

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



LÊ VĂN TRUNG

**NÂNG CẤP TÍNH NĂNG TÌM KIẾM CỦA PHẦN MỀM THƯƠNG MẠI
ĐIỆN TỬ MAGENTO SỬ DỤNG SPHINX SEARCH**

LUẬN VĂN THẠC SĨ NGÀNH CÔNG NGHỆ THÔNG TIN

Hà Nội - 2016

LỜI CAM ĐOAN

Tôi xin cam đoan rằng đây là công trình nghiên cứu của cá nhân tôi dưới sự hướng dẫn giúp đỡ của TS. Trần Trúc Mai. Các kết quả được viết chung với các tác giả khác đều được sự đồng ý của tác giả trước khi đưa vào luận văn. Trong toàn bộ nội dung nghiên cứu của luận văn, các vấn đề được trình bày đều là những tìm hiểu và nghiên cứu của chính cá nhân tôi hoặc là được trích dẫn từ các nguồn tài liệu có ghi tham khảo rõ ràng, hợp pháp.

Trong luận văn, tôi có tham khảo đến một số tài liệu của một số tác giả được liệt kê tại mục tài liệu tham khảo.

Hà nội, tháng 5 năm 2016

Tác giả luận văn

Lê Văn Trung

LỜI CẢM ƠN

Để hoàn thành tốt luận văn này, đầu tiên Tôi xin bày tỏ lòng biết ơn chân thành và sâu sắc đến Tiến sĩ Trần Trúc Mai, người đã tận tình và trực tiếp hướng dẫn Tôi trong suốt quá trình triển khai và nghiên cứu đề tài, tạo điều kiện để Tôi hoàn thành luận văn này.

Thứ hai, Tôi xin bày tỏ lòng biết ơn chân thành tới toàn thể các thầy cô giáo trong khoa Công nghệ thông tin, trường Đại học Công nghệ Hà Nội, Đại học Quốc gia Hà Nội đã dạy bảo tận tình Tôi trong suốt quá trình Tôi học tập tại khoa.

Cuối cùng Tôi xin chân thành cảm ơn tới gia đình, bạn bè, đồng nghiệp đã luôn bên em cổ vũ, động viên, giúp đỡ Tôi trong suốt quá trình học tập và thực hiện luận văn.

Mặc dù đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Tôi rất mong được sự góp ý chân thành của thầy cô và các bạn để Tôi hoàn thiện luận văn của mình.

Xin chân thành cảm ơn!

Hà Nội, tháng 5 năm 2016

Học viên

Lê Văn Trung

MỤC LỤC

LỜI CẢM ƠN.....	3
Chương 1. TỔNG QUAN VỀ MAGENTO VÀ MÔI TRƯỜNG CÀI ĐẶT	3
1.1. Tổng quan về Magento.....	3
1.1.1. Giới thiệu chung	3
1.1.2. Lịch sử phát triển	4
1.1.3. Tính năng của hệ thống mã nguồn mở Magento	5
1.2. Ưu điểm và nhược điểm của hệ thống mã nguồn mở Magento.....	8
1.3. Môi trường cài đặt.....	10
1.3.1. Virtual Private Server của DigitalOcean.	10
1.3.2. LEMP Server	10
1.4. Kết luận chương 1	11
Chương 2. TÌM KIẾM TRÊN MAGENTO	12
2.1. Giới thiệu.....	12
2.2. Thuật toán nền tảng của các loại tìm kiếm trong Magento	14
2.2.1. Tìm kiếm like.....	14
2.2.2. Tìm kiếm toàn văn bản	15
2.2.3. Tìm kiếm kết hợp.....	19
2.3. Giải pháp nâng cấp tính năng tìm kiếm trên Magento	20
2.3.1. Phương pháp tiếp cận.....	20
2.3.2. Các giải pháp mở rộng tính năng tìm kiếm cho Magento	21
2.3.3. Sử dụng Sphinx để nâng cấp tính năng tìm kiếm	25
2.4. Kết luận chương 2	26
Chương 3. SỬ DỤNG SPHINX SEARCH NÂNG CẤP TÍNH NĂNG TÌM KIẾM TRÊN MAGENTO.....	27
3.1. Một số vấn đề về tìm kiếm toàn văn bản	27
3.1.1. Khái niệm tìm kiếm toàn văn bản.....	27
3.1.2. Khái niệm chỉ số và thuật toán đánh chỉ số	27
3.2. Tổng quan về Sphinx Search.....	31

3.2.1. Định nghĩa:	31
3.2.2. Tính năng của Sphinx Search.	31
3.2.3. Lịch sử phát triển	32
3.2.4. Cài đặt sphinx trong môi trường Linux	32
3.2.5. Cách sử dụng Sphinx Search	33
3.3. Cơ chế hoạt động của Sphinx Search.....	34
3.3.1. Các thành phần trong sphinx	34
3.3.2. Tiến trình làm việc.....	34
3.3.3. Lập chỉ số.....	36
3.3.4. Tìm kiếm.....	42
3.3.5. Matching Modes (Chế độ phù hợp).....	44
3.3.6. Xếp hạng kết quả tìm kiếm.....	47
3.3.7. Các Rankers được xây dựng sẵn.....	48
3.3.8. Sắp xếp kết quả tìm kiếm.....	49
3.3.9. Nhóm kết quả tìm kiếm	51
3.4. Kết luận chương 3	51
Chương 4. TRIỂN KHAI NÂNG CẤP TÍNH NĂNG TÌM KIẾM CHO MAGENTO BẰNG CÁCH SỬ DỤNG SPHINX	52
4.1. Yêu cầu.....	52
4.2. Phân tích thiết kế.....	52
4.2.1. Chức năng lập chỉ số từ bảng dữ liệu của Magento.....	52
4.2.2. Chức năng lựa chọn các chế độ tìm kiếm.....	52
4.3. Triển khai xây dựng	53
4.3.1. Tạo bảng chứa dữ liệu các trường sẽ được lập chỉ mục	53
4.3.2. Cài đặt Sphinx trên Server và cấu hình	55
4.3.3. Lập chỉ số dữ liệu.....	57
4.3.4. Lựa chọn các chế độ tìm kiếm.....	57
4.4. Kết luận chương 4:	59
Chương 5. ĐÁNH GIÁ KẾT QUẢ.....	60

5.1. Kết quả xây dựng	60
5.2. Đánh giá về kết quả xây dựng	61
5.3. Kết luận chương.	66
KẾT LUẬN	67
TÀI LIỆU THAM KHẢO	68

DANH SÁCH CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

STT	Từ viết tắt	Từ hoặc cụm từ
1	FTS	Full Text Search (Tìm kiếm toàn văn bản)
2	API	Application Program Interface
3	RT	Realtime (Chỉ số thời gian thực)
4	SPHINX	SQL Phrase Index
5	MVA	Multi-Valued attributes
6	VPS	Virtual Private Server

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 1.1. Thị phần của Magento[16].....	4
Hình 2.1. Chọn chức năng tìm kiếm like.....	12
Hình 2.2. Chọn chức năng tìm kiếm Fulltext	13
Hình 2.3. Chọn chức năng tìm kiếm kết hợp.....	13
Hình 2.4. Kết quả tìm kiếm từ khóa “cotton shirt” ở chế độ like.....	15
Hình 2.5. Kết quả tìm kiếm từ khóa “cotton shirt” ở chế độ fulltext.	18
Hình 3.1. Sơ đồ làm việc của Sphinx Search.....	35
Hình 3.2. Dữ liệu trong bảng post	38
Hình 3.3. Kết quả tạo chỉ số	39
Hình 3.4. Kết quả tìm kiếm từ khóa ‘php’	39
Hình 3.5. Kết quả xây dựng lại chỉ số	40
Hình 3.6. Kết quả tìm kiếm từ khóa ‘php’ được viết bởi tác giả Aditya Mooley	40
Hình 4.1. Kết quả xây dựng bảng sphinx_catalogSearch_fulltext	54
Hình 4.2. Lập chỉ số trong Sphinx	57
Hình 5.1. Giao diện quản trị	60
Hình 5.2. Giao diện tìm kiếm	60
Hình 5.3. Giao diện trang kết quả tìm kiếm	61
Hình 5.4. Kết quả search like.....	63
Hình 5.5. Kết quả tìm kiếm fulltext search trong mysql	64
Hình 5.6. Kết quả lựa chọn chế độ kết hợp	65
Hình 5.7. Kết quả tìm kiếm trên Sphinx Search.....	66

DANH SÁCH BẢNG BIỂU

Bảng 2.1. Bảng một danh sách một số stopwords.....	16
Bảng 3.1. Danh sách các document cần tìm kiếm	28
Bảng 3.2. Danh sách term.....	28

Bảng 3.3. Kết quả tìm kiếm dựa vào term	29
Bảng 3.4. Bảng kết quả chuyển term sang định dạng tiêu chuẩn.	30
Bảng 3.5. Bảng post trong cơ sở dữ liệu myblog	37
Bảng 3.6. Bảng so sánh thời gian index trong các trường hợp “stop word”	41
Bảng 3.7. Các loại truy vấn Boolean	45
Bảng 3.8. Các truy vấn mở rộng	47
Bảng 3.9. Các loại sắp xếp trong Sphinx	50
Bảng 3.10. Các phương thức nhóm kết quả tìm kiếm	51
Bảng 4.1. Các trường dữ liệu trong bảng sphinx_catalogSearch_fulltext.....	53
Bảng 5.1. Kết quả tìm kiếm từ khóa	62

MỞ ĐẦU

1. ĐẶT VẤN ĐỀ

Ngày nay thương mại điện tử phát triển vô cùng mạnh mẽ bởi tốc độ sử dụng internet cùng với ngày càng nhiều các công nghệ hiện đại ra đời. Mọi người ngày càng ưa thích giao dịch dưới hình thức này bởi những thuận lợi của nó mang lại như tiện lợi, nhanh chóng, tiết kiệm chi phí, dễ dàng để tìm kiếm các mặt hàng cần mua, bán. Do đó yêu cầu đặt ra là xây dựng các gian hàng trực tuyến có khả năng hỗ trợ tìm kiếm hiệu quả và nâng cao để có thể thực hiện mua bán trực tuyến dễ dàng. Magento là phần mềm thương mại điện tử mã nguồn mở hoàn toàn miễn phí và cho phép người dùng thiết kế, cài đặt và chạy các gian hàng thương mại điện tử của riêng mình với sự linh hoạt cao. Tuy nhiên trong Magento chức năng tìm kiếm mặc định có nhiều hạn chế vì vậy ta cần nâng cấp tính năng tìm kiếm này để đáp ứng các yêu cầu ngày càng cao của khách hàng.

2. MỤC ĐÍCH CỦA LUẬN VĂN

Đề xuất sử dụng công cụ tìm kiếm toàn văn bản Sphinx để nâng cấp tính năng tìm kiếm mặc định trong Magento.

Xây dựng một chương trình mở rộng tích hợp vào mã nguồn Magento dựa trên công cụ tìm kiếm toàn văn bản Sphinx Search để nâng cấp tính năng tìm kiếm mặc định trên Magento. Đặt tên chương trình này là Extension Sphinx Search.

Đánh giá kết quả tìm kiếm trước và sau khi sử dụng Extension Sphinx Search.

3. BỐ CỤC CỦA LUẬN VĂN

Ngoài phần mở đầu, kết luận, danh mục, các đề tài có liên quan, luận văn gồm 05 chương được tổ chức như sau:

Chương 1: *Tổng quan về Magento và môi trường cài đặt*: Giới thiệu chung về hệ thống mã nguồn mở thương mại điện tử Magento và môi trường cài đặt Magento trong luận văn.

Chương 2: Tìm kiếm trên Magento: Trình bày về các công cụ tìm kiếm có sẵn trong Magento, những hạn chế của các loại tìm kiếm mặc định này và đặt ra bài toán cần nâng cấp tính năng tìm kiếm mặc định của hệ thống Magento.

Chương 3: Sử dụng Sphinx Search nâng cấp tính năng tìm kiếm trong Magento: Trình bày các đặc điểm của công cụ tìm kiếm sử dụng Sphinx đồng thời giải thích lý do sử dụng công cụ này.

Chương 4: Triển khai nâng cấp tính năng tìm kiếm cho Magento bằng cách sử dụng Sphinx Search : Các bước triển khai xây dựng một chương trình mở rộng tính năng tìm kiếm tích hợp vào Magento.

Chương 5: Đánh giá kết quả: Tổng kết đánh giá kết quả việc nâng cấp tính năng tìm kiếm trên hệ thống mã nguồn mở Magento.

Nội dung chi tiết từng chương sẽ được trình bày ở các phần sau:

Chương 1. TỔNG QUAN VỀ MAGENTO VÀ MÔI TRƯỜNG CÀI ĐẶT

1.1. Tổng quan về Magento

1.1.1. Giới thiệu chung

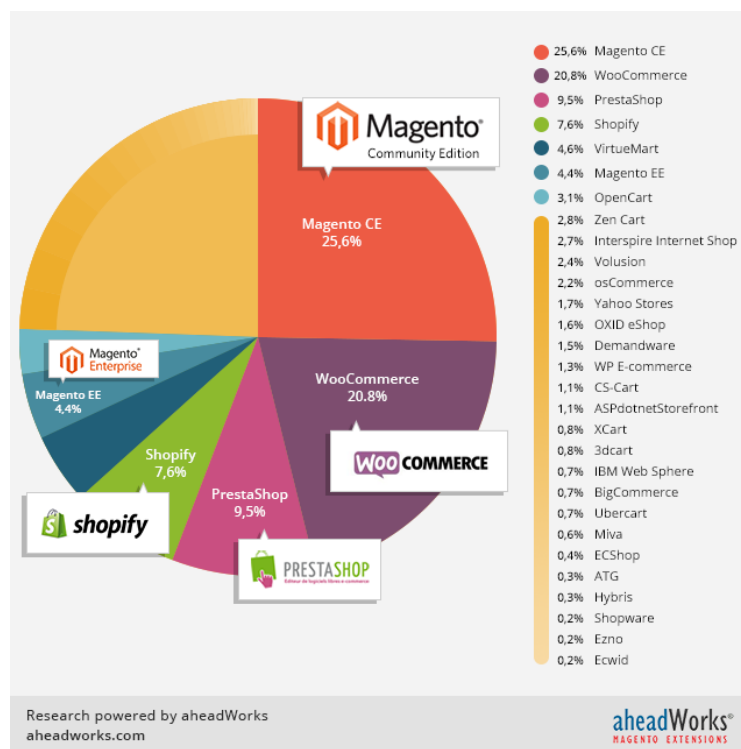
Thương mại điện tử hay còn gọi là E-commerce là sự mua bán sản phẩm hay dịch vụ trên các hệ thống điện tử như Internet và các mạng máy tính. Thương mại điện tử dựa trên một số công nghệ như chuyển tiền điện tử, quản lý chuỗi dây chuyền cung ứng, tiếp thị internet, quá trình giao dịch trực tuyến... Ngày nay, cùng với sự phát triển mạnh mẽ của internet và các công nghệ hiện đại khác. Con người ngày càng ưa thích giao dịch dưới hình thức này bởi những thuận lợi của nó mang lại. Để đáp ứng các yêu cầu này ngày càng có nhiều các giải pháp thương mại điện tử ra đời, trong đó nổi bật nhất là phần mềm thương mại điện tử Magento.

Magento là phần mềm thương mại điện tử mã nguồn mở được sử dụng bởi các thương hiệu hàng đầu trên thế giới. Magento được cho là hệ thống thương mại điện tử linh hoạt và hoàn chỉnh nhất hiện nay với nhiều tính năng tiên tiến để thiết lập các gian hàng ảo, độc đáo, an toàn, thân thiện với các công cụ tìm kiếm, tạo điều kiện thuận lợi cho việc lập chỉ số trong công cụ tìm kiếm Google, giảm chi phí đầu tư vào quảng cáo. Magento cho phép người dùng tạo và quản lý các gian hàng trực tuyến một cách dễ dàng, nhanh chóng.

Hiện nay, Không chỉ có Magento là giải pháp thương mại điện tử mã nguồn mở duy nhất trên thị trường, có những phần mềm mã nguồn mở khác như:

- Opencart.
- PrestaShop.
- osCommerce (more or less obsolete).
- Joomla with VirtureMart.
- Wordpress với e-commerce plugins like WooCommerce.
- Zen Cart.

Theo nghiên cứu của AheadWorks[16] tiến hành vào tháng 10 năm 2014, thị phần của Magento trong số 30 nền tảng thương mại điện tử phổ biến nhất chiếm 25,6 %, đứng đầu thế giới.



