TRƯỜNG ĐẠI HỌC SÀI GÒN

**KHOA ĐIỆN TỬ VIỄN THÔNG**

**ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH CƠ SỞ DỮ LIỆU**

TÊN ĐỀ TÀI:

**XÂY DỰNG HỆ CƠ SỞ DỮ LIỆU QUẢN LÝ**

**CỬA HÀNG CÀ PHÊ**

**Niên khóa: 2016 - 2020**

Trưởng nhóm: Nguyễn Thiện Hảo - 3116500009

Thành viên tham gia:

* + - * 1. Nguyễn Thanh Lưu - 3115520039
        2. Phạm Hồng Diêu - 3116500004

Giảng viên hướng dẫn: **ThS. Trần Thanh Nam**

**Tp. Hồ Chí Minh, 12/2019**

TRƯỜNG ĐẠI HỌC SÀI GÒN

**KHOA ĐIỆN TỬ VIỄN THÔNG**



**ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH CƠ SỞ DỮ LIỆU**

TÊN ĐỀ TÀI:

**XÂY DỰNG HỆ CƠ SỞ DỮ LIỆU QUẢN LÝ**

**CỬA HÀNG CÀ PHÊ**

**Niên khóa: 2016 - 2020**

|  |  |  |
| --- | --- | --- |
| **Xác nhận của**  **Giảng viên hướng dẫn**  *(ký, họ tên)* |  | **Trưởng nhóm**  *(ký, họ tên)* |

**Tp. Hồ Chí Minh, 12/2019**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA ĐIỆN TỬ VIỄN THÔNG**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc Lập – Tự Do – Hạnh Phúc**

**PHIẾU GIAO NHIỆM VỤ**

**THIẾT KẾ ĐỒ ÁN MÔN HỌC: “LẬP TRÌNH CƠ SỞ DỮ LIỆU”**

**1-Tên đề tài:** Xây dựng hệ cơ sở dữ liệu quản lý cửa hàng cà phê

**2-Thời gian thực hiện:** Ngày 10 tháng 10 năm 2019 – ngày 12 tháng 12 năm 2019

**3-Nhóm sinh viên thực hiện:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên sinh viên** | **Mã số sinh viên** | **Nhiệm vụ** | **Tỷ lệ**  **thực hiện** | **Chữ ký** |
| 1 | Nguyễn Thiện Hảo | 3116500009 | Dàn layout, viết PHP truy vấn CSDL, hoàn chỉnh bài báo cáo | 100% |  |
| 2 | Nguyễn Thanh Lưu | 3115520039 | Phân tích thiết kế CSDL, viết báo cáo | 100% |  |
| 3 | Phạm Hồng Diêu | 3116500004 | Thêm dữ liệu vào CSDL, viết báo cáo | 100% |  |

**4-Ngày báo cáo:** Ngày 13 tháng 12 năm 2019

**5-Kết quả báo cáo:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Nhóm/Sinh viên** | **Điểm báo cáo** | **Điểm cộng** | **Điểm tổng** |
| 0 | **Cả nhóm** |  |  |  |
| 1 | Nguyễn Thiện Hảo |  |  |  |
| 2 | Nguyễn Thanh Lưu |  |  |  |
| 3 | Phạm Hồng Diêu |  |  |  |

*TP.HCM, Ngày … tháng … năm ...*

**Giảng viên hướng dẫn**

**ThS. Trần Thanh Nam**

**LỜI CẢM ƠN**

Báo cáo đề tài “Xây dựng hệ cơ sở dữ liệu quản lý cửa hàng cà phê” thuộc môn học lập trình cơ sở dữ liệu được hoàn thành vào ngày 12/12/2019. Nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy phụ trách ThS. Trần Thanh Nam đã trực tiếp giảng dạy và hướng dẫn, giúp đỡ nhóm với những chỉ dẫn khoa học quý giá trong suốt quá trình nghiên cứu, và hoàn thành đề tài “Xây dựng hệ cơ sở dữ liệu quản lý cửa hàng cà phê” .

Xin ghi nhận công sức và những đóng góp quý báu của tất cả thành viên nhóm đã đóng góp công sức, ý kiến, thu thập tài liệu để hoàn thành bài cáo này.

Nhóm xin chân thành cảm ơn thầy!

**Trưởng nhóm**

*(ký, họ tên)*

Nguyễn Thiện Hảo

**MỤC LỤC**

**DANH MỤC HÌNH ẢNH**

# **MỞ ĐẦU**

1. **LÝ DO CHỌN ĐỀ TÀI**

Với sự phát triển rất nhanh của công nghệ thông tin, ngày nay mọi lĩnh vực đều cố gắng xây dựng các phần mềm quản lý để tiết kiệm chi phí và gia tăng hiệu quả lao động. Về phía những người dùng Internet, những tiện ích họ cần khi truy cập Internet chính là để tiếp cận thông tin một cách nhanh nhất. Hiện nay, việc sử dụng Internet đã trở thành thói quen vì sự phát triển của công nghệ thông tin giúp cho việc tiếp cận với lượng thông tin vô cùng lớn trên Internet đã trở nên dễ dàng hơn rất nhiều. Nhu cầu đã trở nên to lớn như vậy, nên việc đầu tư vào các mảng dịch vụ liên quan đến ứng dụng web trở nên màu mỡ hơn bao giờ hết. Nền kinh tế hiện nay đã thay đổi chóng mặt, những cửa hàng kể cả bán các mặt hàng thời trang hay đồ ăn, thức uống cho đến bất động sản như nhà hàng, khách sạn, nhà ở chia sẻ,… đều dần chuyển sang mô hình mới mà ở đó yếu tố công nghệ là không thể thiếu. Một ứng dụng web có thể giúp cho cả người quản lý doanh nghiệp và người dùng Internet thông thường tiết kiệm được rất nhiều thời gian và chi phí.

