường
* Và SELECT UPPER ('ABCDE') để trả về chuỗi gồm các ký tự HOA
* Còn **Hàm SPACE** trong SQL Server được sử dụng để trả về một chuỗi có số lượng khoảng trắng được chỉ định trong câu lệnh.

Cú pháp SPACE(sokhoangtrang). Tham số đầu vào phải là số không âm (>=0). Nếu tham số đầu vào là số âm hàm sẽ trà vể NULL

VD SELECT SPACE (5) => trả về chuỗi với 5 ký tự khoảng trắng

VD SELECT SPACE (-5) => NULL

# SLIDE 27: DEMO SỬ DỤNG CÁC HÀM

* VD này ta sẽ hiện thị 4 cột,
* cột đầu là DCHI,
* cột 2 lấy 2 ký tự đầu tiên từ trái qua của DCHI nhờ hàm LEFT(DCHI,2) dùng mệnh đề AS đặt tên là LEFT2
* cột 3 lấy 3 ký tự đầu tiên từ phải qua của DCHI nhờ hàm RIGHT(DCHI,3) dùng mệnh đề AS đặt tên là RIGHT3
* cột 4 với hàm SUBSTRING ta sẽ lấy chuỗi con từ chuỗi DCHI, từ vị trí thứ LEN(DCHI)-2 và lấy 3 ký tự. len(DCHI) trả về độ dài chuỗi DCHI ta trừ cho 2 tìm được vị trí cần lấy chuỗi con.
* Ở hàng đầu tiên, LEN(DCHI)=22, 22-2=20, vậy ta sẽ trong chuỗi DCHI từ vị trí 20, ta lấy 3 ký tự nên kq trả về chuỗi HCM

# SLIDE 28: SẮP THỨ TỰ KHI XỬ LÝ CHUỖI

* VD: Khi thực hiện truy vấn hiển thị danh sách nhân viên và sắp xếp theo MANV.

SELECT \*FROM NHANVIEN ORDER BY MANV

* Kết quả truy vấn ta thấy nhân viên có manv 1000 xuất hiện trước nhân viên có manv 999.
* Lý do là gì? Thì trong trong csdl QLDA MANV kiểu varchar và đối với kiểu varchar nó sẽ xét chuỗi từ trái qua phải, và sắp xếp theo thứ tự từ trái qua phải, manv 1000 (bắt đầu là 1) sẽ xuất hiện trước manv 999 (bắt đầu là 9) do vậy ta sẽ nhận được kết quả không như ta mong đợi.
* Vậy nên kKhi sắp xếp một cột kiểu chuỗi chứa số, cột cần GROUP BY ta sẽ thực hiện chuyển đổi kiểu chuỗi thành giá trị số. Trong trường hợp này ta sẽ dùng hàm CAST hoặc CONVERT để ép kiểu cột MANV về kiểu int rồi mới thực hiện sắp xếp.

SELECT \*FROM NHANVIEN ORDER BY CAST(MANV AS INT)

* Thì kết quả truy vấn lúc này đã đúng như ý định của chúng ta, nv mã 999 xuất hiện trước nv mã 1000.

# SLIDE 29: PHÂN TÁCH KÝ TỰ KHI XỬ LÝ CHUỖI

* Một vấn đề thường gặp khi làm việc với chuỗi ký tự đó là phân tách ký tự.
* Thông thường khi lập một danh sách, một số csdl thường nhập cả họ và tên vào một cột. Vậy để tách họ, tên một người nào đó trong danh sách thì ta phải làm thế nào?
* Mình nhờ 1 bạn giúp mình đọc câu lệnh SQL tạo biến bảng @NV gồm 1 cột HOVATEN chứa họ, tên lót và tên của nhân viên trong bảng NHANVIEN
* Ta thực hiện truy vấn từ biến bảng @NV, hiển thị 3 cột: cột 1 chứa Họ và tên, cột 2 chứa Họ, cột 3 chứa Tên

