

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
Khoa Khoa học - Kỹ thuật Máy tính



BÁO CÁO BÀI TẬP LỚN SỐ 1
HỆ CƠ SỞ DỮ LIỆU

ĐỀ TÀI: : E-BOOKSTORE

GVHD: Võ Thị Ngọc Châu

SV thực hiện: Tô Duy Hưng – 1810198
Trương Thanh Lộc – 1812982

TP. HỒ CHÍ MINH, THÁNG 11/2020

Mục lục

1	Giới thiệu đề tài	2
2	Mô tả dữ liệu	3
2.1	Các thực thể (Entities)	3
2.2	Các liên kết (Relations)	3
2.3	Các thuộc tính	4
2.4	Các ràng buộc	4
2.5	Các ràng buộc về giá trị	4
2.6	Lược đồ quan hệ - thực thể (ER diagram)	5
2.7	Lược đồ cơ sở dữ liệu quan hệ	6
3	Lựa chọn DBMS	7
3.1	MySQL	7
3.1.1	Giới thiệu	7
3.1.2	Lý do mà nhóm lựa chọn MySQL	7
3.1.3	Cú pháp lệnh truy vấn Select hỗ trợ bởi MySQL	7
3.2	Workbench	8
4	Định nghĩa trên DBMS	9
4.1	Định nghĩa lược đồ cơ sở dữ liệu quan hệ	9
4.2	Định nghĩa nhóm người dùng	9
5	Viết câu lệnh SQL	10
5.1	Ngôn ngữ định nghĩa dữ liệu - DDL	10
5.2	Ngôn ngữ thao tác dữ liệu - DML	14
6	Công nghệ phát triển ứng dụng cơ sở dữ liệu	18
6.1	NodeJS	18
6.1.1	NodeJs là gì	18
6.1.2	Cài đặt	18
6.1.3	Ưu điểm & Nhược điểm	19
6.1.4	Module	19
6.2	Mô hình MVC	21
7	Phân chia công việc	23

1 Giới thiệu đề tài

Xem xét quản lý mua/ bán sách tại hiệu sách trực tuyến.

Hiệu sách giới thiệu và bán qua mạng nhiều sách. Các sách được lưu trữ trong các kho hàng của hiệu sách ở nhiều nước trên thế giới để phục vụ khách hàng đặt mua gần nhất có thể. Số lượng sách hiển thị trên website bán hàng là tổng lượng sách đang có từ tất cả các kho hàng của hiệu sách. Khi không còn sách trong bất kỳ kho hàng nào, sách được ghi chú là “hết hàng”. Khi này, hiệu sách sẽ đặt hàng với số lượng cụ thể với nhà xuất bản tương ứng để nhập hàng vào kho hàng được chỉ định.

Sách được viết bởi một hay nhiều tác giả. Một tác giả có một hay nhiều sách được xuất bản.

Sách được phân theo các lĩnh vực khác nhau như khoa học tự nhiên, khoa học xã hội, văn học, lịch sử, ... Một số từ khóa phản ánh nội dung của sách được trích và lưu kèm với sách. Ngoài sách dạng bản in giấy truyền thống, hiệu sách cũng giới thiệu, bán, và cho thuê sách điện tử. Các sách điện tử không có lưu kho vật lý mà được quản lý trên kho lưu trữ ảo.

Hình thức giao hàng cho mỗi loại sách này (truyền thống hay điện tử) cũng khác nhau. Sách truyền thống sẽ được gửi qua bưu điện đến địa chỉ giao hàng được yêu cầu của khách hàng. Trong khi đó, sách điện tử được đặt ở đường link trên trang web bán hàng được cá nhân hóa cho khách hàng và đồng thời gửi đến email của khách hàng tùy dạng thương mại là “mua” hay “thuê” sách điện tử. Nếu là “mua” thì khách hàng có thể download sách điện tử từ đường link được gửi qua email hoặc trên trang web bán hàng được cá nhân hóa cho khách hàng. Nếu là “thuê” thì khách hàng chỉ có thể xem sách điện tử trên web browser từ đường link trên email hoặc trên trang web bán hàng được cá nhân hóa cho khách hàng.

Để mua sách, khách hàng cần phải đăng ký tài khoản trên website của hiệu sách. Sau đó khách hàng có thể mua 1 hay nhiều sách với số lượng khác nhau cho mỗi ISBN. Cho mỗi lần mua sách, khách hàng có thể chọn hình thức thanh toán: thẻ tín dụng hay chuyển khoản. Khách hàng có thể đăng ký 1 hay nhiều thẻ tín dụng. Thông tin về thẻ tín dụng bao gồm tên chủ thẻ, mã thẻ, tên ngân hàng, tên chi nhánh ngân hàng mà thẻ liên kết tài khoản, ngày hết hạn. Khi khách hàng đăng ký chuyển khoản, phiếu mua hàng sẽ xuất cho khách hàng bao gồm thông tin về ngân hàng của hiệu sách để nhận thanh toán của khách hàng và giá trị đơn hàng. Khi này các sách được chọn mua chưa được ghi nhận là “xuất kho” mà chỉ được ghi nhận “chờ mua”. Khi thanh toán hoàn thành, các sách được ghi nhận là “xuất kho” và quá trình chuyển hàng được kích hoạt.

Chương trình ứng dụng được phát triển từ đề tài này dự kiến được dùng bởi:

1. Nhân viên của hiệu sách
2. Khách hàng
3. Tác giả

Nhân viên của hiệu sách sẽ kiểm tra và phản hồi các yêu cầu mua sách của các khách hàng khi các giao dịch trực tuyến gặp sự cố. Nhân viên của hiệu sách sẽ kiểm tra và chuẩn bị sách trong các kho sách của hiệu sách. Do đó, nhân viên của hiệu sách cập nhật các giao dịch mua hàng, trong đó có phần thanh toán. Nhân viên của hiệu sách cũng cập nhật các thông tin liên quan đến các kho sách khi nhập kho và xuất kho. Khách hàng có thể cập nhật thông tin cá nhân khi và sau khi đăng ký với website của hiệu sách. Khách hàng có thể cập nhật các giao dịch mua hàng khi chưa được xử lý. Khách hàng có thể phản hồi về sản phẩm đã mua cũng như các giao dịch đã thực hiện với hiệu sách.