Đề tài “Xây dựng hệ cơ sở dữ liệu quản lý cửa hàng cà phê” của chúng em tập trung vào ngành hàng là cà phê với bố cục web được lấy ý tưởng từ ứng dụng web của “The Coffee House” để xây dựng hệ cở sở dữ liệu cho việc quản lý hàng hóa và giao dịch với người dùng. Về mặt UX/UI của ứng dụng này cũng được xây dựng bắt mắt giúp cho người dùng có được trải nghiệm tốt nhất khi sử dụng.

1. **NỘI DUNG VÀ BỐ CỤC**

**Chương 1:** Cơ sở lý thuyết

**Chương 2:** Phân tích thiết kế hệ thống

**Chương 3:** Thực nghiệm

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

## **1.1. Tổng quan về cơ sở dữ liệu**

### *1.1.1. Cơ sở dữ liệu (Database)*

Là một tập hợp các [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u_(m%C3%A1y_t%C3%ADnh)) có tổ chức, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức.

### *1.1.2. Hệ quản trị cơ sở dữ liệu (Database Management System - DBMS)*

Là [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) tương tác với người dùng cuối, ứng dụng và chính cơ sở dữ liệu để thu thập và phân tích dữ liệu. Phần mềm DBMS bao gồm các tiện ích cốt lõi được cung cấp để quản trị cơ sở dữ liệu. Tổng cộng của cơ sở dữ liệu, DBMS và các ứng dụng liên quan có thể được gọi là "hệ thống cơ sở dữ liệu". Thông thường thuật ngữ "cơ sở dữ liệu" cũng được sử dụng để nói đến bất kỳ DBMS, hệ thống cơ sở dữ liệu hoặc ứng dụng nào được liên kết với cơ sở dữ liệu.

Các nhà khoa học máy tính có thể phân loại các hệ thống quản lý cơ sở dữ liệu theo các mô hình cơ sở dữ liệu mà họ hỗ trợ. Những dữ liệu mô hình này dưới dạng các hàng và cột trong một loạt các bảng và phần lớn sử dụng [SQL](https://vi.wikipedia.org/wiki/SQL) (*Structured Query Language* - ngôn ngữ truy vấn cấu trúc) để ghi và truy vấn dữ liệu. Vào những năm 2000, các cơ sở dữ liệu phi quan hệ đã trở nên phổ biến, được gọi là [NoSQL](https://vi.wikipedia.org/wiki/NoSQL) vì chúng sử dụng các [ngôn ngữ truy vấn](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_truy_v%E1%BA%A5n) khác.

Chính thức, một "cơ sở dữ liệu" đề cập đến một tập hợp các dữ liệu liên quan và cách thức tổ chức. Truy cập vào dữ liệu này thường được cung cấp bởi "hệ thống quản lý cơ sở dữ liệu" (DBMS) bao gồm một bộ phần mềm máy tính tích hợp cho phép người dùng tương tác với một hoặc nhiều cơ sở dữ liệu và cung cấp quyền truy cập vào tất cả dữ liệu có trong cơ sở dữ liệu (mặc dù hạn chế có thể tồn tại giới hạn truy cập vào dữ liệu cụ thể). DBMS cung cấp các chức năng khác nhau cho phép nhập, lưu trữ và truy xuất số lượng lớn thông tin và cung cấp các cách để quản lý cách thức tổ chức thông tin đó.

Do mối quan hệ chặt chẽ giữa chúng, thuật ngữ "cơ sở dữ liệu" thường được sử dụng tình cờ để chỉ cả cơ sở dữ liệu và DBMS được sử dụng để xử lý và truy vấn nó.

Các DBMS hiện tại cung cấp các chức năng khác nhau cho phép quản lý cơ sở dữ liệu và dữ liệu có thể được phân loại thành bốn nhóm chức năng chính:

* Định nghĩa dữ liệu - Tạo, sửa đổi và loại bỏ các định nghĩa xác định tổ chức dữ liệu.
* Cập nhật - Chèn, sửa đổi và xóa dữ liệu thực tế.
* Truy xuất - Cung cấp thông tin dưới dạng có thể sử dụng trực tiếp hoặc để xử lý thêm bởi các ứng dụng khác. Dữ liệu được truy xuất có thể được cung cấp ở dạng cơ bản giống như được lưu trữ trong cơ sở dữ liệu hoặc ở dạng mới thu được bằng cách thay đổi hoặc kết hợp dữ liệu hiện có từ cơ sở dữ liệu.
* Quản trị - Đăng ký và giám sát người dùng, thực thi bảo mật dữ liệu, giám sát hiệu suất, duy trì tính toàn vẹn dữ liệu, xử lý kiểm soát đồng thời và khôi phục thông tin đã bị hỏng do một số sự kiện như lỗi hệ thống không mong muốn.

