Bài: Insert, Delete, Update Table trong SQL Server

Xem bài học trên website để ủng hộ Kteam: Insert, Delete, Update Table trong SQL Server

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Như trong bài KHỞI TẠO, XÓA, SỬA TABLE đã đề cập, một Table (Bảng) bao gồm các Column (Trường thuộc tính) và các Record (Bản ghi). Trong đó, các Record chính là dữ liệu đưa vào các Table tương ứng cấu trúc của dữ liệu định sẵn của các Column.

Tuy nhiên, trong quá trình lưu trữ dữ liệu, ta thấy rõ không chỉ lưu trữ mà Table còn cần phát sinh các hoạt động **THÊM, XÓA, SỬA DỮ LIỆU** với một hay nhiều dữ liệu trong Table. Sau đây, chúng ta sẽ cùng tìm hiểu các thao tác trên.

Nội dung chính

Để theo dõi tốt nhất bài này, bạn nên xem qua các bài:

- Khởi tạo DATABASE trong SQL.
- Khởi tạo, xóa, sửa TABLE trong SQL.
- KIỂU DỮ LIỆU trong SQL

Trong bài này, chúng ta sẽ cùng nhau tìm hiểu một số vấn đề sau:

- Database mẫu
- Thao tác với dữ liệu bằng giao diện Table.
- Thao tác với dữ liệu bằng Code.

Database mẫu

Để thao tác tốt với bài này, chúng ta sử dụng database **TRUONGHOC** sau. Hoặc bạn có thể tự khởi tạo Database có các Table tượng tự để nhớ bài tốt hơn.

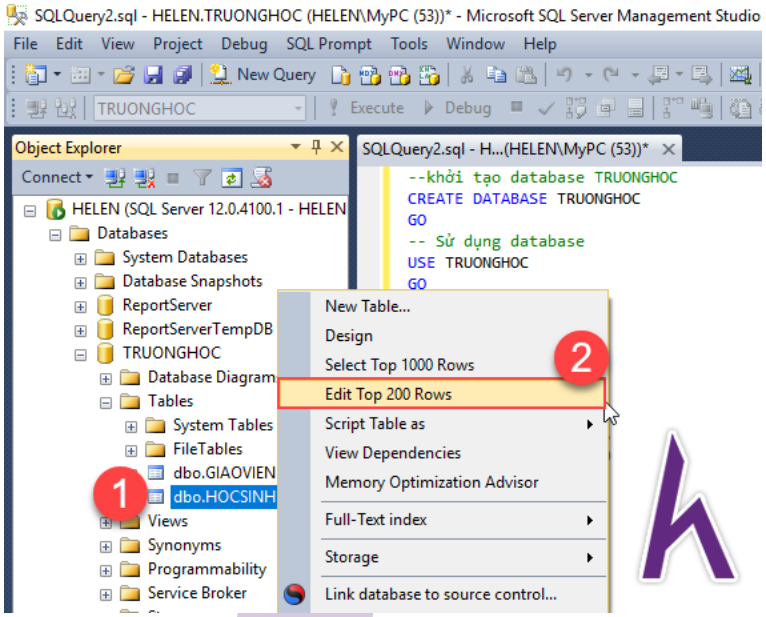
SQL:

```
--khởi tạo database TRUONGHOC
CREATE DATABASE TRUONGHOC
GO
-- Sử dụng database
USE TRUONGHOC
GO
-- Tạo bảng HOCSINH
CREATE TABLE HOCSINH
(
        MAHS CHAR(5),
        TEN NVARCHAR(30),
        NAM BIT, -- Column giới tính Nam: 1 - đúng, 0 - sai
        NGAYSINH DATETIME,
        DIACHI VARCHAR(20),
        DIEMTB FLOAT,
)
GO
-- Tạo bảng GIAOVIEN
CREATE TABLE GIAOVIEN
(
        MAGV CHAR(5),
        TEN NVARCHAR(30),
        Nam BIT, -- Column giới tính Nam: 1 - đúng, 0 - sai
        NGAYSINH DATETIME,
        DIACHI VARCHAR(20),
        LUONG MONEY
)
GO
-- Tạo bảng LOPHOC
CREATE TABLE LOPHOC
(
        MALOP CHAR(5),
        TENLOP NVARCHAR(30),
        SOLUONG INT
)
GO
```

