

CÁC GIẢI PHÁP KẾT NỐI WEB VỚI CƠ SỞ DỮ LIỆU

ĐOÀN VĂN BAN, HỒ VĂN HUONG

Abstract. A part from putting information into the network under the available Web pages, the putting of informations in databases into the network under the dynamic Web pages is very necessary. Since it help us to supply much more informations from databases. To that end, we must find the solution to the issue of database access for Web service. This paper presents a number of solution to linking web with databases and provides several comment, evaluation of linking method in the designing, application and recommend a solution to development of a software product to serve the purpose of commerce electronic.

1. MỞ ĐẦU

Công nghệ Web ngày nay không chỉ giới hạn phạm vi tồn tại trên mạng Internet mà đã trở thành công cụ giao tiếp lý tưởng của các hệ thống thông tin, đặc biệt là các ứng dụng xây dựng hệ thống thông tin trên mạng cục bộ, mạng diện rộng, mạng Internet [10, 11].

Để đưa thông tin trong các cơ sở dữ liệu (CSDL) lên mạng dưới dạng các trang Web, chúng ta phải giải quyết bài toán truy nhập CSDL cho dịch vụ Web. Cụ thể là chúng ta phải nghiên cứu các phương pháp kỹ thuật kết nối nhiều hệ thống CSDL khác nhau nhằm đáp ứng những yêu cầu trao đổi và xử lý thông tin trong các hệ thống phân tán.

Trong bài này chúng tôi trình bày một số phương pháp kết nối Web với CSDL, đưa ra một số nhận xét, đánh giá về các phương pháp kết nối trong thiết kế và ứng dụng, và đề xuất một giải pháp xây dựng một sản phẩm phần mềm phục vụ cho mục đích thương mại điện tử.

2. MÔ HÌNH KẾT NỐI WEB VỚI CSDL

Mô hình Client/Server truyền thống hai tầng (2-tiers) nhường chỗ cho mô hình Client/Server 3 hay nhiều tầng hơn (n -tiers), vì mô hình nhiều tầng thích hợp trong việc xây dựng hệ thống thông tin cho mạng cục bộ, mạng diện rộng và Internet. Trong mô hình 3 tầng, hệ thống thông tin được xây dựng trên Web Server và có thể truy nhập ở mọi nơi nhờ trình duyệt Web.

Hình 1 giới thiệu mô hình Client/Server 3 tầng: tầng Client, tầng giữa (Middle tier) và tầng nguồn dữ liệu (Data source tier).

Tầng 1 là tầng Client trên máy tính gọi là máy trạm thông qua trình duyệt Web hay trình ứng dụng người sử dụng thao tác, xử lý và khai thác dữ liệu.

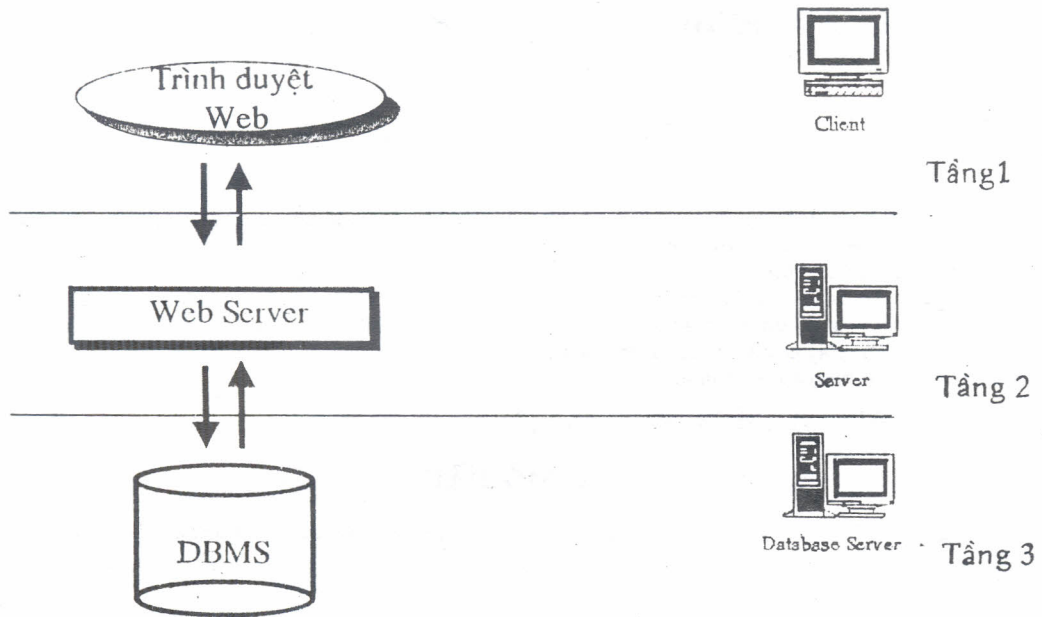
Tầng 2 là máy tính với hệ điều hành chứa toàn bộ chương trình quản trị, tổ chức và khai thác CSDL.

