

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Bài tập lớn môn Hệ cơ sở dữ liệu nâng cao
TÌM HIỂU CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI
TƯỢNG TRÊN DB4O

GVHD

PGS. TS. Đặng Trần Khánh

Nhóm 6

Diệp Hưng (1570209)

Biện Lê Anh Hưng (7140235)

Nguyễn Thanh Hải (7140230)

Đặng Danh Hữu (1570210)

Tp. Hồ Chí Minh, Tháng 1/2016

MỤC LỤC

1.	Giới thiệu.....	6
2.	Tổng quan về cơ sở dữ liệu hướng đối tượng.....	8
2.1.	Mô hình hướng đối tượng.....	8
2.2.	Các hệ quản trị cơ sở dữ liệu hướng đối tượng	9
2.3.	Chuẩn ODMG.....	10
3.	Cơ sở dữ liệu hướng đối tượng db4o	11
3.1.	Tổng quan db4o	11
3.2.	Cài đặt và sử dụng db4o	11
3.2.1.	Cài đặt	11
3.2.2.	Cấu trúc bộ máy db4o.....	11
3.2.3.	Sử dụng db4o	12
3.2.4.	Cài đặt Object Manager Enterprise	12
3.3.	Tổng quan về API.....	12
3.3.1.	<i>Db4objects.Db4o</i>	12
3.3.2.	<i>Db4objects.Db4o.Ext</i>	13
3.3.3.	<i>Db4objects.Db4o.Config</i>	13
3.3.4.	<i>Db4objects.Db4o.Query</i>	13
3.3.5.	<i>Db4objects.Db4o.Linq</i>	13
3.4.	Cơ chế bảo mật	13
3.5.	Triển khai ứng dụng với db4o.....	14
3.5.1.	Mở cơ sở dữ liệu	15
3.5.2.	Lưu trữ các đối tượng	15
3.5.3.	Tìm kiếm các đối tượng	15
3.5.4.	Cập nhật các đối tượng	16
3.5.5.	Xóa đối tượng	17
3.6.	Công cụ quản lý đối tượng.....	17
3.6.1.	Duyệt cơ sở dữ liệu bằng OME.....	18
3.6.2.	Truy vấn	19
3.7.	Các hệ thống truy vấn	21
4.	Phát triển ứng dụng sử dụng CSDL db4o.....	23
4.1.	Giới thiệu.....	23

4.2.	Thiết kế mô hình đối tượng	23
4.2.1.	<i>Các đối tượng chính</i>	23
4.2.2.	<i>Các đối tượng trong phân quyền</i>	25
4.3.	Hiện thực	25
4.3.1.	<i>Môi trường hiện thực</i>	25
4.3.2.	<i>Cấu trúc chương trình.....</i>	25
4.3.3.	<i>Mô hình phân quyền</i>	26
4.3.4.	<i>Giao diện chương trình</i>	27
5.	Kết quả	33
6.	Kết luận	33
7.	Phụ lục	34
7.1.	Hướng dẫn cài đặt ứng dụng	34
7.2.	Hướng dẫn cài đặt môi trường phát triển ứng dụng.....	35
7.2.1.	<i>Cài đặt môi trường phát triển.....</i>	35
7.2.2.	<i>Biên dịch mã nguồn</i>	35
Tham khảo		36

MỤC LỤC HÌNH

Hình 1. Cấu trúc thư mục cài đặt db4o	11
Hình 2. Mô hình CSDL HĐT của db4o so với hệ quản trị CSDL phổ biến khác.....	14
Hình 3. Object Manager Enterprise trên Visual Studio	17
Hình 4. Giao diện duyệt CSDL trên OME	18
Hình 5. Tạo thư mục quản lý đối tượng trong OME.....	19
Hình 6. Giao diện truy vấn của OME	20
Hình 7. Cửa sổ Query Builder của OME.....	20
Hình 8. Kết quả Query Builder trên OME	21
Hình 9. Lịch sử truy vấn trên Query Builder	21
Hình 10. Sơ đồ các đối tượng chính trong ứng dụng	24
Hình 11. Các đối tượng lưu trữ phân quyền	25
Hình 12. Cấu trúc chính của chương trình	26
Hình 13. Mô hình phân quyền người dùng đối với các đối tượng	27
Hình 14. Giao diện đăng nhập	28
Hình 15. Giao diện chính sau khi đăng nhập thành công	28
Hình 16. Giao diện quản lý thông tin sinh viên	30
Hình 17. Giao diện quản lý thông tin giáo viên	30
Hình 18. Giao diện quản lý thông tin điểm sinh viên	31
Hình 19. Giao diện quản lý thông tin môn học.....	31
Hình 20. Giao diện quản lý thông tin lớp học.....	31
Hình 21. Giao diện quản lý thông tin khoa.....	32
Hình 22. Giao diện phân quyền người dùng	32
Hình 23. Kết quả giải nén thư mục cài đặt	34
Hình 24. Tiến trình cài đặt phần mềm.....	34
Hình 25. Kết quả giải nén thư mục mã nguồn.....	35

