

# Bài: Tạo khóa chính trong SQL Server

Xem bài học trên website để ủng hộ Kteam: [Tạo khóa chính 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

Trong những bài trước, chúng ta đã học cách [KHỞI TẠO, XÓA, SỬA TABLE](#), cũng như cách [THÊM, XÓA, CẬP NHẬP DỮ LIỆU](#). Tuy nhiên, việc quản lý một Table không dừng lại ở đó, trong bài này Kteam sẽ giới thiệu đến bạn một yếu tố rất quan trọng của Table là **KHÓA CHÍNH!**

Chúng ta hãy cùng tìm hiểu khóa chính là gì? Và tại sao cần có khóa chính nhé!

## Nội dung chính

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

- Khởi tạo [DATABASE, TABLE](#) trong SQL.
- [KIỂU DỮ LIỆU](#) trong SQL.
- [Insert, delete, update Table trong SQL](#).

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

- Khóa chính là gì? Tại sao phải sử dụng khóa chính
- Thao tác với khóa chính bằng giao diện.
- Thao tác với khóa chính bằng code.

## Database mẫu

Để thao tác tốt với bài này chúng ta sử dụng database **TRƯỜNG HỌC** sau. Hoặc bạn có thể tự khởi tạo Database và các Table tương 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
```

## Khóa chính là gì? Tại sao phải tạo khóa chính?

Như chúng ta đã tìm hiểu trong bài [KHỞI TẠO, XÓA, SỬA TABLE](#), bạn có thể thấy chúng ta hoàn toàn có thể tạo một Table không có yếu tố gọi là **KHÓA CHÍNH**.

Tuy nhiên, sẽ gặp vấn đề là có thể tồn tại hai hoặc nhiều Record chứa dữ liệu giống nhau ở tất cả các trường thuộc tính > trùng lặp dữ liệu > không xác định được một đối tượng có tồn tại hay chưa, có là duy nhất trong một Table không.

Vậy ta thấy rõ cần tồn tại một/ một nhóm trường thuộc tính định danh cho Table gọi là **KHÓA CHÍNH** (primary key):