Tầng 3 là hệ quản trị CSDL như Informix, SQL Server, Oracle, v.v...

Thực tế các ứng dụng xây dựng hệ thống thông tin trên mạng cục bộ mạng diện rộng và mạng Internet thường áp dụng mô hình 3 tầng nhờ khả năng và tính ưu việt của chúng.

3. CÁC PHƯƠNG PHÁP KẾT NỐI WEB VỚI CSDL

Dịch vụ Web là dịch vụ thông tin mới và có ưu điểm là nó có thể chuyển tải thông tin đến một khối lượng người sử dụng rất lớn, phong phú, đa dạng và dễ sử dụng. Việc đưa thông tin trong các CSDL lên mạng dưới dạng các trang Web tĩnh hay động là một nhu cầu thiết yếu, giúp chúng ta cung cấp thêm cho người dùng rất nhiều thông tin hữu ích trong các CSDL và dựa vào đó chúng ta có thể xây dựng những ứng dụng CSDL rất đa dạng trên nền dịch vụ Web [10, 11].



Hình 1. Mô hình Client/Server 3 tầng

1. Trình duyệt Web gửi yêu cầu cho Web Server,
2. Web Server truy cập CSDL,
3. Hệ quản trị CSDL gửi kết quả về cho Web Server,
4. Web Server trả kết quả về cho trình duyệt Web.

Mô hình hoạt động của thông tin Web là mô hình Client/Server [10, 11], trong đó trình duyệt Web sẽ gửi các yêu cầu tới máy phục vụ Web và máy phục vụ Web sẽ gửi các thông tin trả lời. Nhưng bản thân một máy phục vụ Web ngay từ đầu lại không có khả năng truy nhập và lấy thông tin từ các cơ sở dữ liệu. Vì vậy nếu muốn giải quyết bài toán truy nhập CSDL cho dịch vụ Web, ta phải mở rộng khả năng của máy phục vụ Web để nó có thể xử lý yêu cầu truy nhập vào một hay nhiều CSDL nào đó, lấy các thông tin từ đó ra và sau đó các thông tin này dưới dạng HTML được trả về cho trình duyệt Web đã gửi yêu cầu. Cho tới nay để làm được điều này chúng ta có thể thực hiện bằng một trong hai phương pháp sau:

• **Phương pháp 1:** Viết chương trình để mở rộng khả năng của máy phục vụ Web. Chương trình này phải một mặt giao tiếp với máy phục vụ Web để lấy các yêu cầu truy nhập CSDL được gửi tới từ trình duyệt Web và mặt khác giao tiếp được với CSDL để lấy các thông tin theo yêu cầu rồi sau đó trả về cho máy phục vụ Web dưới dạng HTML để máy phục vụ Web trả về cho trình duyệt Web [7, 10]. Các chương trình như vậy có thể được gọi là các ứng dụng tích hợp giữa máy phục vụ Web và CSDL.