Thao tác với dữ liệu bằng giao diện Table

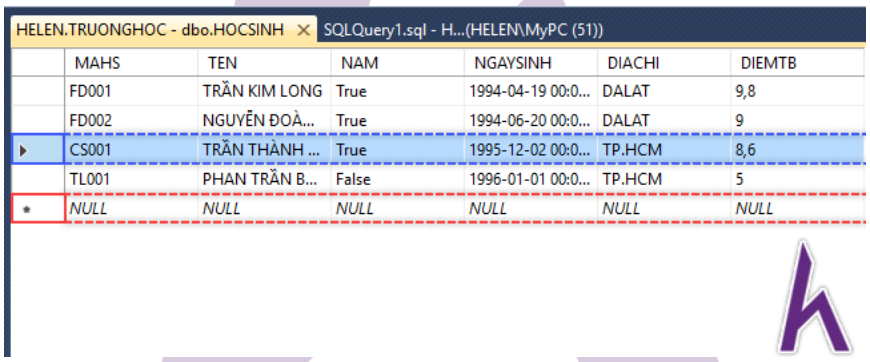
Thêm/ Sửa/ Cập nhập dữ liệu vào Table (Insert/ Update Record)

Để thêm hoặc sửa dữ liệu bằng giao diện, chúng ta tìm tới Table cần thêm/ sửa > Chuột phải > Edit top 200 rows.



Giao diện thêm/sửa dữ liệu trên table hiển thị như sau

- Tạo một Record mới tại dòng có vị trí dấu * phía trước. Sau khi nhập đủ dữ liệu thành phần cần thiết > Enter.
- Giá trị **mặc định** tại các trường thuộc tính ban đầu là **NULL**
- Dòng có dấu mũi tên phía trước, biểu thị Record hiện hành bạn đang chọn. Tại đây bạn có thể thêm/sửa một hay nhiều thành phần trong Record.



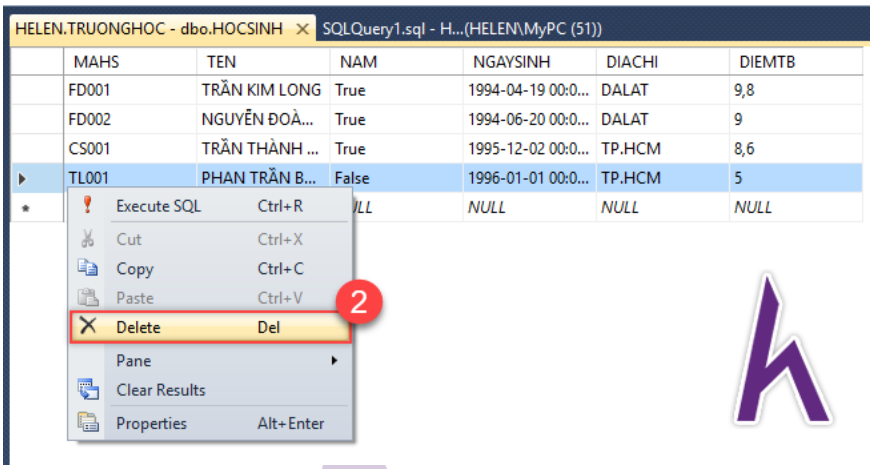
	MAHS	TEN	NAM	NGAYSINH	DIACHI	DIEMTB
	FD001	TRẦN KIM LONG	True	1994-04-19 00:0...	DALAT	9,8
	FD002	NGUYỄN ĐOÀ...	True	1994-06-20 00:0...	DALAT	9
▶	CS001	TRẦN THÀNH ...	True	1995-12-02 00:0...	TP.HCM	8,6
	TL001	PHAN TRẦN B...	False	1996-01-01 00:0...	TP.HCM	5
*	NULL	NULL	NULL	NULL	NULL	NULL

Trong quá trình thêm/ sửa dữ liệu, bạn cần lưu ý nhập dữ liệu theo cấu trúc của kiểu dữ liệu đã chọn trong quá trình khởi tạo Table (Các kiểu dữ liệu đã giới thiệu trong bài KIỂU DỮ LIỆU TRONG SQL)

