# Chương 1: TỔNG QUAN(Sửa lại văn nói, giải thích các từ tiếng anh)

## **Đặt vấn đề**

Đối với sinh viên việc học từng môn học thì thông tin về môn học vô cùng cần thiết, cũng như việc giáo viên lên kế hoạch về việc dạy môn học cho sinh viên biết thông tin về môn học khá là quan trọng trong quá trình cho sinh viên tiếp cận với kiến thức một cách hiệu quả hơn. Do đó, khi khoa phân công giáo viên giảng dạy thì đầu mỗi học kì giáo viên sẽ soạn ra đề cương chi tiết gửi cho bộ môn hoặc khoa phê duyệt. Từ việc đó, sinh sẽ viên thuận lợi hơn trong việc tra cứu chính xác hơn về thông tin môn học, công việc quản lý đề cương chi tiết môn học của các ngành học, bậc học, hệ đào tạo mà giảng viên phụ trách càng hiệu quả và chính xác hơn, do đó cần có một hệ thống quản lý công việc trên.

## **Đối tượng nghiên cứu (NodeJS)**

## **Phương pháp nghiên cứu**

Phương pháp nghiên cứu lý thuyết: Tìm hiểu lý thuyết, phân tích cơ sở dữ liệu,tìm kiếm tài liệu liên quan về ngôn ngữ để lập trình hổ trợ web như HTML, thư viện Bootstrap, Javascrip, NodeJS....

Phương pháp thực nghiệm: Xây dựng hệ thống quản lý đề cương chi tiết cho Bộ môn Công nghệ thông tin.

# Chương 2: CƠ SỞ LÝ THUYẾT

## **JavaScript và NodeJS**

### **Tổng quan:**

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ. Khi nằm bên trong một môi trường, JavaScript có thể kết nối tới các đối tượng của môi trường đó và cung cấp các cách quản lý chúng.

Javascript ngày càng trở nên phổ biến hơn với nhiều tính năng và các thư viện được hỗ trỡ cho các lập trình viên, điều đó khiến cho các giao diện web càng trở nên sinh động hơn. Mọi thứ mà chúng ta có thể làm được trên web ngày nay là Javascript có thể chạy được trên server, cũng như chạy được trên browser, điều này là khó tưởng tượng trong những năm trở lại đây. Thật vậy, rõ ràng rằng từ xưa đến nay chúng ta vẫn quan niệm rằng lập trình bên phía server chỉ dùng được những ngôn ngữ như php, ruby, ... nhưng từ khi NodeJs ra đời nó mang đến tư tưởng mới cho việc lập trình cả bên phía server cũng như bên phía client. Và đối với NodeJS thì chúng ta biến những điều mà trước kia chỉ có thể thao tác được bên phía client thì nay cũng thao tác và xử lý được trên server, và ngược lại.

NodeJS được viết bằng ngôn ngữ javascript, nó là một trình biên đóng gói của Google’s V8 JavaScript engine, và một thư viện lõi được viết bằng Javascript. NodeJS cung cấp công cụ giúp lập trình viên có thể làm việc trong non-blocking, mô hình I/O . Sau hơn 20 năm nghiên cứu, xây dựng và phát triển, nhóm kĩ sư đã cho ra đời sản phẩm ứng dụng web NodeJS chạy thời gian thực và kết nối 2 chiều client và server, cho phép trao đổi dữ liệu một cách tự do.