Hiện nay có hai hướng chính để viết một chương trình ứng dụng như trên: Viết một ứng dụng theo chuẩn CGI (Common Gateway Interface) [7, 10] hoặc viết một ứng dụng cho máy phục vụ Web dưới dạng một thư viện liên kết động DLL (Dynamic Link Library) [10] sử dụng phương pháp lập trình ISAPI (Internet Server Application Programming Interface) hay NSAPI (Netscape Server Application Programming Interface).

• **Phương pháp 2:** Sử dụng các công nghệ hỗ trợ việc truy nhập CSDL một cách vạn năng. Các công nghệ hỗ trợ này thường được viết để truy nhập CSDL một cách tổng quát, có thể được dùng trong nhiều trường hợp khác nhau. Các công nghệ chúng ta có thể sử dụng là OLE DB, ADO [12], ODBC [10], ASP [1], IDC [10], COLD FUSION [4], JDBC [2, 3].

3.1. Sử dụng chuẩn CGI

Chuẩn CGI là một giao diện để chạy các chương trình ứng dụng dưới một máy phục vụ thông tin [7, 10]. Ứng dụng theo chuẩn CGI là một chương trình thực hiện được, nó có thể nhận được các yêu cầu từ máy phục vụ Web, thực hiện các yêu cầu này và trả kết quả về cho máy phục vụ dưới dạng một văn bản HTML hoặc cất giữ vào CSDL. Bằng cách dùng ứng dụng CGI, máy phục vụ Web có thể thâm nhập được những dữ liệu mà trình duyệt Web không đọc được một cách trực tiếp.

Sự trao đổi dữ liệu giữa chương trình CGI, trình duyệt Web và máy phục vụ Web được mô tả theo một quá trình gồm 5 bước như sau:

Bước 1: Máy trạm gửi yêu cầu tới máy phục vụ thông qua trình duyệt Web.

Bước 2: Web Server nhận yêu cầu, phân tích và chuyển yêu cầu.

Bước 3: Chương trình lấy thông số của yêu cầu thông qua các biến môi trường của Web Server, thực hiện các nhiệm vụ tùy thuộc vào yêu cầu nhận được.

Bước 4: Chương trình gửi kết quả về cho máy phục vụ.

Bước 5: Web Server trả dữ liệu về cho người dùng.

Các chương trình CGI có thể được viết bằng ngôn ngữ lập trình như Perl, C, C++, Visual Basic, Java [7], v.v..

Thông qua các cơ chế truy nhập CSDL của các ngôn ngữ, chúng ta có thể khai thác, xử lý, cập nhật thông tin tùy theo yêu cầu.

3.2. Sử dụng chuẩn ISAPI

ISAPI cho phép viết những chương trình mở rộng khả năng của một máy phục vụ thông tin. Một chương trình ISAPI thực chất là một thư viện liên kết động [10], nó có khả năng thực hiện mọi công việc mà ta có thể làm được khi sử dụng mô hình CGI. Hoạt động thông qua ISAPI tương tự như CGI và có điểm khác biệt khi máy chủ nhận được yêu cầu chạy chương trình ISAPI nó sẽ không tạo ra một tiến trình mới riêng cho chương trình đó mà sẽ đọc chương trình vào bộ nhớ tại cùng không gian địa chỉ với máy phục vụ Web và chương trình ISAPI sẽ có chung một tiến trình với máy phục vụ Web.

3.3. Công nghệ ASP

ASP (Active Server Pages) là một ngôn ngữ lập trình dạng kịch bản (Script) cho phép tạo những ứng dụng năng động có tính tương tác và hiệu suất cao chạy trên máy phục vụ Web [1]. Trong ASP có thể sử dụng VBScript, hoặc JScript [1, 7], v.v.. Hoạt động của ASP như sau:

Bước 1: Người dùng gửi tới máy phục vụ Web yêu cầu về một tệp asp (*.asp). Các tệp này có chứa các thẻ HTML và chương trình viết bằng ngôn ngữ Script nào đó.