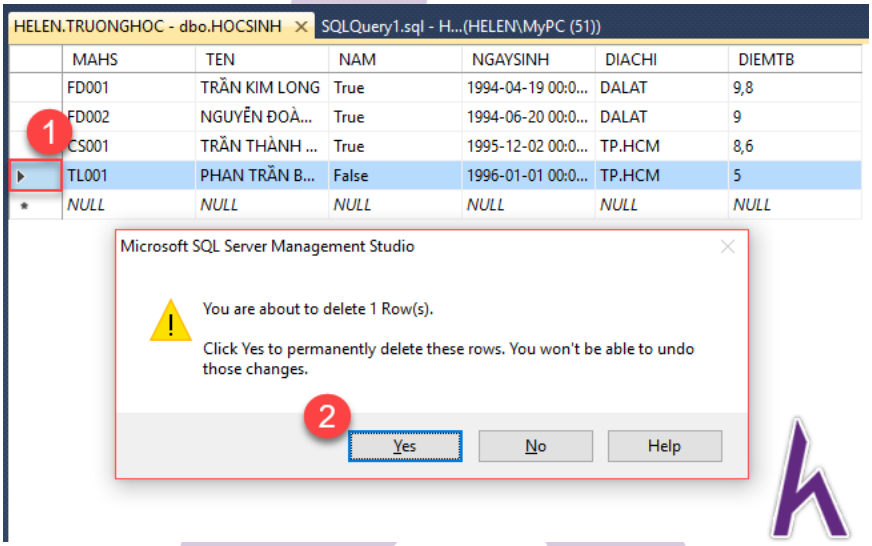
Xoá dữ liệu trên Table (Delete Record)

Xóa một record

Để xóa một record đã có trong Table, tại record cần xóa > chuột phải > Delete.

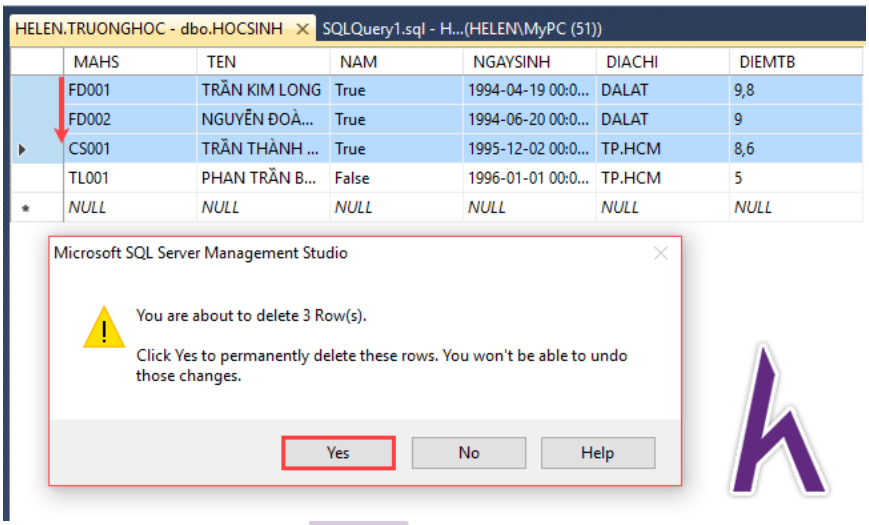


Hoặc bạn có thể nhấp chuột trái vào phía trước record cần xóa. > phím Delete. Cửa sổ thông báo xác nhận xuất hiện > Yes để hoàn tất thao tác xóa một record.



Xóa đồng thời nhiều Record

Nếu bạn muốn xóa cùng lúc nhiều Record, nhấp giữ chuột trái vào phía trước Record cần xóa kéo đến hết các record muốn xóa > phím Delete. Tương tự, cửa sổ thông báo yêu cầu xác nhận thao tác xóa.



Thao tác với dữ liệu bằng code

Để việc thao tác với dữ liệu bằng code được dễ dàng và nhanh chóng hơn, **Kteam** khuyến khích bạn dùng công cụ hỗ trợ nhắc lệnh đã được hướng dẫn cài đặt trong bài HƯỚNG DẪN CÀI ĐẶT CÔNG CỤ SQL TOOLBELT.

Một số cấu trúc nhập dữ liệu bằng code cần lưu ý

Như bài KIỂU DỮ LIỆU TRONG SQL đã đề cập, trong quá trình khởi tạo Table, mỗi column được gán một kiểu dữ liệu riêng. Vì vậy, quá trình nhập liệu cũng cần tuân thủ nguyên tắc để dữ liệu nhập vào không bị lỗi. Sau đây là một số cấu trúc nhập liệu cơ bản:

Kiểu dữ liệu	Cấu trúc gợi ý	Dữ liệu nhập vào
char/varchar (n)	'', -- TÊN COLUMN – char/varchar(n)	Dữ liệu nhập vào nằm trong dấu nháy đơn ' ' Ví dụ: 'FD001' --MAHS - char(5)
nchar/nvarchar (n)	N'', -- TÊN COLUMN – nchar/nvarchar(30)	Dữ liệu nhập vào nằm trong dấu nháy đơn ' ', có thể sử dụng tiếng việt. Ví dụ: N' Trần Kim Long' --TEN – nvarchar(30)
Int	0, -- TÊN COLUMN – int	Giá trị mặc định là 0 Giá trị nhập vào cần thuộc kiểu số nguyên vào vị trí 0. Ví dụ: 1994, --SOLUONG -int
float	0.0, -- TÊN COLUMN - float	Giá trị mặc định là 0.0 Giá trị nhập vào cần thuộc kiểu số nguyên vào vị trí 0.0 Ví dụ: 9.04, --DIEMTB -float

bit	NULL, -- TÊN COLUMN - bit	Giá trị mặc định là NULL (rỗng) Chỉ nhập được giá trị 0 hoặc 1 tương ứng true và false tại vị trí NULL Ví dụ: 1, --NAM -bit
money	NULL -- TÊN COLUMN - money	Giá trị mặc định là NULL (rỗng) Gía trị nhập vào kiểu số thực tại vị trí NULL Ví dụ: 9500000, --LUONG -money
datetime/date	GETDATE(), -- TÊN COLUMN – datetime Hoặc ' ', -- TÊN COLUMN - datetime	Mặc định của kiểu datetime là lệnh **GETDATE()** để lấy ngày hiện tại theo hệ thống. Để nhập ngày tháng năm theo ý muốn, bạn nhập theo cấu trúc **yyyymmdd** bên trong dấu nháy đơn ' ' Ví dụ: '19940419' -- NGAYSINH -datetime

Thêm dữ liệu vào Table (Insert Record)

Cú Pháp:

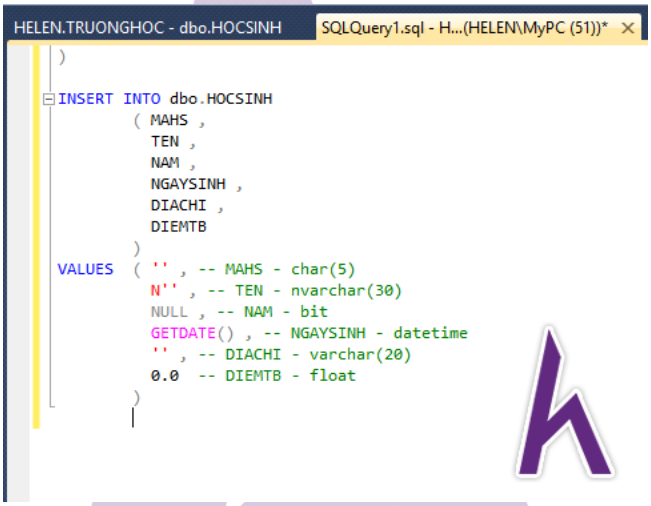
Thêm Record theo column tùy chọn:

```
INSERT INTO <Tên Table>
    ( column1, column2, column3, ... , columnn, )
VALUES  (
        Gợi ký nhập dữ liệu, -- Tên column1 – kiểu dữ liệu tương ứng column1
        Gợi ký nhập dữ liệu, -- Tên column2 – kiểu dữ liệu tương ứng column2
        Gợi ký nhập dữ liệu, -- Tên column3 – kiểu dữ liệu tương ứng column3
        ...
    )
```

Thêm Record theo thứ tự cấu trúc mặc định Column:

```
INSERT INTO <Tên Table>

VALUES  (

            Gợi ký nhập dữ liệu, -- Tên column1 – kiểu dữ liệu tương ứng column1

            Gợi ký nhập dữ liệu, -- Tên column2 – kiểu dữ liệu tương ứng column2

            Gợi ký nhập dữ liệu, -- Tên column3 – kiểu dữ liệu tương ứng column3

             ...

        )
```

Khi sử dụng công cụ nhắc lệnh, thao tác thêm record > SQL sẽ hiển thị cấu trúc mặc định của dữ liệu mới theo kiểu dữ liệu của column trong Table:



Ví dụ:

Thêm một Record mới vào table HOCSINH theo đúng thứ tự mặc định của Table

SQL:

```
--Thêm một Record mới vào Table HOCSINH theo đúng thứ tự mặc định của Table

INSERT dbo.HOCSINH

VALUES  ( 'CS002' , -- MAHS - char(5)
          N'KIM LONG' , -- TEN - nvarchar(30)
          1 , -- NAM - bit
          '19940226', -- NGAYSINH - datetime
          'DONGNAI' , -- DIACHI - varchar(20)
          9.0  -- DIEMTB - float
        )
```

Lưu ý:

- Từ khóa INTO có hoặc không đều **không gây ảnh hưởng** đến cấu trúc lệnh.
- Dữ liệu nhập liệu cần được lưu tuần tự tương ứng thứ tự column đã chọn.