Hình 1.1. Thị phần của Magento [16]

1.1.2. Lịch sử phát triển

Bắt đầu từ năm 2001. Roy Rubin và Yoav Kutner thành lập công ty Varien. Họ bắt đầu triển khai nhiều hệ thống thương mại điện tử, đặc biệt là sử dụng osCommerce. Nhưng Họ không hài lòng về việc sử dụng giải pháp đó, vì sự thiếu linh hoạt và ổn định làm cho họ không thực hiện được các dự án của mình một cách tốt nhất. Vì vậy, Họ đã bắt đầu quyết định xây dựng một hệ thống thương mại điện tử riêng trong năm 2007 và đặt tên nó là Magento. Phiên bản dùng thử đầu tiên được phát hành vào tháng 8/2007 và đến 3/2008 bản chính thức version 1.0 được phát hành. Từ mốc thời gian đó các bản cập nhật sau được phát hành một cách nhanh chóng. Trong một vài năm đầu mỗi năm các phiên bản cập nhật được phát hành một hoặc hai lần. Phiên bản hiện tại là 2.0 phát hành tháng 11/2015. [15]

Hiện nay công ty Magento Inc cung cấp ba phiên bản phần mềm:

- MAGENTO ENTERPRISE EDITION: Đây là phiên bản trả phí, có hiệu suất cao đồng thời là giải pháp thương mại điện tử phù hợp cho các doanh nghiệp lớn.

- MAGENTO COMMUNITY EDITION: Đây là giải pháp mã nguồn mở hoàn toàn miễn phí.

- MAGENTO GO: Là giải pháp hosted, tương tự các nền tảng phổ biến. Magento phù hợp với các nhà bán lẻ và cho phép họ nâng cấp.

Trong khuôn khổ đề tài này Tôi chỉ nghiên cứu phiên bản *Magento Community Edition*.

1.1.3. Tính năng của hệ thống mã nguồn mở Magento

Magento là giải pháp nền tảng thương mại điện tử có tính năng phong phú, cung cấp các công cụ điều khiển các gian hàng trực tuyến linh hoạt. Magento tối ưu hóa công cụ tìm kiếm, quản lý danh mục sản phẩm và các khả năng marketing mạnh mẽ để tạo ra các website bán hàng tốt nhất phù hợp với yêu cầu của khách hàng.

Giao diện admin cho phép bạn điều chỉnh trang của bạn phù hợp với nhu cầu kinh doanh.

Các tính năng chính :

Các công cụ liên quan Marketing, Promotions và Conversion:

- Các sản phẩm liên quan, up-sells và cross-sells.
- Danh mục các sản phẩm khuyến mại.
- Phiếu giảm giá linh hoạt.
- Tạo ra một tập hợp các mã giảm giá và xuất danh sách offline tới email, newsletters... Để dễ dàng quản lý, giám sát và sử dụng.
- Tạo ra nhiều mức độ giảm giá khác nhau cho từng sản phẩm.
- MAP.
- Nhóm danh sách khách hàng.
- Liên quan đến hiển thị và so sánh các sản phẩm.
- Công cụ quảng cáo các sản phẩm mới.
- Công cụ quản lý giỏ hàng.
- Lựa chọn miễn phí vận chuyển.
- Thăm dò ý kiến khách hàng.
- Quản lý bản tin.
- Gửi thông tin sản phẩm tới bạn bè.
- Gửi thông tin sản phẩm qua danh sách email.

Tối ưu hóa công cụ tìm kiếm:

- Thân thiện với các máy tìm kiếm.
- Hỗ trợ tính năng quản lý Google Site Map.
- Công cụ tìm kiếm thân thiện với URL's.
- Rewrite URL để kiểm soát đầy đủ URL's.
- Sử dụng thẻ meta để đưa ra thông tin đầy đủ các thông tin về sản phẩm, danh mục và nội dung của trang.
- Tự động sinh site map cho các site.
- Tự động sinh Search term phổ biến.

Quản lý site:

- Có khả năng kiểm soát nhiều website và gian hàng từ administration Pannel với khả năng chia sẻ nhiều hoặc ít thông tin nếu cần thiết.
- Quy định quyền quản trị hệ thống.
- Tùy biến thiết kế 100% bằng cách sử dụng templates.
- Hỗ trợ nhiều ngôn ngữ và tiền tệ. Dịch vụ API cho phép dễ dàng kết hợp giữa Magento và các ứng dụng của bên thứ ba.
- Cung cấp chức năng Import và Export các thông tin về khách hàng.
- Hệ thống quản trị nội dung cho các trang.
- Hệ thống quản lý thuế giá trị gia tăng, phân loại sản phẩm cho mỗi nhóm khách hàng.
- Chức năng CAPTCHA để ngăn chặn các phần mềm tự động đăng nhập giả lập.
- Quản lý danh mục.
- Tiện ích giúp thay đổi kích thước, màu sắc...sản phẩm, và đóng gói, phân loại sản phẩm.
- Tạo giá khác nhau cho các nhóm khách hàng ví dụ như giá bán lẻ, giá buôn.
- Tạo các sản phẩm ảo.
- Không giới hạn các thuộc tính của sản phẩm.
- Hỗ trợ tập hợp nhiều thuộc tính để giúp tạo các danh mục sản phẩm khác nhau.
- Hỗ trợ quản lý hàng tồn kho.
- Cập nhật hàng loạt các sản phẩm trong bảng quản trị.
- Tự động thay đổi kích thước ảnh sản phẩm.

- Phân loại các sản phẩm cho từng khách hàng khác nhau (giá cả, thương hiệu...vv).

- Hỗ trợ cảnh báo hàng tồn kho.

Chức năng thanh toán và giao hàng:

- Bao gồm một trang để thanh toán.

- Hỗ trợ bảo mật SSL cho các đơn đặt hàng.

- Lưu giỏ hàng của bạn.

- Tích hợp nhiều cổng PayPal.

- Tích hợp Authorize.net.

- Cung cấp hóa đơn sau mỗi lần thanh toán.

- Vận chuyển tích hợp với in nhĩn- xem, chỉnh sửa, in nhĩn cho tất cả các nhĩn hàng lớn.

Quản lý đặt hàng:

- Xem chỉnh sửa và tạo các đơn đặt hàng từ bảng quản trị.

- In hóa đơn, phiếu đóng gói và nhĩn vận chuyển.

- Các dịch vụ giúp đỡ khách mua hàng.

- Tạo lại các đơn hàng cho khách hàng từ bảng quản trị.

- Thông báo email của các đơn đặt hàng.

Quản lý tài khoản khách hàng:

- Bảng điều khiển tài khoản.

- Danh bạ khách hàng không giới hạn.

- Danh sách mong muốn với khả năng thêm bình luận.

- Lịch sử đơn đặt hàng.

- Đặt hàng lại từ tài khoản.

- Danh sách các mặt hàng đã đặt gần đây.

- Gửi email cho danh sách khách hàng mong muốn.

Dịch vụ chăm sóc khách hàng:

- Hỗ trợ chức năng liên hệ với shop từ form.

- Tạo và sửa các đơn đặt hàng từ bảng quản trị.

- Lịch sử đơn đặt hàng và cập nhật trạng thái.

- Theo dõi đơn đặt hàng từ tài khoản.

- Reset Password từ front-end và admin panel.

Hỗ trợ quốc tế:

- Hỗ trợ cho nhiều khu vực, nhiều loại tiền tệ và thuế suất.
- Hỗ trợ cho các kí tự có dấu.

Phân tích và báo cáo:

- Tích hợp Google Analytics.
- Bảng điều khiển báo cáo trong Admin.
- Báo cáo bán hàng.
- Báo cáo thuế.
- Báo cáo sản phẩm xem nhiều nhất.
- Báo cáo sản phẩm mua nhiều nhất.
- Báo cáo sử dụng phiếu giảm giá.
- Tổng hợp hóa đơn.

Tích hợp trên Mobile:

- Tích hợp sẵn HTML-5: Giúp nhanh chóng và dễ dàng tạo ra các store tối ưu hóa cho các thiết bị di động. Hỗ trợ thiết bị iPhone, Android và trình duyệt mobile opera.

- Hỗ trợ thiết kế Responsive: Nhanh chóng tạo ra một trang web tối ưu hóa cho bất kì thiết bị nào. [14]

1.2. Ưu điểm và nhược điểm của hệ thống mã nguồn mở Magento.

Magento là hệ thống quản trị nội dung dựa trên nền PHP và MySQL, Tương thích với Linux. Một số ưu điểm và nhược điểm của hệ thống như sau:

Ưu điểm:

- Mã nguồn mở: Magento là nền tảng thương mại điện tử mã nguồn mở. Do đó người sử dụng có thể toàn quyền tùy biến tính năng của Magento bằng cách tạo mới hay cài thêm các phần mở rộng từ bên ngoài, và đặc biệt là hoàn toàn miễn phí khi sử dụng.

- Thân thiện với người dùng: Giao diện Back-End và Front-end trong Magento rất trực quan dễ hiểu và phù hợp với mọi người dùng phổ thông.

- Khách hàng mua sắm trực tuyến có thể dễ dàng sử dụng bởi các sản phẩm được thiết kế hiển thị một cách khoa học, hỗ trợ khả năng phóng to thu nhỏ hình

ảnh sản phẩm, các thông tin về sản phẩm được trình bày một cách chi tiết giúp người sử dụng tìm hiểu được các thông tin về sản phẩm một cách rõ ràng.

- Giao diện quản trị website cũng được thiết kế một cách rất trực quan, cho phép người quản trị có thể kiểm soát hoàn toàn các chức năng của Magento.

- Nhiều tính năng phong phú.

- Thân thiện với các công cụ tìm kiếm (SEO): Ngày nay việc sử dụng các công cụ tìm kiếm như google hay bring...để tìm kiếm thông tin là nhu cầu thiết yếu. Website Magento hỗ trợ rất tốt việc (SEO), đáp ứng các tiêu chuẩn tối ưu của SEO.

- Hỗ trợ nhiều phương thức thanh toán.

- Thân thiện với các thiết bị di động: như smartphones, ipad, Notebook.

- Hệ thống ổn định và bảo mật cao.

- Khả năng mở rộng: Magento có thể sử dụng cho các doanh nghiệp nhỏ đến lớn. Thường các doanh nghiệp nhỏ bắt đầu với việc sử dụng các giải pháp phần mềm nhỏ, nếu công ty phát triển lên sẽ đòi hỏi cần đổi sang một hệ thống mới. Nếu sử dụng các giải pháp phần mềm khác thì chúng ta có thể xây dựng lại từ đầu. Nhưng đối với Magento doanh nghiệp có thể giữ nguyên hệ thống, vì Magento có khả năng cài đặt thêm các ứng dụng mở rộng một cách dễ dàng để đáp ứng các yêu cầu của doanh nghiệp.

- Có nhiều nhà phát triển: Hiện nay có hàng ngàn các nhà phát triển trên toàn thế giới.

Nhược điểm:

- Bên cạnh những ưu điểm đã nêu ở trên Magento có một số các nhược điểm chính như sau:

- Chậm chạp: Magento là một hệ thống mạnh mẽ và phức tạp. Hiện nay một cài đặt Magento có hoảng 30000 file và hơn 1,2 triệu dòng code. nên chúng yêu cầu server mạnh mẽ và không gian đĩa lưu trữ mã nguồn lớn thường chỉ có các VPS, Server chuyên dụng hoặc Cloud Server mới đáp ứng được yêu cầu.

- Phức tạp đối với người dùng mới: Magento được viết dựa trên nền tảng PHP Zend Framework, ngay cả một lập trình viên giàu kinh nghiệm cũng cần có thời gian để tìm hiểu Magento.

1.3. Môi trường cài đặt

Website thương mại điện tử Magento trong đề tài nghiên cứu sử dụng phiên bản Magento Community Edition 1.9.1.0 được cài trên LEMP Server (L-Linux, E-Nginx, M- Mysql, P-php) trong VPS(Virtual Private Server) của DigitalOcean.

1.3.1. Virtual Private Server của DigitalOcean.

VPS - Virtual Private Server là phương pháp phân chia máy chủ vật lý thành nhiều máy chủ ảo. Mỗi máy chủ là một hệ thống hoàn toàn riêng biệt, có hệ điều hành riêng, có toàn quyền quản lý root và có thể reset lại hệ thống bất kỳ lúc nào. VPS hoạt động hoàn toàn như một server riêng nên sở hữu một phần CPU, dung lượng Ram, dung lượng ổ cứng, địa chỉ ip và hệ điều hành riêng.

DigitalOcean là một nhà cung cấp dịch vụ máy chủ ảo (VPS- virtual Private Server) của Mỹ có trụ sở tại thành phố New York. DigitalOcean được thành lập vào năm 2011 bởi Ben Uretsky. DigitalOcean có mức giá rẻ và linh hoạt với những gói cung cấp dịch vụ đa dạng. Từ mức phí rẻ nhất 5USD/ tháng với cấu hình 512 RAM, 1CPU, 20GB ổ cứng SSD và 1TB băng thông đủ để chạy hầu hết những dịch vụ không đòi hỏi hiệu suất quá cao. Đến những gói có cấu hình cao như 64GB RAM, 20 CPU, 640GB ổ cứng SSD và 9TB băng thông. Thời gian khởi tạo VPS trên DigitalOcean rất nhanh, chỉ với 55 giây người dùng đã có 1VPS để chạy các dịch vụ. Ngoài ra, rất dễ dàng để quản lý hệ điều hành với DigitalOcean có nhiều trung tâm dữ liệu trên thế giới.

Website trong nghiên cứu được cài đặt trên Virtual Private Server của DigitalOcean với địa chỉ IP là 45.114.117.18, với cấu hình phần cứng như sau: 1GB RAM, 30GB ổ cứng SSD, CPU 1 core và 2TB transfer. Trung tâm dữ liệu đặt tại Singapore.

1.3.2. LEMP Server

LEMP server là một server chạy Linux (có thể là Ubuntu, CentOS...). Máy chủ web Nginx, cơ sở dữ liệu MySQL, Ngôn ngữ lập trình PHP (hoặc Perl/Python).

Trong đó, phần rất quan trọng trong LEMP server đó là Nginx là một máy chủ web và reverse proxy miễn phí, mã nguồn mở, hiệu năng cao, sử dụng phổ

biến giao thức HTTP, HTTPS, SMTP, POP3 và IMAP, cũng như dùng làm cân bằng tải(load balancer). Rất nhiều nhà quản trị website lựa chọn Nginx làm máy chủ web vì tốc độ của nó. Trong một kết quả đánh giá của Astarion cho Magento, máy chủ website Nginx nhanh hơn 20 lần so với máy chủ website Apache. Nginx được biết đến bởi sự ổn định cao, nhiều tính năng, cấu hình đơn giản và tiết kiệm tài nguyên.

Website thương mại điện tử mã nguồn mở Magento trong nghiên cứu được cài đặt trên LEMP server với hệ điều hành CentOS 6.5 64 bit, nginx phiên bản 1.0.15, MySQL phiên bản 14.14 và php phiên bản 5.6.16.

1.4. Kết luận chương 1

Chương này tập trung giới thiệu 2 vấn đề chính. Vấn đề đầu tiên, giới thiệu tổng quan về Magento và những ưu điểm của mã nguồn mở Magento. Vấn đề tiếp theo, trình bày về môi trường cài đặt website Magento áp dụng trong luận văn.

Trong chương 2 luận văn sẽ trình bày vấn đề tìm kiếm trên Magento, Nhược điểm của các phương pháp tìm kiếm mặc định trên Magento từ đó đưa ra giải pháp sử dụng công cụ tìm kiếm toàn văn bản Sphinx Search để khắc phục những nhược điểm của việc tìm kiếm trên website Magento.

Chương 2. TÌM KIẾM TRÊN MAGENTO

2.1. Giới thiệu

Ngày nay việc mua sắm trực tuyến ngày càng phát triển và trở nên phổ biến trên thế giới, việc tìm kiếm nhanh một mặt hàng sẽ làm tăng khả năng mua hàng của người dùng. Vì vậy nếu thiếu một công cụ tìm kiếm hiệu quả sẽ tác động tiêu cực tới việc mua bán hàng hóa. Giả sử chúng ta có hơn 10000 mặt hàng cần bán nếu chúng ta sử dụng phương pháp tìm kiếm tuần tự đơn thuần sẽ rất khó khăn và tốn rất nhiều thời gian để thực hiện, Mục đích của bất kỳ website thương mại điện tử nào là làm cho việc tìm kiếm và mua sản phẩm dễ dàng, thuận tiện nhất cho khách hàng, làm cho khách hàng mua hàng nhiều hơn và nhiều hơn nữa, người tiêu dùng sẽ sử dụng công cụ tìm kiếm để tìm trên website những mặt hàng họ cần mua. Và nếu tìm được sản phẩm đó ngay lập tức thì có khả năng mua hàng sẽ rất cao.