- Chứa các giá trị không rỗng (NULL)
- Chỉ tồn tại dữ liệu duy nhất, không trùng lặp.

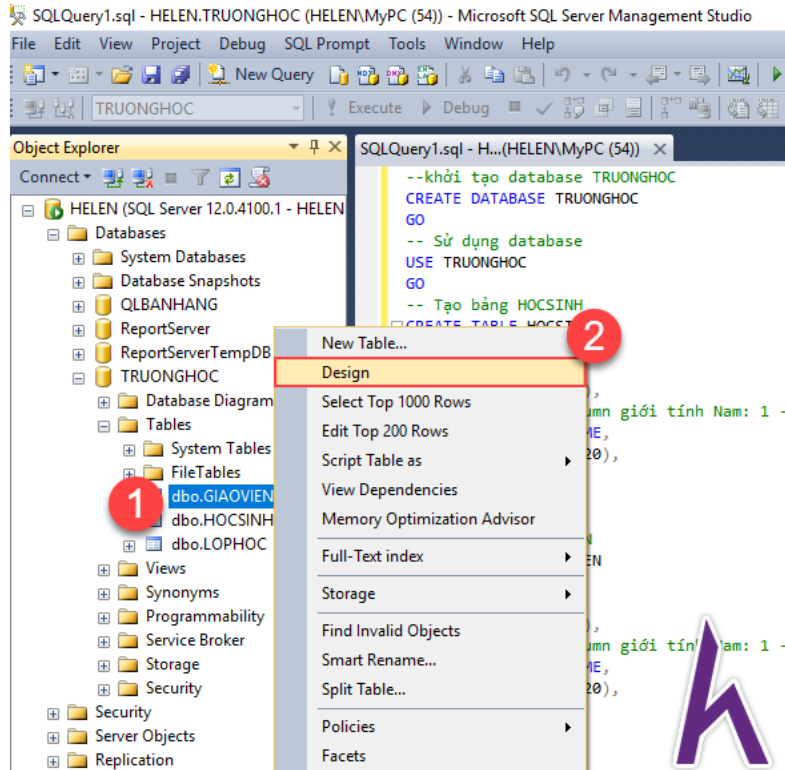
Ngoài ra, thông qua khóa chính của Table, mà các Table khác có thể tham chiếu tới bằng Khóa ngoại (sẽ được giới thiệu chi tiết trong bài [KHÓA NGOẠI TRONG SQL](#))

## Thao tác với khóa chính bằng giao diện Table

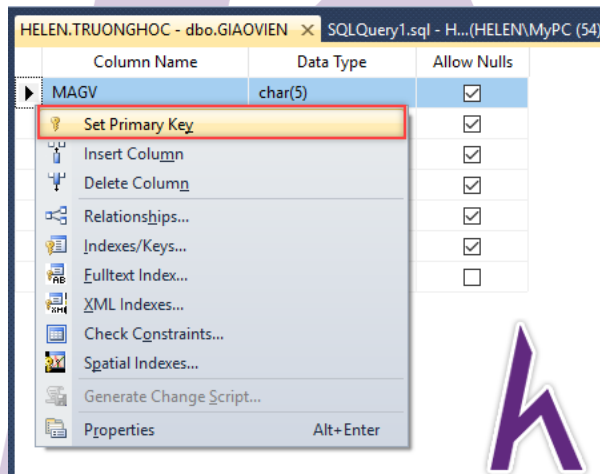
### Tạo khóa chính từ Table có sẵn/Table mới

Thao tác phần này dựa trên Database **TRUONGHOC** ở đầu bài

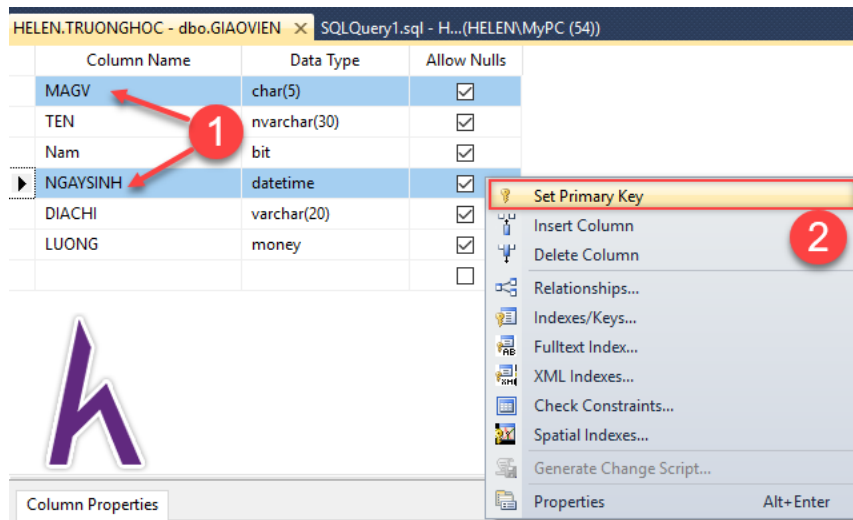
**Bước 1:** Mở Table cần tạo khóa ở giao diện thiết kế > nhấp chuột phải > Design



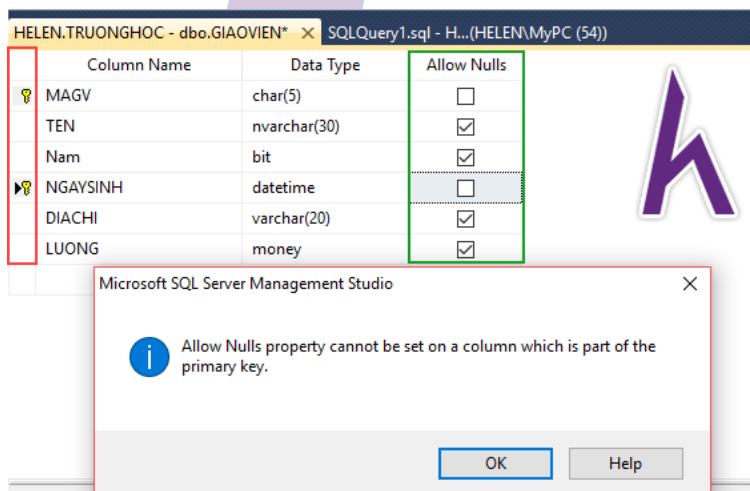
**Bước 2:** Nhấp chuột phải vào mũi tên hiện hành trước column chỉ định làm khóa chính > Set Primary key



Để chọn nhiều hơn một column cùng định danh cho Table bạn giữ phím **Ctrl** > chọn các column làm khóa chính > chuột phải > Set Primary key.



**Kết quả:** Sau khi tạo khóa chính thành công. Phía trước Column định danh có **biểu tượng chia khóa**. Ở cột **Allow Nulls** tương ứng **không** có dấu tick > không cho nhận giá trị NULL ở khóa chính.



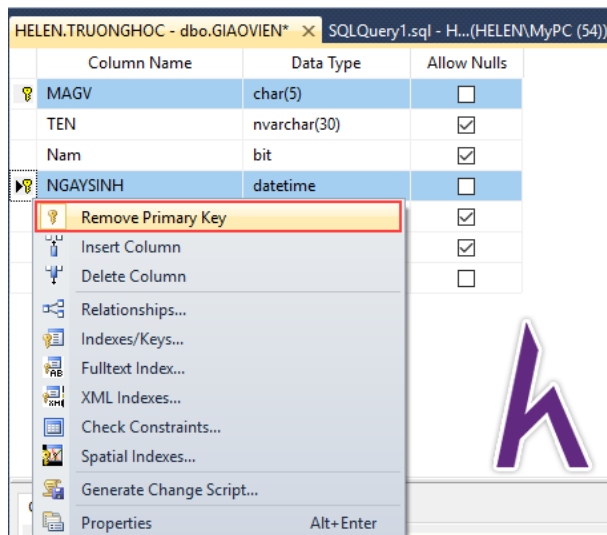
- Nếu bạn cố tình chọn để giá trị NULL, SQL Manager sẽ hiện thông báo lỗi như trên.

**Lưu ý:**

- Khi Table đã tồn tại khóa chính, việc tạo một khóa chính mới đồng nghĩa với ngầm gỡ khóa chính cũ đã có > đưa column về trạng thái bình thường.
- Một Table chỉ tồn tại duy nhất một khóa chính, mỗi khóa chính có thể được xác định từ một hay nhiều Column trong Table.

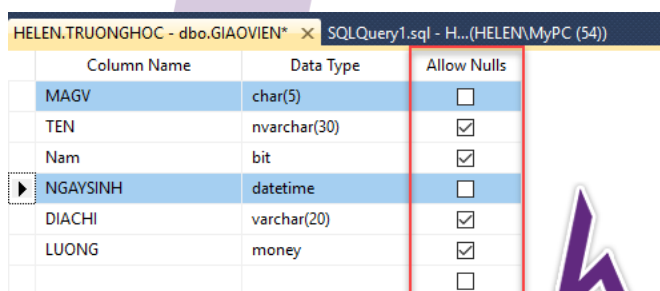