Xóa dữ liệu trong Table (Delete Record)

Xóa toàn bộ dữ liệu bằng lệnh TRUNCATE và DELETE

Nếu bạn để ý, trong bài KHỞI TẠO, XÓA, SỬA TABLE TRONG SQL; Kteam đã đề cập đến lệnh TRUNCATE để xóa tất cả dữ liệu trong Table với cú pháp :

```
TRUNCATE TABLE <Tên TABLE>
```

Trong bài này, Kteam sẽ giới thiệu lệnh DELETE cũng có chức năng xóa tất cả dữ liệu trong Table với cú pháp:

```
DELETE <Tên TABLE>
```

Câu hỏi đặt ra là:

Nếu chúng có cùng chức năng, tại sao cần phải sử dụng hai lệnh xóa dữ liệu riêng biệt? Vậy chúng có gì khác nhau? Sự khác biệt đó có ảnh hưởng gì đến thao tác truy vấn của bạn trong SQL. Hãy cùng Kteam tìm hiểu một số khác biệt cơ bản giữa hai lệnh này!

	TRUNCATE	DELETE
Tổng quan	Cả 2 đều sử dụng để xóa dữ liệu trong một Table mà không làm ảnh hưởng đến cấu trúc của Table đó	
Cú pháp	TRUNCATE TABLE <Tên TABLE>	DELETE <Tên TABLE> [mệnh đề điều kiện]
Xóa có điều kiện	*Không hỗ trợ* mệnh đề WHERE	*Có hỗ trợ* mệnh đề WHERE Ví dụ: DELETE dbo.GIAOVIEN WHERE LUONG >5000
Ghi nhận nhật ký thao tác (transaction log)	Cơ chế xóa của TRUNCATE thực tế TRUNCATE ***không xóa dữ liệu*** mà chỉ ***hủy cấp phát*** đến trang dữ liệu đó và xóa con trỏ đến chỉ mục của trang >ghi lại dữ liệu của trang dữ liệu bị hủy cấp phát vào nhật ký thao tác (Transaction Log).	Cơ chế xóa của DELETE là ***xóa từng dòng*** > ghi lại thay đổi được xóa cho từng dòng trong nhật ký thao tác (Transaction Log)
	Dữ liệu vẫn tồn tại đến khi bị ghi đè	Nếu xóa một lượng lớn Record > ghi nhận nhật ký nhiều lần > nhật ký thao tác phình to > chiếm dụng nhiều tài nguyên của máy chủ
	Vì ít cập nhập nhật ký thao tác, nên TRUNCATE ***xóa toàn bộ dữ liệu*** Table ***nhanh hơn*** DELETE.	Tuy chiếm dụng tài nguyên nhưng nó giúp hệ thống có thể ***khôi phục dữ liệu ở trạng thái gần nhất*** (Record xóa gần nhất)
	***Một số tài liệu cho rằng TRUNCATE không ghi lại nhật ký thao tác (Transaction log) là không chính xác. Tất cả các trang dữ liệu bị hủy cấp phát đều được ghi lại trong tập tin transaction log, trong điều kiện cần thiết để khôi phục dữ liệu, các trang dữ liệu chỉ cần được cấp phát lại > cơ sở dữ liệu trở về trạng thái cũ. Tuy nhiên, TRUNCATE không thể dùng để xóa một hay một vài Record trong Table được.	

Thời hạn phiên làm việc	TRUNCATE ***không thể khôi phục*** dữ liệu ***sau khi đóng kết nối*** của phiên làm việc hiện hành	DELETE ***có thể khôi phục*** dữ liệu trong ***phiên làm việc tiếp theo***.
Thao tác với khóa ngoại	TRUNCATE ***báo lỗi khi*** có khóa ngoại trong Table liên kết đến Table khác	DELETE ***có thể báo lỗi*** khi có khóa ngoại.
	Ngay cả khi không chứa dữ liệu, TRUNCATE vẫn báo lỗi nếu Table tồn tại khóa ngoại	Trong trường hợp xóa nhiều Record, nhưng không xóa được do dữ liệu đang được tham chiếu bởi Table khác > DELETE báo lỗi > hủy thực thi > không có Record nào bị xóa
Hành vi với Trigger	TRUNCATE chỉ hủy cấp phát trang dữ liệu > ***không gọi Trigger***	DELETE ***gọi trigger*** với mỗi Record bị xóa khỏi Table
Hành vi với cột Identity	TRUNCATE ***reset biến đếm*** của cột Identity	DELETE ***không reset biến đếm*** của cột Identity
Indexes - Chỉ mục	TRUNCATE có thể sử dụng lại chỉ mục không sử dụng	DELETE không sử dụng lại được chỉ mục không sử dụng