DECLARE @NV TABLE (HOVATEN NVARCHAR(100))

INSERT INTO @NV

SELECT HONV+ ' '+TENLOT+' '+TENNV

FROM NHANVIEN

SELECT HOVATEN,

LEFT(HOVATEN, CHARINDEX(' ', HOVATEN) - 1) AS Họ,

RIGHT(HOVATEN, LEN(HOVATEN) - CHARINDEX(' ', HOVATEN) ) AS Tên

FROM @NV

* Thì các bạn nhìn vào tập kết quả ta nhận thấy HỌ và tên cách nhau bởi dấu ‘’. Vậy nhiệm vụ của chúng ta là:
* tìm vị trí khoảng trắng đầu tiên trong chuỗi họ tên ta đặt là x (sừ dụng CHARINDEX), sau đó ta sẽ sử dụng hàm LEFT để lấy các ký tự bên trái, và lấy x-1 ký tự.
* để lấy được tên, chúng ta sẽ dùng hàm RIGHT để lấy các ký tự bên phải, và lấy các ký tự còn lại, tức là độ dài chuỗi họ tên-x ký tự.

# SLIDE 30 CÁC HÀM NGÀY THÁNG NĂM

* Phần khó khăn nhất khi làm việc với ngày tháng là đảm bảo rằng định dạng của ngày tháng bạn muốn chèn vào phải ăn khớp với định dạng ngày tháng của cột trong cơ sở dữ liệu.
* Miễn là dữ liệu của bạn chỉ chứa các thông tin ngày tháng, truy vấn của bạn sẽ làm việc như mong đợi. Tuy nhiên, nếu một phần thời gian (giờ phút giây) có liên quan, nó sẽ trở nên phức tạp.
* Cũng giống như các hàm xử lý chuỗi, các hàm ngày tháng thường đi cùng với lệnh SELECT
* Trước khi nói về sự phức tạp của truy vấn ngày tháng, chúng ta sẽ xem thử các hàm quan trọng nhất được xây dựng để làm việc với ngày tháng.
* Hàm GETDATE(): trả về ngày tháng năm và thời gian hiện tại, tức là bao gồm ngày tháng năm giờ phút giây. Hàm GETDATE() không có đối số.

VD: SELECT GETDATE() =>

* Trường hợp ta chỉ cần lấy ngày tháng năm, ta sử dụng CONVERT() để ép kiểu dữ liệu về kiểu date SELECT CONVERT(date,GETDATE())
* Hoặc ta chỉ cần hiển thị thời gian hiện tại, ta sử dụng CONVERT() để ép kiểu dữ liệu về kiểu time SELECT CONVERT(time,GETDATE())
* Ta cũng có thể lấy được năm (nhờ hàm YEAR()), tháng (nhờ hàm MONTH()) hay ngày (nhờ hàm day()) từ ngày hiện tại của hệ thống. Các hàm YEAR(), MONTH(), DAY(): chỉ gồm 1 đối số.

# SLIDE 32: Haàm DATENAME

* **Hàm DATENAME** trong SQL Server trả về một giá trị thời gian của đối số truyền vào, có thể là ngày, tháng, năm, quý, giờ, phút, giây, mili giây… Giá trị trả về là kiểu string (ASCII)
* Cú pháp: DATENAME(dangthoigian, thoigian)
* Hàm DATENAME() gồm 2 đối số đầu vào, đối số 1: dạng thời gian cần chuyển về, đối số 2: khoảng thời gian muốn lấy ra giá trị

SELECT DATENAME(Year,GETDATE()) AS YEAR,

DATENAME(MONTH,GETDATE()) AS MONTH,

DATENAME(DAY,GETDATE()) AS DAY,

DATENAME(DAYOFYEAR,GETDATE()) AS DAYOFYEAR, --Ngày thứ mấy trong năm

DATENAME(WEEK,GETDATE()) AS WEEK, --Tuần thứ mấy trong năm

DATENAME(WEEKDAY,GETDATE()) AS WEEKDAY --Hôm nay là thứ mấy

* Thì kết quả truy vấn chỉ rõ, hiện tại là năm 2022, ngày 19, tháng 5, ngày thứ 139 của năm, tuần thứ 21 của năm và hôm nay thứ 3.