Bước 2: ASP đọc từ đầu tới cuối tệp *.asp, nó xác định các cây lệnh của chương trình được viết bằng ngôn ngữ lập trình nào và dùng máy Script tương ứng thực hiện các câu lệnh đó.

Bước 3: ASP trả về cho người dùng một trang Web.

ASP sử dụng công nghệ OLE DB và ADO để truy cập, xử lý và khai thác CSDL [12].

3.4. Kết nối CSDL trên Internet IDC

IDC (Internet Database Connector) dùng để hỗ trợ cho IIS (Internet Information Sever) trong việc truy nhập các CSDL. IDC là một chương trình ISAPI DLL được viết sẵn tên là httpodbc.dll. Chương trình làm cầu nối giữa máy phục vụ Web với ODBC (Open Database Connectivity) và sử dụng các trình điều khiển ODBC để truy nhập tới các CSDL [10].

IDC dùng 2 tệp để điều khiển cách truy nhập CSDL. Các tệp này là các tệp kết nối CSDL (*.idc) và các tệp HTML mở rộng (*.htx).

Cơ chế làm việc của IDC như sau:

Bước 1: Máy phục vụ IIS nhận URL chứa thông tin về tệp *.idc.

Bước 2: Máy phục vụ thông tin IIS đọc httpodbc.dll vào bộ nhớ và cung cấp cho nó những thông tin liên quan tới tệp *.idc cần xử lý.

Bước 3: httpodbc.dll đọc tệp *.idc.

Bước 4: IDC thiết lập một liên kết với nguồn dữ liệu ODBC và câu lệnh SQL chỉ ra trong tệp *.idc được thực hiện.

Bước 5: Sau khi câu lệnh SQL được thực hiện, IDC đọc dữ liệu từ CSDL và trộn nó vào tệp *.htx được chỉ ra bằng trường Template trong tệp *.idc.

Bước 6: Sau khi đã trộn xong tất cả dữ liệu, IDC trả về cho IIS văn bản HTML để nó trả về cho máy trạm.

3.5. Sử dụng công cụ Cold Fusion

Cold Fusion là một công cụ phát triển các ứng dụng Web cho phép tạo ra các trang Web động và tương tác với nhau [4,7]. Nó cung cấp các phương pháp tích hợp trình duyệt Web, máy phục vụ Web, và CSDL trong những ứng dụng Web một cách hiệu quả. Cold Fusion không đòi hỏi dùng những ngôn ngữ lập trình truyền thống mà dùng một ngôn ngữ đánh dấu linh hoạt phía máy phục vụ tích hợp với HTML đó là CFML (Cold Fusion Markup Language). Hoạt động của Cold Fusion như sau:

Bước 1: Khi người dùng yêu cầu một trang trong ứng dụng Cold Fusion bằng cách gửi một yêu cầu dưới dạng một Form hoặc kích vào một siêu liên kết, trình duyệt Web gửi một yêu cầu HTTP tới máy phục vụ Web thông qua Internet.

Bước 2: Máy phục vụ Web chuyển những tham số về trang được yêu cầu tới Server ứng dụng Cold Fusion qua một Server API hoặc CGI, các trang Cold Fusion được tự động dịch chuyển và lưu trữ trong bộ nhớ nên tốc độ xử lý rất nhanh.

Bước 3: Cold Fusion đọc dữ liệu gửi từ người dùng và xử lý phần CFML. Dựa trên phần CFML, Server ứng dụng Cold Fusion thực hiện chương trình và tương tác với các kỹ thuật Server.

Bước 4: Cold Fusion tự động tạo ra một trang HTML và trả nó về cho máy phục vụ Web.

Bước 5: Máy phục vụ Web trả trang đó về cho người dùng.