Một số toán tử điều kiện

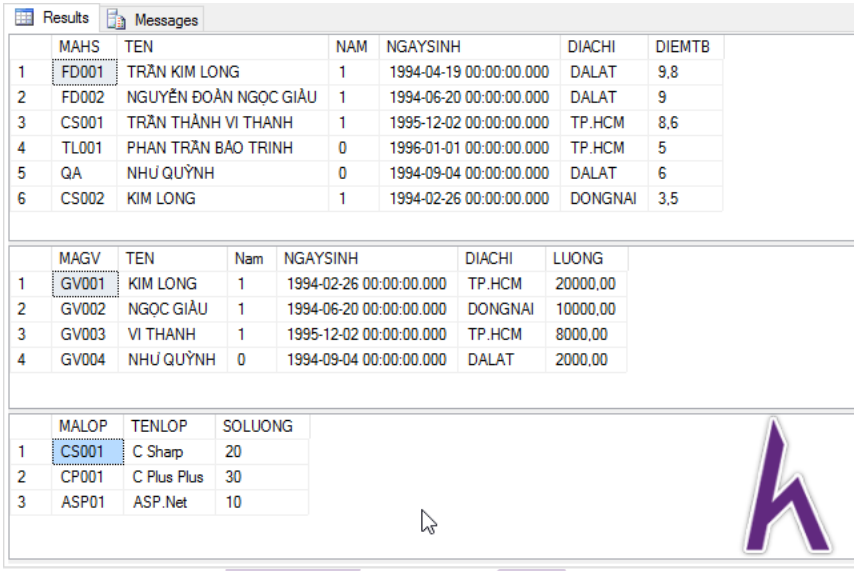
Trong quá trình truy vấn, bạn dễ thấy cần có nhiều hơn một điều kiện cần để truy vấn, vậy để liên kết, kết hợp các điều kiện đó trong SQL, cụ thể trong phần này là ở câu lệnh WHERE chúng ta sử dụng một số toán tử sau:

TOÁN TỬ		MÔ TẢ GIẢN ĐƠN
Toán tử Logic	AND	Lọc các dữ liệu **thỏa tất cả** các vế điều kiện được *ngăn cách* bởi AND đều trả về **giá trị đúng** *(true)*.
	OR	Lọc các dữ liệu **thỏa một trong các vế** điều kiện được ngăn cách bởi OR trả về **giá trị đúng***(true)*
	NOT	Lọc ra các dữ liệu **khác** với dữ liệu thỏa của lệnh/toán tử đi kèm
	BETWEEN	Lọc ra các dữ liệu **thỏa tập giá trị** định sẵn nằm sau BETWEEN.
		Lệnh BETWEEN thường kết hợp với AND để xác định khoảng giá trị đầu cuối
	IN	**So sánh** một giá trị với một danh sách định sẵn
Toán tử so sánh	>	Lọc các dữ liệu thỏa vế trái **lớn hơn** vế phải
	<	Lọc các dữ liệu thỏa vế trái **nhỏ hơn** vế phải
	>=	Lọc các dữ liệu thỏa vế trái **lớn hơn hoặc bằng** vế phải
	<=	Lọc các dữ liệu thỏa vế trái **nhỏ hơn hoặc bằng** vế phải
	=	Lọc các dữ liệu thỏa vế trái **bằng** vế phải. **Không** dùng để so sánh kiểu dữ liệu chuỗi ký tự.
	<> hoặc !=	Lọc các dữ liệu thỏa vế trái **không bằng/khác** vế phải.
		Không dùng để so sánh kiểu dữ liệu chuỗi ký tự.