# SILDE 33: CÁC LƯU Ý KHI LÀM VIỆC VỚI CÁC HÀM NGÀY THÁNG

* Trong SQL Server dữ liệu ngày/giờ được xử lý dưới định dạng tháng/ngày/năm (MM/DD/YYYY)
* Nên khi chúng ta thực hiện INSERT dữ liệu phải nhập đúng định dạng đó.
* Khi truy vấn lấy dữ liệu để hiển thị định dạng ngày tháng năn DD/MM/YYYY ta có thể sử dụng hàm CONVERT với tham số định dạng là
* Khi sử dụng câu lệnh INSERT phải truyền dữ liệu ngày/giờ theo định dạng tháng/ngày/năm
* Khi truy vấn dữ liệu, để lấy về giá trị có định dạng ngày/tháng/năm có thể sử dụng hàm CONVERT với mã định dạng 3(dd/mm/yy) hoặc 103(dd/mm/yyyy)

# LAB 3

## BÀI 3

--Tách các thành phần, số nhà, tên đường, tên thành phố

SELECT MANV,DCHI,

SUBSTRING(DCHI,CHARINDEX(' ', DCHI),CHARINDEX(',', DCHI)-CHARINDEX(' ', DCHI)) AS 'Tên đường'

--LEFT(DCHI,CHARINDEX(' ', DCHI)) AS N'Số Nhà',

--RIGHT(DCHI,LEN(DCHI)-CHARINDEX(',', DCHI)) AS TP,

-- LEFT(RIGHT(DCHI, LEN(DCHI)-CHARINDEX(' ', DCHI)),

-- CHARINDEX(',', RIGHT(DCHI, LEN(DCHI)-CHARINDEX(' ', DCHI)-1))) AS N'Tên đường'

--RIGHT(DCHI, LEN(DCHI)-CHARINDEX(' ', DCHI)),

FROM NHANVIEN

**--C1: Danh sách những nhân viên (HONV, TENLOT, TENNV, DCHI) có trên 2 thân nhân, thỏa các yêu cầu**

* Dữ liệu cột HONV được viết in hoa toàn bộ
* Dữ liệu cột TENLOT được viết chữ thường toàn bộ3
* Dữ liệu cột TENNV có ký tự thứ 2 được viết in hoa, các ký tự còn lại viết  
  thường( ví dụ: kHanh)
* Dữ liệu cột DCHI chỉ hiển thị phần tên đường, không hiển thị các thông tin khác  
  như số nhà hay thành phố.

SELECT b.MA\_NVIEN, UPPER(a.HONV) AS N'Họ',

LOWER(a.TENLOT) AS N'Tên Lót',

LOWER(LEFT(a.TENNV,1)) + UPPER(SUBSTRING(a.TENNV,2,1)) +

LOWER(RIGHT(a.TENNV,LEN(TENNV)-2)) AS N'Tên',

SUBSTRING(DCHI,CHARINDEX(' ', DCHI),CHARINDEX(',', DCHI)-CHARINDEX(' ', DCHI))

AS 'Tên đường'

--,COUNT(b.TENTN) AS N'Số thân nhân'

FROM NHANVIEN a, THANNHAN b

WHERE a.MANV=b.MA\_NVIEN

GROUP BY b.MA\_NVIEN, a.HONV,a.TENLOT, a.TENNV, a.DCHI

HAVING COUNT(b.TENTN)>2

**--C2: Cho biết tên phòng ban và họ tên trưởng phòng của phòng ban có đông nhân viên nhất, hiển thị thêm một cột thay thế tên trưởng phòng bằng tên “Fpoly”**

DECLARE @PBMAX TABLE(

PHONG int,

SoLuong int

)