2 Mô tả dữ liệu

2.1 Các thực thể (Entities)

Thực thể mạnh	Mô tả	Thuộc tính
Tác giả	Chứa thông tin của tác giả	AID - ID của tác giả (key), tên tác giả (thuộc tính hỗn hợp chứa họ và tên của tác giả), năm sinh, email
Khách hàng	Chứa thông tin khách hàng	UID - ID của khách hàng (key), tên khách hàng (thuộc tính hỗn hợp chứa họ và tên của khách hàng), năm sinh, email, phone
Nhân viên	Chứa thông tin nhân viên	SID - ID của nhân viên (key), tên nhân viên (thuộc tính hỗn hợp chứa họ và tên nhân viên), email
Kho	Chứa thông tin kho hàng	Mã kho (key)
Thẻ tín dụng	Chứa thông tin thẻ tín dụng	Mã thẻ (key), tên chủ thẻ, tên ngân hàng, ngày hết hạn
Thanh toán thẻ	Chứa thông tin thanh toán thẻ	Mã thanh toán (key)
Sách	Chứa thông tin của sách	ISBN (key), tên sách, giá sách, từ khóa, lĩnh vực
Bản điện tử	Chứa thông tin sách điện tử	Code (key)
Nhà xuất bản	Chứa thông tin nhà xuất bản	Tên nhà xuất bản (key)
Thực thể yếu	Mô tả	Thuộc tính
Giao dịch	Chứa thông tin giao dịch và được xác định bởi khách hàng và sách, có liên kết ràng buộc với thực thể Sách và Thanh Toán	Ngày/giờ (partial key), giá trị

2.2 Các liên kết (Relations)

Liên kết 2 ngôi

Thực thể 1	Ràng buộc 1	Liên kết	Thực thể 2	Ràng buộc 2
Khách hàng	1 - không bắt buộc	Sở hữu	Thẻ tín dụng	N - bắt buộc
Thanh toán thẻ	1 - không bắt buộc	Sử dụng	Thẻ tín dụng	N - không bắt buộc
Tác giả	N - không bắt buộc	Viết	Sách	M - bắt buộc
Bản in	N - bắt buộc	Được lưu trữ	Kho	N - không bắt buộc
Nhân viên	1 - không bắt buộc	Kiểm kho	Kho	1 - bắt buộc
Sách	1 - bắt buộc	Được xuất bản	Nhà xuất bản	N - không bắt buộc
Sách	1 - bắt buộc	Được phản hồi	Khách hàng	N - không bắt buộc
Thanh toán	1 - bắt buộc	Thanh toán cho	Giao dịch	N - không bắt buộc

Liên kết 3 ngôi

- Liên kết: Thực hiện cho (Đây cũng chính là liên kết ràng buộc dùng để xác định cho thực thể yếu Giao dịch)
 - Thực thể 1: Sách - Ràng buộc: M - Không bắt buộc
 - Thực thể 2: Khách hàng - Ràng buộc: 1 - Không bắt buộc
 - Thực thể 3: Giao dịch - Ràng buộc: N - Bắt buộc
- Liên kết: Quản lý
 - Thực thể 1: Bản in - Ràng buộc N - Không bắt buộc
 - Thực thể 2: Kho - Ràng buộc M - Không bắt buộc
 - Thực thể 3: Nhân viên - Ràng buộc K - Không bắt buộc

2.3 Các thuộc tính

Bên cạnh thuộc tính khóa (key) và các thuộc tính khác được liệt kê trong bảng mô tả thực thể bên trên, cần lưu ý thêm:

- Giao dịch là một thực thể yếu vì giao dịch được xác định bởi các thực thể mạnh khác là Sách và Khách Hàng. Đối với những giao dịch có Mã khách hàng UID và mã ISBN giống nhau thì chúng sẽ được phân biệt bởi thuộc tính (ngày/giờ), do đó (ngày/giờ) chính là khóa liên phần (partial key) của thực thể yếu Giao dịch.
- Thực thể giao dịch còn có một thuộc tính dẫn xuất là giá trị, thuộc tính này có thể tính ra nhờ thuộc tính giá tiền của thực thể sách.
- Thuộc tính hỗn hợp Tên nhân viên/ tên tác giả / tên khách hàng, gồm các giá trị là Họ và Tên ở các thực thể Nhân viên, tác giả, khách hàng.
- Thuộc tính đa trị là từ khóa và lĩnh vực của thực thể sách vì mỗi sách có thể là nhiều lĩnh vực khác nhau và có thể có nhiều từ khóa.
- Thuộc tính đa trị năm xuất bản nằm ở liên kết "được xuất bản" vì một sách có thể có nhiều năm xuất bản khác nhau
- Thuộc tính đa trị Tin phản hồi ở liên kết "được phản hồi" vì một khách hàng có thể thực hiện nhiều phản hồi ở nhiều thời điểm khác nhau cho từng sản phẩm.

2.4 Các ràng buộc

- Mỗi cuốn sách bắt buộc phải có tác giả nhưng một tác giả có thể chưa sáng tác ra cuốn sách nào
- Một cuốn sách có thể được viết bởi một cá nhân hoặc một nhóm tác giả, ngược lại một tác giả cũng có thể viết ra nhiều cuốn sách
- Sách truyền thống bắt buộc lưu giữ tại kho lưu trữ vật lý, còn kho lưu trữ vật lý có thể không bắt buộc lưu giữ sách truyền thống (trong trường hợp kho đã hết hàng hoặc là kho mới được xây dựng và chưa nhập hàng về). Tương tự đối với sách điện tử và kho lưu trữ ảo.
- Nhà kho bắt buộc phải được kiểm tra bởi một nhân viên của hiệu sách, tuy nhiên nhưng không phải nhân viên hiệu sách nào cũng được quản lý kho không bắt buộc nhân viên hiệu sách phải kiểm tra kho.
- Bên cạnh đó, trong lược đồ quan hệ - thực thể trên còn có kiểu liên kết cụ thể hóa. Hai thực thể là *Bản điện tử* và *Bản in* được thừa kế chung thuộc tính ISBN của thực thể cha là *Sách*. Tuy nhiên giữa chúng có điểm khác nhau là thực thể *Bản Điện Tử* có chứa thuộc tính code còn thực thể *Bản In* thì không, thực thể *Bản Điện Tử* không có liên kết với bất kỳ thực thể nào trong khi đó *Bản In* có 2 mối liên kết là *Quản Lý* và *Được Lưu Trữ*. Tương tự, hai thực thể là *Chuyển Khoản* và *Thanh Toán Thẻ* được thừa kế chung thuộc tính Hóa Đơn của thực thể cha là *Thanh Toán*. Tuy nhiên giữa chúng có điểm khác nhau là thực thể *Thanh Toán Thẻ* có chứa thuộc tính Mã Thanh Toán còn thực thể *Chuyển Khoản* thì không, thực thể *Chuyển Khoản* không có liên kết với bất kỳ thực thể nào trong khi đó *Thanh Toán Thẻ* có 1 mối liên kết là *sử dụng* với *Thẻ Tín Dụng*.