TÌM HIỂU CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG TRÊN DB4O

Tháng 1, 2016

Tóm tắt nội dung

Cơ sở dữ liệu hướng đối tượng được ra đời và phát triển đã khá lâu. Đã có rất nhiều công trình khoa học nghiên cứu về lĩnh vực này, cũng như cũng có nhiều ứng dụng áp dụng cơ sở dữ liệu hướng đối tượng. Với ưu điểm lớn nhất là, cơ sở dữ liệu hướng đối tượng sở hữu khả năng lưu trữ thông tin theo cách gần với các sự vật sự việc trong thế giới tự nhiên mà người dùng quen thuộc. Nó cho phép người dùng lưu trữ và truy xuất thông tin đối tượng có kiểu dữ liệu phức tạp mà không cần truy xuất các thuộc tính rời rạc từ nhiều bảng dữ liệu như cơ sở dữ liệu truyền thống. Tuy nhiên ngày nay, cơ sở dữ liệu hướng đối tượng vẫn chưa thật sự phổ biến. Vì sao như vậy? Do cơ sở dữ liệu hướng đối tượng cũng có nhiều khuyết điểm cũng lớn không kém. Trong số đó là, chưa có cơ sở lý thuyết vững chắc (như đại số quan hệ của cơ sở dữ liệu quan hệ truyền thống), khó khăn trong việc bảo trì, nâng cấp ứng dụng sử dụng nó. Trong bài báo cáo này, nhóm thực hiện sẽ thử xây dựng một ứng dụng đơn giản sử dụng cơ sở dữ liệu hướng đối tượng db4o, từ đó đối chiếu lại các đánh giá đã tìm hiểu về cơ sở dữ liệu hướng đối tượng dựa trên kết quả thực tế làm được.

1. Giới thiệu

Trong khóa học năm 2015 này, các học viên đã được hướng dẫn tìm hiểu nhiều mảng kiến thức khác nhau của lĩnh vực Cơ sở dữ liệu cũng như các xu hướng nổi bật về cơ sở dữ liệu trong tương lai. Nhóm 6 đã được giao tìm hiểu về Cơ sở dữ liệu hướng đối tượng (CSDL HĐT). Trong bài thuyết trình đầu tiên của mình, nhóm đã thực hiện được một số việc như sau:

- Giới thiệu tổng quan, khái niệm về CSDL HĐT
- Các tiền đề để xây dựng nên CSDL HĐT
- Các tiêu chuẩn tổng quát mà một hệ CSDL HĐT cần có
- So sánh đánh giá ưu, nhược điểm của CSDL HĐT với các loại CSDL khác

Bối cảnh các bài báo mà nhóm tìm hiểu trong giai đoạn đầu này là vào những năm 1989-1990, khi mà CSDL HĐT còn chưa phát triển. Các tác giả của các bài báo đã đưa ra các tiêu chuẩn, đặc điểm chung nhất mà một hệ CSDL HĐT cần phải đáp ứng. Việc này nhằm định hướng cho sự phát triển của CSDL HĐT về sau, giúp các nhà phát triển có cơ sở tham khảo cho các ứng dụng của mình. Tuy nhiên, ở giai đoạn sơ khai này, chưa có một hệ những hệ CSDL HĐT thật sự thật sự hoàn chỉnh, những ứng dụng sử dụng CSDL HĐT còn hiếm gặp.

Nhóm đã tiếp tục tìm hiểu về CSDL HĐT và quá trình phát triển của nó ở những giai đoạn về sau. Tuy sở hữu nhiều ưu điểm vượt trội so với CSDL truyền thống, CSDL HĐT cũng bộc lộ nhiều khuyết điểm trong quá trình phát triển của mình. Điểm yếu lớn nhất của nó là khả năng thay đổi mềm dẻo theo ứng dụng. Tức là khi cần thay đổi mô hình dữ liệu của ứng dụng thì gần như phải chỉnh sửa mã nguồn và biên dịch lại toàn bộ ứng dụng. Chính vì những điểm yếu đó mà CSDL HĐT gần như không còn được sử dụng trong thời điểm hiện tại.