Phía trên là các toán tử thường sử dụng trong quá trình truy vấn, ngoài ra bạn có thể tham khảo thêm chi tiết các toán tử khác tại TÀI LIỆU THAM KHẢO từ tutorialspoint.com

Một số ví dụ về xóa dữ liệu có điều kiện

Ở phần này, chúng ta thực hiện một số ví dụ sau để hiểu rõ hơn về cách xóa dữ liệu trong Table. Nhập dữ liệu vào Database **TRUONGHOC** đầu bài để bạn có thể thao tác dễ dàng hơn với các lệnh truy vấn sau. Dưới đây là database mẫu đã được cập nhập dữ liệu



Results Messages

	MAHS	TEN	NAM	NGAYSINH	DIACHI	DIEMTB
1	FD001	TRẦN KIM LONG	1	1994-04-19 00:00:00.000	DALAT	9,8
2	FD002	NGUYỄN ĐOÀN NGỌC GIÀU	1	1994-06-20 00:00:00.000	DALAT	9
3	CS001	TRẦN THÀNH VI THANH	1	1995-12-02 00:00:00.000	TP.HCM	8,6
4	TL001	PHAN TRẦN BẢO TRINH	0	1996-01-01 00:00:00.000	TP.HCM	5
5	QA	NHƯ QUỲNH	0	1994-09-04 00:00:00.000	DALAT	6
6	CS002	KIM LONG	1	1994-02-26 00:00:00.000	DONGNAI	3,5

	MAGV	TEN	Nam	NGAYSINH	DIACHI	LUONG
1	GV001	KIM LONG	1	1994-02-26 00:00:00.000	TP.HCM	20000,00
2	GV002	NGỌC GIÀU	1	1994-06-20 00:00:00.000	DONGNAI	10000,00
3	GV003	VI THANH	1	1995-12-02 00:00:00.000	TP.HCM	8000,00
4	GV004	NHƯ QUỲNH	0	1994-09-04 00:00:00.000	DALAT	2000,00

	MALOP	TENLOP	SOLUONG
1	CS001	C Sharp	20
2	CP001	C Plus Plus	30
3	ASP01	ASP.Net	10

Ví dụ 1: Xóa tất cả dữ liệu trong Table HOCSINH, ta sử dụng lệnh:

SQL:

```
DELETE dbo.HOCSINH
```

Hoặc

SQL:

```
TRUNCATE TABLE dbo.HOCSINH
```

Ví dụ 2: Xóa những giáo viên có lương hơn 5000:

SQL:

```
DELETE dbo.GIAOVIEN WHERE LUONG >5000
```

Ví dụ 3: Xóa những giáo viên có lương hơn 5000 và mã số giáo viên <15

SQL:

```
DELETE dbo.GIAOVIEN WHERE LUONG > 5000 AND MAGV < 15
```

Ví dụ 4: Xóa những học sinh có điểm TB là 1; 8; 9.

SQL:

```
DELETE dbo.HOCSINH WHERE DIEMTB IN (1,8,9)
```

Ví dụ 5: Xóa những học sinh có mã học sinh thuộc danh sách FD001, FD002, FD003

SQL:

```
SELECT* FROM dbo.HOCSINH WHERE MAHS IN ('FD002','FD001')
```

Ví dụ 6: Xóa những học sinh có điểm trong khoảng 1 đến 8

SQL:

```
DELETE dbo.HOCSINH WHERE DIEMTB BETWEEN 1 AND 8
```

Ví dụ 7: Xóa những học sinh có địa chỉ không phải ở Đà Lạt.

SQL:

```
DELETE dbo.HOCSINH WHERE DIACHI NOT LIKE 'DALAT'
```

Cập nhập dữ liệu trong Table (Update Record)

Cú pháp:

```
UPDATE <Tên Table>

SET <thuộc tính 1 = giá trị 1>, <thuộc tính 2 = giá trị 2>,...,<thuộc tính n = giá trị n>,

WHERE <điều kiện cập nhập>
```

Lưu ý:

Nếu bạn không sử dụng WHERE trong lệnh sửa dữ liệu > tất cả các Record trong Table đều bị cập nhập dữ liệu.

Ví dụ minh họa:

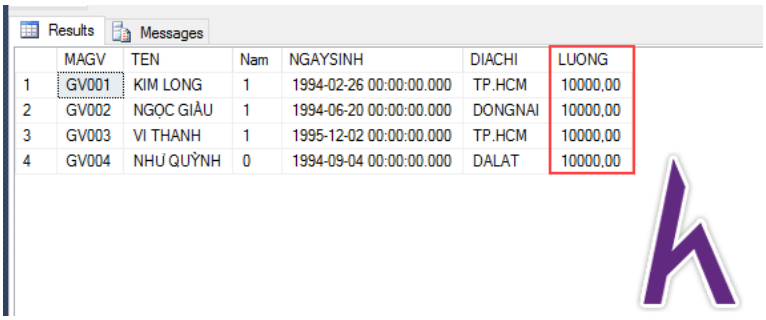
Với Table GIAOVIEN ban đầu có dữ liệu như sau