### *1.1.3. Các mối quan hệ trong cơ sở dữ liệu (Relationship)*

1. Quan hệ 1 – 1: Là quan hệ giữa hai tập thực thể trong đó mỗi thực thể của tập này có thể liên kết với duy nhất một thực thể của tập còn lại. Ví dụ như quan hệ giữa thực thể User và Profile chính là quan hệ 1 – 1 vì một người dùng thì chỉ có một thông tin cá nhân (hồ sơ cá nhân) của người đó hay ngược lại, mỗi hồ sơ cá nhân tương ứng với một người dùng. Quan hệ 1 – 1 được gọi là khả chuyển (transferable) nếu thực thể con có thể liên kết lại với một thực thể cha khác.
2. Quan hệ 1 – nhiều: Là quan hệ giữa hai tập thực thể trong đó một thực thể của tập này có thể liên kết với nhiều thực thể của tập còn lại. Ví dụ như quan hệ giữa thực thể Product Category và Product chính là quan hệ 1 – nhiều vì một danh mục sản phẩm sẽ có nhiều sản phẩm.
3. Quan hệ nhiều – nhiều: Là quan hệ giữa hai tập thực thể trong đó một thực thể của tập này có thể liên kết với không, một hoặc nhiều thực thể của tập kia, và ngược lại. Thường thì ở quan hệ nhiều – nhiều có thêm phần dữ liệu giao nhau để thêm thông tin cụ thể cho mối quan hệ. Ví dụ như quan hệ giữa thực thể Order và Product chính là quan hệ nhiều – nhiều vì mỗi đơn đặt hàng có thể gồm nhiều sản phẩm. Ngược lại, mỗi sản phẩm có thể xuất hiện ở nhiều đơn đặt hàng. Và chúng ta có thể thêm một thực thể Order Item để thêm thông tin cụ thể cho sản phẩm đó hoặc đơn hàng đó ví dụ số như lượng sản phẩm, giá tiền bán ra, chiết khấu bao nhiêu,…
4. Quan hệ đệ quy: Là quan hệ tồn tại giữa hai thực thể thuộc cùng một tập thực thể.

**1.2. Tổng quan về MySQL**

MySQL là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. [RDBMS](https://en.wikipedia.org/wiki/Relational_database_management_system) là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng. 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 có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X), [Unix](https://vi.wikipedia.org/wiki/Unix), [FreeBSD](https://vi.wikipedia.org/wiki/FreeBSD), [NetBSD](https://vi.wikipedia.org/w/index.php?title=NetBSD&action=edit&redlink=1), [Novell NetWare](https://vi.wikipedia.org/wiki/Novell_NetWare), [SGI Irix](https://vi.wikipedia.org/w/index.php?title=SGI_Irix&action=edit&redlink=1), [Solaris](https://vi.wikipedia.org/wiki/Solaris), [SunOS](https://vi.wikipedia.org/w/index.php?title=SunOS&action=edit&redlink=1),..

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 NodeJS, [PHP](https://vi.wikipedia.org/wiki/PHP), [Perl](https://vi.wikipedia.org/wiki/Perl), và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

### *1.2.1. Ưu điểm của MySQL*

* Dễ sử dụng: MySQL là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.
* Độ bảo mật cao: MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều nhiều tính năng bảo mật thậm chí là ở cấp cao.
* Đa tính năng: MySQL hỗ trợ rất nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
* Khả năng mở rộng và mạnh mẽ: MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
* Nhanh chóng: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

### *1.2.2. Nhược điểm của MySQL*

* Giới hạn: Theo thiết kế, MySQL không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.
* Độ tin cậy: Cách các chức năng cụ thể được xử lý với MySQL (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán,…) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.
* Dung lượng hạn chế: Nếu số bản ghi của bạn lớn dần lên thì việc truy xuất dữ liệu của bạn là khá khó khăn, khi đó chúng ta sẽ phải áp dụng nhiều biện pháp để tăng tốc độ truy xuất dữ liệu như là chia tải database này ra nhiều server, hoặc tạo cache MySQL.

### *1.2.3. Các thức vận hành của MySQL*

1. MySQL tạo ra bảng để lưu trữ dữ liệu, định nghĩa sự liên quan giữa các bảng đó.
2. Client sẽ gửi yêu cầu SQL bằng một lệnh đặc biệt trên MySQL.
3. Ứng dụng trên server sẽ phản hồi thông tin và trả về kết quả trên máy client.

### *1.2.4. Khóa chính trong MySQL (Primary key)*

Mỗi table thông thường sẽ có một khóa chính nhằm phân biệt dữ liệu giữa các record trong table. Khi làm các ứng dụng website và để tối ưu tốc độ truy vấn thì thông thường chúng ta sẽ chọn khóa chính là kiểu INT và thiết lập tăng tự động, nghĩa là khi chúng ta thêm một dòng dữ liệu thì không cần phải truyền khóa chính mà nó sẽ tự động tăng lên 1 khi thêm mới.

Giả sử chúng ta có bảng “PRODUCT” và mỗi sản phẩm là một record, lúc này để phân biệt giữa các sản phẩm thì ta không thể dựa vào tên của sản phẩm được vì tên sản phẩm có thể trùng nhau. Chính vì vậy, chúng ta chọn giải pháp là thêm một field “product\_id” dùng để làm khóa chính và giá trị của nó là duy nhất nên có thể phân biệt giữa các sinh viên với nhau.

Sau đây là một số đặc điểm của khóa chính trong MySQL nói chung và trong các mô hình CSDL nói riêng:

* Giá trị của khóa chính là duy nhất cho mỗi “record”, nghĩa là không thể tồn tại hai “record” trùng khóa chính.
* Khóa chính không thể cho phép NULL.
* Mỗi bảng chỉ tồn tại duy nhất một khóa chính, mỗi khóa chính có thể có nhiều cột.

Bởi vì trong MySQL hoạt động với kiểu số sẽ nhanh hơn kiểu chuỗi nên ta sẽ chọn kiểu INT (INT, BIGINT, ..), sau khi chạy code tạo khóa chính thì nó sẽ tạo một chỉ mục có tên là PRIMARYvà đây chính là chỉ mục khóa chính.