Vì thế, trong bài tập lớn của mình nhóm muốn tìm hiểu một CSDL HĐT thực tế, xây dựng thử nghiệm một ứng dụng đơn giản trên CSDL đó, để đánh giá lại một cách thực tế hơn các đặc điểm cũng như các ưu khuyết điểm mà một CSDL HĐT có. Đồng thời làm rõ hơn lý do mà CSDL HĐT vẫn ít được ưa chuộng như hiện nay. Để thực hiện bài tập lớn của mình, nhóm đã tìm hiểu về các hệ CSDL HĐT như: Versant Object Database, ObjectStore, db4o, ObjectDB, odbpp,... Tuy nhiên vì nhiều lý do như: bản quyền thương mại, không còn phát triển,... nhóm đã lựa chọn db4o để thực hiện đề tài của mình.

Đề tài mà nhóm sẽ thực hiện là: Tìm hiểu về cơ sở dữ liệu hướng đối tượng trên db4o.

Mục tiêu của đề tài:

- Tìm hiểu về CSDL HĐT db4o. Tìm hiểu các chức năng mà db4o cung cấp cho người dùng. Tìm hiểu cách thức lưu trữ dữ liệu, cách truy vấn dữ liệu trên db4o.

- Xây dựng ứng dụng “Quản lý hồ sơ trường học” sử dụng db4o để lưu trữ dữ liệu. Thực hiện các chức năng truy vấn dữ liệu trên ứng dụng.
- Thực hiện việc xác thực, phân quyền người dùng để bảo mật dữ liệu.
- Đánh giá các cách hiện thực so với CSDL truyền thống. Đối chiếu lại với các đặc điểm đã tìm hiểu ở bài thuyết trình 1.

2. Tổng quan về cơ sở dữ liệu hướng đối tượng

2.1. Mô hình hướng đối tượng

Một cơ sở dữ liệu hướng đối tượng được xem như là một kho bền vững các đối tượng được tạo ra bởi ngôn ngữ lập trình hướng đối tượng. Với một ngôn ngữ lập trình bất kỳ, các đối tượng sẽ ngừng tồn tại (kết thúc) khi chương trình ứng dụng kết thúc, nhưng trong cơ sở dữ liệu hướng đối tượng, các đối tượng được duy trì ở bên ngoài phạm vi thực hiện của chương trình.

Hệ quản trị cơ sở dữ liệu hướng đối tượng (OODBMS) quản lý dữ liệu, mã chương trình và các cấu trúc kết hợp nhằm thiết lập một cơ sở dữ liệu hướng đối tượng. Khác với các hệ quản trị cơ sở dữ liệu quan hệ, các hệ quản trị cơ sở dữ liệu hướng đối tượng khác nhau rất nhiều về cách thức và khả năng ứng dụng. Tổ chức chuẩn hoá việc quản lý dữ liệu đối tượng ODMG (*Object Data Management Group*) cố gắng tìm cách giải quyết sự khác biệt đó bằng cách thống nhất đưa ra những kỹ thuật mô hình hóa đối tượng OMT (*Object Modeling Techniques*) hay ngôn ngữ mô hình hoá thống nhất UML (*Unified Modeling Language*).

Mặc dù cho đến nay mô hình dữ liệu hướng đối tượng chưa có một chuẩn được chấp nhận chính thức bởi các tổ chức ANSI hay ISO như đối với mô hình dữ liệu quan hệ, nhưng trong suốt những năm qua một số hệ cơ sở dữ liệu hướng đối tượng như O2 hay ObjectStore đã xâm nhập vào thị trường Mỹ một cách chính thức nhờ khả năng mạnh mẽ của việc mô hình hoá thế giới thực. Bên cạnh đó, các đề xuất về một chuẩn cho mô hình dữ liệu hướng đối tượng và ngôn ngữ truy vấn OQL do tổ chức ODMG đã được các nhà tin học quan tâm một cách thực sự. Vấn đề này được đề cập đến thông qua việc giới thiệu “*mô hình hạt nhân*” của mô hình dữ liệu hướng đối tượng, ở đó một số các quy ước tối thiểu về mô hình hướng đối tượng được bàn đến.