### **Ưu điểm NodeJS:**

* Đặc điểm nổi bật của NodeJS là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của Javascript mà NodeJS tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP.
* Ứng dụng trên 1 trang: Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh.
* Shelling tools unix: NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.
* Luồng dữ liệu: NodeJS xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
* Ứng dụng Web thời gian thực: Với sự ra đời của các ứng dụng di động & HTML 5 nên NodeJS rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,…

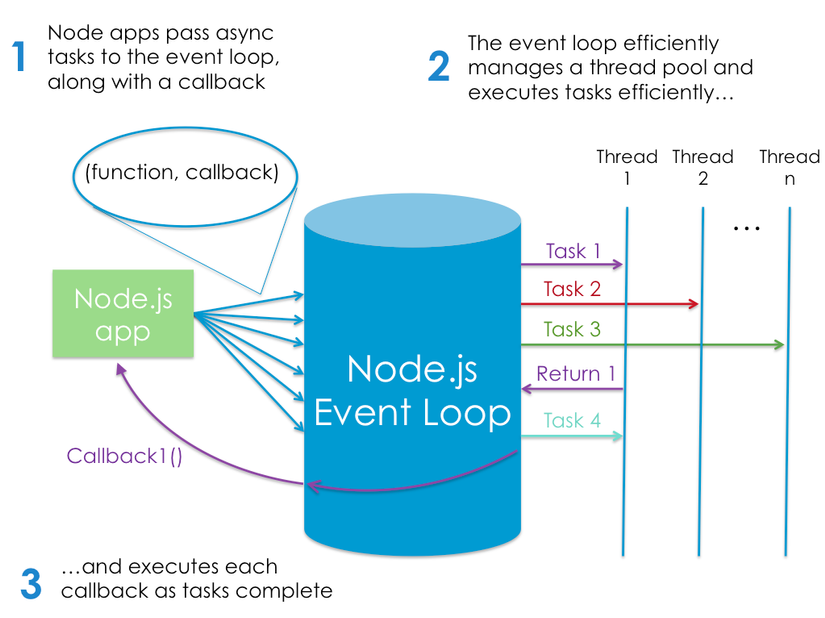
### **Nhược điểm của NodeJS:**

* Ứng dụng tốn tài nguyên: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút nếu xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding, encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng Node.js.
* NodeJS được giới thiệu khá muộn so với các ngôn ngữ khác cho xử lý sever appweb nên cộng đồng hướng dẫn tại Việt Nam còn ít.

### **Mô hình hoạt động của Node.js**

Để cho dễ hiểu, khi người dùng kết nối đến một server truyền thống, chẳng hạn Apache, nó sẽ sinh ra một thread mới để xử lý request. Ở các ngôn ngữ như PHP hay Ruby, mỗi một phép toán I/O (ví dụ truy cập database) sẽ chặn execution trên code của người dùng cho đến khi phép toán đó hoàn thành. Nói cách khác, server sẽ đợi cho đến khi database được duyệt xong mới xử lý kết quả. Nếu có những request mới, server lại tiếp tục sinh những thread mới để xử lý chúng. Điều này dẫn đến nguy cơ kém hiệu quả, khi một lượng lớn thread được tạo ra sẽ khiến cho hệ thống trở nên chậm chạp, tệ hơn nữa có thể khiến site bị sập. Cách thông thường để giải quyết tình trạng này là bổ sung thêm server.

Node.js, mặt khác là single-threaded. Nó cũng thuộc dạng event-driven hay nói cách khác tất cả những gì xảy ra trong Node là để phản hồi lại với một sự kiện. Ví dụ, khi một request được gửi đến, server bắt đầu xử lý nó. Nếu nó gặp phải phép toán I/O, thay vì đợi cho phép toán này kết thúc, nó sẽ đăng ký một callback trước khi tiếp tục xử lý event tiếp theo. Khi phép toán I/O kết thúc, server sẽ chạy callback và tiếp tục làm việc trên request ban đầu. Ở tầng bên dưới, NodeJS sử dụng thư viện libuv để thực hiện hoạt động không đồng bộ này. Mô hình hoạt động này của NodeJS giúp server có thể xử lý một lượng lớn kết nối đến đồng thời. NodeJS thậm chí có module buit-in sẵn để giúp người dùng thực hiện chiến lược clone này trên một server duy nhất. Hình dưới đây mô tả cách hoạt động của Node.