	MAGV	TEN	Nam	NGAYSINH	DIACHI	LUONG
	GV001	KIM LONG	True	1994-02-26 00:0...	TP.HCM	20000,0000
	GV002	NGỌC GIÀU	True	1994-06-20 00:0...	DONGNAI	10000,0000
	GV003	VI THANH	True	1995-12-02 00:0...	TP.HCM	8000,0000
	GV004	NHƯ QUỲNH	False	1994-09-04 00:0...	DALAT	2000,0000
▶*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Ví dụ 1: Cập nhập Lương của tất cả giáo viên thành 10000

SQL:

```
UPDATE dbo.GIAOVIEN SET LUONG = 10000
```

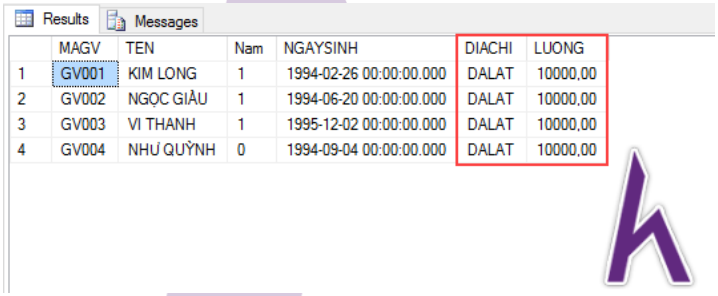


	MAGV	TEN	Nam	NGAYSINH	DIACHI	LUONG
1	GV001	KIM LONG	1	1994-02-26 00:00:00.000	TP.HCM	10000,00
2	GV002	NGỌC GIÀU	1	1994-06-20 00:00:00.000	DONGNAI	10000,00
3	GV003	VI THANH	1	1995-12-02 00:00:00.000	TP.HCM	10000,00
4	GV004	NHƯ QUỲNH	0	1994-09-04 00:00:00.000	DALAT	10000,00

Ví dụ 2: Cập nhập lương của tất cả giáo viên thành 10000 và địa chỉ tại DALAT

SQL:

```
UPDATE dbo.GIAOVIEN SET LUONG = 10000, DIACHI ='DALAT'
```

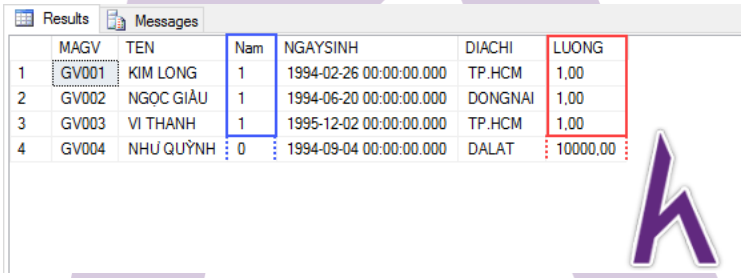


	MAGV	TEN	Nam	NGAYSINH	DIACHI	LUONG
1	GV001	KIM LONG	1	1994-02-26 00:00:00.000	DALAT	10000,00
2	GV002	NGỌC GIÀU	1	1994-06-20 00:00:00.000	DALAT	10000,00
3	GV003	VI THANH	1	1995-12-02 00:00:00.000	DALAT	10000,00
4	GV004	NHƯ QUỲNH	0	1994-09-04 00:00:00.000	DALAT	10000,00

Ví dụ 3: Cập nhập lương của những giáo viên nam thành 1

SQL:

```
UPDATE dbo.GIAOVIEN SET LUONG = 1
WHERE Nam='1'
```



	MAGV	TEN	Nam	NGAYSINH	DIACHI	LUONG
1	GV001	KIM LONG	1	1994-02-26 00:00:00.000	TP.HCM	1,00
2	GV002	NGỌC GIÀU	1	1994-06-20 00:00:00.000	DONGNAI	1,00
3	GV003	VI THANH	1	1995-12-02 00:00:00.000	TP.HCM	1,00
4	GV004	NHƯ QUỲNH	0	1994-09-04 00:00:00.000	DALAT	10000,00

Kết

Trong bài này, chúng ta đã biết cách thêm, xóa sửa dữ liệu SQL.

Bài sau, chúng ta sẽ bắt đầu tìm hiểu về KHÓA CHÍNH TRONG SQL.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của bạn để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".