

Bài: Khóa ngoại trong SQL Server

Xem bài học trên website để ủng hộ Kteam: [Khóa ngoại 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 bài trước, chúng ta đã tìm hiểu về [KHÓA CHÍNH](#) và cách tạo, xóa, đặt tên cho khóa chính trong một Table. Ngoài ra, Kteam cũng đã đề cập đến một thành phần khác có khả năng tham chiếu đến Table thông qua Khóa chính đó là **KHÓA NGOẠI**.

Tại sao có khóa chính lại cần khóa ngoại? Khóa ngoại sử dụng để làm gì? Chúng ta sẽ cùng nhau tìm hiểu về nó 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](#)
- Tạo [KHÓA CHÍNH](#) trong SQL

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

- Khóa ngoại là gì?
- Thao tác với khóa ngoại.
- Sơ đồ diagram.

Khóa ngoại là gì?

Các table trong một database không tồn tại độc lập mà còn có mối quan hệ mật thiết với nhau về mặt dữ liệu.

Mối quan hệ này được thể hiện thông qua **ràng buộc giá trị dữ liệu** xuất hiện ở bảng này phải có xuất hiện trước trong một bảng khác.

Mối quan hệ giữa các table trong database nhằm **đảm bảo được tính đúng đắn và hợp lệ** của dữ liệu trong database.

Điều kiện để tạo khóa ngoại

- Tham chiếu đến khóa chính của table khác, đảm bảo các ràng buộc **UNIQUE, NOT NULL** tương tự khóa chính. (Các ràng buộc UNIQUE, NOT NULL đã được giới thiệu trong bài [KHÓA CHÍNH TRONG SQL](#))
- Khóa ngoại phải có cùng kiểu dữ liệu, cùng số lượng trường có sắp xếp tương ứng khóa chính

Tác dụng

Khóa ngoại giúp đảm bảo toàn vẹn dữ liệu, không có trường hợp tham chiếu đến dữ liệu không tồn tại.

Thao tác với khóa ngoại

Bạn sử dụng Database **Primary_Foreign** sau để dễ dàng thao tác trong quá trình theo dõi bài viết nhé!

SQL:

```

CREATE DATABASE Primary_Foreign
GO

USE Primary_Foreign
GO

CREATE TABLE BoMon
(
    MaBM CHAR(10) PRIMARY KEY,
    Name NVARCHAR(100) DEFAULT N'Tên bộ môn'
)
GO

CREATE TABLE Lop
(
    MaLop CHAR(10) NOT NULL,
    Name NVARCHAR(100) DEFAULT N'Tên lớp'

    PRIMARY KEY(MaLop)
)
GO

```

Quy ước

- **TableKey**: Table chứa khóa chính được tham chiếu đến.
- **Table Foreign**: Table chứa khóa ngoại
- **Column K,K1,K2...:** Column định danh khóa chính thuộc table Key được tham chiếu đến.
- **Column F,F1,F2...:** Column của khóa ngoại thuộc table Foreign.

Số lượng **columnFn** tương ứng số lượng **columnKn** nhằm đảm bảo khóa ngoại có số trường tương ứng khóa chính tham chiếu đến. (n là số)

Tạo khóa ngoại trong lúc tạo bảng

Cú pháp:

```

CREATE TABLE <Table Foreign>

(

...

FOREIGN KEY (<ColumnF, ColumnF1, ColumnF2,...>)

REFERENCES <Table Key> (<ColumnK, ColumnK1, ColumnK2,...>)

)

```

Ví dụ:

SQL:

```

CREATE TABLE GiaoVien
(
    MaGV CHAR(10) NOT NULL,
    Name NVARCHAR(100) DEFAULT N'Tên giáo viên',
    ĐịaChi NVARCHAR(100) DEFAULT N'Địa chỉ giáo viên',
    NgàySinh DATE,
    Sex BIT,
    MaBM CHAR(10),

    -- Tạo khóa ngoại ngay khi tạo bảng
    FOREIGN KEY(MaBM) REFERENCES dbo.BOMON(MaBM)
)
GO

ALTER TABLE dbo.GIAOVIEN ADD PRIMARY KEY(MaGV)

```

Tạo khóa ngoại sau khi tạo bảng

Cú pháp:

```

ALTER TABLE <Table Foreign>

ADD FOREIGN KEY(<ColumnF, ColumnF1, ColumnF2,...>)

REFERENCES <Table Key> (<ColumnK, ColumnK1, ColumnK2,...>)

```

Ví dụ:

SQL:

```

-- Tạo khóa ngoại sau khi tạo bảng
ALTER TABLE dbo.HocSinh FOREIGN KEY(MaLop) REFERENCES dbo.Lop(MaLop)

```

Đặt tên cho khóa ngoại.