3.6. Sử dụng chuẩn JDBC

Chuẩn JDBC được xây dựng từ nhu cầu cần kết nối các ứng dụng Java với CSDL SQL. JDBC là một giao diện lập trình ứng dụng Java để thực hiện các lệnh SQL. JDBC bao gồm một tập các lớp và giao diện được viết bằng ngôn ngữ lập trình Java. JDBC cung cấp một giao diện lập trình ứng dụng API chuẩn cho người xây dựng CSDL và tạo điều kiện để viết các ứng dụng CSDL bằng cách chỉ sử dụng Java API. JDBC là một môi trường SQL đầy đủ các chức năng có thể liên kết với hầu hết các loại CSDL. JDBC sử dụng các lớp Java để thực hiện các liên kết và hỏi đáp SQL, và cho phép chương trình Java truy nhập trực tiếp CSDL tương thích [2,3].

Java là một công cụ giúp cho việc đơn giản hóa việc xây dựng các ứng dụng phân tán chạy trên mạng. Là một công nghệ lập trình tiên tiến, Java hứa hẹn phát triển vai trò của Internet trên đó có thể chạy các chương trình độc lập với cấu hình. Công nghệ này cho phép sử dụng hàng loạt các dịch vụ và thông tin tương tác tức thời trên Internet. Java cho phép sử dụng viết các chương trình ổn định và tin cậy, xây dựng ứng dụng trên hầu hết các máy và chạy chương trình đó trên tất cả các máy mà không cần biên dịch lại mã, phân phối các ứng dụng trên mạng một cách an toàn. Đặc biệt các chương trình Java có thể nhúng vào các tài liệu Web biến các trang tĩnh thành các ứng dụng chạy trên máy tính của người dùng [5,7].

JDBC mở rộng các khả năng có thể thực hiện bằng Java. Chẳng hạn, với Java và JDBC API ta có thể tạo một trang Web chứa một Applet sử dụng thông tin thu được từ một CSDL ở xa. Ngày càng có nhiều lập trình viên sử dụng ngôn ngữ lập trình Java và vì thế nhu cầu truy nhập CSDL một cách dễ dàng từ Java cũng liên tục phát triển.

Về cơ bản, JDBC giúp thực hiện 3 chức năng sau:

- + Thiết lập một liên kết với CSDL.
- + Gửi các lệnh SQL.
- + Xử lý kết quả.

3.6.1. Thực hiện câu lệnh SQL với JDBC

Người lập trình Java sử dụng JDBC thông qua bốn lớp cơ bản sau [2, 3]:

Java.sql.DriverManager: Nạp trình điều khiển JDBC và quản lý các liên kết CSDL.

Java.sql.Connection: Liên kết với CSDL.

Java.sql.Statement: Quản lý các lệnh SQL trên liên kết.

Java.sql.ResultSet: Cho phép truy nhập kết quả của câu lệnh đã thực hiện.

Lớp DriverManager là tầng quản lý của JDBC, giữa người sử dụng và các trình điều khiển. Nó theo dõi các trình điều khiển có mặt và xử lý việc tạo kết nối giữa một CSDL và trình điều khiển tương ứng.

Lớp Connection thể hiện một kết nối với CSDL. Một phiên kết nối gồm các lệnh SQL được thực hiện và kết quả trả lại qua kết nối đó. Một ứng dụng đơn giản có thể có một hoặc nhiều kết nối với một CSDL, hoặc có thể có kết nối với nhiều CSDL.

Lớp Statement được sử dụng để gửi câu lệnh SQL tới CSDL và nhận kết quả từ những câu lệnh khác. Giao diện Statement cung cấp các hàm cơ bản để thực hiện các câu lệnh và lấy kết quả.

Lớp ResultSet chứa tất cả các hàng thỏa mãn điều kiện trong câu lệnh SQL, nó cho phép truy cập dữ liệu trong các hàng đó qua tập các hàm get có thể truy cập tới các cột của hàng hiện tại, đồng thời cũng có thể chuyển tới hàng tiếp theo sử dụng hàm ResultSet.next.