Mô hình hạt nhân đủ mạnh để thoả mãn nhiều đòi hỏi của các ứng dụng mới, hơn nữa còn được dùng làm cơ sở cho việc phân tích những khác biệt chính giữa mô hình dữ liệu hướng đối tượng với các mô hình dữ liệu truyền thống khác như mô hình dữ liệu quan hệ chẳng hạn.

Mô hình hạt nhân dựa trên các khái niệm cơ bản sau:

- Mỗi thực thể của thế giới thực được mô hình hoá bởi một đối tượng. Mỗi đối tượng được xác định với một tên duy nhất được gọi là định danh đối tượng.
- Mỗi đối tượng có một tập các thuộc tính và phương thức. Giá trị của mỗi thuộc tính có thể là một đối tượng hay một tập đối tượng. Đặc trưng này cho phép các đối tượng phức tạp được định nghĩa như sự kết nhập của các đối tượng khác.
- Tập các thuộc tính của một đối tượng và tập các phương thức biểu diễn theo thứ tự cấu trúc và hành vi của đối tượng.

- Các giá trị thuộc tính biểu diễn trạng thái của đối tượng. Trạng thái của đối tượng được truy cập hay sửa đổi bằng việc gửi các thông báo tới đối tượng để viện dẫn các phương pháp tương ứng.
- Các đối tượng có cùng cấu trúc và hành vi được nhóm lại thành một lớp. Một lớp biểu diễn một hình mẫu cho một tập các đối tượng đồng dạng. Mỗi đối tượng là thể hiện của một lớp nào đó.
- Một lớp có thể được định nghĩa như một chuyên biệt hoá của một hay nhiều lớp. Một lớp được định nghĩa như vậy gọi là một lớp con và kế thừa các thuộc tính và phương thức thuộc lớp trên của nó.

Như chúng ta đã biết, các cơ sở dữ liệu quan hệ đã và đang được sử dụng rất hiệu quả trong nhiều bài toán ứng dụng. Cơ sở dữ liệu hướng đối tượng cần phải nghiên cứu và ứng dụng bởi ưu, nhược điểm sau:

- Hỗ trợ các kiểu dữ liệu được định nghĩa bởi người sử dụng
- Cung cấp một mẫu hình phát triển cơ sở dữ liệu cho cả phân tích, thiết kế và cài đặt ứng dụng, không phải chuyển từ mẫu hình này sang mẫu hình khác như trong quá trình phát triển phần mềm.
- Cải tiến đáng kể về chất lượng dữ liệu. Ta có thể đưa ra nhiều ràng buộc vào cấu trúc dữ liệu. Mô hình còn cho phép thể hiện cả những ràng buộc không cấu trúc để chương trình phải thoả mãn khi nó thực hiện trong cơ sở dữ liệu. Một cơ sở dữ liệu hướng đối tượng có thể dẫn về một cơ sở dữ liệu quan hệ được chuẩn hoá.
- Tốc độ phát triển phần mềm nhanh hơn. Cấu trúc cơ sở dữ liệu nhất quán và rõ ràng giúp cho lập trình ứng dụng trở nên đơn giản và nhanh hơn.
- Tích hợp dễ dàng.

2.2. Các hệ quản trị cơ sở dữ liệu hướng đối tượng

Các hệ quản trị cơ sở dữ liệu hướng đối tượng có thể được định nghĩa là hệ quản trị cơ sở dữ liệu trực tiếp hỗ trợ một mô hình dựa trên kiểu hướng đối tượng. Nó phải cung cấp việc lưu trữ bền cho các đối tượng và những mô tả của chúng. Hệ cũng phải cung cấp một ngôn ngữ cho việc định nghĩa lược đồ và cho việc thao tác các đối tượng và lược đồ của chúng. Ngoài ra, nó thường có một ngôn ngữ truy vấn và những cơ chế cơ sở dữ liệu cần thiết cho việc tối ưu hoá truy cập như lập chỉ mục (*indexing*) và gom cụm (*clustering*), các cơ chế điều khiển tương tranh và trao quyền đối với truy cập nhiều người dùng, cơ chế khôi phục khi có sự cố.