Việc NodeJS chạy trên một thread duy nhất dẫn đến một số hạn chế. Ví dụ như nên tránh các phép toán blocking I/O, và lỗi luôn cần được xử lí một cách đúng đắn nếu không có thể dẫn đến toàn bộ tiến trình bị vỡ. Một số lập trình viên không thích callback-based style mà JavaScript sử dụng nhưng với sự xuất hiện của kĩ thuật Promises và async await (được mặc định enable từ Node 7.6), vấn đề này chỉ còn là quá khứ.

## **Giới thiệu về Visual Studio Code**

### **Định nghĩa**

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có đánh dấu cú pháp, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi giao diện, phím tắt, và cá tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở, mặc dù gói tải xuống chính thì là có giấy phép. Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng NodeJS máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco"). Trong cuộc khảo sát vào năm 2018 trên Stack Overflow, Visual Studio Code được xếp hạng là trình biên tập mã phổ biến nhất, với 34.9% của 75398 người trả lời tuyên bố sử dụng nó.

### **Tính năng**

Visual Studio Code là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json (ví dụ như tập tin tùy chỉnh của người dùng). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tốn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.

**Giao diện**

Nó có một vài thanh sidebar, mà có thể được bố trí bên trái hoặc bên phải theo ý thích của người dùng. Cái đầu tiên cung cấp cho người dùng một số biểu tượng để chuyển đổi giữa các bảng điều khiển. Từ Explorer, Search, Git, Debug, và Extensions. Vào View để thay đổi kiểu hiển thị (full screen, zen mode, vân vân). Di chuyển chuột lên trên Explorer nó sẽ cung cấp cho chúng ta một vài lựa chọn: New File, New Folder, Refresh và Collapse All. Nhấp chuột phải mang đến một menu ngữ cảnh.

**Command Palette**

Trong bảng điều khiển này người dùng có thể tìm kiếm bất cứ điều gì người dùng thích ví dụ như những hành động nhanh chóng và dễ dàng; lệnh Git, thay đổi theme, cài đặt extension—đều có thể được thực hiện thông qua Command Palette.

**Terminal**

Điều này sẽ mở ra terminal trực tiếp trong thư mục gốc của dự án của người dùng, giúp người dùng tiết kiệm việc phải điều chuyển. Nhấp vào dấu + mở thêm cửa sổ terminal có thể được hoán đổi từ menu thả xuống. Tất cả điều này cung cấp một tính năng thực sự hữu ích, đặc biệt là nếu người dùng sử dụng compiler khi viết code.

**Git**

Những thay đổi người dùng thực hiện trong các tập tin được đánh dấu khác để người dùng biết chính xác những gì đang xảy ra: người dùng có thể thấy thanh thông báo một sự thay đổi chưa được commit, sự thay đổi được liệt kê trong bảng điều khiển Git, và một dòng đánh dấu chỉ cho người dùng nơi thay đổi trong chính tập tin. VSC cũng sẽ hiển thị phiên bản so sánh nằm cạnh nhau, chỉ để cho thêm rõ ràng. Các lệnh Git tiêu chuẩn khác chẳng hạn như sync, pull, và vân vân, có sẵn trong bảng điều khiển Git.

**Emmet (Zen Coding)**

Emmet đã được tích hợp vào VSC. Không cần cài đặt hoặc cấu hình, người dùng chỉ cần bắt đầu sử dụng nó (với Tab là phím nóng mặc định để kích hoạt các đoạn snippet của người dùng) ngay sau khi người dùng mở editor.

### **Những điểm mạnh của Visual Studio Code**

Trong VS Code, concept “workplace” cho phép chúng ta định nghĩa lại bộ cài đặt. Việc tạo một override cho một workplace cố định là khá đơn giản, tôi chỉ việc tạo folder mới và xác định một vài cái ghi đè. Nó cho phép sử dụng cùng format mà nguòi dùng định nghĩa nó trên cấp user. Nó cho phép tiếp tục dùng cài đặt của tôi cho các project cá nhân khi vẫn tương thích với các quy ước đã thỏa mãn.