Tương tự khóa chính, khóa ngoại cũng đặt tên thông qua từ khóa **CONSTRAINT**. Một Table có thể có một hoặc nhiều khóa chính nên nguyên tắc đặt tên sẽ có chút khác biệt

Tên khóa ngoại: **FK_<Table Foreign>_<Table Key>**

Tùy theo đặc tả và độ phức tạp của từng Database mà ta có thể thay đổi cách đặt tên theo mong muốn

Cú pháp:

```

ALTER TABLE <Table Foreign>

ADD CONSTRAINT <Tên khóa ngoại>

FOREIGN KEY(<ColumnF, ColumnF1, ColumnF2,...>)

REFERENCES <Table Key> (<ColumnK, ColumnK1, ColumnK2,...>)

```

Ví dụ:

SQL:

```
-- Tạo khóa ngoại sau khi tạo bảng

ALTER TABLE dbo.HocSinh ADD CONSTRAINT FK_HS FOREIGN KEY(MaLop) REFERENCES dbo.Lop(MaLop)
```

Hủy khóa ngoại có đặt tên

Cú pháp:

```
ALTER TABLE <Table Foreign> DROP CONSTRAINT <Tên khóa ngoại>
```

Ví dụ:

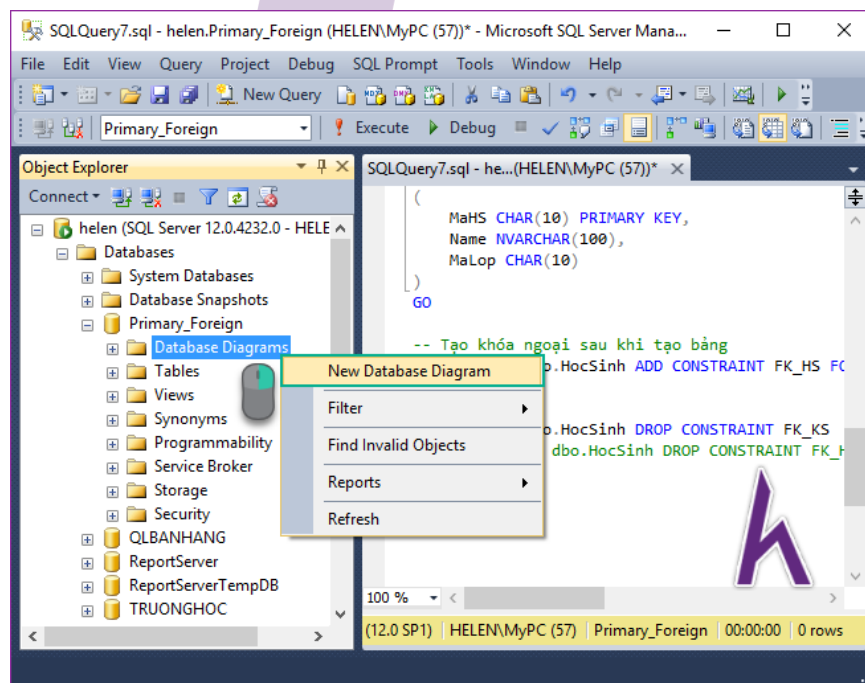
SQL:

```
ALTER TABLE dbo.HocSinh DROP CONSTRAINT FK_KS
```