Trên Magento phiên bản hiện tại cung cấp công cụ tìm kiếm tiêu chuẩn với ba loại như sau:

a) Tìm kiếm 'like'

Để lựa chọn chế độ tìm kiếm like ta tiến hành cấu hình như sau:

Bước 1: Trong Admin menu, lựa chọn System > Configuration

Bước 2: Chọn Catalog > Catalog Search

Hình 2.1. Chọn chức năng tìm kiếm like

b) Tìm kiếm toàn văn bản

Cách lựa chọn chế độ tìm kiếm toàn văn bản tương tự như cách cấu hình trong lựa chọn Search like.

Catalog Search		
Minimal Query Length	<input type="text" value="1"/>	[STORE VIEW]
Maximum Query Length	<input type="text" value="128"/>	[STORE VIEW]
Maximum Query Words Count	<input type="text" value="10"/>	[STORE VIEW]
	<small>▲ Applies for "Like" search type only.</small>	
Search Type	<input type="text" value="Fulltext"/>	[STORE VIEW]
Apply Layered Navigation if Search Results are Less Than	<input type="text" value="2000"/>	[STORE VIEW]
	<small>▲ Enter "0" to enable layered navigation for any number of results.</small>	

Hình 2.2. Chọn chức năng tìm kiếm Fulltext

c) Tìm kiếm kết hợp giữa like và tìm kiếm toàn văn bản:

Kết hợp: tìm kiếm bằng cách kết hợp 2 loại tìm kiếm ở trên

Catalog Search		
Minimal Query Length	<input type="text" value="1"/>	[STORE VIEW]
Maximum Query Length	<input type="text" value="128"/>	[STORE VIEW]
Maximum Query Words Count	<input type="text" value="10"/>	[STORE VIEW]
	<small>▲ Applies for "Like" search type only.</small>	
Search Type	<input type="text" value="Combine (Like and Fulltext)"/>	[STORE VIEW]
Apply Layered Navigation if Search Results are Less Than	<input type="text" value="2000"/>	[STORE VIEW]
	<small>▲ Enter "0" to enable layered navigation for any number of results.</small>	

Hình 2.3. Chọn chức năng tìm kiếm kết hợp

Trong các loại tìm kiếm tiêu chuẩn trên Magento ở trên nếu chọn thiết lập cấu hình tìm trong kiếm chế độ ‘like’ thì trong thực tế sử dụng sẽ không tạo ra kết quả tìm kiếm như ý vì bản chất của tìm kiếm like có nhiều hạn chế. Kết quả trả về của loại tìm kiếm này là giống từ khóa tìm kiếm bạn nhập. Ví dụ khi bạn đang bán trực tuyến các sản phẩm như đồ thể thao và đồ bơi, và các khách hàng trên website nhập vào từ tìm kiếm ‘suit’ kết quả tìm kiếm sẽ trả về các sản phẩm có chứa từ ‘suit’ trong đó, và thời gian tìm kiếm tương đối chậm. Phương pháp tìm kiếm này không dựa vào độ liên quan giữa từ khóa tìm kiếm và kết quả tìm kiếm nên nó không phải là phương pháp tìm kiếm chính xác.

Nếu chọn phương pháp tìm kiếm toàn văn bản (fulltext) thì kết quả trả về sẽ hiệu quả hơn so với loại tìm kiếm “like” ở trên và có đánh giá mức độ liên quan giữa kết quả tìm kiếm và từ khóa tìm kiếm. Nhưng trong chế độ này có nhiều hạn chế về lựa chọn tìm kiếm vì vậy kết quả trả về không như mong đợi của người sử dụng.

Nếu lựa chọn phương pháp tìm kiếm “kết hợp” thì nó sẽ có được ưu điểm của cả loại tìm kiếm theo chế độ ‘like’ và tìm kiếm theo “fulltext” cùng một lúc.

Như đánh giá ở trên cả 3 loại tìm kiếm này đều không đạt hiệu suất tìm kiếm tốt, thời gian tìm kiếm không đạt yêu cầu, kết quả tìm kiếm có sự nhiễu cao. Để giải thích lý do tại sao chúng ta sẽ tìm hiểu chi tiết trong phần tiếp theo.

2.2. Thuật toán nền tảng của các loại tìm kiếm trong Magento

Để tìm hiểu rõ hơn nhược điểm của các loại tìm kiếm trong Magento chúng ta sẽ đi sâu chi tiết vào từng loại tìm kiếm để:

2.2.1. Tìm kiếm like

Đây là phương thức tìm kiếm đơn giản nhất trong các phương thức tìm kiếm ở trên, áp dụng thuật toán tìm kiếm tuần tự để thực hiện việc tìm kiếm trong cơ sở dữ liệu. Trong Magento mỗi khi chúng ta nhập vào từ khóa cần tìm kiếm hệ thống sẽ ngắt chuỗi từ khóa thành các từ riêng và thực hiện tìm kiếm từng từ, sau đó kết hợp các kết quả tìm kiếm bằng biểu thức logic “or” sẽ được kết quả tìm kiếm cuối cùng.

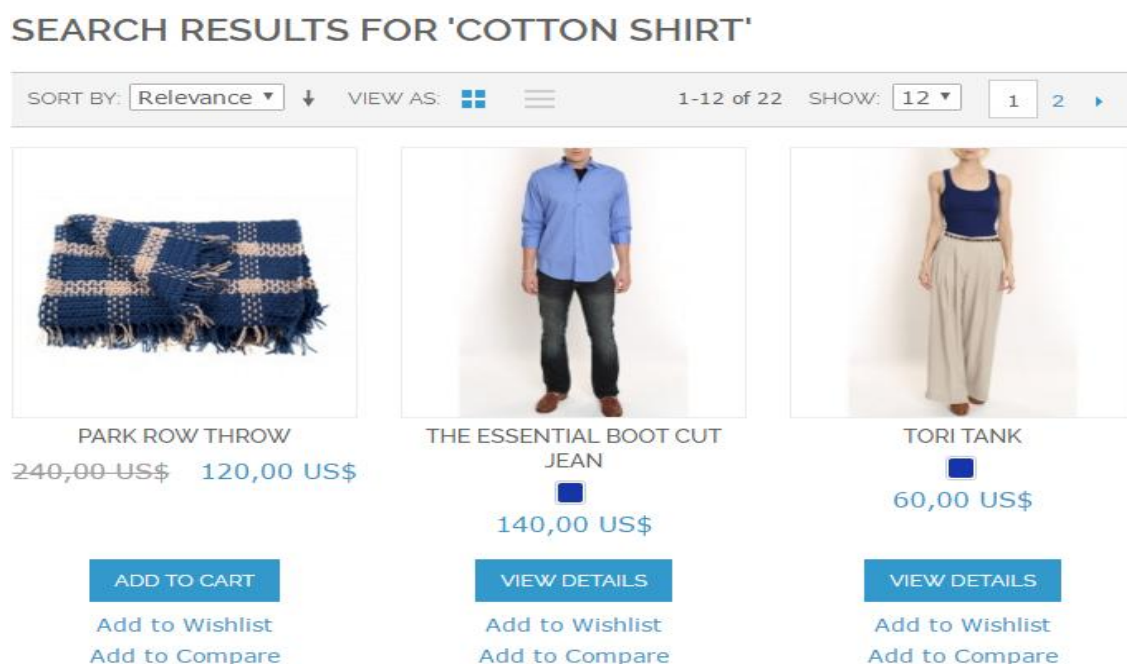
Xem Dòng 326 trong lớp **Mage_CatalogSearch_Model_Resource_Fulltext**:

```
Mage_CatalogSearch_Model_Resource_Fulltext::prepareResult()
$words = Mage::helper('core/string')->splitWords($queryText,
true, $query->getMaxQueryWords());
    foreach ($words as $word) {
        $like[] = $helper->getCILike('s.data_index',
$word, array('position' => 'any'));
    }
    if ($like) {
        $likeCond = '(' . join(' OR ', $like) . ')';
    }
}
```

Bạn có thể nhìn thấy nó chia tách từng từ trong truy vấn và kết nối chúng với nhau trong câu lệnh LIKE. Điều này tương tự như biểu thức truy vấn like trong SQL.

```
WHERE `attribute` LIKE 'my' OR `attribute` LIKE 'Search' OR
`attribute` LIKE 'query'
```

Ví dụ ở chế độ tìm kiếm like nếu cần tìm một áo sơ mi chất liệu bằng cotton, Người sử dụng nhập vào từ khóa tìm kiếm “COTTON SHIRT”. Kết quả như sau:



Hình 2.4. Kết quả tìm kiếm từ khóa “cotton shirt” ở chế độ like.

Kết quả trả về là 22 sản phẩm. Tất cả những sản phẩm có chất liệu bằng cotton hoặc kiểu áo mi sẽ được trả lại. Trong trường hợp này kết quả tìm kiếm được đánh giá là bị nhiễu rất nhiều kết quả không đúng với yêu cầu của khách hàng.

2.2.2. Tìm kiếm toàn văn bản

Magento sử dụng tìm kiếm toàn văn bản trong MySQL để thực hiện tìm kiếm. Vì vậy để tìm hiểu về loại tìm kiếm này chúng ta cần làm rõ một số nội dung về tìm kiếm toàn văn bản trong MySQL.

a) Tìm kiếm toàn văn bản trong MySQL

MySQL hỗ trợ đánh chỉ số và tìm kiếm toàn văn bản:

Chỉ số toàn văn bản trong MySQL chỉ có thể được sử dụng với bảng InnoDB hoặc MyISAM và chỉ có thể tạo cho các cột CHAR, VARCHAR hoặc TEXT.

Việc đánh chỉ số trong MySQL được thực hiện bằng cách sử dụng câu lệnh “CREATE TABLE” khi tạo một bảng mới hoặc sử dụng câu lệnh “ALTER TABLE” hoặc “CREATE INDEX” khi một bảng đã được tạo trong MySQL

Tìm kiếm toàn văn bản trong MySQL được thực hiện bằng cách sử dụng cú pháp”

MATCH()...AGAINST...

Trong đó MATCH() chứa danh sách các cột cần tìm kiếm.

AGAINST chứa từ khóa cần tìm kiếm.

Đánh chỉ số trong MySQL:

MySQL có 2 lựa chọn trong việc đánh chỉ số:

- Giới hạn chiều dài của các từ có thể đánh chỉ số.
- Bảng danh sách các từ không đánh chỉ số hay còn gọi là “stopword”.

Bảng dưới đây cho thấy danh sách mặc định của một số từ dừng cho chỉ số tìm kiếm MyISAM được lưu trữ trong file storage/myisam/ft_static.c.

a's	able	about	above	according
accordingly	across	actually	after	afterwards
again	against	ain't	all	allow
allows	almost	alone	along	already
also	although	always	am	among
amongst	an	and	another	any
anybody	anyhow	anyone	anything	anyway
anyways	anywhere	apart	appear	appreciate
appropriate	are	aren't	around	as
aside	ask	asking	associated	at
available	away	awfully	be	became
because	become	becomes	becoming	been
before	beforehand	behind	being	believe
below	beside	besides	best	better
between	beyond	both	brief	but
by	c'mon	c's	came	can
can't	cannot	cant	cause	causes
certain	certainly	changes	clearly	co
com	come	comes	concerning	consequently
consider	considering	contain	containing	contains
corresponding	could	couldn't	course	currently
definitely	described	despite	did	didn't

Bảng 2.1. Bảng một danh sách một số stopwords.

Lựa Chọn các chế độ tìm kiếm trong MySQL

Lựa chọn chế độ tìm kiếm:

MySQL có thể thực hiện tìm kiếm toàn văn bản bằng cách sử dụng chế độ “IN BOOLEAN MODE”, “IN NATURAL LANGUAGE MODE”, và “QUERY EXPANSION SEARCH”.

Natural language Search: Các từ khóa tìm kiếm được diễn giải như một cụm từ trong ngôn ngữ tự nhiên của con người. Để xác định loại tìm kiếm này người dùng thêm cụm từ “NATURAL LANGUAGE MODE”

Boolean Search: Dịch từ khóa tìm kiếm bằng cách sử dụng các quy tắc của một ngôn ngữ truy vấn đặc biệt

Các quy tắc bao gồm:

><	Tăng, giảm xếp loại
()	Nhóm các chuỗi tìm kiếm
~	Xác định có thể chứa hoặc không chứa từ khóa
*	Đại diện cho một kí tự bất kỳ
‘	Tìm kiếm chính xác các kí tự

Để xác định loại tìm kiếm này người sử dụng thêm cụm từ “IN BOOLEAN MODE”

Query expansion Search: Đây là loại mở rộng của natural Search. Nó rất hữu ích khi cụm từ tìm kiếm quá ngắn. Ví dụ một người tìm kiếm từ “Database” thì tất cả những từ phù hợp như “MySQL”, “Oracle”, “DB2” và “RDBMS” phải được trả lại. Với expansion Search việc tìm kiếm được thực hiện hai lần. Ở lần tìm kiếm thứ 2 từ tìm kiếm ban đầu được nối với các từ liên quan nhiều nhất để thực hiện tìm kiếm.

Để xác định loại tìm kiếm này người sử dụng thêm cụm từ “ WITH QUERY EXPANSION”.

b) Áp dụng tìm kiếm toàn văn bản trong Magento

Trong chế độ tìm kiếm này việc tìm kiếm trên trên trường data_index của bảng catalogSearch_fulltext với biểu thức:

MATCH(data_index) AGAINST('Search query' IN BOOLEAN MODE).

Người sử dụng có thể nhìn thấy cách thức truy vấn này nằm trong lớp: `Mage_Catalog_Search_Model_Resource_Helper_Mysql4`. Trong dòng 44.

```
public function chooseFulltext($table, $alias, $select)
{
    $field = new Zend_Db_Expr('MATCH ( '.$alias.'.data_index)
AGAINST (:query IN BOOLEAN MODE)');
    $select->columns(array('relevance' => $field));
    return $field;
}
```

Bảng cơ sở dữ liệu Magento sử dụng khi thực hiện tìm kiếm fulltext là `catalogSearch_fulltext`. Một ví dụ giá trị:

```
EmCO0014e|Emma Certified|Emma Certified Organic Herbal Tonic
Mist TRIAL/TRAVEL|Australian|Certified Organic|Palm Oil
Free|Nut Free|Vegan Suitable|
```

Giả sử ta cần tìm kiếm từ khóa “cotton shirt” như phần trên kết quả trả về như sau:

SEARCH RESULTS FOR 'COTTON SHIRT'

The screenshot displays search results for 'COTTON SHIRT'. At the top, there are controls for sorting (set to 'Relevance'), viewing (grid view), and pagination (1-12 of 22 items, showing 12 items per page, page 1 of 2). Three product cards are shown:

- FRENCH CUFF COTTON TWILL OXFORD**: Price 190,00 US\$. Includes 'VIEW DETAILS', 'Add to Wishlist', and 'Add to Compare' buttons.
- SLIM FIT DOBBY OXFORD SHIRT**: Price 175,00 US\$ (crossed out) and 140,00 US\$. Includes 'VIEW DETAILS', 'Add to Wishlist', and 'Add to Compare' buttons.
- PLAID COTTON SHIRT**: Price 160,00 US\$. Includes color swatches, 'VIEW DETAILS', 'Add to Wishlist', and 'Add to Compare' buttons.

Hình 2.5. Kết quả tìm kiếm từ khóa “cotton shirt” ở chế độ fulltext.

Kết quả trả về hiệu quả hơn chế độ like. Có sắp xếp theo mức độ liên quan của từ khóa tìm kiếm và mức độ tìm kiếm.

2.2.3. Tìm kiếm kết hợp

Tương tự ở chế độ full-text cũng trả về kết quả là 22 sản phẩm và rất nhiều kết quả không mong muốn. Tuy nhiên có cải thiện hơn kết quả tìm kiếm bằng like là có xếp hạng các sản phẩm liên quan lên vị trí đầu tiên.

Bạn tìm trong dòng 354: Mage_CatalogSearch_Model_Resource_Fulltext

```
if ($likeCond != '' && $SearchType ==
Mage_CatalogSearch_Model_Fulltext::SEARCH_TYPE_COMBINE) {
    $where .= ($where ? ' OR ' : '') . $likeCond;
} elseif ($likeCond != '' && $SearchType ==
Mage_CatalogSearch_Model_Fulltext::SEARCH_TYPE_LIKE) {
    $select->columns(array('relevance' => new
Zend_Db_Expr(0)));
    $where = $likeCond;
}
```

Với chế độ kết hợp: Kết quả trả về tương tự

Cả ba loại tìm kiếm có sẵn trong Magento có những hạn chế nhất định vì vậy yêu cầu đặt ra là nâng cấp tính năng tìm kiếm có sẵn để cải thiện trải nghiệm tìm kiếm của người dùng, và trả về các kết quả tìm kiếm tối ưu nhất. Có nhiều hơn một cách để tối ưu hóa công cụ tìm kiếm của của Magento. Để thực hiện được điều này chúng ta có thể xây dựng và cài đặt vào Magento thêm phân mở rộng tìm kiếm để bổ xung thêm chức năng tìm kiếm tiên tiến làm giảm thiểu thời gian tìm kiếm và nâng cao chất lượng của kết quả tìm kiếm.