### Xóa khóa chính trong giao diện Table

Để xóa một khóa chính đã tạo, bạn chọn **chuột phải** vào **biểu tượng khóa chính** trước column cần xóa > **Remove Primary key**



Sau khi xóa khóa chính, column trở về trạng thái bình thường, biểu tượng khóa chính biến mất. Ở cột **Allow Nulls** vẫn không có dấu tick.

- Tick vào ô trống tương ứng để column nhận giá trị NULL nếu muốn.



## Tạo khóa chính bằng code

### Các ràng buộc (constraint) thường dùng trong SQL

Để đảm bảo tính chính xác và đáng tin cậy của dữ liệu khi nhập vào Table, người ta cần có các quy tắc cho dữ liệu đó. Trong SQL gọi các quy tắc này là các ràng buộc toàn vẹn, có thể đi kèm Table hoặc column, và tùy theo mức độ của ràng buộc mà quy định giới hạn dữ liệu cho toàn Table hay chỉ ở column xác định.

Sau đây là một số ràng buộc thường gặp trong SQL:

Ràng buộc	Công dụng
UNIQUE	- Đảm bảo dữ liệu nhập vào là duy nhất, không trùng lặp.
NOT NULL	- Đảm bảo column không để trống/ không nhận giá trị rỗng (NULL)
DEFAULT	- Đặt một giá trị làm mặc định trong trường hợp không có giá trị chỉ định được nhập vào
CHECK (gợi thiệu chi tiết trong bài <a href="#">Check trong SQL</a> )	- Kiểm tra ràng buộc miền giá trị > đảm bảo các dữ liệu nhập vào thỏa điều kiện cụ thể.
INDEX (Gợi thiệu chi tiết trong bài <a href="#">Indexes trong SQL</a> )	- Tạo & lấy dữ liệu một cách nhanh chóng từ Database. - Khi Index (chỉ mục) được gọi, nó gán 1 RowID cho mỗi record trước khi sắp xếp dữ liệu > Index (chỉ mục) hợp lý > tăng hiệu suất truy xuất với các Database lớn
IDENTITY	- Tạo dãy số liên tục tăng cho column chỉ định.

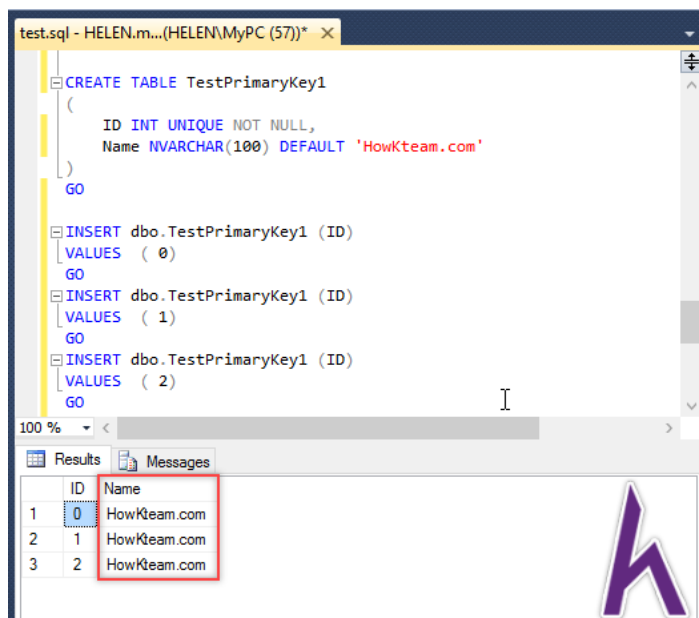
**Lưu ý:**

Cặp ràng buộc **UNIQUE** và **DEFAULT** không thể đi kèm với nhau.

**Ví dụ 1:****SQL:**