2.5 Các ràng buộc về giá trị

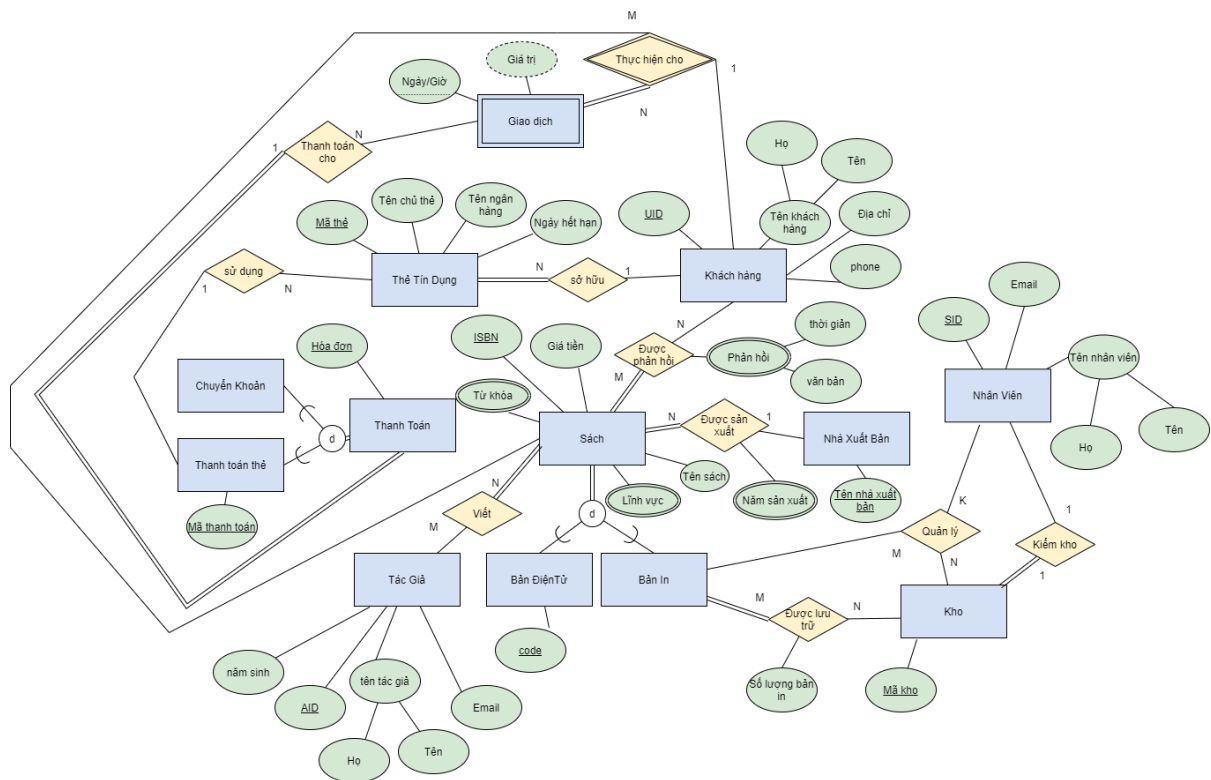
1. Theo mô tả

- Ngày/Giờ phản hồi sách/giao dịch lớn hơn Ngày/Giờ được thanh toán cho các sách liên quan phản hồi.
- Tổng lượng sách hiện có trong tất cả các kho của một sách bản in không kể số lượng đang được xử lý trong các giao dịch chờ mua sách.
- Thông tin khách hàng giống với thông tin thẻ tín dụng được khách hàng dùng trong tất cả các thanh toán.
- Thuê chỉ được áp dụng cho sách điện tử.

2. Phần mở rộng

- ID phải là một chuỗi kí tự (có thể là chữ cái hoặc số) có độ dài là 10. Riêng ID tác giả phải được bắt đầu bằng chữ A và ID nhân viên bắt đầu bằng chữ S để phân biệt chức năng của các tài khoản này. *giá tiền* chỉ được nhận giá trị là một số nguyên không âm, không nhận giá trị null.
- Thuộc tính *phone* chỉ được phép chứa kí tự là số và có độ dài từ 9 đến 11 số.
- Thuộc tính *Địa chỉ* và *Phone* của *Khách hàng* không được phép nhận giá trị null để tránh trường hợp khách đặt sách mà sau đó nhân viên không biết địa chỉ để gửi đi.
- Thuộc tính *Năm sản xuất* của liên kết *Được sản xuất* giữa thực thể *Sách* và *Nhà xuất bản* phải là một số nguyên nhỏ hơn 9999. Tương tự, thuộc tính *Năm sinh* của *Tác giả* cũng có ràng buộc giống vậy.
- Thuộc tính *Giá tiền* của *Sách* phải là một số nguyên dương và không được phép là null.
- Thuộc tính *Số lượng bản in* của liên kết *Được lưu trữ* giữa thực thể *Bản in* và *Kho* là một số nguyên không âm, có thể bằng 0 được.

2.6 Lược đồ quan hệ - thực thể (ER diagram)



2.7 Lược đồ cơ sở dữ liệu quan hệ

NhanVien(SID, Email, Ho, Ten)
Kho(MaKho)
NhaXuatBan(TenNhaXuatBan)
TacGia(AID, Ho, Ten, NamSinh, Email)
Sach(ISBN, TenSach, GiaTien)
BanIn(ISBN)
FK: ISBN to Sach.ISBN
BanDienTu(ISBN, code)
Secondary key: code
FK: ISBN to Sach.ISBN
KhachHang(UID, Ho, Ten, DiaChi, Phone)
TheTinDung(MaThe, TenChuThe, TenNganHang, NgayHetHan)
ThanhToan(HoaDon)
ChuyenKhoan(HoaDon)
FK: HoaDon to ThanhToan.HoaDon
ThanhToanThe(HoaDon, MaThanhToan)
Secondary key: MaThanhToan
FK: HoaDon to ThanhToan.HoaDon
GiaoDich(UID, ISBN, NgayGio)
FK: UID to KhachHang.UID, ISBN to Sach.ISBN

- Kho(MaKho, SID)
FK: SID to NhanVien.SID
Not null: SID
- Sach(ISBN, TenSach, GiaTien, TenNhaXuatBan)
FK: TenNhaXuatBan to NhaXuatBan.TenNhaXuatBan
Not null: TenNhaXuatBan
- TheTinDung(MaThe, TenChuThe, TenNganHang, NgayHetHan, UID)
FK: UID to KhachHang.UID
Not null: UID
- TheTinDung(MaThe, TenChuThe, TenNganHang, NgayHetHan, UID, HoaDon)
FK: UID to KhachHang.UID, HoaDon to ThanhToanThe(HoaDon)
- GiaoDich(UID, ISBN, NgayGio, HoaDon)
FK: UID to KhachHang.UID, ISBN to Sach.ISBN, HoaDon to ThanhToan(HoaDon)

LuuTru(MaKho, ISBN, SoLuongBanIn)
FK: MaKho to Kho.MaKho, ISBN to BanIn.ISBN
QuanLy(MaKho, SID, ISBN)
FK: MaKho to Kho.MaKho, SID to NhanVien.SID, ISBN to BanIn.ISBN
Viet(AID, ISBN)
FK: AID to TacGia.AID, ISBN to Sach.ISBN
PhanHoi(UID, ISBN)
FK: UID to KhachHang.UID, ISBN to Sach.ISBN
TuKhoa(ISBN, ATuKhoa)
FK: ISBN to Sach.ISBN
LinhVuc(ISBN, ALinhVuc)
FK: ISBN to Sach.ISBN
NamSanXuat(ISBN, TenNhaXuatBan, ANamSanXuat)
FK: ISBN to Sach.ISBN, TenNhaXuatBan to NhaXuatBan.TenNhaXuatBan
TimPhanHoi(UID, ISBN, ThoiGian, VanBan)
FK: (UID, ISBN) to PhanHoi.(UID, ISBN)

3 Lựa chọn DBMS

3.1 MySQL

3.1.1 Giới thiệu

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở hoạt động theo mô hình client-server, được sử dụng để tạo ra và quản lý các cơ sở dữ liệu theo hình thức quản lý các mối liên hệ giữa chúng.