2.3. Giải pháp nâng cấp tính năng tìm kiếm trên Magento

2.3.1. Phương pháp tiếp cận

Nâng cấp tính năng tìm kiếm là cải thiện về kết quả hiển thị và thời gian tìm kiếm. Để đáp ứng yêu cầu này thì ta quan tâm đến một số các lĩnh vực sau:

- Mức độ liên quan của từ khóa tìm kiếm và kết quả tìm kiếm.
- Khả năng sửa lỗi chính tả khi người dùng nhập vào từ khóa tìm kiếm.
- Khả năng tìm kiếm các từ đồng nghĩa.
- Tự động hoàn thành hoặc gợi ý cho các từ khóa tìm kiếm.
- Tăng tốc độ tìm kiếm.

Độ liên quan của từ khóa tìm kiếm và kết quả tìm kiếm:

Sự phù hợp của kết quả tìm kiếm là chìa khóa cho người dùng tin tưởng và trung thành với cửa hàng trực tuyến của bạn. Rõ ràng đây là điều quan trọng để khách hàng của bạn có thể tìm thấy những sản phẩm mà họ đang tìm kiếm khi thực hiện tìm kiếm trên cửa hàng trực tuyến của bạn. Nếu không họ sẽ tìm nơi khác.

Khả năng sửa lỗi chính tả khi nhập vào từ khóa tìm kiếm:

Nó phổ biến cho người dùng để sử dụng những sai sót chính tả khi tìm kiếm các mặt hàng. Khả năng chịu lỗi và sửa lỗi cải thiện đáng kể trải nghiệm của người dùng và giữ người sử dụng trên website của bạn.

Khả năng tìm kiếm các từ đồng nghĩa đồng nghĩa:

Đó là khả năng tuyệt vời để thêm các từ đồng nghĩa cho các truy vấn tìm kiếm để khách hàng sẽ được trình bày và thuật ngữ sử dụng phổ biến trên khắp các cửa hàng. Ví dụ bạn nhập vào từ khóa ‘bike’ không được sử dụng phổ biến trong các thuộc tính được lập chỉ mục tìm kiếm bạn có thể sử dụng một từ đồng nghĩa ‘bicycle’ nếu đó là một thuật ngữ rất phổ biến được sử dụng trong tên sản phẩm và mô tả.

Gợi ý và tự động hoàn thành từ khóa tìm kiếm:

Đề xuất tự động hoàn thành là một cách tiết kiệm thời gian tuyệt vời cho người sử dụng. Tính năng này cho phép danh sách các kết quả có liên quan nhất ngay sau khi nhập vào từ tìm kiếm đầu tiên trong hộp tìm kiếm. Các kết quả tìm

kiếm thường được sắp xếp và hiển thị trong một hộp tìm kiếm menu thả xuống. Tự động hoàn chỉnh cho tìm kiếm website đặc biệt giá trị nếu bạn có nhiều sản phẩm. Bạn cũng có thể hiển thị kết quả đề nghị bên trong hộp tìm kiếm phù hợp với truy vấn tìm kiếm tự động hoàn hành, cho phép khách hàng bấm trực tiếp vào trang chi tiết sản phẩm mà không cần phải xem xét thông qua một trang kết quả tìm kiếm.

Tốc độ tìm kiếm:

Không muốn chờ đợi quá lâu là tâm lý chung của mọi khách hàng, giả sử khách hàng nhập vào một từ khóa để tìm kiếm nếu phải chờ đợi kết quả trả về quá lâu thì khách hàng sẽ bỏ đi. Vì vậy vấn đề này cần được đặt ra trong các cách nâng cao tìm kiếm trong Magento.

2.3.2. Các giải pháp mở rộng tính năng tìm kiếm cho Magento

Dưới đây là danh sách các nền tảng hay được sử dụng nhất tìm để nâng cấp cũng như mở rộng tính năng tìm kiếm cho cửa hàng trực tuyến trên nền tảng Magento.

- Algolia Search.
- Solr.
- Sphinx.
- ElasticSearch.
- Amazon CloudSearch.
- Sooqr Search.
- SLI learning Search.

a) Algolia Search

Algolia là một giải pháp lưu trữ và đồng thời là một công cụ tìm kiếm toàn văn bản. Dữ liệu cần tìm kiếm của chúng ta sẽ được đánh chỉ số và lưu trên server riêng của Algolia. Trên server này có cài đặt các thuật toán tìm kiếm và lựa chọn tìm kiếm hướng tới mục tiêu làm tăng hiệu quả tìm kiếm của người dùng. Algolia Có khả năng cung cấp kết quả tìm kiếm thời gian thực khi bạn nhập vào kí tự tìm kiếm đầu tiên. Algolia API cho phép bạn nhanh chóng và liên tục thực hiện tìm kiếm thời gian thực bên trong website của bạn và các ứng dụng di động.

Tính năng quan trọng của Algolia Search:

Tùy biến mức độ liên quan và xếp hạng tìm kiếm.

Sửa lỗi chính tả:

Việc đánh sai từ cần tìm kiếm đặc biệt là người sử dụng thiết bị động thường xuyên xảy ra. Thay vì để những sai lầm phổ biến này làm hỏng kết quả tìm kiếm của bạn. Algolia có khả năng sử dụng lỗi đánh máy tự phát hiện ra những gì người dùng đang tìm kiếm để trả về kết quả chính xác.

Đánh dấu thông minh:

Giúp bạn đánh dấu những kết quả nổi bật sau mỗi truy vấn tìm kiếm.

Và một số tính năng khác như: tìm kiếm theo khu vực, tùy chọn cách hiển thị các giao diện tìm kiếm.

Tốc độ:

Trả lại kết quả trong mili giây.

Algolia Search được xây dựng dưới dạng extension nên việc cài đặt Algolia Search vào Magento được thực hiện như sau:

- Truy cập vào trang <https://www.Magentocommerce.com>.
- Tìm kiếm từ khóa Algolia.
- Chọn nút cài đặt (chú ý bạn phải đăng kí và đăng nhập vào Magento Connect).
- Nhập key từ bảng quản trị Magento của bạn vào Magento Connect.
- Tạo một tài khoản Aloglia : https://www.algolia.com/users/sign_up
- Nhận chứng nhận tài khoản của bạn để cấu hình trong Magento extension.
- Trong bảng quản trị tới System > Index Management và chọn Reindex data để xây dựng lại tất cả các chỉ số.

Hạn chế của phương pháp này: Đây là một công cụ tìm kiếm bản quyền phải trả phí chúng ta không thể thay đổi thuật toán tìm kiếm tất cả phụ thuộc vào nhà phát hành phần mềm.

b) Solr Search

Solr là một nền tảng tìm kiếm mã nguồn mở, được viết bằng Java. Các tính năng chính bao gồm: Tìm kiếm toàn văn bản, lập chỉ số thời gian thực...Solr được thiết kế phù hợp với khả năng mở rộng và chịu lỗi cao.

Lịch sử phát triển: Năm 2004 Solr được tạo bởi Yonik seely tại CNET Networks lúc đầu dự án này chỉ phục vụ khả năng tìm kiếm cho website của công ty.

Tháng riêng năm 2006, CNET Networks đã quyết định công khai công bố mã nguồn dự án. Tháng 9 năm 2008, Soler 1.3 đã được phát hành với nhiều cải tiến. Qua nhiều phiên bản được cập nhật và cải tiến. Vào tháng 2/2015, Solr 5.0 đã được phát hành, Solr được đóng gói như là một ứng dụng độc lập.

Một số đặc tính của Solr:

- Tích hợp khả năng tìm kiếm Full-Text nâng cao.
- Lập chỉ mục thời gian thực.
- Highlighting.
- Dynamic clustering.
- Database integration.
- Geospatial Search.

Solr có khả năng mở rộng và được sử dụng trong rất nhiều website trên thế giới.

Hiện tại công cụ tìm kiếm solr cho Magento có nhiều extension được xây dựng, một phần mở rộng cơ bản như:

- Search Suite by MageWorx.
- Solr Bridge Search.
- SolrGento.
- Advance Search with Solr.

Các cài đặt các extension này được thực hiện giống cách cài đặt Extension Algolia.

c) ElasticSearch

ElasticSearch là một mã nguồn mở tìm kiếm toàn văn bản và máy phân tích dữ liệu. Nó cho phép bạn lưu trữ, tìm kiếm và phân tích khối dữ liệu lớn một cách nhanh trong và gần như trong thời gian thực, được viết trên nền java. Phiên bản đầu tiên được phát hành ra công chúng là version 0.4 vào tháng 2/2010.

Dưới đây là một vài trường hợp sử dụng ElasticSearch:

Nếu bạn có một cửa hàng trực tuyến, ở đây bạn cho phép khách hàng tìm kiếm các sản phẩm. Trong trường hợp này, bạn có thể sử dụng ElasticSearch để lưu toàn bộ danh mục sản phẩm và hàng tồn kho của bạn và cung cấp các gợi ý tìm kiếm và tự động hoàn thiện gợi ý cho khách hàng.

Nếu bạn có nhu cầu phân tích/ kinh doanh thông minh và muốn nhanh chóng điều tra, phân tích đặt câu hỏi trên hàng triệu hoặc hàng tỷ hồ sơ. Trong trường hợp này, Bạn có thể sử dụng ElasticSearch để lưu trữ dữ liệu và sau đó sử dụng Kibana (một phần của ElasticSearch) để xây dựng các bảng điều khiển để tìm các dữ liệu quan trọng với bạn.

Các tính năng chính:

- Gợi ý tự động khi người dùng nhập vào từ khóa tìm kiếm.
- Autocomplete rất nhanh.
- Chỉnh sửa thông minh khi truy vấn sai chính tả.
- Tù ý kiếm trên tùy chọn thuộc tính sản phẩm.
- Xử lý hàng ngàn sản phẩm một cách dễ dàng.

Một số extension cho Magento như sau:

Extension for Elastic Search.

Qbox ElasticSearch Extension.

d) Amazon CloudSearch

Amazon cloudSearch là một giải pháp tìm kiếm hiệu quả cho trang web hoặc ứng dụng của bạn. Amazon CloudSearch hỗ trợ 34 ngôn ngữ và các tính năng tìm kiếm phổ biến như highlighting, autocomplete, tìm kiếm theo khu vực địa lý.

e) Sooqr

Extensions: . Sooqr for Magento

Sooqr là một dịch vụ trả phí cho phép tìm kiếm nhanh chóng cho các cửa hàng trực tuyến. Nó tạo ra kết quả tìm kiếm liên quan, sửa lỗi chính tả đảm bảo rằng việc tìm kiếm diễn ra bằng cách sử dụng thuật ngữ đúng theo dõi những cách người dùng thực hiện tìm kiếm, do đó các chủ sở hữu trang web có thể đọc một cái nhìn sâu sắc thực sự tốt của việc sử dụng tìm kiếm trang web của bạn.

Ngoài ra còn một số giải pháp tìm kiếm khác như SLI Learning Search...

f) Sphinx :

Sphinx là một mã nguồn mở tìm kiếm toàn văn bản đầy đủ. Search sphinx có nhiều ưu điểm như độ liên quan cao, sắp xếp kết quả tìm kiếm linh hoạt, hiệu suất cao và cho phép sửa lỗi chính tả, viết tắt. Công cụ tìm kiếm này được sử dụng bởi các trang web phổ biến khác nhau như: Craigslist.org mysql.com, joomla.org, tumblr.com, Dailymotion.com

Qua tìm hiểu các giải pháp nâng cấp các tính năng tìm kiếm mặc định cho Magento. Tôi chọn giải pháp sử dụng Sphinx Search vì sphinx là một phần mềm mã nguồn mở cho phép chúng ta có thể mở rộng, phát triển và đặc biệt có các tính năng phù hợp để đáp ứng yêu cầu của bài toán nâng cấp tính năng tìm kiếm trên Magento.

2.3.3. Sử dụng Sphinx để nâng cấp tính năng tìm kiếm

Để giải quyết vấn đề nâng cấp tính năng tìm kiếm, Tôi đã xây dựng một chương trình mở rộng tính năng tìm kiếm riêng cho Magento trong đó sử dụng công cụ tìm kiếm toàn văn bản Sphinx.

Ưu điểm chính của sphinx như sau:

- Tốc độ xây dựng lại chỉ số lớn (quan trọng trong trường hợp nếu số lượng sản phẩm lớn hơn 50000) sản phẩm.
 - Hỗ trợ hình thái ngôn ngữ khác nhau.
 - Và ấn tượng nhất là hỗ trợ cú pháp truy vấn mở rộng.
 - Các thủ tục chung cho chuỗi tìm kiếm.
 - Người sử dụng nhập vào một cụm từ tìm kiếm.
 - Chúng tôi đang tìm kiếm cho cụm từ này trong các truy vấn được thực thi.
- Nếu có một kết quả, sau đó chúng tôi lập thức trả lại một tập các kết quả của sản phẩm đã tìm thấy.
- Nếu không chúng ta sẽ chuyển đổi truy vấn tìm kiếm.
 - Gửi yêu cầu và nhận được kết quả từ máy chủ sphinx
 - Lưu kết quả liên quan của nó vào cơ sở dữ liệu.
 - Điều đáng nói đến là yêu cầu máy chủ Sphinx được xử lý trong một phần nhỏ của một giây. Trong dự án thực tế với danh mục sản phẩm của 60000 với 10000 truy vấn chỉ có 12 truy vấn được thực hiện nhiều hơn 0,005 giây.

- Cải thiện cơ chế xếp hạng và sắp xếp kết quả.
- Hỗ trợ các hình thái ngôn ngữ khác nhau.

2.4. Kết luận chương 2

Sau khi cài đặt và cấu hình Magento trên LEMP server sử dụng VPS của Digital Ocean, Nghiên cứu sử dụng tính năng tìm kiếm cơ bản có sẵn trong Magento. Kết quả cho thấy:

- Tốc độ lập chỉ số (indexing) chậm so với các công cụ tìm kiếm khác.
- Tốc độ tìm kiếm cùng 1 từ khóa với lượng dữ liệu như nhau chậm.
- Hiệu suất tìm kiếm kém, các lựa chọn tìm kiếm rất hạn chế.

Với kết quả như vậy, Khi người sử dụng thực hiện chức năng tìm kiếm sản phẩm trên gian hàng được xây dựng bằng Magento sẽ trả lại những kết quả không mong muốn, và thời gian trả lại kết quả chậm.

Để giải quyết vấn đề này, Nghiên cứu cải thiện tính năng tìm kiếm của Magento bằng cách sử dụng Sphinx là nhu cấp thiết và phù hợp với thực tế. Sphinx đáp ứng được hầu hết các yêu cầu nâng cấp tìm kiếm, đồng thời là chương trình mã nguồn mở, rất phù hợp để nghiên cứu và phát triển.

Nội dung của Sphinx Search được giới thiệu trong chương 3.

Chương 3. SỬ DỤNG SPHINX SEARCH NÂNG CẤP TÍNH NĂNG TÌM KIẾM TRÊN MAGENTO

3.1. Một số vấn đề về tìm kiếm toàn văn bản

Sphinx là công cụ tìm kiếm toàn văn bản, nên trước khi đi xa hơn chúng ta cần tìm hiểu một số vấn đề chung về tìm kiếm toàn văn bản và vì sao nó vượt trội hơn so với các tìm kiếm truyền thống.

3.1.1. Khái niệm tìm kiếm toàn văn bản

Tìm kiếm toàn văn bản tên tiếng anh Full Text Search là một kỹ thuật được sử dụng để tìm kiếm một tài liệu được lưu trữ trên máy tính hoặc cơ sở dữ liệu. Trong FTS các công cụ tìm kiếm xem xét các tất cả các từ trong mỗi tài liệu được lưu trữ và cố gắng tìm ra những từ phù hợp yêu cầu của người sử dụng.

3.1.2. Khái niệm chỉ số và thuật toán đánh chỉ số

Khái niệm: Một database index là một cấu trúc dữ liệu để cải thiện tốc độ tìm kiếm dữ liệu, Nếu không có index, các công cụ tìm kiếm sẽ quét tất cả các tài liệu điều đó sẽ tốn thời gian và khả năng tính toán.

Thuật toán đánh chỉ số: Các thuật toán xây dựng chỉ số bao gồm 1 số loại như sau: Suffix tree , Inverted index , Citation index, Ngram index , Document-term matrix .