Notepad là một text editor tuyệt vời cho mọi loại text có các lines kết thúc bằng CRLF, nhưng còn những loại còn lại thì sao? Edit mọi thứ mà không có các line-breaks phù hợp nghe như một cơn ác mộng vậy. Có một option đó là mở Visual Studio cho file hoặc folder đơn, nhưng người dùng sẽ phải đợi một lúc

Visual Studio chỉ hỗ trợ trên Windows, nhưng một bản khác dựa trên MonoDevelop thì hỗ trợ trên Mac. Trong khi đó Windows, Mac, và Linux đều được hỗ trợ chính thức bởi VS Code team. Điều này đồng nghĩa rằng có ít nhất thêm một tool mà chúng ta không còn bị phụ thuộc vào Windows!

## **Giới thiệu MySQL**

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên người dùng có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,… MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL). MySQL được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,…

**Ưu điểm:**

**Linh hoạt**

Sự linh hoạt về flatform là 1 đặc tính nổi bật của MySQL với các phiên bản đang được hỗ trợ của Linux, Unix, Windows, MySQL cho phép tùy biến hoàn toàn theo ý muốn, thêm vào các yêu cầu thích hợp cho database server.

**Thực thi cao**

Các chuyên gia cơ sỡ dữ liệu có thể cấu hình máy chủ cơ sở dữ liệu MySQL đặc trưng cho các ứng dụng đặc thù thông qua kiến trúc storage-engine. MySQL có thể đáp ứng khả năng xử lý những yêu cầu khắt khe nhất của từng hệ thống, MySQL còn đưa ra các “công cụ” cần thiết cho các hệ thống doanh nghiệp khó tính bằng tiện ích tải tốc độ cao, bộ nhớ cache và các cơ chế xử lý nâng cao khác.

**Sử dụng ngay**

Các tiêu chuẩn đảm bảo của MySQL  giúp cho người dùng vững tin và chọn sử dụng ngay, MySQL đưa ra nhiều tùy chọn và các giải pháp để người sử dụng dùng ngay cho server cơ sở dữ liệu MySQL

**Hỗ trợ giao dịch**

MySQL hỗ trợ giao dịch mạnh 1 cách tự động, thống nhất, độc lập và bền vững, ngoài ra khả năng giao dịch cũng được phân loại và hỗ trợ giao dịch đa dạng mà người viết không gây trở ngại cho người đọc và ngược lại. Các dữ liệu được đảm bảo toàn vẹn trong suốt quá trình server có hiệu lực và các mức giao dịch độc lập được chuyên môn hóa cao.

**Nơi tin cậy để lưu trữ web và dữ liệu**

Do MySQL có engine xử lý tốc độ cao và khả năng chèn dữ liệu nhanh, hỗ trợ tốt cho các chức năng chuyên dùng cho web,…nên MySQL là lựa chọn tốt nhất cho các ứng dụng web và các ứng dụng web doanh nghiệp.

**Bảo mật tốt**

Doanh nghiệp nào cũng cần tính năng bảo mật dữ liệu tuyệt đối vì đó chính là lợi ích quan trọng hàng đầu, và đó cũng là lý do mà các chuyên gia về cơ sỡ dữ liệu chọn dùng MySQL. MySQL có các kỹ thuật mạnh trong việc xác nhận truy cập cơ sở dữ liệu và chỉ có người dùng đã được xác nhận mới có thể truy cập vào server cơ sở dữ liệu. Bên cạnh đó, tiện ích backup và recovery cho phép backup logic và recovery toàn bộ hoặc tại 1 thời điểm nào đó.

**Phát triển ứng dụng hỗn hợp**

MySQL cung cấp hỗ trợ hỗn hợp cho bất kỳ sự phát triển ứng dụng nào nên MySQL được xem là cơ sở dữ liệu mã nguồn mở phổ biến nhất thế giới. Thư viện plugin có sẵn để nhúng vào cơ sở dữ liệu MySQL hỗ trợ trong bất kỳ ứng dụng nào. MySQL còn cung cấp các bộ kết nối cho phép tất cả các form của ứng dụng ưu tiên sử dụng MySQL như 1 server quản lý dữ liệu.