### *1.2.5. Khóa ngoại trong MySQL (Foreign key)*

Khóa ngoại là mối quan hệ giữa hai bảng và mối quan hệ này ta hay gọi là cha - con, nghĩa là nếu bảng A có một thuộc tính liên kết tới bảng B thì lúc này bảng B đóng vai trò là cha và bảng A đóng vai trò là con.

Khái niệm khóa ngoại không chỉ có ở MySQL mà nó là một thành phần của tất cả các hệ quản trị CSDL như SQL Server, Oracle, Access...

Có hai loại khóa ngoại đó là khóa ngoại giữa hai bảng và khóa ngoại đệ quy. Khóa ngoại của bảng này được xem như con trỏ trỏ tới khóa chính của bảng khác chính là khóa ngoại giữa hai bảng. Người sử dụng cũng không thể nhập vào cột khóa ngoại một giá trị mà giá trị đó không xuất hiện ở cột khóa chính mà khóa này trỏ tới.

**1.3. Lập trình cơ sở dữ liệu**

Với sự phát triển của công nghệ thông tin, ngày nay mọi lĩnh vực đều cố gắng xây dựng các phần mềm quản lý để tiết kiệm chi phí và gia tăng hiệu quả lao động. Làm sao chúng ta có thể lập trình được một ứng dụng sử dụng dữ liệu được lưu trữ ở một hệ quản trị cơ sở dữ liệu nào đó ví dụ như MySQL chẳng hạn. Với sự ra đời của PHP, chúng ta có thể xây dựng những ứng dụng lập trình với cơ sở dữ liệu một cách dễ dàng và nhanh chóng. Bài báo cáo của chúng em sẽ trình bày chi tiết việc lập trình với cơ sở dữ liệu sử dụng ngôn ngữ PHP và hệ quản trị cơ sở dữ liệu MySQL.

PHP là một ngôn ngữ lập trình dùng để xây dựng các ứng dụng Website, hay nói cách khác nó là ngôn ngữ chính dùng để lập trình phía Server nhằm xử lý các yêu cầu của client. Còn MySQL là một hệ quản trị cơ sở dữ liệu dùng để lưu trữ dữ liệu và nó thường được dùng kèm theo với PHP ngay từ thuở khai sinh, đây là một cặp đôi hoàn hảo dùng để xây dựng website.

### *1.3.1. Mối liên hệ giữa PHP và MySQL*

Thông thường, khi ta làm việc với MySQL thì chắc hẳn ta đã quá quen với những câu truy vấn T-SQL như select, insert, update, delete. Nếu kết hợp với PHP thì những câu truy vấn này sẽ trông như thế nào và có công dụng gì? Dưới đây là một ví dụ điển hình cho câu truy vấn ta thực thi ở phpMyAdmin để lấy danh sách các sản phẩm của cửa hàng cà phê:

SELECT \* FROM products;

MySQL là một hệ quản trị cơ sở dữ liệu nên nó có tính an toàn về bảo mật, nghĩa là nó có chức năng quản lý và phân quyền người dùng. Vì vậy trước khi vào phpMyAdmin chúng ta phải đăng nhập. Như vậy với PHP cũng cần có bước đăng nhập, sau đó sẽ là bước định nghĩa câu truy vấn và cuối cùng là thực thi và lấy kết quả.

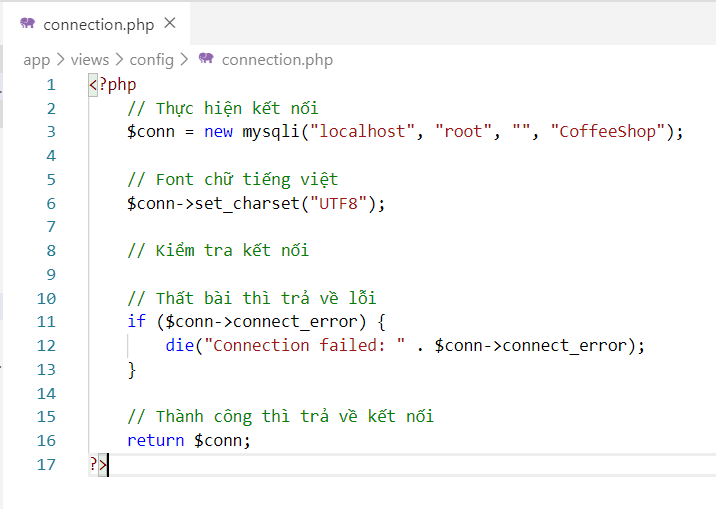
Tóm lại, PHP sẽ có nhiệm vụ kết nối MySQL và yêu cầu MySQL thực thi các câu truy vấn và trả kết quả về cho PHP để từ đó PHP lấy kết quả để xử lý.

### *1.3.2. Kết nối PHP và MySQL*

Kể từ phiên bản PHP5 trở đi chúng ta có thể kết nối với MySQL bằng cách sử dụng hai thư viện chính đó là:

* MySQLi (i là viết tắt của improved)
* PDO (PHP Data Objects)

Ở dự án này, nhóm chúng em đã chọn và sử dụng MySQLi. Ở thư viện này thì chúng ta chỉ có thể kết nối đến một hệ quản trị cơ sở dữ liệu duy nhất đó là MySQL - một nhược điểm so với việc sử dụng PDO. Thư viện này hỗ trợ Prepared Statements, đây là vấn đề quan trọng để tránh lỗi [SQL Injection](https://freetuts.net/ky-thuat-tan-cong-sql-injection-va-cach-phong-chong-trong-php-107.html). Hầu hết các Hosting, VPS hay Server đều đã cài đặt sẵn cho chúng ta bộ thư viện này. Và dưới đây là cách mà chúng em kết nối đến MySQL bằng MySQLi hướng đối tượng (Object-Oriented) trong tệp connection.php:



# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **2.1. Tổng quan về hệ thống**

Đề tài tập trung xây dựng hệ cơ sở dữ liệu và những chức năng liên quan phía khách hàng. Bố cục và trang trí trên trình duyệt (browser) được lấy ý tưởng từ ứng dụng web của “The Coffee House”. Giúp cho khách hàng dễ dàng tiếp cận đến sản phẩm và thực hiện các bước giao dịch cơ bản.