3.6.2. Thiết kế ứng dụng JDBC

Phương pháp truy nhập dữ liệu là phương pháp Client/Server thông thường, với máy của người sử dụng là Client và máy lưu giữ CSDL là Server. Một ứng dụng Java sẽ trao đổi trực tiếp với CSDL đòi hỏi phải truy cập trình điều khiển JDBC, trao đổi với hệ thống quản trị CSDL cụ thể. Các lệnh SQL của người dùng được gửi tới CSDL và kết quả của lệnh này được gửi trả lại cho người dùng. Đây là phương pháp thích hợp cho hầu hết các ứng dụng. Tuy nhiên phương pháp này có một số hạn chế, chẳng hạn như phải cài phần mềm khai thác thông tin trên mọi máy. Để khắc phục hạn chế này phải sử dụng phương pháp 3 tầng.

Xử lý ứng dụng Java ba tầng là bổ sung một tầng nữa vào mô hình Client/Server chuẩn. Trong mô hình này, các lệnh được gửi tới tầng trung gian và sau đó tầng này sẽ gửi các lệnh SQL tới CSDL. CSDL sẽ xử lý các lệnh SQL và gửi kết quả cho tầng trung gian, và tầng này sẽ gửi cho người sử dụng.

Mô hình 3 tầng rất phù hợp vì tầng trung gian cho phép duy trì điều khiển truy nhập và các loại cập nhật thực hiện trên CSDL.

Trên cơ sở nghiên cứu các phương pháp kết nối Web với CSDL, chúng tôi đã đề xuất xây dựng một sản phẩm phần mềm phục vụ cho Web khai thác CSDL, kết quả sản phẩm và chương trình được trình bày chi tiết trong tài liệu [8].

4. MỘT SỐ NHẬN XÉT ĐÁNH GIÁ VỀ GIẢI PHÁP KẾT NỐI WEB VỚI CSDL

Mô hình kết nối Web với CSDL chúng tôi trình bày không phụ thuộc vào môi trường lập trình cụ thể nào. Với công nghệ Microsoft hay công nghệ Java thì bản chất của vấn đề tích hợp Web với CSDL là như nhau:

- + Web Server,
- + CGI,
- + ADO hay JDBC.

Chúng ta đã xem xét các phương pháp kết nối Web với CSDL: CGI, ISAPI, ASP, IDC, COLD FUSION, JDBC. Mỗi phương pháp đều có đặc điểm riêng ưu và có nhược điểm khác nhau. Sau đây

là một số điểm khác biệt của mỗi phương pháp dựa trên mô hình thực hiện các ứng dụng kết nối CSDL.

+ Theo phương pháp CGI chương trình ứng dụng được viết dưới dạng này là một chương trình thực hiện được. Mỗi khi có yêu cầu thực hiện CGI từ phía người dùng máy phục vụ tạo ra một tiến trình mới cho chương trình CGI và truyền các dữ liệu từ người dùng qua biến môi trường. Như vậy số lượng các tiến trình mới do máy tạo ra bằng với số yêu cầu thực hiện của người dùng. Đây là nhược điểm của CGI.

+ Chương trình viết theo Web Server API thực chất là một thư viện liên kết động. Với phương pháp này, khi có một yêu cầu từ phía người dùng thì máy phục vụ sẽ không tạo ra một tiến trình mới mà đọc thư viện vào bộ nhớ tại cùng không gian địa chỉ với máy phục vụ Web và xử lý các yêu cầu. Như vậy các thư viện này cùng chung một tiến trình với máy phục vụ Web. Một khi thư viện đã được đọc vào bộ nhớ nó có thể phục vụ một lúc nhiều yêu cầu.

Các phương pháp ASP, IDC, Cold Fusion đều dùng ngôn ngữ Script nhúng vào các trang HTML để thực hiện ứng dụng.

+ Phương pháp IDC là phương pháp đơn giản nhất chỉ gồm một tệp chứa các lệnh SQL để truy nhập CSDL ODBC, tệp mẫu và kết xuất kết quả cho người dùng.