Chúng ta tìm hiểu một loại thuật toán xây dựng chỉ số mà Sphinx sử dụng để hiểu rõ hơn cách xây dựng chỉ số:

Inverted index:

Inverted index là kỹ thuật thay vì lập chỉ số theo đơn vị dòng chúng sẽ tiến hành lập chỉ số theo đơn vị Term. Cụ thể hơn, Inverted Index là một cấu trúc dữ liệu, nhằm mục đích kết hợp giữa term và các document chứa các term đó.

Ví dụ chúng ta có 2 tài liệu như sau:

Document	String
D1	The quick brown fox jumped over the lazy dog
D2	Quick brown foxes leap over lazy dogs in summer

Bảng 3.1. Danh sách các document cần tìm kiếm

Inverted index của 3 document đó sẽ được lưu dưới dạng như sau:

Term	Doc_1	Doc_2
Quick		x
The	x	
brown	x	x
dog	x	
dogs		x
fox	x	
foxes		x

Bảng 3.2. Danh sách term

Bây giờ nếu chúng ta muốn tìm quick brow, chúng ta chỉ cần tìm các tài liệu trong đó mỗi term xuất hiện.

Term	Doc_1	Doc_2
brown	x	x
quick	x	

Total	2	1
-------	---	---

Bảng 3.3. Kết quả tìm kiếm dựa vào term

Cả hai document đều phù hợp, nhưng các tài liệu đầu tiên có kết quả phù hợp hơn so với tài liệu thứ 2. Nếu chúng ta áp dụng một thuật toán đánh giá đơn giản đếm số lượng term phù hợp trong mỗi tài liệu. Chúng ta có thể kết luận điều trên.

Nhưng có một vài vấn đề với inverted index đó là:

- Quick and quick xuất hiện như 2 term riêng biệt, trong khi 2 từ này là giống nhau.
- Fox và foxes là 2 từ khá giống nhau, dog và dogs cũng vậy. Chúng có cùng gốc.
- Jumped và leap không cùng gốc nhưng chúng có nghĩa giống nhau.
- Với các chỉ số trên. Một tìm kiếm +Quick + fox (+ có nghĩa là bắt buộc phải có mặt) không phù hợp với bất kỳ tài liệu nào.

Nếu chúng ta chuyển các term sang một định dạng tiêu chuẩn, sau đó chúng ta có thể tìm thấy các tài liệu phù hợp.

- Dùng lowercased chuyển Quick thành quick.
- Foxes cắt hậu tố 'es' thành fox. Dogs → dog
- jumped và leap chuyển thành jump.

Term	Doc_1	Doc_2
brown	x	x
dog	x	x
fox	x	x
in		x
lazy	x	x

over	x	x
quick	x	x
summer		x
leap	x	x

Bảng 3.4. Bảng kết quả chuyển term sang định dạng tiêu chuẩn.

Nhưng chúng ta vẫn không thể tìm kiếm +Quick +fox vẫn thất bại vì sẽ không có chính xác từ “Quick”. Tuy nhiên nếu chúng ta áp dụng chuyển đổi chuỗi “Quick fox” thành chữ thường sẽ thành “quick fox” , việc tìm kiếm sẽ thực hiện được.

Có 2 vấn đề cần giải quyết để việc đánh chỉ số có thể thực hiện được đó là:

Bài toán tách string (gọi là tokenize)

Có 2 kỹ thuật tokenize cơ bản là

- N-Gram.
- Morphological Analysis.

Quá trình chuẩn hóa các tokenization được gọi là phân tích (Analysis).

N-Gram:

N-gram là kỹ thuật tokenize một chuỗi thành các chuỗi con, bằng cách chia đều chuỗi đã có thành các chuỗi con đều nhau, có độ dài là N.

Về cơ bản thì N thường nằm từ 1-3, với các tên gọi tương ứng là unigram(N=1), Bigram(N=2), trigram(N=3).

Ví dụ đơn giản chúng ta có chuỗi “good morning”, được phân tích thành bigram:

“good morning” => { “go”, “oo”, “od”, “d”, “ m”, “mo”, “or”, “rn”, “ni”, “in”, “ng” }

Morphological Analysis:

Là một kỹ thuật phổ biến trong xử lý ngôn ngữ tự nhiên. MA sẽ là kỹ thuật tokenize để tách một chuỗi ra thành các chuỗi có ý nghĩa. Ví dụ như cụm từ “good morning” ở trên chúng ta sẽ tách như sau:

“good morning” => “good”, “morning”

Để được kết quả như trên, ngoài việc sở hữu một bộ từ điển tốt để phân biệt được từ nào có nghĩa, MA phải sử dụng các kết quả nghiên cứu sâu về ngôn ngữ tự nhiên.

Analysis

Phân tích là một quá trình bao gồm các nội dung sau đây:

Đầu tiên, tokenizing(tách) một khối văn bản thành các term phù hợp cho việc inverted index. Tối ưu hóa các term này để thỏa mãn các tiêu chuẩn phục vụ cho việc cải thiện chức năng tìm kiếm hoặc truy hồi.

3.2. Tổng quan về Sphinx Search

3.2.1. Định nghĩa:

Sphinx Search: là một công cụ tìm kiếm toàn văn bản, cung cấp khả năng tìm kiếm nhanh chóng, độ chính xác cao, phù hợp với các ứng dụng của bên thứ ba. Nó có nhiều đặc điểm phù hợp và hỗ trợ cho việc tìm kiếm trên cơ sở dữ liệu SQL, tích hợp tốt vào các ngôn ngữ kịch bản như PHP, Python, Ruby, Perl và Java.

3.2.2. Tính năng của Sphinx Search.

- Lập chỉ mục và tìm kiếm hiệu suất cao.
- Lập chỉ mục nâng cao và chứa công cụ truy vấn linh hoạt, tính năng phong phú tokenizer văn bản, ngôn ngữ truy vấn, có nhiều chế độ xếp hạng kết quả tìm kiếm khác nhau.
- Đạt được kết quả truy vấn hiệu quả sau khi sử dụng (SELECT với các biểu thức, WHERE, ORDER BY, GROUP BY, HAVING ...trên kết quả tìm kiếm văn bản).
- Có khả năng mở rộng đã được kiểm chứng lên đến hàng tỷ tài liệu, terabyte dữ liệu và hàng ngàn truy vấn mỗi giây.
- Dễ dàng kết hợp với nguồn dữ liệu SQL, XML và giao diện tìm kiếm SphinxQL, SphinxAPI, hoặc SphinxSE.

- Dễ dàng mở rộng quy mô với các tìm kiếm phân tán.
- Có tốc độ lập chỉ mục cao (tối đa 10-15 MB/giây cho mỗi core trên một tiêu chuẩn nội bộ).
- Có tốc độ tìm kiếm cao (tối đa 150-250 truy vấn/ giây cho mỗi core tìm kiếm trên 1.000.000 tài liệu, 1,2 GB dữ liệu trong tiêu chuẩn nội bộ).
- Có khả năng mở rộng cao (Chỉ số cụm được biết đến nhiều nhất qua 3.000.000.000 tài liệu, đỉnh cao là 50.000.000 truy vấn/ ngày).
- Cung cấp khả năng xếp hạng mức độ phù hợp của tài liệu tốt bằng cách kết hợp xếp hạng cụm từ gần và xếp hạng thống kê (BM25).
- Hỗ trợ nhiều trường full-text cho mỗi tài liệu (mặc định lên tới 32 trường)
- Hỗ trợ nhiều thuộc tính bổ sung cho mỗi tài liệu (ví dụ group, timestamps, etc...)
- Hỗ trợ stopwords.
- Hỗ trợ morphological word từ các từ điển.
- Hỗ trợ loại trừ tokenizing.
- Hỗ trợ UTF-8 encoding.[3]

3.2.3. Lịch sử phát triển

Bắt đầu từ năm 2001, Andrew Aksyonoff (Một nhà phát triển website của nga) Đã gặp phải nhiều vấn đề trong việc thực hiện tìm kiếm trong các ứng dụng web, thời điểm hiện tại không có công cụ tìm kiếm nào đạt được cao yêu cầu tìm kiếm như tốc độ tìm kiếm cao, yêu cầu tài nguyên thấp.

Cuối cùng Andrew Aksyonoff đã quyết định xây dựng một công cụ tìm kiếm riêng mà sau này đặt tên là Sphinx. Sau một vài phiên bản đầu tiên nhận được phản hồi tích cực từ người dùng, Ông đã quyết định tiếp tục phát triển công cụ này và thành lập công ty Sphinx Technologies.[4]

3.2.4. Cài đặt sphinx trong môi trường Linux

Trong phạm vi đề tài này tôi thực hiện cài đặt trên linux nên nội dung trình bày chỉ giới thiệu việc cài đặt trên linux.

(1) Tải phiên bản mới nhất của sphinx Search từ trang web:

<http://sphinxSearch.com/downloads/release/>

(2) Giải nén tập tin vừa tải về, và đi đến thư mục chứa file vừa được giải nén:

```
$ tar -xvzf sphinx-0.9.9.tar.gz
```

```
$ cd sphinx-0.9.9
```

(3) Chạy tiện ích configure:

```
$ ./configure --prefix=/usr/local/sphinx
```

(4) Chạy câu lệnh “make”:

```
$ make
```

(5) Cài đặt ứng dụng với quyền “root”:

```
$ make install
```

3.2.5. Cách sử dụng Sphinx Search

Để sử dụng Sphinx, bạn cần thực hiện theo các bước sau:

Bước 1: Tạo file cấu hình

Mặc định tên tập tin cấu hình là sphinx.conf. Tất cả các chương trình Sphinx tìm file này trong thư mục làm việc mặc định.

Sphinx chứa một tập tin cấu hình mẫu là sphinx.conf.dist, trong đó tất cả các tùy chọn tài liệu đã được cấu hình sẵn. Copy và sử dụng tập tin cấu hình mẫu để sửa đổi và tạo tập tin cấu hình của bạn.

```
$ cd /usr/local/sphinx/etc
```

```
$ cp sphinx.conf.dist sphinx.conf
```

```
$ vi sphinx.conf
```

Tập tin cấu hình mẫu được cài đặt để xây dựng các bảng chỉ số trên cơ sở dữ liệu MySQL, do đó chúng ta sử dụng tập tin example.sql mẫu chứa một số bảng để chạy thử nghiệm.

```
$ mysql -u test < /usr/local/sphinx/etc/example.sql
```

Bước 2: Chạy indexer để tạo full-text index từ dữ liệu của bạn.

```
$ cd /usr/local/sphinx/etc
```

```
$ /usr/local/sphinx/bin/indexer --all
```

Bước 3: Truy vấn các chỉ số vừa mới được tạo ra.

Kết nối tới server:

```
$ mysql -h0 -P9306
```

Thực hiện truy vấn:

```
SELECT * FROM test1 WHERE MATCH('my document');
```

3.3. Cơ chế hoạt động của Sphinx Search

3.3.1. Các thành phần trong sphinx

Hiện tại gói đầy đủ Sphinx Search bao gồm các chương trình sau đây:

Indexer : là một tiện ích tạo ra fulltext indexes.

Searchd: là một daemon cho phép phần mềm bên ngoài (ví dụ các ứng dụng website) có thể tìm kiếm thông qua fulltext indexes (Chỉ số toàn văn bản).

SphinxAPI: là một tập hợp các thư viện client API cho các ngôn ngữ kịch bản web phổ biến (PHP, Python, Perl, Ruby).

Spelldump: Một công cụ dòng lệnh (command-line) để trích xuất các items từ một ispell hoặc MySpell định dạng từ điển để giúp tùy chỉnh chỉ số(index) của bạn.

Indextool: Là một tiện ích sửa lỗi về các chỉ số (index) được thêm vào trong phiên bản 0.9.9-RC2.

Wordbreaker: một tiện ích để phá vỡ các từ ghép thành các từ đơn riêng biệt, được thêm vào trong phiên bản 2.1.1

3.3.2. Tiến trình làm việc

Một giải pháp tìm kiếm đầy đủ bao gồm bốn thành phần chính sau:

Trình duyệt website máy khách:

Người dùng nhập một chuỗi tìm kiếm vào chương trình máy khách(ví dụ như trình duyệt web) Sau đó chương trình sẽ gửi một truy vấn tìm kiếm đến Searchd, sau đó hiển thị các kết quả tìm kiếm.

Data source (Nguồn dữ liệu).

Đây là nơi lưu trữ dữ liệu của bạn và được truy vấn bởi các chương trình indexer. Nguồn dữ liệu được sử dụng phổ biến là MySQL. Hoặc một SQL server để lưu trữ. Nhưng đây không phải yêu cầu bắt buộc. Nó có thể làm việc tốt với các nguồn dữ liệu không phải là SQL.

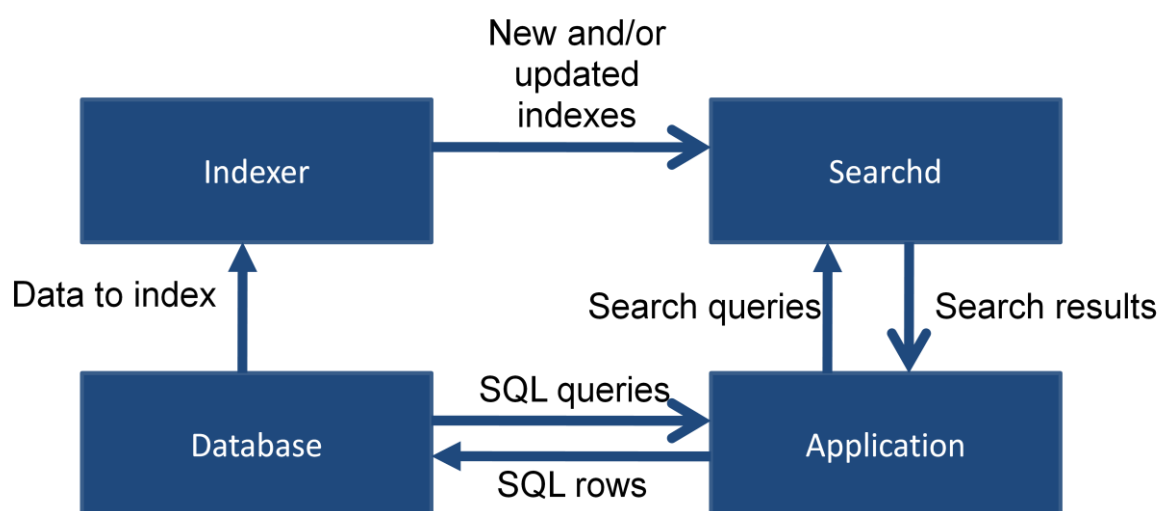
Indexer:

Chương trình này tìm nạp dữ liệu từ các nguồn dữ liệu (Data source) và tạo ra một full-text index của dữ liệu đó. Bạn sẽ cần phải chạy indexer định kỳ, tùy thuộc vào yêu cầu cụ thể của bạn.

Searchd:

Đây là một chương trình nói chuyện với your client program (ví dụ trình duyệt website của bạn), và sử dụng các full-text được xây dựng bởi indexer để nhanh chóng xử lý các truy vấn tìm kiếm. Tuy nhiên Searchd thực hiện nhiều việc hơn là chỉ tìm kiếm ví dụ như tách một chuỗi thành các từ khóa (hay còn gọi là tokenizing) hoặc xây dựng snippets(đoạn ngắn) và một số nhiệm vụ khác.

Chúng ta có thể nhìn thấy sơ đồ làm việc của Sphinx như sau:



Hình 3.1. Sơ đồ làm việc của sphinx Search

3.3.3. Lập chỉ số

Quá trình lập chỉ số:

Khi indexer khởi chạy, một kết nối với cơ sở dữ liệu được thiết lập bằng cách sử dụng các thông tin cung cấp trong tập tin cấu hình. Sau đó truy vấn chính `sql_query` được sử dụng để xác định các trường dữ liệu sẽ lập chỉ mục. Một khi việc xác định các trường cần lập chỉ mục được hoàn thành tiến hành đóng kết nối tới cơ sở dữ liệu và chuyển sang giai đoạn sắp xếp, phân loại index.

Kết nối tới nguồn dữ liệu để lập chỉ mục

Các nguồn dữ liệu dùng để lập chỉ số có thể đến từ nhiều nguồn khác nhau như cơ sở dữ liệu SQL, các file văn bản đơn thuần, các dịch vụ website, các nguồn thư điện tử...

Trong sphinx nguồn dữ liệu được xác định trong tập tin cấu hình nằm trong thư mục:

`/etc/sphinx/sphinx.conf`. Mỗi chỉ số có thể được xác định từ nhiều nguồn khác nhau.

Vì Magento sử dụng nguồn dữ liệu MySQL vì vậy Tôi sẽ trình bày chi tiết về nguồn dữ liệu này.