Cách vận hành chính trong môi trường của MySQL là:

- MySQL tạo ra bảng để lưu dữ liệu, định nghĩa sự liên quan giữa các bảng đó.
- Client (có thể là MySQL Workbench, Sequel ro, HeidiSQL, phpMyAdmin,...) sẽ gửi yêu cầu SQL bằng một lệnh đặc biệt trên MySQL.
- Sau đó, ứng dụng trên server (cũng chính là máy tính của mình) sẽ phản hồi thông tin và trả về kết quả trên client.

3.1.2 Lý do mà nhóm lựa chọn MySQL

- Đây là một phần mềm mã nguồn mở nên có thể tải về miễn phí từ trang chủ <http://dev.mysql.com/downloads/>
- MySQL có nhiều phiên bản khác nhau và hỗ trợ cho nhiều hệ điều hành khác nhau như: Window, Linux, Mac OS X, Unix, Solaris ... Do đó, các thành viên trong nhóm dù không sử dụng chung một hệ điều hành đều có thể sử dụng được MySQL.
- MySQL là cơ sở dữ liệu có tốc độ cao, ổn định và dễ sử dụng. Quá trình cài đặt và học cách sử dụng vô cùng đơn giản, không tốn nhiều thời gian để thích nghi.
- Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho việc truy cập các CSDL trên internet
- MySQL có thể xử lý một lượng lớn dữ liệu một cách hiệu quả và tiết kiệm chi phí, Ngoài ra, nó có khả năng mở rộng mạnh mẽ nếu cần thiết.
- Tuy nhiên bên cạnh đó, MySQL có một hạn chế là dung lượng khá hạn chế. Nếu số bản ghi lớn dần lên thì việc truy xuất dữ liệu có thể sẽ gặp một số 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. Nhưng với quy mô thực hiện ứng dụng chỉ là trong môn học Cơ sở dữ liệu (tn), hạn chế này có thể xem như không ảnh hưởng đáng kể.

3.1.3 Cú pháp lệnh truy vấn Select hỗ trợ bởi MySQL

1. **Select** (bắt buộc)
2. **Expression** (bắt buộc)
3. **From** <bảng dữ liệu>
4. **Where** <điều kiện truy vấn> (không bắt buộc)
5. **Group By** (không bắt buộc)
6. **Having** (không bắt buộc)
7. **Order By** (không bắt buộc)
8. **Limit** (không bắt buộc)