Một số hệ quản trị cơ sở dữ liệu hướng đối tượng phổ biến trên thị trường như *ObjectStore*, *GemStore*, *Objectivity*, *O2*, *Jasmine*, *Versant*, *Db4o* và *POET*. Hệ quản trị cơ

sở dữ liệu hướng đối tượng hỗ trợ rất nhiều kiểu dữ liệu phong phú và thông thường là kết hợp chặt chẽ với ít nhất một ngôn ngữ lập trình. Các hệ quản trị cơ sở dữ liệu hướng đối tượng thực hiện nhanh hơn những ứng dụng điều khiển từ các đối tượng tham chiếu đến đối tượng, hỗ trợ kế thừa, quản lý định danh đối tượng và nhiều tính chất khác.

Các hệ quản trị cơ sở dữ liệu hướng đối tượng cũng có các yếu điểm sau:

- Thiếu cơ sở lý thuyết và chuẩn hoá
- Chưa có cơ chế đủ mạnh để bảo vệ cơ sở dữ liệu
- Không có khả năng mở rộng logic
- Chưa hỗ trợ các siêu (meta) ứng dụng

Hệ quản trị cơ sở dữ liệu hướng đối tượng là thích hợp hơn đối với những ứng dụng mới như:

- Những ứng dụng thiết kế công nghệ
- Các ứng dụng đa phương tiện
- Các cơ sở tri thức
- Những ứng dụng đòi hỏi xử lý phân tán và tương tranh.
- Các phần mềm nhúng

2.3. Chuẩn ODMG

Như đã biết, chưa có một định nghĩa tiêu chuẩn thống nhất cho các mô hình đối tượng. Các ngôn ngữ lập trình hướng đối tượng và các hệ cơ sở dữ liệu hướng đối tượng hỗ trợ nhiều mô hình đối tượng khác nhau. Để giải quyết vấn đề này, ODMG (*Object Database Management Group* – Nhóm quản trị cơ sở dữ liệu đối tượng), một tổ chức mà các thành viên là các nhà sản xuất của nhiều hệ quản trị cơ sở dữ liệu hướng đối tượng thương mại khác nhau đã đề xuất một cơ sở dữ liệu tiêu chuẩn. Mục tiêu của ODMG là thống nhất mô hình đối tượng hạt nhân của nhiều hệ quản trị cơ sở dữ liệu khác nhau.

Chuẩn ODMG – 93 được công bố năm 1993 và được chỉnh sửa thành phiên bản 1.1 năm 1994. Phiên bản 2.0 (1997) của chuẩn định nghĩa một mô hình đối tượng dựa trên mô hình đối tượng hạt nhân, được đề xuất bởi nhóm quản trị đối tượng (OMG – *Object Management Group*). Phiên bản mới nhất là 3.0 (1999) nâng cao khả năng kết nối với Java.

Những thành phần cơ bản của ODMG 3.0 là:

- Mô hình hướng đối tượng
- Ngôn ngữ đặc tả đối tượng
- Ngôn ngữ truy vấn đối tượng
- Ngôn ngữ kết hợp C++, Smalltalk, Java

3. Cơ sở dữ liệu hướng đối tượng db4o

3.1. Tổng quan db4o

Db4o (*database for objects*) là một cơ sở dữ liệu hướng đối tượng ra đời năm 2000 bởi *Carl Rosenberger*. Đây là CSDL HĐT mã nguồn mở hỗ trợ Java và .NET. Năm 2004, phiên bản thương mại đầu tiên của db4o được phát hành bởi *db4objects, Inc.* Năm 2008, db4o được mua lại bởi *Versant Corporation*, một công ty con của *Actian*. Vào tháng 10 năm 2014, *Actian* đã không tiếp tục theo đuổi và thúc đẩy thương mại hóa db4o cho các khách hàng mới nữa. Phiên bản mới nhất của db4o là 8.0 năm 2011.