### *2.1.1. Các chức năng chính của hệ thống*

* Cho phép khách hàng đăng ký tài khoản mới.
* Cho phép khách hàng đăng nhập vào tài khoản đã đăng ký trên cơ sở dữ liệu.
* Hiển thị các sản phẩm nổi bật trong danh mục nổi bật lên trang chủ.
* Hiển thị tất cả sản phẩm chia theo danh mục sản phẩm ở trang thực đơn.
* Khách hàng có thể trực tiếp đặt hàng ở thực đơn hiện trên trang chủ.
* Khách hàng có thể chọn sản phẩm ở thực đơn hiện trên trang chủ để xem thông tin chi tiết của sản phẩm ở trang chi tiết sản phẩm rồi mới tiến hành đặt hàng.
* Khách hàng mua sản phẩm thông qua giỏ hàng.

### *2.1.2. Những xử lí cở bản ở phía client (client-side)*

* Ứng dụng web được thiết kế chú trọng về trải nghiệm người dùng và giao diện người dùng (UX/UI của ứng dụng).
* Sử dụng HTML, CSS, jQuery, AJAX để lập trình Frontend (xử lí logic ở client-side). Giúp tăng hiệu quả xử lí cho ứng dụng web. Đặc biệt là giúp tối ưu trải nghiệm cho người dùng nhờ công nghệ AJAX của jQuery để gửi các yêu cầu đến server.
* Ứng dụng web được responsive tương thích với mọi kích thước màn hình cho tất cả các thiết bị từ máy tính, laptop, các thiết bị tablet cho đến các thiết bị di động.

## **2.2. Phân tích hệ thống**

Sau quá trình khảo sát hiện trạng của hệ thống và xây dựng hệ thống, nhóm đã hoàn thành tệp cơ sở dữ liệu của hệ thống và dùng phpMyAdmin để xuất tệp. (Tệp coffeeshop.sql được lưu ở thư mục database).

### *2.2.1. Xây dựng mô hình thực thể quan hệ*

### *2.2.2. Các bảng và trường dữ liệu của bảng*

### *2.2.3. Các lệnh tạo bảng*

* **Bảng thông tin khách hàng (users)**

Bảng thông tin khách hàng có một khóa chính là trường “username” và không có khóa ngoại. Trong đó, giá trị ở trường “username” và trường “email” là duy nhất. Bảng này liên kết với bảng “orders” với quan hệ 1 – nhiều. Nghĩa là, một khách hàng có thể đặt nhiều đơn hàng, nhưng một đơn hàng chỉ có thể được một khách hàng đặt mà thôi. Trường “full\_name” có giá trị là họ và tên đầy đủ của khách hàng. Trường “password” có giá trị là mật khẩu của khách đã được hệ thống mã hóa nhờ hàm md5 của thư viện có sẵn. Nó được mã hóa ngay khi người dùng đăng ký tài khoản mới. Trường “role” có giá trị phân quyền người dùng bao gồm: quyền admin và user.

* Lệnh tạo bảng:

CREATE TABLE users (

username VARCHAR (255),

full\_name VARCHAR (255) NOT NULL,

password VARCHAR (255) NOT NULL,

email VARCHAR (255) NOT NULL,

role VARCHAR (11) DEFAULT NULL,

PRIMARY KEY (username)

);

* **Bảng sản phẩm (products)**

Bảng sản phẩm có một khóa chính là trường “product\_id” và khóa ngoại là trường “category\_id”. Trong đó, giá trị ở trường “product\_id” là duy nhất. Bảng này liên kết với bảng “products\_categories” với quan hệ 1 – nhiều. Nghĩa là, một danh mục sản phẩm sẽ có thể bao gồm nhiều sản phẩm, nhưng một sản phẩm không thuộc nhiều danh mục (vẫn có nhưng hiếm). Trường “product\_name” có giá trị là tên của sản phẩm. Trường “product\_img” có giá trị là liên kết đến hình ảnh minh họa cho sản phẩm đó (hình ảnh được tải ở server). Trường “list\_price” có giá trị là giá bán của sản phẩm cho khách hàng (chưa được chiết khấu). Trường “product\_detail” có giá trị là mô tả chi tiết về tính chất, mùi vị của sản phẩm.

* Lệnh tạo bảng:

CREATE TABLE products (

product\_id INT (11),

product\_name VARCHAR (255) NOT NULL,

product\_img VARCHAR (255) NOT NULL,

category\_id INT (11) NOT NULL,

list\_price DECIMAL (10,2) NOT NULL,

PRIMARY KEY (product\_id),

FOREIGN KEY (category\_id)

REFERENCES products\_categories (category\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

* **Bảng cửa hàng (stores)**

Bảng của hàng có một khóa chính là trường “store\_id” và không có khóa ngoại. Trong đó, giá trị ở trường “store\_id” là duy nhất. Bảng này không liên kết với bảng khác. Trường “store\_name” có giá trị là tên của cửa hàng (và cũng là địa chỉ của cửa hàng). Trường “store\_img” có giá trị là liên kết đến hình ảnh minh họa cho cửa hàng đó (hình ảnh được tải ở server). Các trường “phone”, “email”, “state”, “city”, “zip\_code” có giá trị là những thông tin liên lạc của cửa hàng.