Khởi nguồn dữ liệu sẽ được xác định như sau:

```
source src1 {
    type          = mysql
    sql_host      = localhost
    sql_user      = test
    sql_pass      =
    sql_db        = test
    sql_port      = 3306 # optional, default is 3306
}
```

Trong đó: Type: loại dữ liệu dùng để lập chỉ số

Sql_host: Máy chủ sql.

Sql_user: Tên người sử dụng.

Sql_pass: mật khẩu.

Sql_db: tên cơ sở dữ liệu.

Sql_port: cổng kết nối mặc định là 3306.

Xác định các trường trong cơ sở dữ liệu sẽ được lập chỉ mục.

Để thực hiện việc này sphinx xây dựng các tùy trọng trong tập tin cấu hình:

Main query: Truy vấn chính chứa yêu cầu xác định các trường dữ liệu.

Pre-queries: Chạy trước truy vấn chính, Thực hiện các thiết lập cần thiết trước khi chạy truy vấn chính ví dụ thiết lập bảng mã.

Post-queries: Chạy sau truy vấn chính.

Post-index queries: Chạy khi quá trình lập chỉ số đã hoàn thành.

Truy vấn chính:

```
sql_query          = \
SELECT id, title, content, UNIX_TIMESTAMP(publish_date) \
  AS publish_date, author_id FROM posts
Pre-Queries, Post-Queries, and Post-Index Queries
```

Quá trình tạo một chỉ số sử dụng nguồn dữ liệu SQL.

Giả sử ta tạo một cơ sở dữ liệu myblog gồm các bảng sau:

Bảng post:

Trường	Loại dữ liệu
Id	INT
Title	VARCHAR(255)
Content	TEXT
Author_id	INT
Publish_date	DATETIME

Bảng 3.5. Bảng post trong cơ sở dữ liệu myblog

id	title	content	author_id	publish_date
1	Electronics For You	EFY- Electronics For You is a magazine for people ...	2	2016-01-20 13:03:14
2	What is PHP?	PHP Hypertext Preprocessor (the name is a recursi...	3	2016-01-20 13:03:58
3	Nintendo	Games that are easy to play and fun for anyone. Ni...	4	2016-01-20 13:04:44
4	Sony PlayStation - Full of life	Sony Playstation is one of the leading gaming cons...	1	2016-01-20 13:05:38
5	Namespaces in PHP 5.3	One of the most significant and welcome features a...	2	2016-01-20 13:06:30
6	Leadership Skills	Leadership skill is the key to success in any fiel...	2	2016-01-20 13:07:07
7	Ruby on Rails	RoR is a rapid web application development framewo...	4	2016-01-20 13:07:51
8	Sphinx search engine	Sphinx was created by Andrew Aksyonoff and it can ...	1	2016-01-20 13:08:37
9	PHP scripting language	PHP is a web scripting language originally created...	0	2016-04-21 07:46:35
10	Programming Languages	There are many languages available to cater any ki...	2	2016-04-21 07:48:49
11	My Life	This post is about my life which in a sense is bea...	3	2016-04-21 07:49:28
12	Life on Mars	Is there any life on mars?	4	2016-04-21 07:50:08

Hình 3.2. Dữ liệu trong bảng post

Tạo file cấu hình /etc/sphinx/sphinx.conf

```
source blog
{
    type                = mysql
    sql_host            = localhost
    sql_user            = root
    sql_pass            = ''
    sql_db              = myblog
    sql_port            = 3306 # optional, default is 3306
    sql_query           = SELECT id, title, content FROM
posts
    sql_query_info      = SELECT id, title FROM posts
WHERE id=$id
}
index post
{
    source              = blog
```

```

path                = /var/lib/sphinx/blog
docinfo              = extern
charset_type        = sbcs
}
indexer
{
    mem_limit        = 32M
}

```

(1) Tạo chỉ số sử dụng câu lệnh sau:

```
/usr/bin/indexer --config /etc/sphinx/sphinx.conf --all
```

```

collected 12 docs, 0.0 MB
sorted 0.0 Mhits, 100.0% done
total 12 docs, 1830 bytes
total 0.004 sec, 446450 bytes/sec, 2927.54 docs/sec
total 2 reads, 0.000 sec, 1.5 kb/call avg, 0.0 msec/call avg
total 6 writes, 0.000 sec, 1.0 kb/call avg, 0.0 msec/call avg

```

Hình 3.3. Kết quả tạo chỉ số

(2) Kiểm tra chỉ số vừa được tạo ra bằng cách sử dụng lệnh:

Tìm kiếm từ khóa ‘php’ trong bảng posts.

```
/usr/bin/Search php
```

```

using config file '/etc/sphinx/sphinx.conf'...
index 'post': query 'php ': returned 3 matches of 3 total in 0.000 sec

displaying matches:
1. document=5, weight=2667
   id=5
   title=Namespaces in PHP 5.3
2. document=2, weight=2646
   id=2
   title=What is PHP?
3. document=9, weight=2646
   id=9
   title=PHP scripting language

words:
1. 'php': 3 documents, 7 hits

```

Hình 3.4. Kết quả tìm kiếm từ khóa ‘php’

Thêm thuộc tính vào chỉ số:

Các chỉ số chúng ta tạo ra từ cơ sở dữ liệu myblog là đủ tốt cho tìm kiếm, nhưng nó chỉ hoạt động để tìm kiếm văn bản đầy đủ. Nếu chúng ta muốn lọc các kết quả tìm kiếm ví dụ như lọc theo tác giả hoặc lọc theo ngày. Điều đó không thể được thực hiện với các chỉ số mà chúng ta đã tạo ra trước đó. Để giải quyết vấn đề này, Sphinx cung cấp các trường lĩnh vực đặc biệt trong các chỉ số được gọi là thuộc tính.

Hãy thêm các thuộc tính vào chỉ số của bạn là `author_id` và `publish_date`.

(3) Sửa file cấu hình thêm các thuộc tính vào index.

```
sql_query      =      SELECT      id,      title,      content,
UNIX_TIMESTAMP(publish_date)\
AS publish_date, author_id FROM posts
sql_attr_unit  =  author_id
sql_attr_timestamp =publish_date
sql_query_info= SELECT id, title FROM posts WHERE id=$id
```

(4) Xây dựng lại chỉ số:

```
using config file '/etc/sphinx/sphinx.conf'...
indexing index 'post'...
collected 12 docs, 0.0 MB
sorted 0.0 Mhits, 100.0% done
total 12 docs, 1830 bytes
total 0.009 sec, 192591 bytes/sec, 1262.89 docs/sec
total 3 reads, 0.000 sec, 1.0 kb/call avg, 0.0 msec/call avg
total 9 writes, 0.000 sec, 0.7 kb/call avg, 0.0 msec/call avg
```

Hình 3.5. Kết quả xây dựng lại chỉ số

(5) Tìm kiếm từ khóa ‘php’ của được viết bởi tác giả Aditya Mooley (`author_id=2`)

```
using config file '/etc/sphinx/sphinx.conf'...
index 'post': query 'php ': returned 1 matches of 1 total in 0.000 sec

displaying matches:
1. document=5, weight=2667, publish_date=Wed Jan 20 13:06:30 2016, author_id=2
   id=5
   title=Namespaces in PHP 5.3

words:
1. 'php': 3 documents, 7 hits
```

Hình 3.6. Kết quả tìm kiếm từ khóa ‘php’ được viết bởi tác giả Aditya Mooley

Đã sửa nội dung file cấu hình sphinx.conf và thay đổi sql_query thêm hai trường mới là author_id và publish_date. Thêm hai lựa chọn mới để xác định hau thuộc tính (attributes) là sql_attr_unit, sql_attr_timestamp. Trong đó author_id là kiểu dữ liệu: unsigned integer và publish_date có kiểu dữ liệu là: TIMESTAMP

Xử lý stop words và short words

Stop words: là những từ khóa thường xuyên xảy ra, bạn có thể bỏ qua những từ này khi lập chỉ mục và tìm kiếm. Ví dụ “i, the, and, to, a, it...).

Loại bỏ stop word có thể cải thiện đáng kể thời gian lập chỉ mục và kích thước chỉ mục. Trong bảng 3-1. Là bảng so sánh sự khác nhau giữa thời gian lập chỉ mục và kích thước chỉ mục khi số stop word khác nhau bằng cách sử dụng 100000 document index.

N (Số stop words)	Thời gian tạo index (seconds)	Index size (millions of bytes)
0	12.2	73.6
10	11.1	67.2
20	10.5	63.8
30	10.4	61.4
100	9.6	51.8

Bảng 3.6. Bảng so sánh thời gian index trong các trường hợp “stop word”

Để xử lý stop word chúng ta tiến hành tạo file cấu hình sphinx.conf như sau:

```
index test1
{
path = /var/lib/sphinx/data/test1
source = src1
stopwords = /var/lib/sphinx/stopwords.txt
}
```

Tập tin stopwords.txt là tập tin văn bản bao gồm các từ khóa được bỏ qua khi làm việc với index test1.

Short words: Là lựa chọn xác định chiều dài từ khóa tối thiểu được lập chỉ mục. Từ khóa ngắn hơn giới hạn này sẽ không được lập chỉ mục.

Sử dụng `min_word_len` để xác như sau:

```
index test1
{
path = /var/lib/sphinx/data/test1
source = src1
min_word_len = 3
}
```

Ví dụ từ khóa “i” và “am” sẽ không được lập chỉ mục. Từ khóa “you” sẽ được lập chỉ mục vì có độ dài 3.

Tìm kiếm

Trong phần trước chúng ta đã tạo ra được các chỉ số, trong phần này sẽ thực hiện viết truy vấn tìm kiếm để tìm kiếm trên các chỉ số đã được tạo ra

3.3.4. Tìm kiếm

Để truy cập Searchd, sphinx có hỗ trợ một số cách sau:

- API
- SphinxSE
- SphinxQL

Hay nói cách khác bạn có thể gửi các tìm kiếm thông qua API hay qua các truy vấn của MySQL.

Triển khai API cho sphinx

Sphinx đi kèm với một số Searchd API, Tại thời điểm hiện tại Sphinx API được triển khai cho ngôn ngữ PHP, Python, Java.

Tìm kiếm sử dụng client API

Các bước để thực hiện tìm kiếm sử dụng API. Sử dụng các chỉ số và cấu hình đã tạo ra ở phần trước, bổ sung cấu hình liên quan đến Searchd và tạo một file PHP để tìm kiếm các chỉ số sử dụng Sphinx thực hiện client API cho PHP.

Các bước triển khai như sau:

(1) Thêm cấu hình Searchd vào file cấu hình:

```
Searchd {
listen          = 9312
log             = /usr/local/sphinx/var/log/Searchd.log
query_log      = /usr/local/sphinx/var/log/query.log
max_children   = 30
pid_file       = /usr/local/sphinx/var/log/Searchd.pid
}
```

Giải thích các tùy chọn:

Listen: Lựa chọn này xác định địa chỉ ip và cổng được Searchd sử dụng.

Log: Nơi chứa file log chứa tất cả các sự kiện của Searchd.

Query_log: Tên tập tin chứa các lịch sử truy vấn tìm kiếm.

max_children: Xác định số lượng tối đa các tìm kiếm chạy đồng thời. Mặc định là 0 (không giới hạn)

pid_file: Tên file của quá trình Searchd ID.

(2) Bắt đầu chạy Searchd.

```
/usr/bin/searchd -c /etc/sphinx/sphinx.conf
```

(3) Sao chép tập tin sphinxapi.php vào thư mục đang làm việc.

```
mkdir /path/to/your/webroot/sphinx
```

```
cd /path/to/your/webroot/sphinx
```

```
cp /path/to/sphinx-0.9.9/api/sphinxapi.php ./
```

(4) Tạo một kịch bản simple_search.php để sử dụng lớp PHP client API thực hiện tìm kiếm trên chỉ số đã được tạo ra trong phần trước.

```
<?php require_once('sphinxapi.php');
// Instantiate the sphinx client
$client = new SphinxClient();
// Set Search options
```

```

$client->SetServer('localhost', 9312);
$client->SetConnectTimeout(1);
$client->SetArrayResult(true);
// Query the index
$results = $client->Query('php');
// Output the matched results in raw format
print_r($results['matches']);

```

Các phương thức được sử dụng :

`SetServer($host, $port)`. Đây là phương thức xác định tên máy chủ và cổng giao tiếp của Searchd. Mặc định `host='localhost'`, và `port='9312'`.

`SetConnectTimeout($timeout)`. Xác định thời gian tối đa để kết nối tới máy chủ.

`SetArrayResult($arrayresult)`. Xác định kết quả tìm kiếm sẽ trả về dưới dạng một mảng hoặc một hàm băm (hash). Nếu (\$arrayresult là true thì kết quả tìm kiếm sẽ được trả lại giá trị dưới dạng một mảng.

`$client->Query('php')`: Thực tìm kiếm từ khóa 'php'. Nó sẽ trả về một mảng các kết quả phù hợp.

Trong sphinx cung cấp một số chế độ tìm kiếm để xác định các kết quả phù hợp với các yêu cầu của người sử dụng. Chúng được gọi là Matching Modes (chế độ phù hợp) được xác định bằng cách sử dụng phương thức `setMatchMode()`.

3.3.5. Matching Modes (Chế độ phù hợp)

Khi các chỉ số toàn văn bản được tạo ra ở phần trên. Để thực hiện việc tìm kiếm trên các chỉ số này, Sphinx có các chế độ phù hợp khác nhau để tìm kiếm kết quả.

Các chế độ phù hợp có trong sphinx là:

`SPH_MATCH_ALL`: Kết quả trả về phù hợp với tất cả các từ khóa tìm kiếm

`SPH_MATCH_ANY`: Kết quả trả về phù hợp với bất kỳ từ khóa tìm kiếm nào

SPH_MATCH_PHRASE: Kết quả trả về phải phù hợp với cả cụm từ tìm kiếm

SPH_MATCH_BOOLEAN: Chọn chế độ tìm kiếm được xây dựng bằng biểu thức boolean

SPH_MATCH_EXTENDED: Chế độ tìm kiếm được xây dựng bằng các biểu thức trong ngôn ngữ riêng của sphinx:

SPH_MATCH_EXTENDED2: Chế độ tìm kiếm mở rộng từ chế độ phù hợp từ SPH_MATCH_EXTENDED

SPH_MATCH_FULLSCAN : Ở chế độ tìm kiếm này, tất cả các dữ liệu được lập chỉ số sẽ được trả về, áp dụng trong trường hợp từ khóa tìm kiếm là rỗng

Bạn có thể thay đổi giữa các chế độ này bằng cách sử dụng phương thức SetMatchMode() trong SphinxAPI.

Cú pháp truy vấn Boolean:

Truy vấn boolean cho phép sử dụng các phương thức sau:

Phương thức	Ví dụ	Ý nghĩa
OR	hello world	Tìm kiếm từ khóa hello “hoặc” world
NOT	hello –world, hello !world	Tìm kiếm từ khóa hello và “không chứa” từ khóa world
Grouping	(hello world)	Tìm kiếm cả 2 từ khóa hello và world

Bảng 3.7. Các loại truy vấn Boolean

Cú pháp truy vấn mở rộng

Sphinx có hỗ trợ một số phương pháp truy vấn mở rộng như sau:

Phương thức	Ví dụ	Ý nghĩa
OR	hello world	Tìm kiếm từ khóa hello hoặc world
NOT	Hello –world, Hello!world	Phủ định tìm kiếm
Field Search	@title hello @body world	Giới hạn trường tìm kiếm
field position limit modifier	@body[50] hello	Giới hạn vị trí trong trường tìm kiếm
multiple-field Search	@(title,body) hello world	Giới hạn nhiều trường tìm kiếm
ignore field Search	@!title hello world	Bỏ qua việc tìm kiếm trong một trường xác định
ignore multiple-field Search	@!(title,body) hello world	Bỏ qua việc tìm kiếm trong nhiều trường
all-field Search	@* hello	Tìm kiếm trong tất cả các trường
phrase Search	"hello world"	Tìm kiếm theo cụm từ
proximity Search	"hello world"~10	Giới hạn kết quả tìm kiếm là phù hợp nếu nằm trong một khoảng cách nhất định
quorum matching	"the world is a wonderful place"/3	Kết quả là phù hợp khi chứa ít nhất N từ khóa
strict order	aaa << bbb << ccc	Kết quả được coi là phù hợp nếu xuất hiện đúng thứ tự

Phương thức	Ví dụ	Ý nghĩa
field-start and field-end modifier	^hello world\$	Xác định kết quả tìm kiếm bắt đầu và kết thúc là từ khóa tìm kiếm
ZONE limit	ZONE:(h3,h4)	Giới hạn khu vực tìm kiếm nằm trong thẻ html

Bảng 3.8. Các truy vấn mở rộng

Lọc kết quả tìm kiếm:

Trong một số trường hợp cụ thể tìm kiếm toàn văn bản không đáp ứng được hoàn toàn yêu cầu tìm kiếm. Ví dụ tìm kiếm các bài viết được biết bởi một tác giả, hoặc tìm kiếm các mặt hàng có giá nằm trong một giới hạn nhất định. Để giải quyết vấn đề này Sphinx có chứa các thuộc tính để thực hiện lọc trên đó.