**Dễ quản lý**

Quá trình cài đặt MySQL diễn ra khá nhanh chóng trên Microsoft Windows, Linux, Macintosh hoặc Unix. Sau khi cài đặt, các tính năng tự động mở rộng không gian, tự khởi động lại và cấu hình động được thiết lập sẵn sàng cho người quản trị cơ sở dữ liệu làm việc.

MySQL còn có các công cụ quản lý đồ họa mà 1 DBA có thể quản lý, sửa chữa và điều khiển hoạt động của nhiều server, điều khiển tác vụ thiết kế dữ liệu và ETL, quản trị cơ sỡ dữ liệu hoàn thiện cũng như quản lý công việc và thực hiện kiểm tra.

**Mã nguồn mở tự do và hỗ trợ xuyên suốt**

Nhiều doanh nghiệp lo lắng việc sử dụng mã nguồn mở là không an toàn và không được hỗ trợ tốt vì đa số tin vào các phần mềm có bản quyền, nhưng đối với MySQL, các nhà doanh nghiệp hoàn toàn có thể yên tâm về điều này, MySQL có chính sách bồi thường hẳn hoi và luôn hỗ trợ tối đa cho quý doanh nghiệp.

**Chi phí thấp**

Đối với các dự án phát triển mới, nếu các doanh nghiệp sử dụng MySQL thì đó là 1 chọn lựa đúng đắn vừa tiết kiệm chi phí vừa đáng tin cậy. Mức duy trì của MySQL không chiếm nhiều thời gian sửa chữa của người quản trị cơ sở dữ liệu và các doanh nghiệp thật sự hài lòng về khả năng xử lý thông qua việc sử dụng server cơ sở dữ liệu MySQL và kiến trúc scale-out.

# Chương 3: KẾT QUẢ NGHIÊN CỨU

## **Yêu cầu chức năng**

### **Yêu cầu lưu trữ**

Hệ thống cần lưu trữ những thông tin :

* File đề cương chi tiết (Word hoặc PDF).

### **Yêu cầu tra cứu**

Hệ thống yêu cầu chức năng tra cứu thông tin:

* Tra cứu đề cương chi tiết theo: Giảng viên, học kì.

## **Yêu cầu phi chức năng**

*Giao diện*: Dễ nhìn, dễ sử dụng, gần gũi với người sử dụng đảm bảo có chức năng quản lý (tạo mới, sửa, xóa, lưu) hoặc tìm kiếm, thống kê để cho người dùng có thể truy xuất thông tin một cách dễ dàng, người quản trị hệ thống có thể cập nhật thông tin một cách thuận tiện.

*Khả năng truy xuất*: Việc truy xuất từng bảng dữ liệu phải nhanh chóng.

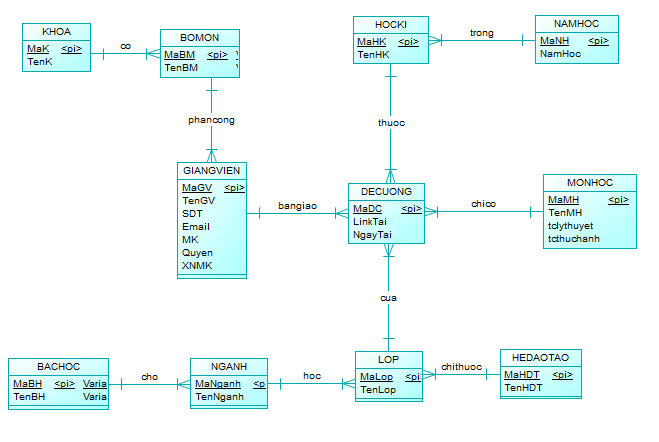
*Khả năng mở rộng bảo trì*: Chỉ có người quản trị được phép thực hiện công việc này trong quá trình hoạt động, hoặc tăng thêm tính năng cho chương trình.

