Tạo câu lệnh SQL cho mối quan hệ giữa giảng viên và sinh viên cho 2 bảng bên dưới: -- Mỗi giảng viên có thể hướng dẫn nhiều sinh viên

-- Mỗi sinh viên chỉ được hướng dẫn bởi một giảng viên.

CREATE TABLE GiangVien(

MaGV int primary key,

HoTenGv char(30),

Luong decimal(5, 2),

MaKhoa char(10),

foreign key (MaKhoa) references Khoa(MaKhoa)

);

CREATE TABLE SinhVien(

Masv int primary key,

HoTenSV char(40),

MaKhoa char(10),

NamSinh int,

QueQuan char(30),

foreign key (MaKhoa) references Khoa(MaKhoa)

);

Dựa trên yêu cầu của bạn, tôi sẽ sửa trực tiếp vào định nghĩa bảng **SinhVien** để thêm cột **MaGVHuongDan** tham chiếu đến bảng **GiangVien**.

CREATE TABLE GiangVien(

MaGV int primary key,

HoTenGv char(30),

Luong decimal(5, 2),

MaKhoa char(10),

foreign key (MaKhoa) references Khoa(MaKhoa)

);

CREATE TABLE SinhVien(

Masv int primary key,

HoTenSV char(40),

MaKhoa char(10),

NamSinh int,

QueQuan char(30),

MaGVHuongDan int,

foreign key (MaKhoa) references Khoa(MaKhoa),

foreign key (MaGVHuongDan) references GiangVien(MaGV)

);

Trong định nghĩa bảng **SinhVien**, tôi đã thêm cột **MaGVHuongDan** và đồng thời thêm ràng buộc khóa ngoại cho cột này tham chiếu đến cột **MaGV** của bảng **GiangVien**.

Đánh giá câu lệnh SQL sau cho yêu câu trên của tôi. Nó đúng hay sai ? CREATE TABLE HuongDan(

Masv int primary key,

MaGV int,

foreign key (MaGV) references GiangVien(MaGV)

);

ChatGPT

Câu lệnh SQL bạn cung cấp tạo một bảng mới tên là **HuongDan** để thể hiện mối quan hệ giữa **GiangVien** và **SinhVien**.

Cấu trúc này:

1. Sử dụng một bảng **HuongDan** mà mỗi hàng trong đó thể hiện một mối quan hệ giữa một **GiangVien** và một **SinhVien**.
2. Với **Masv** làm khóa chính, điều này đảm bảo rằng mỗi sinh viên chỉ xuất hiện một lần trong bảng **HuongDan**, đồng nghĩa với việc mỗi sinh viên chỉ được hướng dẫn bởi một giảng viên.

Câu lệnh của bạn cũng đúng và phù hợp với yêu cầu:

* Mỗi giảng viên có thể hướng dẫn nhiều sinh viên.
* Mỗi sinh viên chỉ được hướng dẫn bởi một giảng viên.

Khác với cách tiếp cận trước, cấu trúc này chỉ cần một cột khóa ngoại (**MaGV**) trong bảng **HuongDan** để thể hiện mối quan hệ giữa sinh viên và giảng viên. Bạn không cần đến cặp khóa như cách tiếp cận trước.

Tuy nhiên, có một điều bạn cần lưu ý: với cấu trúc này, bạn sẽ cần thêm và cập nhật dữ liệu cho bảng **HuongDan** để thể hiện mối quan hệ giữa **GiangVien** và **SinhVien**Top of Form

Sửa câu lệnh SQL trong 2 bảng bên dưới cho mối quan hệ giữa sinh viên và đề tài

-- Một đề tài có thể được nhiều sinh viên thực tập

-- Một sinh viên chỉ có thể thực tập một đề tài

CREATE TABLE DeTai(

MaDeTai char(10) primary key,

TenDeTai char(30),

Kinhphi int,

Noithuctap char(30)

);

CREATE TABLE SinhVien(

MaSV int primary key,

HoTenSV char(40),

MaKhoa char(10),

NamSinh int,

QueQuan char(30),

foreign key (MaKhoa) references Khoa(MaKhoa)

);

Để thể hiện mối quan hệ giữa sinh viên và đề tài theo yêu cầu của bạn, chúng ta cần thêm một cột khóa ngoại vào bảng **SinhVien** để tham chiếu đến bảng **DeTai**. Cụ thể, chúng ta có thể thêm một cột **MaDeTai** vào bảng **SinhVien** để thể hiện mỗi sinh viên chỉ có thể thực tập một đề tài, và một đề tài có thể được nhiều sinh viên thực tập. Dưới đây là cách sửa đổi các câu lệnh SQL của bạn:

CREATE TABLE DeTai(

MaDeTai char(10) primary key,

TenDeTai char(30),

Kinhphi int,

Noithuctap char(30)

);

CREATE TABLE SinhVien(

MaSV int primary key,

HoTenSV char(40),

MaKhoa char(10),

MaDeTai char(10), -- Thêm cột MaDeTai

NamSinh int,

QueQuan char(30),

foreign key (MaDeTai) references DeTai(MaDeTai) -- Thêm khóa ngoại MaDeTai

);

Bằng cách này, bạn đã tạo một mối quan hệ một-nhiều giữa **DeTai** và **SinhVien**, đồng thời đảm bảo rằng mỗi **SinhVien** chỉ có thể tham gia vào một **DeTai**.

-- 1. Cho biết mã số và tên của các đề tài do giảng viên ‘Tran son’ hướng dẫn

-- CREATE TABLE DeTai(

-- MaDeTai char(10) primary key,

-- TenDeTai char(30),

-- Kinhphi int,

-- NoiThucTap char(30)

-- );

-- --

-- CREATE TABLE HuongDan(

-- MaSV int primary key,

-- MaDeTai char(10),

-- MaGV int,

-- KetQua decimal(5, 2),

-- foreign key (MaDeTai) references DeTai(MaDeTai),

-- foreign key (MaGV) references GiangVien(MaGV)

-- );

-- Nguyên nhân phải tách ra 1 bảng HuongDan vì có những sinh viên ko được hướng dẫn

-- Mỗi giảng viên có thể hướng dẫn nhiều sinh viên

-- Mỗi sinh viên chỉ được hướng dẫn bởi một giảng viên.

-- Một sinh viên chỉ có thể thực tập một đề tài (do cột MaSV là primary key)

-- Một đề tài có thể được nhiều sinh viên thực tập

-- => Giảng viên và Đề tài:

-- Mỗi giảng viên có thể hướng dẫn nhiều đề tài,

-- Mỗi đề tài chỉ được hướng dẫn bởi một giảng viên.

-- CREATE TABLE GiangVien(

-- MaGV int primary key,

-- HoTenGv char(30),

-- Luong decimal(5, 2),

-- MaKhoa char(10),

-- foreign key (MaKhoa) references Khoa(MaKhoa)

-- );

-- Một giảng viên thuộc về một Khoa.

-- Mỗi Khoa có nhiều giảng viên.

-- CREATE TABLE SinhVien(

-- MaSV int primary key,

-- HoTenSV char(40),

-- MaKhoa char(10),

-- NamSinh int,

-- QueQuan char(30),

-- foreign key (MaKhoa) references Khoa(MaKhoa)

-- );

-- Bảng DeTai và bảng HuongDan có mối quan hệ một-nhiều, nghĩa là một đề tài có thể được nhiều sinh viên thực tập, nhưng một sinh viên chỉ có thể thực tập một đề tài.

-- Bảng GiangVien và bảng HuongDan có mối quan hệ một-nhiều, nghĩa là một giảng viên có thể hướng dẫn nhiều đề tài, nhưng một đề tài chỉ có thể được một giảng viên hướng dẫn.

-- Giảng viên và Đề tài: Mỗi giảng viên có thể hướng dẫn nhiều đề tài, nhưng mỗi đề tài chỉ được hướng dẫn bởi một giảng viên.

-- Sinh viên và Đề tài: Mỗi sinh viên chỉ thực tập trên một đề tài, và mỗi đề tài có thể được thực tập bởi nhiều sinh viên. Tuy nhiên, dựa trên định nghĩa của bảng HuongDan, mỗi sinh viên chỉ thực tập một đề tài (do cột MaSV là primary key). Mối quan hệ này là mối quan hệ một-một (1-1) giữa SinhVien và DeTai.

-- Cho biết mã số và tên của các đề tài do giảng viên ‘Tran son’ hướng dẫn

-- SELECT dt.MaDeTai, dt.TenDeTai

-- FROM DeTai dt

-- JOIN HuongDan hd ON dt.MaDeTai = hd.MaDeTai

-- JOIN GiangVien gv ON hd.MaGV = gv.MaGV

-- WHERE gv.HoTenGv = 'Tran Son';