INSERT INTO @PBMAX

SELECT PHG, COUNT(MANV)

FROM NHANVIEN

GROUP BY PHG

SELECT \* FROM @PBMAX

SELECT PHONG, b.TENPHG,HONV+ ' '+TENLOT+ ' '+TENNV AS N'Tên Trưởng Phòng',

HONV+ ' '+TENLOT+ ' '+REPLACE(TENNV,TENNV,'FPoly') AS N'TÊN MỚI'

FROM @PBMAX a, PHONGBAN b,NHANVIEN c

WHERE a.PHONG=b.MAPHG

AND b.TRPHG=c.MANV

AND SoLuong=(

SELECT MAX(SoLuong)

FROM @PBMAX

)

C2:

SELECT TENPHG,HONV,TENLOT,TENNV,REPLACE(TENNV,TENNV,'Fpoly') AS TENMOI

FROM PHONGBAN A, NHANVIEN B

WHERE A.MAPHG=B.PHG

AND A.TRPHG=B.MANV

AND MAPHG= (

SELECT TOP 1 WITH TIES MAPHG

FROM PHONGBAN A, NHANVIEN B

WHERE A.MAPHG=B.PHG

GROUP BY MAPHG

ORDER BY COUNT(\*) DESC

)

## BÀI 4

* Cho biết các nhân viên có năm sinh trong khoảng 1960 đến 1965.

SELECT \*

FROM NHANVIEN

WHERE YEAR(NGSINH) BETWEEN 1960 AND 1965

--WHERE DATENAME(year,NGSINH) BETWEEN 1960 AND 1965

* Cho biết tuổi của các nhân viên tính đến thời điểm hiện tại.

SELECT HONV,TENLOT,TENNV,YEAR(GETDATE())-YEAR(NGSINH) AS N'Tuổi'

FROM NHANVIEN

* Dựa vào dữ liệu NGSINH, cho biết nhân viên sinh vào thứ mấy.

SELECT HONV,TENLOT,TENNV,DATENAME(WEEKDAY,NGSINH) AS N'THỨ'

FROM NHANVIEN

* Cho biết số lượng nhân viên, tên trưởng phòng, ngày nhận chức trưởng phòng và ngày nhận chức trưởng phòng hiển thi theo định dạng dd-mm-yy (ví dụ 25-04-2019)

SELECT a.PHG, b.TENPHG,c.HONV,c.TENLOT,c.TENNV,

b.NG\_NHANCHUC,

CONVERT(VARCHAR,b.NG\_NHANCHUC,105) AS 'DD-MM-YYYY',

COUNT(a.MANV) AS N'Số Lượng'

FROM NHANVIEN a, PHONGBAN b, NHANVIEN c

WHERE a.PHG=b.MAPHG

AND b.TRPHG=c.MANV

GROUP BY a.PHG,b.TENPHG,c.HONV,c.TENLOT,c.TENNV,b.NG\_NHANCHUC,

CONVERT(VARCHAR,b.NG\_NHANCHUC,105)

## BÀI 5

**Danh sách những nhân viên (HONV, TENLOT, TENNV, SONHA, TP) có trên 2 thân nhân.**

**Cột SONHA chỉ chứa số nhà. Cột TP chỉ chứa tên TP**

SELECT b.MA\_NVIEN,a.HONV,a.TENLOT,a.TENNV,

LEFT(DCHI,CHARINDEX(' ',DCHI)) AS 'SONHA',

RIGHT(DCHI,LEN(DCHI)-CHARINDEX(',',DCHI)-1) AS 'TP'

FROM NHANVIEN a, THANNHAN b

WHERE a.MANV=b.MA\_NVIEN

GROUP BY b.MA\_NVIEN,a.HONV,a.TENLOT,a.TENNV,

LEFT(DCHI,CHARINDEX(' ',DCHI)),RIGHT(DCHI,LEN(DCHI)-CHARINDEX(',',DCHI))

HAVING COUNT(b.TENTN)>2