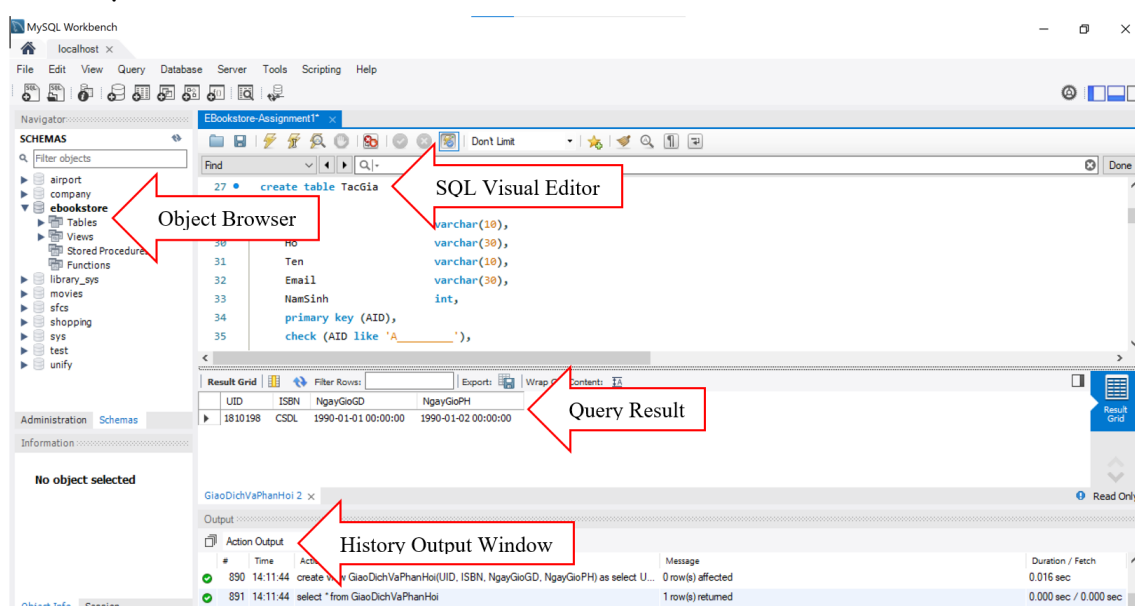
3.2 Workbench

Định nghĩa

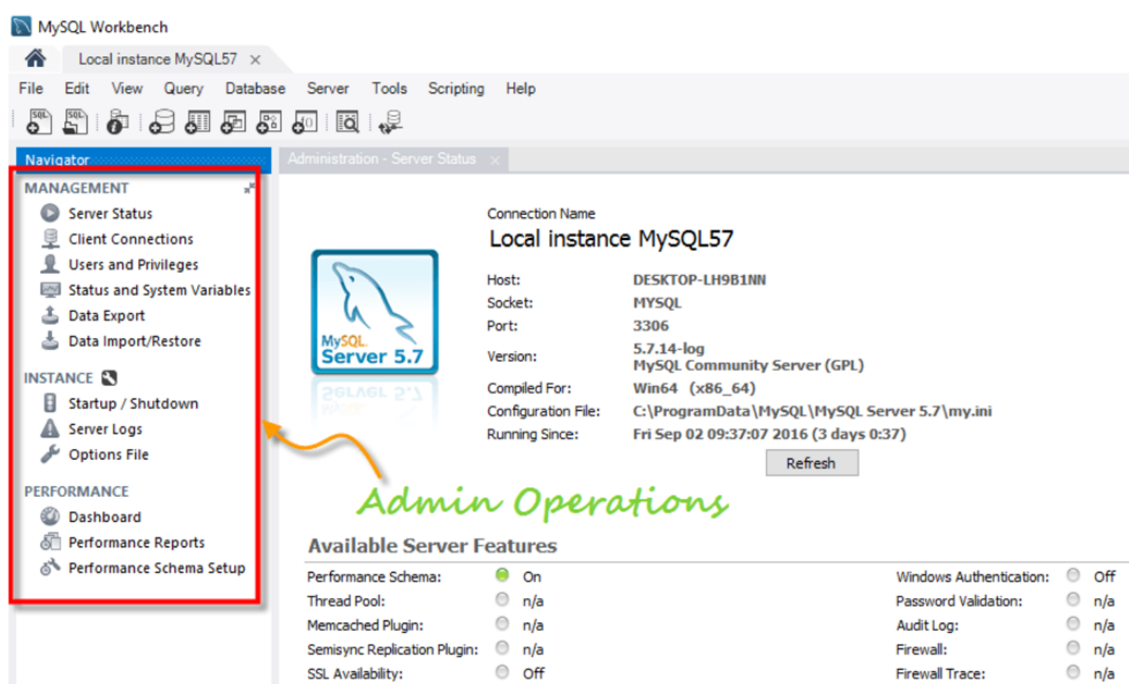
- **MySQL Workbench** tạo ra các thiết kế mô hình database một cách trực quan và rất đơn giản khi dịch sang cơ sở dữ liệu MySQL.
- **MySQL Workbench** còn hỗ trợ người dùng tạo nhiều modeling trong cùng một môi trường cũng như là hỗ trợ các đối tượng khác như trình xem, bảng biểu, trình kích hoạt, phương pháp lưu trữ, ... để tạo nên một cơ sở dữ liệu. Nó còn cho phép chỉnh sửa các modeling đồ họa của bạn bằng cách dùng Table Editor, một công cụ thường sử dụng để sửa các Index, Column, Privileges, Routines and Views, ...

Hiển thị

1. Giao diện



2. Bảng quản trị



4 Định nghĩa trên DBMS

4.1 Định nghĩa lược đồ cơ sở dữ liệu quan hệ

Để định nghĩa lược đồ cơ sở dữ liệu quan hệ trên MySQL command line, ta cần phải:

- Bỏ đi cơ sở dữ liệu quan hệ có tên là "eBookstore" nếu có tồn tại bằng lệnh:
DROP DATABASE IF EXISTS eBookStore;
- Tạo cơ sở dữ liệu quan hệ mới có tên là "eBookstore" bằng lệnh
CREATE DATABASE eBookStore;
- Sử dụng cơ sở dữ liệu quan hệ này để tạo các bảng dữ liệu bằng lệnh:
USE eBookStore;

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| airport  |
| company  |
| ebookstore |
| information_schema |
| library_sys |
| movies   |
| mysql     |
| performance_schema |
| sfcs      |
| shopping  |
| sys       |
| test      |
| unify     |
+-----+
13 rows in set (0.02 sec)
```

Kết quả khi tạo thành công cơ sở dữ liệu ebookstore

4.2 Định nghĩa nhóm người dùng

Tiếp đó, bằng cách sử dụng lệnh:

```
CREATE USER '<user_name>'@'localhost' IDENTIFIED BY 'mypass';
```

ta có thể tạo ra 2 user có tên là staff và customer - tượng trưng cho 2 nhóm người dùng lần lượt là Nhân viên và Khách Hàng.

```
mysql> create user 'staff'@'localhost' identified by 'mypass';
Query OK, 0 rows affected (0.03 sec)

mysql> select host,user from mysql.user;
+-----+-----+
| host      | user                |
+-----+-----+
| %         | root                |
| localhost | mysql.infoschema    |
| localhost | mysql.session       |
| localhost | mysql.sys           |
| localhost | root                |
| localhost | staff               |
+-----+-----+
6 rows in set (0.00 sec)

mysql> create user 'customer'@'localhost' identified by 'mypass';
Query OK, 0 rows affected (0.02 sec)

mysql> select host,user from mysql.user;
+-----+-----+
| host      | user                |
+-----+-----+
| %         | root                |
| localhost | customer            |
| localhost | mysql.infoschema    |
| localhost | mysql.session       |
| localhost | mysql.sys           |
| localhost | root                |
| localhost | staff               |
+-----+-----+
7 rows in set (0.00 sec)
```

Kết quả khi tạo ra 2 user

Với 2 nhóm người dùng như trên, Nhân Viên là những người duy nhất có khả năng cập nhật và thay đổi các dữ liệu của Kho và việc Quản Lý Sách. Do đó, ta sẽ phân quyền cho staff có khả năng UPDATE bảng dữ liệu của thực thể *Kho* và *Quản Lý* bằng lệnh:

GRANT + <privilege list> + ON + <database objects> + TO <user list>

Trong đó:

- privilege list là danh sách những đặc quyền có thể phân chia cho người dùng. Ví dụ như: UPDATE, INSERT, DELETE,...
- database objects là những đối tượng, hay nói chính xác hơn là những bảng dữ liệu mà user được phân quyền có khả năng tác động lên.
- user list là danh sách những user được phân quyền

```
mysql> grant update on ebookstore.QuanLy to 'staff'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> grant update on ebookstore.Kho to 'staff'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Phân quyền UPDATE 2 bảng QuanLy và Kho cho staff bằng lệnh GRANT

Còn lại nhóm người dùng là Khách Hàng thì không được phép thay đổi hay thêm xóa sửa dữ liệu trong CSDL nên user customer sẽ không được phân quyền ở đây.

Tuy nhiên, trong trường hợp nếu muốn hủy đi những quyền đã phân phát, ta có thể sử dụng lệnh REVOKE thay cho GRANT như sau:

REVOKE + <privilege list> + ON + <database objects> + TO <user list>

5 Viết câu lệnh SQL

5.1 Ngôn ngữ định nghĩa dữ liệu - DDL

- DROP DATABASE IF EXISTS eBookStore;
- CREATE DATABASE eBookStore;
- USE eBookStore;

- create table NhanVien (
SID varchar(10),
Email varchar(20),
Ho varchar(20),
Ten varchar(10),
primary key (SID),
check (SID like 'S_____')
);
- create table Kho(
MaKho varchar(20),
primary key (MaKho)
);
- create table NhaXuatBan(
TenNhaXuatBan varchar(30),
primary key (TenNhaXuatBan)
);
- create table TacGia(
AID varchar(10),
Ho varchar(30),
Ten varchar(10),
Email varchar(30),
NamSinh int,
primary key (AID),
check (AID like 'A_____'),
check (NamSinh > 0 AND NamSinh < 9999)
);
- create table Sach(
ISBN varchar(10),
TenSach varchar(30),
GiaTien int not null,
primary key (ISBN),
check (GiaTien > 0)
);
- create table BanIn(
ISBN varchar(10),
primary key (ISBN),
foreign key (ISBN) references Sach(ISBN) on delete cascade on update cascade
);
- create table BanDienTu(
ISBN varchar(10),
Bcode varchar(20) unique not null,
primary key (ISBN),
foreign key (ISBN) references Sach(ISBN) on delete cascade on update cascade
);