Ví dụ từ chỉ số được tạo ra ở trên lọc theo tác giả:

```
$client->SetFilter('author_id', array(2, 4));
```

3.3.6. Xếp hạng kết quả tìm kiếm

Tổng quan về ranking (xếp hạng)

Ranking (hay còn gọi là weighting-trọng số) của các kết quả tìm kiếm có thể được định nghĩa là một quá trình tính một mức độ phù hợp (hay còn gọi là trọng số) cho mỗi tài liệu phù hợp nhất định liên quan đến một truy vấn. Kết quả tìm kiếm có thể được sắp xếp dựa trên con số này hoặc một thông số bổ sung.

Xếp hạng trong Sphinx là quá trình cấu hình. Nó kèm theo một khái niệm gọi là ranker. Một Ranker có thể được định nghĩa là một hàm có đầu vào là document và query và đầu ra là một giá trị liên quan. Sphinx sử dụng một thuật toán cụ thể để xác định ranker.

Trước đây, chức năng ranking phụ thuộc vào matching mode. Vì vậy trong các matching mode:

- SPH_MATCH_ALL.
- SPH_MATCH_ANY.
- SPH_MATCH_PHRASE.

- SPH_MATCH_BOOLEAN.

Bạn không thể chọn ranker. Bạn chỉ có thể chọn ranker trong chế độ SPH_MATCH_EXTENDED. Bạn có thể chọn RANKER bằng cách sử dụng SetRankingMode() với SphinxAPI, hoặc sử dụng OPTION trong câu lệnh SELECT khi sử dụng SphinxQL.

3.3.7. Các Rankers được xây dựng sẵn

Các ranker hiện có là:

SPH_RANK_PROXIMITY_BM25: Chế độ xếp hạng mặc định, kết hợp cả hai cụm từ gần và xếp hạng BM25.

SPH_RANK_BM25: Chế độ xếp hạng thống kê, sử dụng thuật toán xếp hạng BM25 (tương tự các máy tìm kiếm full-text khác). Chế độ xếp hạng này là nhanh nhưng có thể dẫn đến kém hiệu quả nếu truy vấn chứa hơn một từ khóa.

SPH_RANK_NONE: Đây là chế độ không xếp hạng. Đây là chế độ nhanh nhất. Chúng ta không gán trọng số cho các bản ghi phù hợp. Điều này đôi khi được gọi là tìm kiếm boolean mà chỉ phù hợp với các tài liệu nhưng không xếp hạng chúng.

SPH_RANK_WORDCOUNT: Xếp hạng các từ khóa bằng cách đếm số lần xuất hiện. Ranker này tính số lần xảy ra từ khóa cho mỗi field, sau đó nhân chúng bởi trọng trường, và tổng các giá trị kết quả.

SPH_RANK_PROXIMITY: Thêm vào trong phiên bản 0.9.9-rc1, trả về giá trị cụm từ gần thô như một kết quả. Chế độ này được sử dụng trong nội bộ để mô phỏng với truy vấn SPH_MATCH_ALL.

SPH_RANK_MATCHANY: Thêm vào trong phiên bản 0.9.9-rc1, trả về rank như nó được tính toán ở chế độ SPH_MATCH_ANY trước đó, và là nội bộ được sử dụng để mô phỏng các truy vấn SPH_MATCH_ANY.

SPH_RANK_FIELDMASK, thêm vào trong phiên bản 0.9.9-RC2, trả về một mặt nạ 32-bit với bit thứ N tương ứng với lĩnh vực fulltext N-th, đánh số từ 0. Các bit sẽ chỉ được thiết lập khi các lĩnh vực tương ứng có bất cứ sự thỏa mãn từ khóa truy vấn.

SPH_RANK_SPH04: Thêm vào trong phiên bản 1.10 beta, nói chung là dựa trên ranker SPH_RANK_PROXIMITY_BM25 mặc định, nhưng làm tăng thêm các bản ghi phù hợp khi chúng xuất hiện trong đầu hoặc cuối của một trường văn bản. Như vậy, nếu một field tương đương với các truy vấn chính xác, SPH04 nên rank nó cao hơn một lĩnh vực có chứa các truy vấn chính xác nhưng không bằng nó. (ví dụ, truy vấn là “Hyde Park”, một tài liệu mang tên “Hyde Park” nên được xếp hạng cao hơn so với một tựa đề “Hyde Park, London” hoặc “The Hyde Park Cafe”).

SPH_RANK_EXPR: Được thêm vào trong phiên bản 2.0.2 –beta, cho phép bạn chỉ định các công thức xếp hạng trong thời gian chạy.

Cú pháp để cấu hình ranker là:

Trong SphinxAPI

```
$client->SetRankingMode ( SPH_RANK_SPH04 );
```

Trong SphinxQL

```
mysql_query ( "SELECT ... OPTION ranker=sph04" );
```

Chế độ phù hợp lựa chọn một ranker mặc định như sau:

- SPH_MATCH_ALL sử dụng SPH_RANK_PROXIMITY ranker.
- SPH_MATCH_ANY sử dụng SPH_RANK_MATCHANY ranker.
- SPH_MATCH_PHRASE sử dụng SPH_RANK_PROXIMITY ranker.
- SPH_MATCH_BOOLEAN sử dụng SPH_RANK_NONE ranker.

3.3.8. Sắp xếp kết quả tìm kiếm

Có đôi lúc bạn muốn sắp xếp các kết quả tìm kiếm theo giá trị khác sự sắp xếp theo sự liên quan mặc định. Sphinx có các chế độ sắp xếp như sau:

Chế độ sắp xếp	Ý nghĩa
SPH_SORT_RELEVANCE:	Sắp xếp theo độ liên quan theo thứ tự giảm dần (phù hợp tốt nhất nằm đầu tiên)

SPH_SORT_ATTR_DESC:	Sắp xếp theo thứ tự giảm dần của thuộc tính (thuộc tính có giá trị lớn nằm đầu tiên);
SPH_SORT_ATTR_ASC:	Sắp xếp theo thứ tự tăng dần của thuộc tính (thuộc tính có giá trị nhỏ nằm đầu tiên)
SPH_SORT_TIME_SEGMENTS:	Sắp xếp theo các loại của thời gian theo mức độ giảm dần và sau đó là sự liên quan theo mức độ giảm dần.
SPH_SORT_EXTENDED:	Sắp xếp bằng cách kết hợp trong SQL của cột theo thứ tự tăng hoặc giảm.
SPH_SORT_EXPR	Sắp xếp theo một biểu thức số học.

Bảng 3.9. Các loại sắp xếp trong Sphinx

Chế độ sắp xếp SPH_SORT_TIME_SEGMENTS:

Trong chế độ này, giá trị của các thuộc tính được phân thành các khoảng thời gian. Và sắp xếp theo khoảng thời gian này.

Các khoảng thời gian như sau: Last hour, Last day, Last week, Last month, Last 3 month. Hoặc 1 số khoảng thời gian khác.

Chế độ sắp xếp SPH_SORT_EXTENDED

Trong chế độ SPH_SORT_EXTENDED bạn có thể chỉ định một biểu thức SQL để sắp xếp. Ví dụ như sau:

@relevance DESC, price ASC, @id DESC

Chế độ sắp xếp SPH_SORT_EXPR

Khi sử dụng chế độ sắp xếp này cho phép bạn sắp xếp kết quả tìm kiếm bằng các **biểu** thức số học ví dụ:

```
$cl->SetSortMode ( SPH_SORT_EXPR,
    "@weight + ( user_karma + ln(pageviews) ) * 0.1" );
```

3.3.9. Nhóm kết quả tìm kiếm

Sphinx cung cấp các chế độ group khác nhau. Chúng được kích hoạt bằng cách gọi phương thức SetGroupBy() trong API. Các chức năng được xây dựng sẵn trong Sphinx là:

Phương thức	Ý nghĩa:
SPH_GROUPBY_DAY	Nhóm theo ngày
SPH_GROUPBY_WEEK	Nhóm theo tuần
SPH_GROUPBY_MONTH	Nhóm theo tháng
SPH_GROUPBY_YEAR	Nhóm theo năm
SPH_GROUPBY_ATTR	Sử dụng giá trị attribute để nhóm

Bảng 3.10. Các phương thức nhóm kết quả tìm kiếm

3.4. Kết luận chương 3

Nội dung trong chương trình bày đặc điểm của công cụ tìm kiếm toàn văn bản sphinx Search, các tính năng chính và cơ chế hoạt động.

Nội dung của chương tiếp theo sẽ trình bày việc triển khai tích hợp Sphinx Search vào Magento để nâng cấp tính năng tìm kiếm của Magento.

Chương 4. TRIỂN KHAI NÂNG CẤP TÍNH NĂNG TÌM KIẾM CHO MAGENTO BẰNG CÁCH SỬ DỤNG SPHINX

4.1. Yêu cầu

Bài toán nâng cấp tính năng tìm kiếm trên Magento sử dụng công cụ Sphinx Search được giải quyết bằng cách xây dựng phần mở rộng theo chuẩn của Magento để tích hợp vào mã nguồn. Phần mở rộng này sẽ thay thế tìm kiếm mặc định của Magento tạo ra một tìm kiếm thân thiện và mạnh mẽ. Đáp ứng một số yêu cầu cụ thể sau:

- Trả về kết quả tìm kiếm nhanh chóng.
- Trả về các kết quả liên quan phù hợp nhất.

Khi một người dùng thực hiện tìm kiếm trong gian hàng trực tuyến của bạn, Extension Magento Sphinx Search sẽ truy vấn tìm kiếm trong sphinx để trả về kết quả cho người dùng.

4.2. Phân tích thiết kế

4.2.1. Chức năng lập chỉ số từ bảng dữ liệu của Magento

- Trong cơ sở dữ liệu của Magento xây dựng một bảng dữ liệu mới. Đặt tên bảng là `sphinx_catalogSearch_fulltext` Bảng này chứa các trường dữ liệu cần lập chỉ số.

- Cài đặt công cụ Sphinx Search trên máy chủ và cấu hình trong file `sphinx.conf`.

- Chạy chức năng index trong sphinx để tạo chỉ số từ bảng *dữ liệu vừa tạo ra ở trên*.

4.2.2. Chức năng lựa chọn các chế độ tìm kiếm

Sau khi các chỉ số được tạo ra bằng sphinx ta tiến hành lựa chọn các chế độ tìm kiếm phù hợp với yêu cầu bao gồm:

- Chọn chế độ tìm kiếm.
- Đánh trong số các trường để ưu tiên hiển thị kết quả.
- Chọn chế độ xếp hạng tìm kiếm.
- Chọn chế độ liên quan của các từ khóa tìm kiếm.

4.3. Triển khai xây dựng

4.3.1. Tạo bảng chứa dữ liệu các trường sẽ được lập chỉ mục

Xây dựng bảng sphinx_catalogSearch_fulltext để lưu các dữ liệu cần index:

Gồm các trường như sau:

Tên	Loại dữ liệu
product_id	Int(10)
Store_id	Smallint(5)
Name	Varchar(255)
Name_attributes	longtext
Category	Varchar(255)
Data_index	longtext



Bảng 4.1. Các trường dữ liệu trong bảng sphinx_catalogSearch_fulltext

Tạo bảng trên bằng cách sử dụng code theo chuẩn xây dựng extension của Magento như sau:

```
<?php
$installer = $this;
$installer->startSetup();
$installer->run("
CREATE TABLE IF NOT EXISTS {$this->
>getTable('sphinx_catalogSearch_fulltext')} (
    `product_id` int(10) unsigned NOT NULL,
    `store_id` smallint(5) unsigned NOT NULL,
    `name` varchar(255) NOT NULL,
    `name_attributes` longtext NOT NULL,
    `category` varchar(255) DEFAULT NULL,
    `data_index` longtext NOT NULL,
    PRIMARY KEY (`product_id`,`store_id`),
    FULLTEXT KEY `data_index` (`data_index`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
");
$installer->endSetup();
```

Kết quả xây dựng được bảng dữ liệu như sau:

#	Name	Type	Collation	Attributes	Null	Default
1	<u>product_id</u> 	int(10)		UNSIGNED	No	None
2	<u>store_id</u> 	smallint(6)			No	None
3	<u>name</u>	varchar(255)	utf8_general_ci		No	None
4	<u>name_attributes</u>	text	utf8_general_ci		No	None
5	<u>category</u>	varchar(255)	utf8_general_ci		Yes	NULL
6	<u>data_index</u>	text	utf8_general_ci		No	None
7	<u>sku</u>	varchar(255)	utf8_general_ci		No	None

Hình 4.1. Kết quả xây dựng bảng sphinx_catalogSearch_fulltext

Cập nhật dữ liệu vào bảng dữ liệu vừa xây dựng:

Sau khi xây dựng bảng sphinx_catalogSearch_fulltext để chứa các dữ liệu cần index.

Cần chèn dữ liệu trong cơ sở dữ liệu của Magento vào bảng này.

Code chèn dữ liệu vào bảng:

```
public function saveEntityIndexes($storeId, $entityIndexes,
    $entity = 'product')
    {
        $adapter = $this->_getWriteAdapter();
        $data      = array();
        $storeId = (int)$storeId;
        foreach ($entityIndexes as $entityId => &$index) {
            $data[] = array(
                'product_id'      =>
(int)$entityId,
                'store_id'       => $storeId,
```

```

        'data_index' =>
$index['data_index'],

        'name' =>
$index['name'],

        'name_attributes' =>
$index['name_attributes'],

        'category' =>
$index['category'],

    );

    }

    if ($data) {
        $adapter->
>insertOnDuplicate('sphinx_catalogSearch_fulltext', $data,
array('data_index', 'name', 'name_attributes', 'category'));
    }

    return $this;
}

```

4.3.2. Cài đặt Sphinx trên Server và cấu hình

- Cài đặt sphinx Search trên server
- Cài đặt file cấu hình.
- Cấu hình sphinx để lập chỉ số dựa vào nguồn dữ liệu vừa tạo ra.
- File sphinx.config.

```

source Magento_fulltext
{
    type                = mysql
    sql_host             = localhost
    sql_user             = root
    sql_pass             =
    sql_db               = demo.vitrys
}

```



```

        sql_port                = 3306 # optional, default is
3306

        sql_query_pre  = SET NAMES utf8

        sql_query      = SELECT product_id, name,
name_attributes, category, data_index FROM
sphinx_catalogSearch_fulltext

        sql_query_info = SELECT product_id, name,
name_attributes, category, data_index FROM
sphinx_catalogSearch_fulltext WHERE product_id=$id
}

index fulltext
{
        source                = Magento_fulltext

        path                  =
/var/lib/sphinx/Magento.sphinx.index

        docinfo               = extern

        charset_type         = sbcs
}

indexer
{
        mem_limit             = 128M
}

Searchd
{
        listen                = 9312

        listen                = 9306:mysql41

        log                   =
/var/log/sphinx/Searchd1.log

        query_log             = /var/log/sphinx/query1.log

        read_timeout         = 5

        max_children         = 30

        pid_file              = /var/run/sphinx/Searchd1.pid
}

```

```

max_matches          = 1000
seamless_rotate      = 1
preopen_indexes      = 1
unlink_old           = 1
workers              = threads # for RT to work
binlog_path          = /var/lib/sphinx/
}

```

4.3.3. Lập chỉ số dữ liệu

Chạy indexing:

```

[root@Tuan4 ~]# /usr/bin/indexer --all
Sphinx 2.0.9-id64-release (rel20-r4115)
Copyright (c) 2001-2013, Andrew Aksyonoff
Copyright (c) 2008-2013, Sphinx Technologies Inc (http://sphinxsearch.com)

using config file '/etc/sphinx/sphinx.conf'...
indexing index 'fulltext'...
WARNING: Attribute count is 0: switching to none docinfo
collected 270 docs, 0.4 MB
sorted 0.1 Mhits, 100.0% done
total 270 docs, 401817 bytes
total 0.064 sec, 6193710 bytes/sec, 4161.84 docs/sec
total 2 reads, 0.000 sec, 30.2 kb/call avg, 0.0 msec/call avg
total 6 writes, 0.000 sec, 21.8 kb/call avg, 0.0 msec/call avg

```

Hình 4.2. Lập chỉ số trong Sphinx

4.3.4. Lựa chọn các chế độ tìm kiếm

Cài đặt chế độ tìm kiếm:

Đánh trọng số các trường tìm kiếm để ưu tiên hiện thị kết quả.

```

'name'=7
'data_index'=3
'category'=1
'name_attributes'=1

```

Lựa chọn chế độ tìm kiếm: SPH_MATCH_EXTENDED2

Lựa chọn chế độ xếp hạng: SPH_RANK_SPH04

```
public function getSphinxAdapter() {
    require_once(Mage::getBaseDir('lib') .
DIRECTORY_SEPARATOR . 'sphinxapi.php');

    $sphinx = new SphinxClient();

    $host =
Mage::getStoreConfig('sphinxSearch/server/host');
    $port =
Mage::getStoreConfig('sphinxSearch/server/port');
    if (empty($host)) {
        return $sphinx;
    }
    if (empty($port)) {
        $port = 9312;
    }
    $sphinx->SetServer($host, $port);
    $sphinx->SetMatchMode(SPH_MATCH_EXTENDED2);
    $sphinx->setFieldWeights(array(
        'name' => 7,
        'category' => 1,
        'name_attributes' => 1,
        'data_index' => 3
    ));
    $sphinx->setLimits(0, 200, 1000, 5000);

    $sphinx->SetRankingMode(SPH_RANK_SPH04, "");

    return $sphinx;
}
```

Dựa vào các cài đặt trên để hiển thị kết quả truy vấn phù hợp.