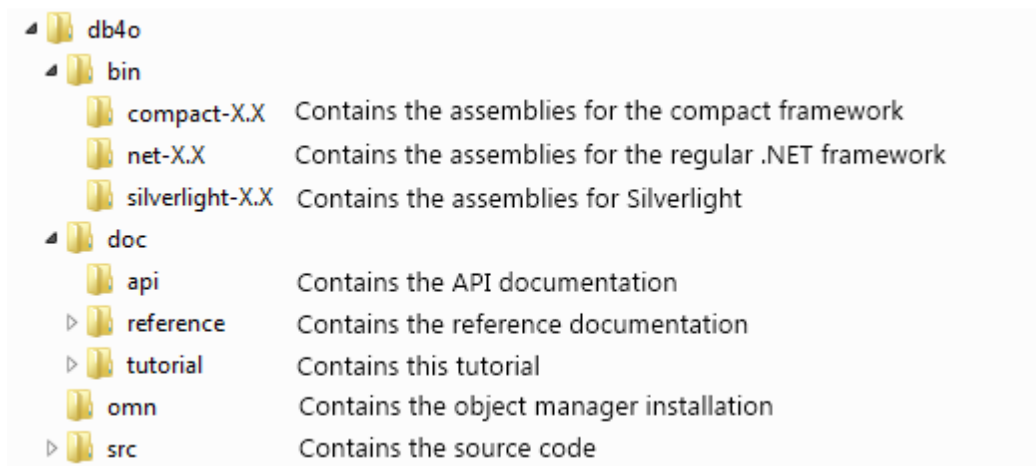
Một số đặc điểm chính của db4o:

- Tốc độ nhanh
- Đơn giản, dễ sử dụng
- Yêu cầu bộ nhớ tương đối nhỏ
- Không cần ánh xạ chuyển đổi đối tượng
- Không thay đổi các lớp tạo ra các đối tượng bền vững
- Làm việc với chế độ đơn hoặc client/server
- Lưu trữ và kết hợp với bộ sưu tập các đối tượng không thuần nhất
- Tự động quản lý và xác định phiên bản của lược đồ dữ liệu

3.2. Cài đặt và sử dụng db4o

3.2.1. Cài đặt

Db4o .NET chứa trong một file cài đặt MSI, *db4o-8.0-net.msi*. Sau khi cài đặt, ta có cấu trúc thư mục như sau:



Hình 1. Cấu trúc thư mục cài đặt db4o

3.2.2. Cấu trúc bộ máy db4o

Bộ máy cơ sở dữ liệu db4o bao gồm có một file DLL chính. Các phiên bản được chứa trong thư mục */db4o-8.0/bin/*.

Db4o hỗ trợ cho nhiều phiên bản *Microsoft .NET*, *.NET Framework 3.5* hoặc *.NET Framework 4.0*. Phải chắc chắn dùng đúng trong từng môi trường.

/db4o-8.0/bin/net-X.X/Db4objects.Db4o.dll là bộ máy db4o chuẩn cho *.NET framework* tương ứng.

/db4o-8.0/bin/compact-X.X/Db4objects.Db4o.dll được xây dựng cho *compact Framework*.

/db4o-8.0/bin/silverlight-X.X/Db4objects.Db4o.dll là bộ máy db4o cho *Silverlight*.

3.2.3. Sử dụng db4o

Để sử dụng db4o khi phát triển dự án, chỉ cần thêm tham chiếu đến một trong các file *Db4objects.Db4o.dll* bên trên.

3.2.4. Cài đặt Object Manager Enterprise

Object Manager Enterprise (OME) là chương trình quản lý đối tượng của cơ sở dữ liệu db4o, được cung cấp như là một plugin của Visual Studio. OME được cài đặt như là một phần của bộ cài đặt db4o.

3.3. Tổng quan về API

Tài liệu API cung cấp cái nhìn tổng thể về API. Các gói *Db4objects.Db4o* và *Db4objects.Db4o.Query* là tất cả những gì chúng ta cần quan tâm.

3.3.1. Db4objects.Db4o

Db4objects.Db4o gồm hầu hết các hàm thường dùng khi làm việc với db4o. Hai đối tượng cần chú ý là: *Db4objects.Db4o.Db4oEmbedded* và *Db4objects.Db4o.IObjectContainer*. Các phương thức tĩnh trong lớp *Db4oEmbedded* cho phép mở các tập tin cơ sở dữ liệu, bắt đầu với một *server* hoặc kết nối tới *server* đã tồn tại. Nó cũng giúp cấu hình môi trường db4o trước khi mở cơ sở dữ liệu.

Giao diện quan trọng nhất mà ta hay sử dụng là *IObjectContainer*. Nó như là CSDL db4o của ta.

- Một *IObjectContainer* có thể sử dụng cơ sở dữ liệu ở chế độ đơn hoặc *client* kết nối với db4o server.