```
CREATE TABLE TestPrimaryKey1
(
    ID INT UNIQUE NOT NULL,
    Name NVARCHAR(100) DEFAULT 'Howkteam.com'
)
GO

INSERT dbo.TestPrimaryKey1 (ID)
VALUES ( 0)
GO
INSERT dbo.TestPrimaryKey1 (ID)
VALUES ( 1)
GO
INSERT dbo.TestPrimaryKey1 (ID)
VALUES ( 2)
GO
```



**Ví dụ 2:**

**SQL:**

```

CREATE TABLE TestPrimaryKey2
(
    ID INT UNIQUE NOT NULL IDENTITY(1,1),
    Name NVARCHAR(100) CHECK (Name <> 'howkteam')
)
GO
INSERT dbo.TestPrimaryKey2( Name )
VALUES (
    N'KIM LONG' -- Name - nvarchar(100)
)
GO
INSERT dbo.TestPrimaryKey2( Name )
VALUES (
    N'NGOC GIAU' -- Name - nvarchar(100)
)
GO
INSERT dbo.TestPrimaryKey2( Name )
VALUES (
    N'VI THANH' -- Name - nvarchar(100)
)
GO
INSERT dbo.TestPrimaryKey2( Name )
VALUES (
    N'HOWKTEAM' -- Name - nvarchar(100)
)
GO
    
```

 Kteam, howkteam, free education, sql server, primary key, table, constraint sql, delete primary key,

Kteam khuyến khích các bạn thực hành các thao tác trong hai ví dụ trên để nhận biết kết quả và hiểu bài một cách tốt nhất.

**Tạo khóa chính trong lúc khởi tạo Table**

## Tạo khóa chính trong quá trình khai báo column

```
CREATE TABLE <Tên Table>
(
    <column1> <kiểu dữ liệu> PRIMARY KEY,
    <column2> <kiểu dữ liệu>,
    ...
    <columnn> <kiểu dữ liệu>
)
```

### Ví dụ:

#### SQL:

```
CREATE TABLE HOCSINH
(
    MAHS CHAR(5) PRIMARY KEY,
    TEN NVARCHAR(30),
    NAM BIT, -- Column giới tính Nam: 1 - đúng, 0 - sai
    NGAYSINH DATETIME,
    DIACHI VARCHAR(20),
    DIEMTB FLOAT
)
GO
```

## Tạo khóa chính sau khi khai báo tất cả column

```
CREATE TABLE <Tên Table>
(
    <column1> <kiểu dữ liệu>,
    <column2> <kiểu dữ liệu>,
    ...
    <columnn> <kiểu dữ liệu>

    PRIMARY KEY (columnKey1, columnKey2..., columnKeyn)
)
```

### Ví dụ:

#### SQL:



```

CREATE TABLE GIAOVIEN
(
    TEN NVARCHAR(30) NOT NULL,
    Nam BIT, -- Column giới tính Nam: 1 - đúng, 0 - sai
    NGAYSINH DATETIME NOT NULL,
    DIACHI VARCHAR(20),
    LUONG MONEY,
    PRIMARY KEY (TEN, NGAYSINH)
)
GO

```

**Kết quả:** Khóa chính được thiết lập ở hai ví dụ trên

The screenshot shows a SQL query window with the following code:

```

GO
-- Tạo bảng LOPHOC
CREATE TABLE LOPHOC
(
    MALOP CHAR(5),
    TENLOP NVARCHAR(30),
    SOLUONG INT
)
GO
ALTER TABLE dbo.LOPHOC ADD PRIMARY KEY(MALOP)
GO

```

Below the code, the Messages window displays the following error:

```

Msg 8111, Level 16, State 1, Line 38
Cannot define PRIMARY KEY constraint on nullable column in table 'LOPHOC'.
Msg 1750, Level 16, State 0, Line 38
Could not create constraint or index. See previous errors.

```

## Tạo khóa chính cho Table sẵn có

### Khóa chính có một column

```
ALTER TABLE <Tên Table> ADD PRIMARY KEY(column)
```

### Khóa chính có nhiều column

```
ALTER TABLE <Tên Table> ADD PRIMARY KEY(column1,column2,...columnn)
```

#### Lưu ý:

- Khuyến khích tạo khóa chính trước khi bạn nhập dữ liệu cho Table.
- Khi thiết lập khóa chính cho một hoặc nhiều column đã tồn tại sẵn, cần đảm bảo tất cả column có ràng buộc không rỗng (NOT NULL) và không đồng thời trùng lặp dữ liệu (UNIQUE) ở tất cả các column tạo thành khóa chính.

#### Ví dụ:

#### SQL:

```

CREATE TABLE DETAI
(
    MADT NCHAR(4) NOT NULL UNIQUE,
    TENDT NVARCHAR(50),
    CAPQL NCHAR(20),
    KINHPhi FLOAT,
    NGAYBD DATE,
    NGAYKT DATE,
    MACD NCHAR(4),
    GVCNDT NCHAR(3),
)
GO

-- thêm khóa chính vào bảng
ALTER TABLE dbo.DETAi ADD PRIMARY KEY(MADT)
GO

```

## Đặt tên cho khóa chính với CONSTRAINT

Để việc quản lý khóa chính được dễ dàng hơn và có hệ thống hơn trong quá trình thao tác thêm, xóa, sửa dữ liệu của khóa chính. Chúng ta sử dụng từ khóa **CONSTRAINT** để đặt tên cho khóa chính.

Nguyên tắc đặt tên cho khóa chính: **PK\_<tên Table>**

### Cú pháp:

```
CONSTRAINT PK_<Tên Table> PRIMARY KEY (ColumnKey1, ColumnKey2,..., ColumnKeyn)
```

### Lưu ý:

Khai báo các column tồn tại trong Table trước khi khai báo khóa chính có đặt tên.

**Ví dụ 1:** Thiết lập khóa chính có đặt tên ngay trong quá trình khởi tạo Table

### SQL:

```

CREATE TABLE LOPHOC
(
    MALOP CHAR(5) NOT NULL UNIQUE,
    TENLOP NVARCHAR(30),
    SOLUONG INT,
    CONSTRAINT PK_LOPHOC PRIMARY KEY (MALOP)
)
GO

```

**Ví dụ 2:** Thêm khóa chính có đặt tên cho Table sẵn có

### SQL:

```

CREATE TABLE DIEM
(
    MAHS INT NOT NULL,
    MALOP CHAR NOT NULL,
    DIEMTB FLOAT CHECK(DIEMTB <=10 AND DIEMTB >=0)
)
GO

ALTER TABLE dbo.DIEM ADD CONSTRAINT PK_DIEM PRIMARY KEY(MAHS,MALOP)

```

## Xóa khóa chính có đặt tên

Cú pháp:

```
ALTER TABLE <Tên Table> DROP CONSTRAINT <Tên Khóa chính>
```

Ví dụ:

SQL:

```
ALTER TABLE dbo.DIEM DROP CONSTRAINT PK_DIEM  
GO
```

## Kết

Trong bài này, chúng ta đã biết cách KHỞI TẠO KHÓA CHÍNH trong SQL Server.

Bài sau chúng ta sẽ bắt đầu tìm hiểu về cách [KHỞI TẠO KHÓA NGOẠI 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**"