|  |  |
| --- | --- |
| **Yêu cầu** | **Mô tả** |
| Tính tiến hóa | * Dễ dàng bảo trì và nâng cấp; |
| Tính tiện dụng | * Giao diện thân thiện, dễ sử dụng; * Truy cập các chức năng một cách nhanh chóng, linh hoạt; |
| Tính hiệu quả | * Tìm kiếm thông tin nhanh chóng; * Thực hiện ghi nhận thông tin dữ liệu nhanh, hiệu quả; * Lưu lại quá trình lưu trữ, cập nhật thông tin của người dùng được phân quyền trong hệ thống; |
| Tính tương thích | * Tương thích nhiều trình duyệt khác nhau; |
| Tính bảo mật | * Đối với quản trị viên: Quản lý toàn bộ hệ thống, phân quyền cho người dùng, thống kê đề cương chi tiết; * Đối với giảng viên: Tải đề cương chi tiết; * Đối với giảng viên được phân quyền: Tải đề cương chi tiết, thống kê đề cương chi tiết. |

Bảng 3. Yêu cầu phi chức năng

## **Thiết kế dữ liệu**

### **Lược đồ thực thể kết hợp (ERD)**



Hình 3. Lược đồ thực thể kết hợp

### **Danh sách các bảng**

KHOA(**MaK**, TenK)

BOMON(**MaBM,** TenBM, MaK)

GIANGVIEN(**MaGV**, TenGV, SDT, Email, MK, Quyen, XNMK, MaBM)

DECUONG(**MaDC**, LinkTai, NgayTai, MaGV, MaHK, MaMH, MaLop)

HOCKI(**MaHK**, TenHK, MaNH)

NAMHOC(**MaNH**, NamHoc)

MONHOC(**MaMH**, TenMH, tclythuyet, tcthuchanh)

LOP(**MaLop**, TenLop, MaHDT, MaNganh)

HEDAOTAO(**MaHDT**, TenHDT)

NGANH(**MaNganh**, TenNganh, MaBH)

BACHOC(**MaBH**, TenBH)

### **Thiết kế cấu trúc bảng**

Bảng KHOA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaKhoa | Mã khoa | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenKhoa | Tên khoa | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng BOMON

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaBM | Mã bộ môn | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenBM | Tên bộ môn | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng GIANGVIEN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaGV | Mã giảng viên | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenGV | Tên giảng viên | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |
| SDT | Số điên thoại giảng viên | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 10 |  |
| Email | Email giảng viên | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 30 |  |
| MK | Mật khẩu | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 100 |  |
| XNMK | Xác nhận mật khẩu | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 100 |  |
| Quyen | Quyền | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 5 |  |

Bảng DECUONG

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaDC | Mã giảng viên | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| LinkTai | Đường dẫn tải đề cương | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |
| NgayTai | Ngày tải đề cương | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 10 |  |

Bảng HOCKI

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaHK | Mã học kì | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenHK | Tên học kì | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng NAMHOC

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaNH | Mã năm học | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenNH | Tên năm học | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng MONHOC

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaMH | Mã môn học | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenMH | Tên môn học | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng LOP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaLop | Mã lớp | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenLop | Tên lớp | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng HEDAOTAO

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaHDT | Mã hệ đào tạo | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenHDT | Tên hệ đào tạo | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng NGANH

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaNganh | Mã ngành | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenNganh | Tên ngành | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |

Bảng BACHOC

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên tắt thuộc tính | Diễn giải | Loại giá trị | Kiểu dữ liệu | Miền giá trị | Chiều dài(ký tự) | Ghi chú |
| MaBH | Mã bậc học | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 15 | Khóa chính, duy nhất |
| TenBH | Tên bậc học | Bắt buộc | Varchar | Ràng buộc toàn vẹn | 50 |  |