- create table KhachHang(
 UID varchar(10),
 Ho varchar(20),
 Ten varchar(10),
 DiaChi varchar(30) not null,
 Phone varchar(11) not null,
 primary key (UID),
 check (length(Phone) >= 9 AND length(Phone) <= 11)
);
- create table TheTinDung(
 MaThe varchar(10),
 TenChuThe varchar(20),
 TenNganHang varchar(20),
 NgayHetHan date,
 primary key (MaThe)
);
- create table ThanhToan(
 HoaDon varchar(20),
 primary key (HoaDon)
);
- create table ChuyenKhoan(
 HoaDon varchar(20),
 primary key (HoaDon),
 foreign key (HoaDon) references ThanhToan(HoaDon) on delete cascade on update cascade
);
- create table ThanhToanThe(
 HoaDon varchar(20),
 MaThanhToan varchar(10) unique not null,
 primary key (HoaDon),
 foreign key (HoaDon) references ThanhToan(HoaDon) on delete cascade on update cascade
);
- create table GiaoDich(
 UID varchar(10),
 ISBN varchar(10),
 NgayGio datetime,
 primary key (UID,ISBN,NgayGio),
 foreign key (UID) references KhachHang(UID) on delete cascade on update cascade,
 foreign key (ISBN) references Sach(ISBN) on delete cascade on update cascade
);
- alter table Kho
 add SID varchar(10) not null,
 add foreign key(SID) references NhanVien(SID) on delete cascade on update cascade;

- alter table Sach
add TenNhaXuatBan varchar(30) not null,
add foreign key(TenNhaXuatBan) references NhaXuatBan(TenNhaXuatBan) on delete cascade on update cascade;
- alter table TheTinDung
add UID varchar(10) not null,
add foreign key(UID) references KhachHang(UID) on delete cascade on update cascade,
add HoaDon varchar(20),
add foreign key(HoaDon) references ThanhToanThe(HoaDon) on delete cascade on update cascade;
- alter table GiaoDich
add HoaDon varchar(20),
add foreign key(HoaDon) references ThanhToan(HoaDon) on delete cascade on update cascade;
- create table LuuTru(
MaKho varchar(20),
ISBN char(13),
SoLuongBanIn int,
primary key (MaKho,ISBN),
foreign key (Makho) references Kho(MaKho) on delete cascade on update cascade,
foreign key (ISBN) references BanIn(ISBN) on delete cascade on update cascade,
check (SoLuongBanIn >= 0)
);
- create table QuanLy(
MaKho varchar(20),
SID varchar(10),
ISBN char(13),
primary key (MaKho,SID,ISBN),
foreign key (MaKho) references Kho(MaKho) on delete cascade on update cascade,
foreign key (SID) references NhanVien(SID) on delete cascade on update cascade,
foreign key (ISBN) references BanIn(ISBN) on delete cascade on update cascade
);
- create table Viet(
AID varchar(10),
ISBN varchar(10),
primary key (AID, ISBN),
foreign key (AID) references TacGia(AID) on delete cascade on update cascade,
foreign key (ISBN) references Sach(ISBN) on delete cascade on update cascade
);
- create table PhanHoi(
UID varchar(10),
ISBN varchar(10),
primary key(UID, ISBN),
foreign key(UID) references KhachHang(UID) on delete cascade on update cascade,
foreign key(ISBN) references Sach(ISBN) on delete cascade on update cascade
);

- create table TuKhoa(
ISBN varchar(10),
ATuKhoa varchar(20),
primary key(ISBN, ATuKhoa),
foreign key(ISBN) references Sach(ISBN) on delete cascade on update cascade
);
- create table LinhVuc(
ISBN varchar(10),
ALinhVuc varchar(20),
primary key(ISBN, ALinhVuc),
foreign key(ISBN) references Sach(ISBN) on delete cascade on update cascade
);
- create table NamSanXuat(
ISBN varchar(10),
TenNhaXuatBan varchar(30),
ANamSanXuat year,
primary key(ISBN, TenNhaXuatBan, ANamSanxuat),
foreign key(ISBN) references Sach(ISBN) on delete cascade on update cascade,
foreign key(TenNhaXuatBan) references NhaXuatBan(TenNhaXuatBan) on delete cascade on update cascade,
check (ANamSanXuat > 0 AND ANamSanXuat < 1999)
);
- create table TinPhanHoi(
UID varchar(10),
ISBN varchar(10),
ThoiGian datetime,
VanBan varchar(500),
primary key(UID, ISBN, ThoiGian, VanBan),
foreign key(UID, ISBN) references PhanHoi(UID, ISBN) on delete cascade on update cascade
);

5.2 Ngôn ngữ thao tác dữ liệu - DML

1. Thêm dữ liệu bằng lệnh INSERT

EBookstore-Assignment1

Find: Q: -QuanLy

```

209
210 • insert into TacGia(AID, NamSinh) values('A123456789',1999);
211 • insert into NhanVien(SID) values('S123456789');
212 • insert into KhachHang values('1810198','to','hung','KTX','0905983795');
213 • insert into Sach(ISBN, GiaTien) values('CSDL',50000);
214 • insert into GiaoDich(UID, ISBN, NgayGio) values('1810198','CSDL','1990-01-10');
215 • insert into PhanHoi values('1810198','CSDL');
216 • insert into TinPhanHoi values('1810198','CSDL','1990-01-01','sounds interesting');
217 • insert into Kho values('Kho1','S123456789');
218 • insert into NhaXuatBan values('KimDong');
219 • insert into BanIn values('CSDL');
220 • insert into BanDienTu values('CSDL','XYZ');
221 • insert into LinhVuc values('CSDL','CNTT');
222 • insert into LinhVuc values('CSDL','KHMT');
223 • insert into NamSanXuat values('CSDL','KimDong',2019);
224 • insert into NamSanXuat values('CSDL','KimDong',2020);

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
237	17:42:00	insert into NamSanXuat values('CSDL','KimDong',2019)	1 row(s) affected	0.015 sec
238	17:42:00	insert into NamSanXuat values('CSDL','KimDong',2020)	1 row(s) affected	0.016 sec

2. Lấy tên khách hàng, mã ISBN của sách, ngày giờ giao dịch, ngày giờ phản hồi và nội dung phản hồi của mỗi giao dịch