- Mọi *IObjectContainer* đều chứa một giao tác (*transaction*). Khi mở một *IObjectContainer*, là ta đang trong một giao tác, khi sử dụng *Commit()* hoặc *Rollback()*, giao tác tiếp theo bắt đầu thực hiện ngay lập tức.
- Mọi *IObjectContainer* đảm bảo nó tham chiếu tới các đối tượng lưu trữ và trừu tượng. Mặt khác, nó cũng quản lý các đối tượng định danh và có thể đạt được tới mức thực thi cao.
- Khi kết thúc một *IObjectContainer*, tất cả cơ sở dữ liệu tham chiếu tới đối tượng trong RAM sẽ bị huỷ bỏ.

3.3.2. *Db4objects.Db4o.Ext*

Gói *Db4objects.Db4o.Ext* chứa các phương thức mở rộng.

3.3.3. *Db4objects.Db4o.Config*

Gói *Db4objects.Db4o.Config* chứa các kiểu để cấu hình db4o

3.3.4. *Db4objects.Db4o.Query*

Gói *Db4objects.Db4o.Query* chứa lớp lưu trữ để cấu trúc *Native Queries*. Giao diện *Native Query* là giao diện truy vấn chính của db4o và ưu tiên hơn *Soda Query API*.

3.3.5. *Db4objects.Db4o.Linq*

Đây là một giao diện query khác, kết hợp ưu thế của *Native Queries* và loại hỗ trợ CSDL rộng rãi khác.

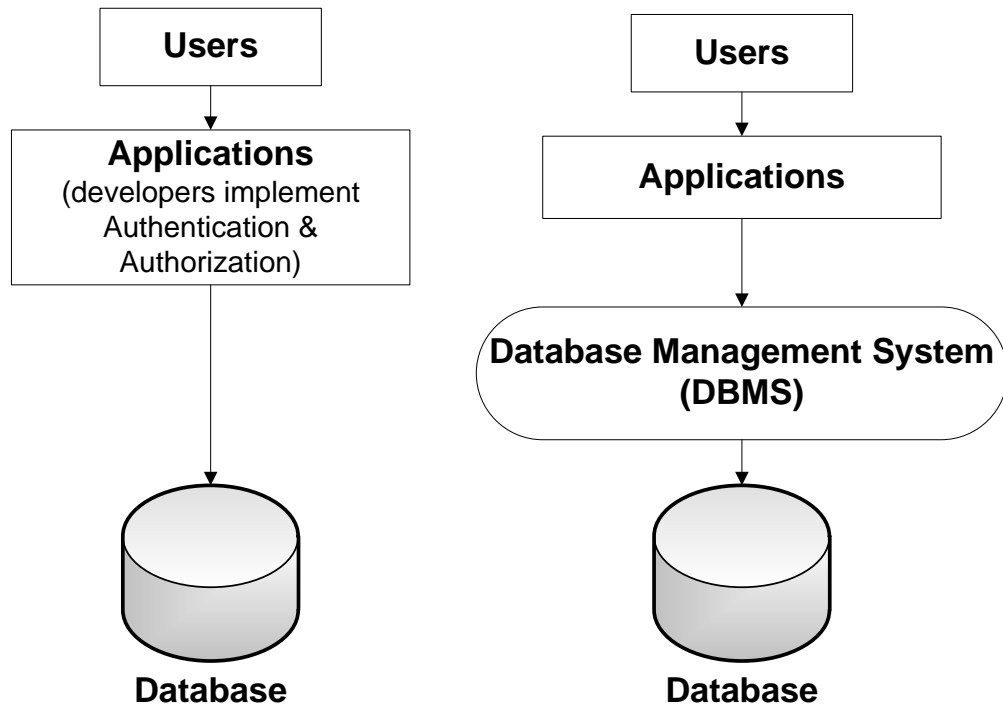
3.4. Cơ chế bảo mật

Db4o chỉ là một ODB và không phải là một ODBMS, nó không độc lập, nghĩa là mọi thứ đều thông qua một ứng dụng. Đó đó ứng dụng có thể kiểm soát hoàn toàn cơ chế xác thực (*authentication*) và phân quyền (*authorization*).

Trong chế độ nhúng (*embedded mode*), không có cơ chế bảo mật nào được tích hợp, không có quản lý *user* nào cả. Do đó, db4o trở nên rất linh hoạt, từ việc sử dụng cơ chế không cần xác thực gì cho đến tất cả các cơ chế *authentication* có thể thực hiện được.