* Lệnh tạo bảng:

CREATE TABLE products (

store\_id INT (11) PRIMARY KEY,

store\_name VARCHAR (255) NOT NULL,

store\_img VARCHAR (255) NOT NULL,

phone VARCHAR (25) NOT NULL,

email VARCHAR (255) NOT NULL,

state VARCHAR (10) NOT NULL,

zip\_code VARCHAR (5) NOT NULL,

);

* **Bảng danh mục sản phẩm (products\_categories)**

Bảng danh mục sản phẩmcó một khóa chính là trường “category\_id” và không có khóa ngoại. Trong đó, giá trị ở trường “category\_id” là duy nhất. Bảng này liên kết với bảng “products” như đã trình bày ở mục bảng sản phẩm. Trường “category\_name” có giá trị là tên của danh mục.

* Lệnh tạo bảng:

CREATE TABLE products\_categories (

category\_id INT (11) PRIMARY KEY,

category \_name VARCHAR (255) NOT NULL

);

* **Bảng đơn đặt hàng (orders)**

Bảng đơn đặt hàng có một khóa chính là trường “order\_id” và khóa ngoại là trường “username”. Trong đó, giá trị ở trường “order\_id” là duy nhất. Bảng này liên kết với bảng “order\_items” với quan hệ 1 - 1. Bởi vì bảng “order\_items” phần dữ liệu giao nhau để thêm thông tin cụ thể cho mối quan hệ giữa 2 bảng “products” và bảng “orders”. Bảng này còn liên kết với bảng “users” với quan hệ 1 – nhiều. Nghĩa là một đơn hàng chỉ có thể cho một khách hàng đặt, và khách hàng thì có thể đặt nhiều đơn hàng. Trạng thái đơn hàng: 1 = Chờ; 2 = Đang xử lí; 3 = Hủy bỏ; 4 = Hoàn thành. Trường “order\_date” có giá trị là thời gian mà đơn hàng được đặt.

* Lệnh tạo bảng:

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

username VARCHAR (255),

status tinyint NOT NULL,

order\_date DATE NOT NULL,

FOREIGN KEY (username)