```

232 • create view GiaoDichVaPhanHoi(UID, ISBN, NgayGioGD, NgayGioPhanHoi, NoiDungPhanHoi)
233 • as select UID, ISBN, NgayGio, ThoiGian, VanBan from GiaoDich natural join TinPhanHoi;
234
235 • select * from GiaoDichvaPhanHoi;

```

Result Grid

	UID	ISBN	NgayGioGD	NgayGioPhanHoi	NoiDungPhanHoi
▶	1810198	CSDL	1990-01-10 00:00:00	1990-01-01 00:00:00	sounds interesting
	1810198	CSDL	1990-01-10 00:00:00	1990-01-03 00:00:00	love it
	1810198	HD qua mon	1991-01-10 00:00:00	1991-12-10 00:00:00	uuuuuu emmm

3. Lấy mã ISBN và số lượng sách cho mỗi sách khách nhau, và tổng số tiền cho mỗi loại sách

```

244 • select ISBN, count(*) as SoLuong, sum(GiaTien) as ThanhTien from GiaoDich natural join Sach group by ISBN;
245
246

```

Result Grid

	ISBN	SoLuong	ThanhTien
▶	CSDL	2	100000
	HD qua mon	4	400000

4. Lấy tất cả mã sách và giá tiền tương ứng của mỗi sách trong kho

```

251 • select ISBN, GiaTien from LinhVuc natural join Sach
252 | where ALinhVuc = 'KHMT';
253

```

ISBN	GiaTien
SphLaDL	100000
CSDL	50000
HD qua mon	100000
VaTheLaHet	100000

5. Lấy tên nhà xuất bản và số lượng xuất bản được đặt

```

261 • select TenNhaXuatBan, count(*) from Sach group by TenNhaXuatBan having count(*) > 1
262
263
264
265

```

TenNhaXuatBan	count(*)
HCM	2
KimDong	2

6. Truy vấn mã ISBN và Giá tiền của cuốn sách có giá trị thấp nhất.

```

266 • select ISBN, GiaTien from Sach where GiaTien in (select min(GiaTien) as minGiaTien from Sach);

```

ISBN	GiaTien
CSDL	50000
NULL	NULL

7. Truy vấn mã ISBN và Tên nhà xuất bản của những cuốn sách được viết bởi tác giả có AID là "A123456789"

```

269 -- 6
270 • select ISBN, TenNhaXuatBan from Sach natural join Viet where AID = 'A123456789';
271
272

```

ISBN	TenNhaXuatBan
SphLaDL	HCM
CSDL	KimDong
VaTheLaHet	HCM

8. Cập nhật tên sách là "Cơ sở dữ liệu" cho cuốn sách có mã ISBN là "CSDL"

```

272 -- 7
273 • update Sach
274 set TenSach = 'Cơ sở dữ liệu'
275 where ISBN = 'CSDL';
276 • select * from Sach;
277

```

ISBN	TenSach	GiaTien	TenNhaXuatBan
5phLaDL	NULL	100000	HCM
CSDL	Cơ sở dữ liệu	50000	KimDong
HD qua mon	NULL	100000	KimDong
VaTheLaHet	NULL	100000	HCM
NULL	NULL	NULL	NULL

9. Cập nhật giá tiền tăng gấp đôi cho những cuốn sách có chung Nhà xuất bản với cuốn sách có mã ISBN là "CSDL" mà không phải là chính cuốn đó.

```

279 • update Sach
280 set GiaTien = GiaTien*2
281 where TenNhaXuatBan in (select distinct TenNhaXuatBan from NamSanXuat where ISBN = 'CSDL')
282 and ISBN != 'CSDL';
283 • select * from Sach;
284

```

ISBN	TenSach	GiaTien	TenNhaXuatBan
5phLaDL	NULL	100000	HCM
CSDL	Cơ sở dữ liệu	100000	KimDong
HD qua mon	NULL	200000	KimDong
VaTheLaHet	NULL	100000	HCM
NULL	NULL	NULL	NULL

10. Truy vấn toàn bộ thông tin của Tác giả có cụm "345" trong AID

```

286 • select * from TacGia
287 where AID like '%345%';

```

AID	Ho	Ten	Email	NamSinh
A123456789	NULL	NULL	NULL	1999
NULL	NULL	NULL	NULL	NULL

11. Xóa đi sách có mã ISBN là "VaTheLaHet"

```

289 -- 10
290 • delete from Sach
291 where ISBN = 'VaTheLaHet';
292 • select * from Sach;

```

ISBN	TenSach	GiaTien	TenNhaXuatBan
5phLaDL	NULL	100000	HCM
CSDL	Cơ sở dữ liệu	50000	KimDong
HD qua mon	NULL	200000	KimDong
NULL	NULL	NULL	NULL

6 Công nghệ phát triển ứng dụng cơ sở dữ liệu

6.1 NodeJS

6.1.1 NodeJs là gì

Sơ lược

- NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.
- NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.



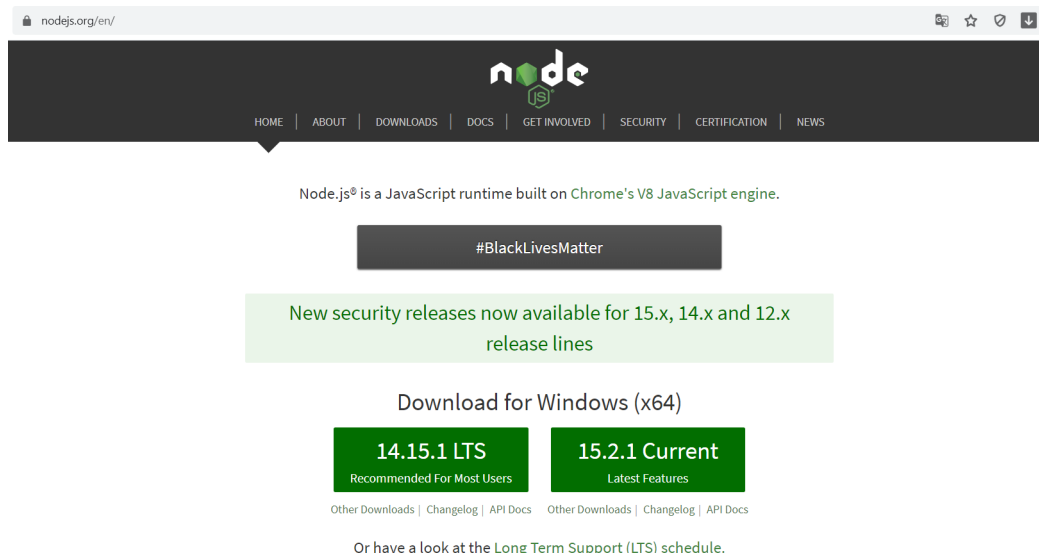
Cách hoạt động

- Ý tưởng chính của Node.js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng.
- Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả.
- Chính vì lẽ đó giải pháp mà Node.js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ rất nhiều kết nối đồng thời

6.1.2 Cài đặt

Cho Window

- Truy cập vào đường link <https://nodejs.org/>
- Chọn phiên bản phù hợp với máy tính vào tải về



Cho Linux

- Mở terminal lên
- Gõ command
 1. **sudo apt update**
 2. **sudo apt install nodejs**
 3. **sudo apt install npm**
 4. **nodejs -v**

To get this version, you can use the `apt` package manager. Refresh your local package index by typing:

```
$ sudo apt update
```

Install Node.js from the repositories:

```
$ sudo apt install nodejs
```

If the package in the repositories suits your needs, this is all you need to do to get set up with Node.js. In most cases, you'll also want to install `npm`, the Node.js package manager. You can do this by typing:

```
$ sudo apt install npm
```

This will allow you to install modules and packages to use with Node.js.

Because of a conflict with another package, the executable from the Ubuntu repositories is called `nodejs` instead of `node`. Keep this in mind as you are running software.

To check which version of Node.js you have installed after these initial steps, type:

```
$ nodejs -v
```

6.1.3 Ưu điểm & Nhược điểm

Ưu điểm

1. Ngôn ngữ cấp cao
2. Có nhiều thư viện hỗ trợ
3. Dễ dàng mở rộng và kết hợp với những ngôn ngữ khác

Nhược điểm

1. Không xử lý hiệu quả các ứng dụng tốn tài nguyên CPU như encoding video, convert file,...

6.1.4 Module

Các module chính sử dụng trong webapp

1. MySQL
2. Express
3. Express-handlebars

Module mysql

- Dùng để connected node với mysql
- Cung cấp interface để tương tác với database
- Ví dụ về việc connect và xử dụng các lệnh query

```
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "localhost",
  user: "yourusername",
  password: "yourpassword",
  database: "mydb"
});

con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT * FROM customers", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
  });
});
```

Module Express

- Là framework dùng để làm webserver
- Ví dụ về sử dụng express

1. Code

```
const express = require('express') ( 4.17.1 )
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`)
})
```

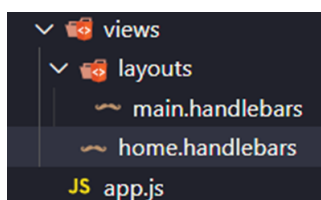
2. Kết quả



Module Express-handlebars

- Handlebars là một thư viện javascript rất mạnh mẽ hỗ trợ người dùng có thể binding data vào một template để hiển thị ra website.
- Handlebar hỗ trợ thêm các tính năng cho Express về việc chỉnh sửa các giao diện html, css, js...
- Ví dụ về việc sử dụng handlebars

1. Đầu tiên xây dựng directory structure như hình



2. Sau đó lần lượt thêm vào cái file app.js, main.handlebars, home.handlebars và nhận được kết quả

```
var express = require('express');
var exphbs = require('express-handlebars');

var app = express();

var hbs = exphbs.create({
  // Specify helpers which are only registered on this instance.
  helpers: {
    foo: function () { return 'FOO!'; },
    bar: function () { return 'BAR!'; }
  }
});

app.engine('handlebars', hbs.engine);
app.set('view engine', 'handlebars');

app.get('/', function (req, res, next) {
  res.render('home', {
    showTitle: true,

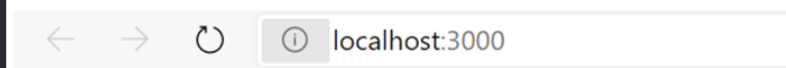
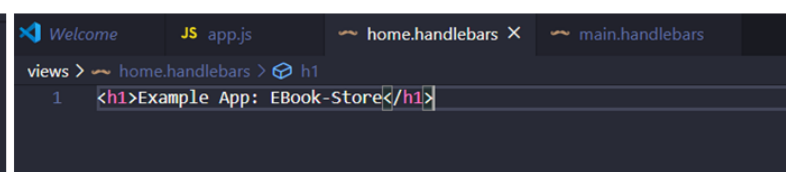
    // Override `foo` helper only for this rendering.
    helpers: {
      foo: function () { return 'foo.'; }
    }
  });
});

app.listen(3000);
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Example App</title>
</head>
<body>

  {{{body}}}

</body>
</html>
```



Example App: EBook-Store

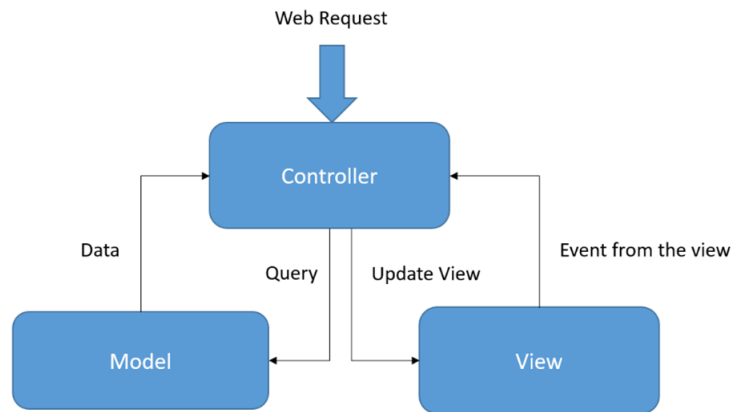
6.2 Mô hình MVC

Định nghĩa

- MVC là viết tắt của Model - View - Controller
- Mô hình MVC là một mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó đóng vai trò quan trọng trong quá trình xây dựng - phát triển - vận hành và bảo trì một hệ thống hay một ứng dụng

phần mềm.

- Hay nói cách khác, nó là một mô hình 3 lớp Model-View-Controller, mỗi lớp có một nhiệm vụ riêng biệt và độc lập với các lớp khác



Các thành phần trong MVC

Module

- Là nơi chứa tất cả nghiệp vụ tương tác với cơ sở dữ liệu hay là hệ quản trị cơ sở dữ liệu. Nó bao gồm các class và function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm và xóa dữ liệu ...

View

- Đảm nhận nhiệm vụ hiển thị dữ liệu, giúp người dùng tương tác với hệ thống. là nơi chứa tất cả các file html, form, image ...

Controller

- Là nơi tiếp nhận và điều hướng những yêu cầu từ phía người dùng bao gồm các class hoặc function xử lý yêu cầu giúp lấy dữ liệu từ Model và trả kết quả cho người dùng thông qua View.

7 Phân chia công việc

Tô Duy Hưng (50%)

- Ánh xạ lược đồ quan hệ
- Mở rộng yêu cầu dữ liệu
- Định nghĩa lược đồ cơ sở dữ liệu quan hệ trên DBMS
- Định nghĩa và phân quyền các nhóm người dùng trên DBMS
- Viết các câu lệnh query

Trương Thanh Lộc (50%)

- Liệt kê các kiểu thực thể mạnh và mối liên kết
- Liệt kê các kiểu thực thể yếu và mối liên kết xác định
- Xác định thuộc tính và mô tả về các thuộc tính
- Xác định ràng buộc
- Vẽ lược đồ thực thể-mối liên kết (EERD)

Trong quá trình làm việc thì cả nhóm có trao đổi với nhau. Và cả nhóm đều tham gia vào phần viết báo cáo.