+ ASP là môi trường Script được nhúng trong máy phục vụ Web của Microsoft và chỉ thực hiện được thông qua cơ chế ISAPI. Ưu điểm nổi bật của ASP là hỗ trợ ngôn ngữ kịch bản, chương trình nguồn sẽ được tích hợp trong trang HTML.

+ Cold Fusion cũng là một ngôn ngữ giống ASP. Tuy nhiên điểm mạnh của nó là có thể thực hiện trên hầu hết các môi trường máy phục vụ Web khác nhau như Netscape Web Server, Microsoft Web Server, v.v.. Bằng nhiều cơ chế như CGI, Web Server API.

ASP, Cold Fusion, IDC đều có thể kết nối trực tiếp với các CSDL hoặc thông qua các trình điều khiển kết nối các dữ liệu trung gian.

+ Phương pháp sử dụng JDBC - Java Applet khác về cơ bản so với các phương pháp khác. Java Applet coi là một ứng dụng chạy độc lập, nó chỉ nhờ máy trạm Web để làm môi trường thực hiện, do đó dễ dàng xây dựng ứng dụng truy nhập CSDL Client/Server bằng Java Applet.

5. KẾT LUẬN

Trong thời đại hiện nay thông tin đã trở thành một trong những nhu cầu thiết yếu của con người. Việc nghiên cứu các phương pháp kết nối Web với CSDL nhằm khai thác, truy nhập và xử lý thông tin trên môi trường mạng cục bộ, mạng diện rộng, và mạng Internet cũng như đưa thông tin trong các CSDL lên mạng dưới dạng các trang Web là một nhu cầu cần thiết và đáng quan tâm. Tùy thuộc vào các lĩnh vực ứng dụng và môi trường phát triển phần mềm, chúng ta có thể chọn một trong các phương pháp này để kết nối nhiều hệ CSDL phục vụ cho việc quản lý, trao đổi thông tin, tạo ra các trang Web hoặc xây dựng các phần mềm phục vụ thương mại điện tử. Trong [8] chúng tôi đã sử dụng phương pháp kết nối Web với CSDL dùng JDBC để xây dựng một sản phẩm phần mềm phục vụ cho Web truy nhập tới một CSDL bất kỳ.

TÀI LIỆU THAM KHẢO

- [1] *Active Server Pages*, Microsoft Corporation, 1998.
- [2] Abston Hobbs, *Database Programming with JDBC*, Sams.net Publishing, 1997.
- [3] Art Taylor, *Jdbc Developers' Resource*, Informix Press, Prentice Hall PRT, 1997.
- [4] *Cold Fusion User Guide, Getting Started with Cold Fusion, Cold Fusion Language Reference*, Allaire Corporation, 1998.
- [5] Darel Ince and Adam Freeman, *Programming the Internet with Java*, Addison-Wesley, 1997.

- [6] Đoàn Văn Ban, *Phân tích thiết kế và Lập trình hướng đối tượng*, NXB Thống kê, Hà Nội, 1997.
- [7] Eric Ladd et al., *Using HTML 3.2, Java 1.1 and CGI*, Que Corporation, 1996.
- [8] Hồ Văn Hương, "Kết nối Web với cơ sở dữ liệu", luận án thạc sĩ khoa học, Đại học Khoa học tự nhiên, 1999.
- [9] Jeffrey D. Ullman, *Principles of Database and Knowledge System*, Prentice Hall, 1987.
- [10] *Microsoft Internet Information Server*, Microsoft Corporation, 1996.
- [11] Nguyễn Thúc Hải, *Mạng máy tính và các hệ thống mở*, NXB Giáo dục, 1997.
- [12] OLE DB/ADO: *Making Universal Data Access a Reality*, Microsoft Corporation, 1998.

Nhận bài ngày 10-8-1999

Nhận lại sau khi sửa ngày 12-11-1999

*Đoàn Văn Ban - Viện Công nghệ thông tin.
Hồ Văn Hương - Ban Cơ yếu Chính phủ.*