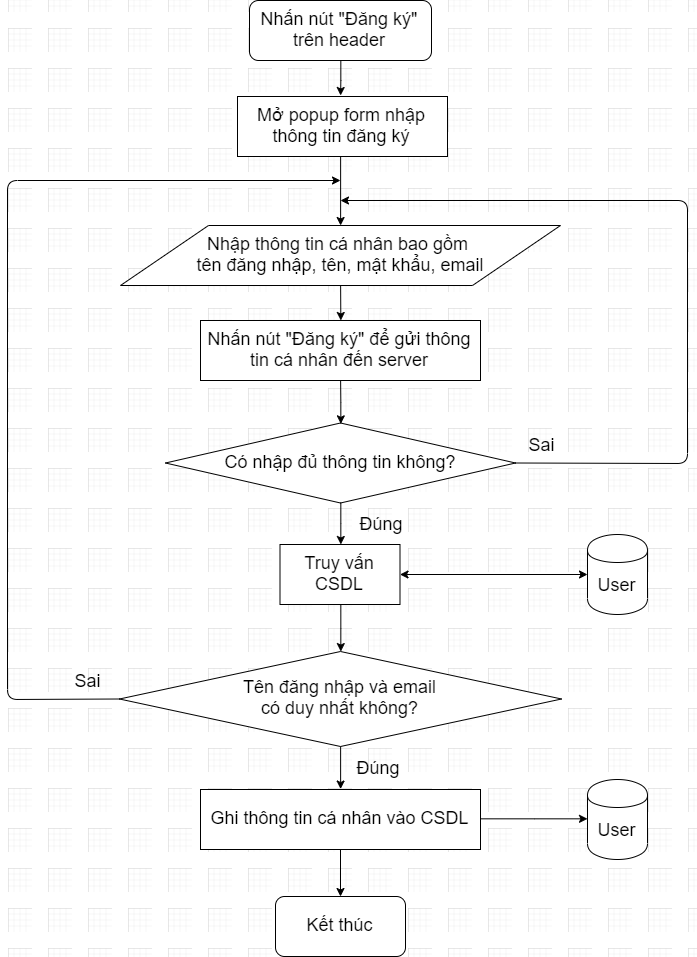
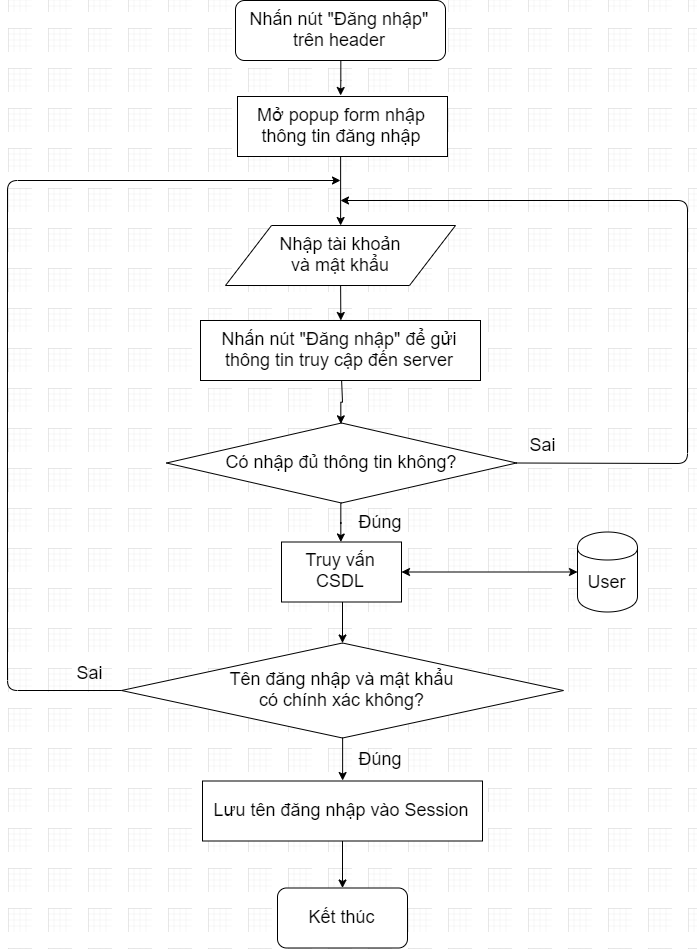
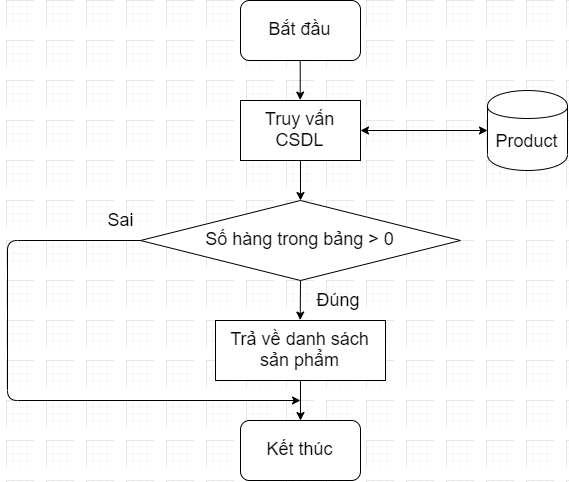
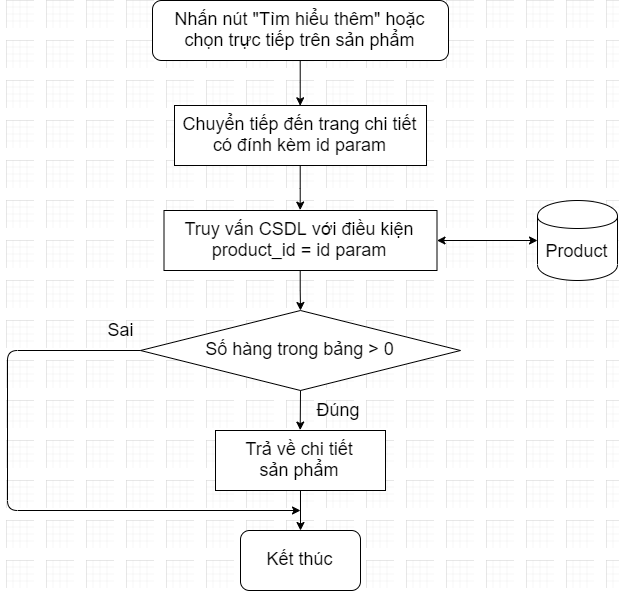
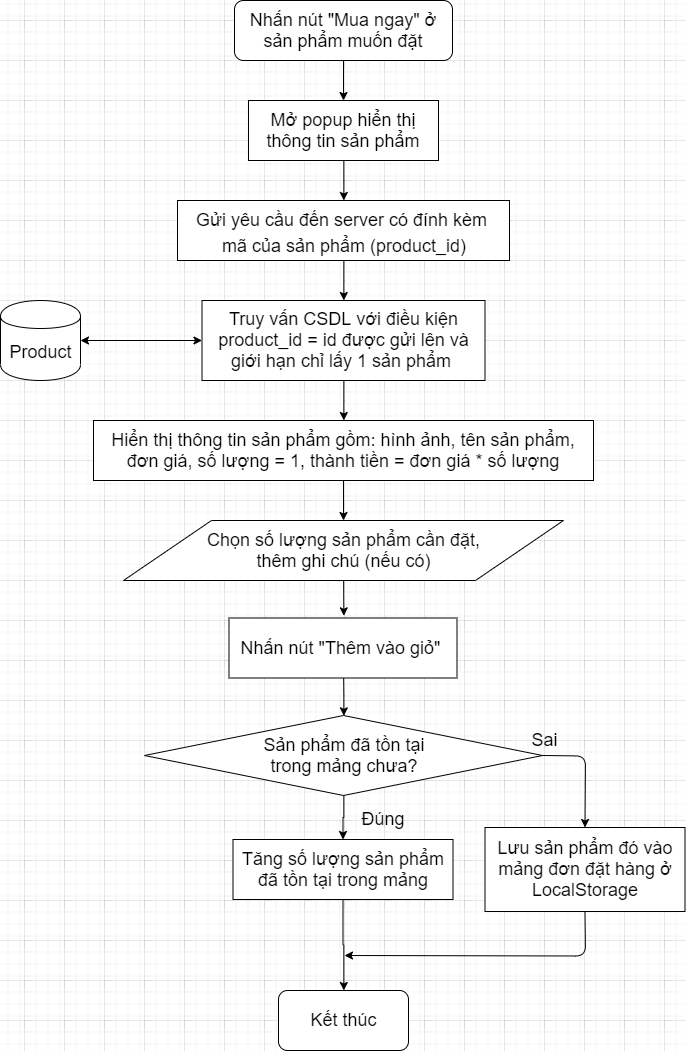
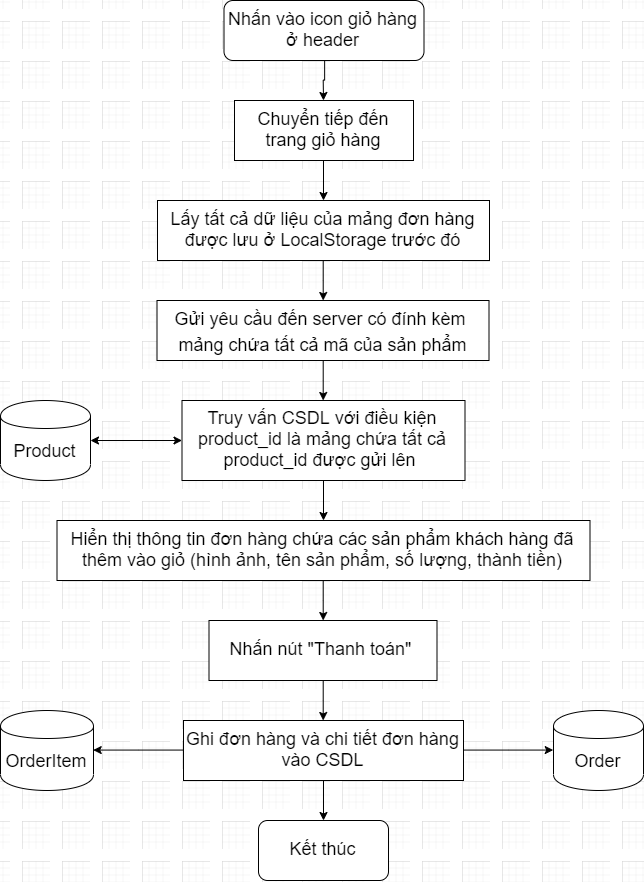
REFERENCES users (username)