4.4. Kết luận chương 4:

Chương này của luận văn đã sử dụng các phần nội dung lý thuyết tìm hiểu ở phần trên để xây dựng một chương trình tích hợp vào Magento để nâng cấp tính năng tìm kiếm mặc định sẵn có dựa trên nền tảng công cụ tìm kiếm sphinx Search đáp ứng các yêu cầu tìm kiếm của khách hàng trên các website thương mại điện tử sử dụng Magento.

Chương 5. ĐÁNH GIÁ KẾT QUẢ

5.1. Kết quả xây dựng

a) Giao diện trang quản trị:

Chứa các lựa chọn:

- Kích hoạt extension.
- Chạy index
- Cấu hình host, port
- Chọn thêm trường để index

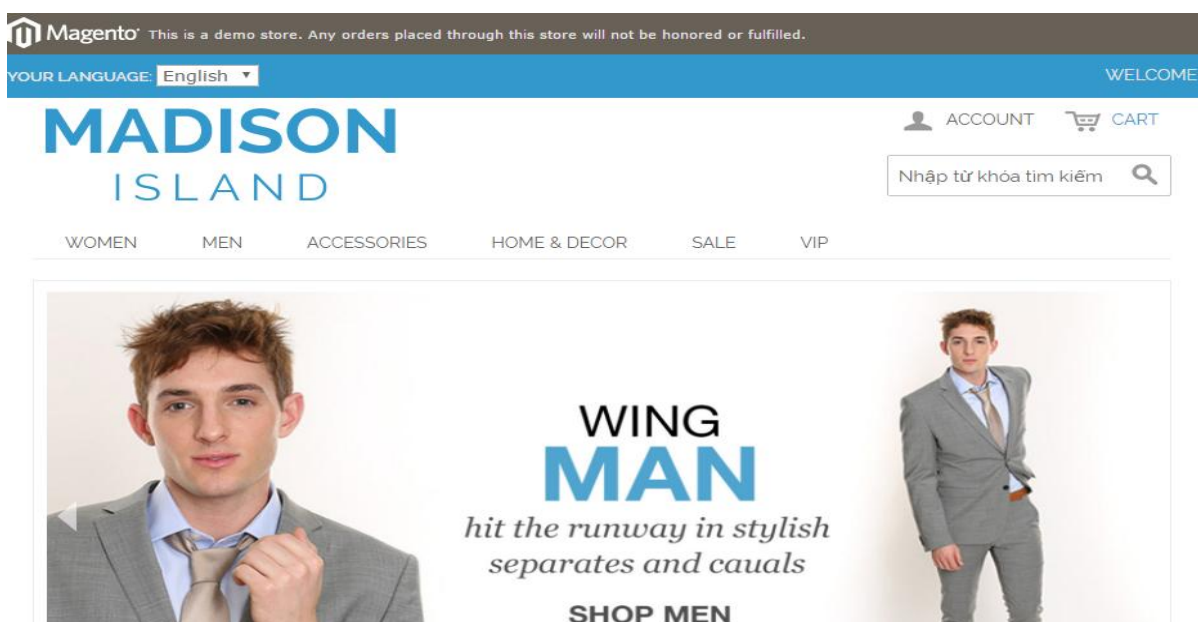
The screenshot displays two sections of the Magento Admin interface:

- Activate Sphinx Search:**
 - Enable Search:** Set to 'Yes' (with a dropdown arrow). Below it, a small triangle icon indicates 'Use sphinx search in frontend'. A '[STORE VIEW]' link is to the right.
 - Enable Index:** Set to 'Yes' (with a dropdown arrow). Below it, a small triangle icon indicates 'Index when "Catalog Search" is reindexed'. A '[STORE VIEW]' link is to the right.
- Server Configuration:**
 - Host:** Input field containing 'localhost'. A '[STORE VIEW]' link is to the right.
 - Port:** Input field containing '9312'. A '[STORE VIEW]' link is to the right.
 - Index:** Empty input field. A '[STORE VIEW]' link is to the right.

Hình 5.1. Giao diện quản trị

b) Giao diện trang tìm kiếm

Bao gồm hộp thoại nhập từ khóa tìm kiếm.




Hình 5.2. Giao diện tìm kiếm

c) Giao diện trang kết quả tìm kiếm

SEARCH RESULTS FOR 'JEAN'

SORT BY: Relevance ▾ ↓ VIEW AS: 🗪 ☰

4 Item(s) SHOW: 12 ▾



TRIBECA SKINNY JEAN


■

185,00 đ

[VIEW DETAILS](#)

[Add to Wishlist](#)

[Add to Compare](#)



THE ESSENTIAL BOOT CUT JEAN


■

140,00 đ

[VIEW DETAILS](#)

[Add to Wishlist](#)

[Add to Compare](#)



DUMBO BOYFRIEND JEAN

■

210,00 đ

[VIEW DETAILS](#)

[Add to Wishlist](#)

[Add to Compare](#)

Hình 5.3. Giao diện trang kết quả tìm kiếm

5.2. Đánh giá về kết quả xây dựng

Từ khóa tìm kiếm	Sphinx Search	Search default in Magento
ANN ANKLE BOOT	Trả về 01 mặt hàng phù hợp. 1.ANN ANKLE BOOT	Trả về 6 mặt hàng 1.ANN ANKLE BOOT 2.THE ESSENTIAL BOOT CUT JEAN. 3.FLAT FRONT TROUSER. 4. BARCLAY D'ORSAY PUMP, NUDE. 5.A TALE OF TWO CITIES.



Từ khóa tìm kiếm	Sphinx Search	Search default in Magento
		6.FIRE [KALIMA REMIX] BY UNANNOUNCED GUEST
LINEN BLAZER	Trả về 01 mặt hàng LINEN BLAZER	Trả về 04 mặt hàng LINEN BLAZER STRETCH COTTON BLAZER ELIZABETH KNIT TOP SULLIVAN SPORT COAT











Bảng 5.1. Kết quả tìm kiếm từ khóa



Từ khóa tìm kiếm: ANN ANKLE BOOT

Chế độ Search like: Kết quả trả về 6 mặt hàng:

SEARCH RESULTS FOR 'ANN ANKLE BOOT'

SORT BY: **Relevance** ↓ VIEW AS:   **6 Item(s)** SHOW: **12** ▼

		
<p>THE ESSENTIAL BOOT CUT JEAN</p> <p></p> <p>140,00 đ</p>	<p>FLAT FRONT TROUSER</p> <p></p> <p>195,00 đ</p>	<p>BARCLAY D'ORSAY PUMP, NUDE</p> <p></p> <p>390,00 đ</p>
<p>VIEW DETAILS</p> <p>Add to Wishlist</p> <p>Add to Compare</p>	<p>VIEW DETAILS</p> <p>Add to Wishlist</p> <p>Add to Compare</p>	<p>VIEW DETAILS</p> <p>Add to Wishlist</p> <p>Add to Compare</p>
		
<p>ANN ANKLE BOOT</p> <p></p> <p>470,00 đ</p>	<p>A TALE OF TWO CITIES</p> <p>10,00 đ</p>	<p>FIRE [KALIMA REMIX] BY UNANNOUNCED GUEST</p> <p>2,00 đ</p>
<p>VIEW DETAILS</p> <p>Add to Wishlist</p> <p>Add to Compare</p>	<p>VIEW DETAILS</p> <p>Add to Wishlist</p> <p>Add to Compare</p>	<p>VIEW DETAILS</p> <p>Add to Wishlist</p> <p>Add to Compare</p>


SORT BY: **Relevance** ↓ VIEW AS:   **6 Item(s)** SHOW: **12** ▼

Hình 5.4. Kết quả search like

Chế độ full-text MySQL: Kết quả trả về 4 mặt hàng

SEARCH RESULTS FOR 'ANN ANKLE BOOT'

SORT BY: Relevance ▼
VIEW AS: ☐ ☰
4 Item(s) SHOW: 12 ▼




ANN ANKLE BOOT

470,00 đ

VIEW DETAILS

[Add to Wishlist](#)
[Add to Compare](#)




THE ESSENTIAL BOOT CUT
JEAN

140,00 đ

VIEW DETAILS

[Add to Wishlist](#)
[Add to Compare](#)




FLAT FRONT TROUSER

195,00 đ

VIEW DETAILS

[Add to Wishlist](#)
[Add to Compare](#)



BARCLAY D'ORSAY PUMP,
NUDE

390,00 đ

VIEW DETAILS

[Add to Wishlist](#)
[Add to Compare](#)







SORT BY: Relevance ▼
VIEW AS: ☐ ☰
4 Item(s) SHOW: 12 ▼

Hình 5.5. Kết quả tìm kiếm fulltext search trong mysql

Kết hợp like và full-text MySQL: Kết quả trả về 6 mặt hàng

SEARCH RESULTS FOR 'ANN ANKLE BOOT'

SORT BY: ↓ VIEW AS: 6 Item(s) SHOW:

		
<p>ANN ANKLE BOOT <input type="checkbox"/></p>	<p>THE ESSENTIAL BOOT CUT JEAN <input type="checkbox"/></p>	<p>FLAT FRONT TROUSER <input type="checkbox"/></p>
<p>470,00 đ</p>	<p>140,00 đ</p>	<p>195,00 đ</p>
<p>VIEW DETAILS</p>	<p>VIEW DETAILS</p>	<p>VIEW DETAILS</p>
<p>Add to Wishlist</p>	<p>Add to Wishlist</p>	<p>Add to Wishlist</p>
<p>Add to Compare</p>	<p>Add to Compare</p>	<p>Add to Compare</p>
		
<p>BARCLAY D'ORSAY PUMP, NUDE <input type="checkbox"/></p>	<p>A TALE OF TWO CITIES <input type="checkbox"/></p>	<p>FIRE [KALIMA REMIX] BY UNANNOUNCED GUEST <input type="checkbox"/></p>
<p>390,00 đ</p>	<p>10,00 đ</p>	<p>2,00 đ</p>
<p>VIEW DETAILS</p>	<p>VIEW DETAILS</p>	<p>VIEW DETAILS</p>
<p>Add to Wishlist</p>	<p>Add to Wishlist</p>	<p>Add to Wishlist</p>
<p>Add to Compare</p>	<p>Add to Compare</p>	<p>Add to Compare</p>


SORT BY: ↓ VIEW AS: 6 Item(s) SHOW:

Hình 5.6. Kết quả lựa chọn chế độ kết hợp

Chế độ sphinx Search: Kết quả trả về 1 mặt hàng.

SEARCH RESULTS FOR ANN ANKLE BOOT

SORT BY: Relevance ▾ ↓ VIEW AS: ⊞ ☰ 1 Item(s) SHOW: 12 ▾



ANN ANKLE BOOT

■

470,00 đ

[VIEW DETAILS](#)

[Add to Wishlist](#)

[Add to Compare](#)

SORT BY: Relevance ▾ ↓ VIEW AS: ⊞ ☰ 1 Item(s) SHOW: 12 ▾

Hình 5.7. Kết quả tìm kiếm trên Sphinx Search

5.3. Kết luận chương.

Nội dung của chương thể hiện sự so sánh các kết quả tìm kiếm giữa tìm kiếm mặc định trong Magento và tìm kiếm sử dụng Sphinx qua đó thấy được hiệu quả của việc nâng cấp tính năng tìm kiếm của phần mềm thương mại điện tử Magento.

KẾT LUẬN

Tìm kiếm trên một gian hàng trực tuyến là yêu cầu cấp bách của người sử dụng. Tuy nhiên chức năng tìm kiếm có sẵn trong Magento có nhiều hạn chế về chất lượng và thời gian tìm kiếm chính vì vậy đề tài luận văn “**Nâng cấp tính năng tìm kiếm của phần mềm thương mại điện tử Magento sử dụng Sphinx Search**” đã giải quyết yêu cầu thực tế trên, luận văn tập trung tìm hiểu, nghiên cứu và đạt được một số kết quả sau:

Nghiên cứu tài liệu để trình bày các vấn đề sau:

- Trình bày kiến thức cơ bản về hệ thống mã nguồn mở Magento, các chức năng chính, ưu điểm và nhược điểm của hệ thống.

- Tìm hiểu về chức năng tìm kiếm trên hệ thống mã nguồn mở Magento, hạn chế của chức năng tìm kiếm mặc định và các giải pháp nâng cấp tính năng tìm kiếm mặc định này.

- Nghiên cứu các đặc điểm của công cụ tìm kiếm toàn văn bản Sphinx: Các chức năng chính, cơ chế làm việc, cách tích hợp vào Magento.

- Nghiên cứu cách tích hợp sphinx Search vào Magento để nâng cấp tính năng tìm kiếm.

Hướng phát triển tiếp theo của đề tài sẽ là nghiên cứu các hướng sau:

Tối ưu hóa chương trình đã xây dựng, cải thiện tốc độ và chức năng lập chỉ số, cải thiện kết quả tìm kiếm về thời gian và hiệu quả.

TÀI LIỆU THAM KHẢO

1. Tài liệu Tiếng Việt

[1] Nguyễn Kim Quy (2006), “*Một số vấn đề liên quan đến SEARCH ENGINE, ứng dụng trong TEXT*”, Luận văn Thạc sỹ công nghệ thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, tr. 42-57.

[2] Nguyễn Thị Hoàng Yên (2015), “*Cải thiện tốc độ hoạt động của Website thương mại điện tử sử dụng mã nguồn mở Magento thông qua kỹ thuật VARNISH*”, Luận văn Thạc sỹ công nghệ thông tin, Trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội, tr. 1-23

2. Tài liệu tiếng Anh

[3] Abbas Ali (2011), “*Sphinx Search Beginner’s Guide*”, Packt Publishing, Birmingham, B27 6PA, UK.

[4] Andrew Aksyonoff (2011), “*Introduction to Search with Sphinx*”, O’Reilly, United States of America.

3. Website

[5] <http://www.vijaywebsolutions.com/advantages-disadvantages-of-Magento-ecommerce-platform/>

[6] <http://www.shoppimon.com/content/2015/01/optimizing-Magento-performance-solutions-help-speed-shop/>

[7] https://en.wikipedia.org/wiki/Apache_Solr

[8] <https://www.elastic.co>

[9] <http://www.sooqr.com/>

[10] https://en.wikipedia.org/wiki/Full_text_Search

[11] <https://www.mysql.com/>

[12] https://en.wikipedia.org/wiki/Suffix_tree

[13] Digital Ocean. Tham khảo trang

<http://www.digitalocean.com/>

[14] Magento. Được tham khảo từ trang Magento

<http://Magento.com>

[15] Magento. Giới thiệu về Magento được tham khảo từ trang

[http://www.Magentoecommerce.com/knowledge-base/entry/Magento-for-dev-part-1-introduction-to-Magento.](http://www.Magentoecommerce.com/knowledge-base/entry/Magento-for-dev-part-1-introduction-to-Magento)

[16] Magento. Biểu đồ thị phần của Magento

<http://blog.aheadworks.com/2014/10/e-commerce-platforms-share-investigation-october-2014/>

PHỤ LỤC 1

GIẢI THÍCH TỪ NGỮ

Extension: Là phần mở rộng được thêm vào Magento để thực hiện các chức năng riêng biệt khác nhau:

LEMP Server (L-Linux, E-Nginx, M- Mysql, P-php): Đề cập đến một hệ điều hành dựa trên Linux, máy chủ web Nginx (phát âm “engine x”), máy chủ cơ sở dữ liệu MySQL, Ngôn ngữ lập trình PHP.