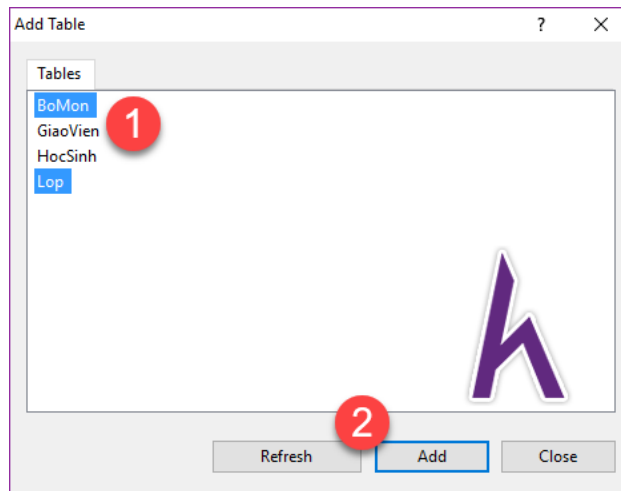
Sơ đồ diagram

Với việc tạo khóa ngoại bằng code tuy đơn giản nhưng lại không cho bạn cái nhìn trực quan về các liên kết trong Database. Vậy để xem mối quan hệ giữa khóa chính khóa ngoại ta đi sẽ mở mô hình diagram như sau

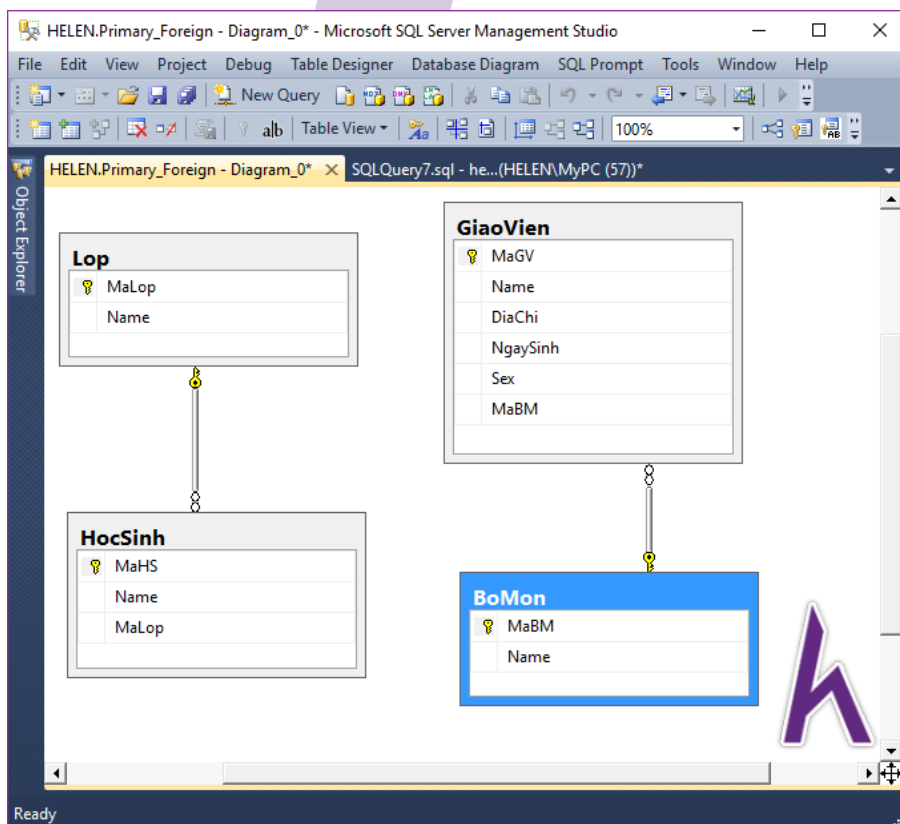
Bước 1: Chọn Database cần xem > **chuột phải** vào Database Diagrams > **New Database Diagram**



Bước 2: Cửa sổ Add Table hiển thị > **Chọn Table** > **Add** > **Close**



Sơ đồ Diagram của Database sẽ hiển thị như sau



Trong đó:

- Ký hiệu **vô cùng** trở về Table có chứa Khóa ngoại.
- Ký hiệu **chìa khóa** trở về Table có chứa Khóa chính.
- Liên kết nối vô cùng và chìa khóa biểu diễn quan hệ tham chiếu của khóa ngoại từ table này đến khóa chính của table khác.

Từ Diagram chúng ta dễ dàng nắm bắt được các thông tin cơ bản như Tên Table, Tên column, khóa chính, khóa ngoại,...

Kết

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

Bài sau, chúng ta sẽ bắt đầu tìm hiểu về [TRUY VẤN CƠ BẢN 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**"