ON DELETE CASCADE ON UPDATE CASCADE

);

### *2.2.4. Sơ đồ phân cấp chức năng*

### *2.2.5. Lưu đồ giải thuật của các chức năng chính*

* **Đăng ký thành viên**
* **Đăng nhập**
* **Hiển thị danh sách sản phẩm**
* **Xem chi tiết sản phẩm**
* **Đặt hàng**
* *Xử lí các sản phẩm được chọn vào giỏ hàng*
* *Xử lí đơn hàng và lưu đơn hàng vào cơ sở dữ liệu*

### *2.2.6. Các lệnh truy vấn cơ sở dữ liệu của hệ thống phân theo chức năng*

* **Đăng ký thành viên**

Khai báo biến để gán dữ liệu được gửi từ người dùng qua phương thức POST. Mật khẩu ngay khi được phía người dùng gửi lên sẽ được mã hóa (hash password).

$username = mysqli\_real\_escape\_string($conn, $\_POST['username']);

$fullname = mysqli\_real\_escape\_string($conn, $\_POST['fullname']);

$password = md5($\_POST['password']);

$email = mysqli\_real\_escape\_string($conn, $\_POST['email']);

Truy vấn cơ sở dữ liệu để lấy giá trị của trường “username” và “email” để so sánh với tài khoản và email của người dùng nhập vào:

**SELECT** \*

**FROM** users

**WHERE** username = '$username' OR email = '$email'

Vì username và email phải duy nhất nên trường hợp nếu câu truy vấn này trả về kết quả số hàng lớn hơn 0, nghĩa là username hoặc email đã tồn tại trên cơ sở dữ liệu rồi => trả về cho người dùng thông báo tài khoản hoặc email đã tồn tại. Yêu cầu người dùng chọn một tài khoản hoặc email khác. Ngược lại, nếu câu truy vấn này trả về kết quả số hàng là 0, nghĩa là username hoặc email chưa tồn tại trên cơ dữ liệu thì sẽ thực hiện lệnh ghi dữ liệu vào cơ sở dữ liệu:

**INSERT INTO** users (username, fullname, password, email)

**VALUES** ('$username', '$fullname' , '$password', '$email')

=> trả về cho người dùng thông báo đã đăng ký thành công.

* **Đăng nhập**

Khai báo biến để gán dữ liệu được gửi từ client qua phương thức POST

$username = $\_POST['username'];

$password = $\_POST['password'];

Sau khi nhận được dữ liệu từ người dùng thì điều đầu tiên là mã hóa mật khẩu thông qua hàm md5 từ thư viện có sẵn: $password = md5($password). Tiếp đến là thực hiện lệnh truy vấn cơ sở dữ liệu để lấy giá trị của trường “username” và “password” để so sánh với tài khoản và mật khẩu của người dùng nhập vào:

**SELECT** username, password

**FROM** users

**WHERE** username = '$username'

Nếu câu truy vấn này trả về kết quả số hàng bằng 0, nghĩa là tên đăng nhập này không tồn tại trên cơ sở dữ liệu. Đồng nghĩa với việc tài khoản này chưa được đăng ký.

Ngược lại, nếu câu truy vấn này trả về kết quả số hàng lớn hơn 0, nghĩa là tài khoản này có tồn tại trên cơ sở dữ liệu. Đồng nghĩa với việc tài khoản này đã được đăng ký. Sau đó lấy mật khẩu trong cơ sở dữ liệu lên và so sánh với mật khẩu được mã hóa trước đó. Nghĩa là, ở cơ sở dữ liệu hiện tại giá trị với trường “password” đang là mật khẩu được mã hóa lấy lên để so sánh với mật khẩu từ người dùng cũng được mã hóa. Nếu đúng mật khẩu thì trả về cho người dùng thông báo đã đăng nhập thành công.