Mọi cơ chế *authentication* hay *authorization* nếu cần sẽ được thực hiện bởi người phát triển ứng dụng. Dữ liệu được chứa trong một file, việc truy cập vào file nên được điều khiển bởi việc dùng cơ chế của nền tảng bên dưới.

Trong chế độ *client/server*, phần *server* lắng nghe kết nối mạng và với cách này có thể áp dụng cơ chế xác thực *username/password* để cho phép truy cập vào *server*. Phía *client* khi đó phải xác định *username* và *password* để có thể kết nối đến *server*. Đây có lẽ là cơ chế bảo mật có thể được hiện thực và chỉ là cơ chế *authentication* chứ không phải *authorization*.



Hình 2. Mô hình CSDL HĐT của db4o so với hệ quản trị CSDL phổ biến khác

3.5. Triển khai ứng dụng với db4o

Bắt đầu bằng một ví dụ đơn giản. Ta sẽ xem xét làm thế nào để lưu trữ, tìm kiếm, cập nhật và xoá dữ liệu trong một lớp đơn chỉ chứa các thành phần nguyên thủy và chuỗi. Trong ví dụ này dùng một pilot chứa thuộc tính tên và điểm.

Đầu tiên hãy tạo một lớp chứa dữ liệu như sau:

```
namespace Db4odoc.Tutorial.F1.Chapter1
{
    public class Pilot
    {
        string _name;
        int _points;
        public Pilot(string name, int points)
        {
            _name = name;
            _points = points;
        }
        public string Name
        {
            get { return _name; }
        }
        public int Points
        {
            get { return _points; }
        }
    }
}
```

```

    }
    public void AddPoints(int points)
    {
        _points += points;
    }
    override public string ToString()
    {
        return string.Format("{0}/{1}", _name, _points);
    }
}
}

```

3.5.1. Mở cơ sở dữ liệu

Để truy cập cơ sở dữ liệu db4o hoặc tạo một cơ sở dữ liệu mới, gọi hàm *Db4oEmbedded.OpenFile()* và cung cấp đường dẫn cho file cơ sở dữ liệu là các tham số và nhận được một đối tượng *IObjectContainer*. *IObjectContainer* mô tả cơ sở dữ liệu và sẽ là giao diện chính của db4o. *IObjectContainer* dùng phương thức *Close()* để đóng file cơ sở dữ liệu và giải phóng tất cả các tài nguyên liên kết với nó.

```

string YapFileName = "";
// truy cập db4o
IObjectContainer db = Db4oFactory.OpenFile(YapFileName);
try
{
    // thực thi công việc với db4o
}
finally
{
    db.Close();
}

```

Tên file cơ sở dữ liệu db4o phải là giá trị chuỗi (*string*). Nếu tên file đã tồn tại thì db4o sẽ mở ra, nếu chưa tồn tại sẽ tạo mới.

3.5.2. Lưu trữ các đối tượng

Các đối tượng trong db4o được lưu trữ dựa trên cấu trúc của lớp đối tượng định nghĩa trên ngôn ngữ của ứng dụng mà không cần phải có cấu trúc lớp đối tượng riêng dành cho CSDL.

Để lưu trữ đối tượng, gọi thủ tục *Store()*

```

Pilot pilot1 = new Pilot("Michael Schumacher", 100);
db.Store(pilot1);
Console.WriteLine("Stored {0}", pilot1);

```

3.5.3. Tìm kiếm các đối tượng

Db4o cung cấp 3 hệ thống truy vấn *Query by Example* (QBE), *Native Queries* (NQ) và *SODA Query API* (SODA).

Khi sử dụng *Query by Example* để tạo ra đối tượng nguyên mẫu cho db4o sử dụng. Db4o sẽ truy vấn tất cả các đối tượng kiểu đã cho. Kết quả sẽ trả về là một dữ liệu *ObjectSet*. Ta sử dụng phương thức *listResult()* để hiển thị nội dung kết quả *ObjectSet*:

```
public static void ListResult(IObjectSet result)
{
    Console.WriteLine(result.Count);
    foreach (object item in result)
    {
        Console.WriteLine(item);
    }
}
```

Để tìm tất cả các pilots từ cơ sở dữ liệu, ta tạo ra một tập proto ban đầu là rỗng:

```
Pilot proto = new Pilot(null, 0);
IObjectSet result = db.QueryByExample(proto);
ListResult(result);
```

Chú ý rằng ta ghi rõ 0 điểm, những kết quả không ràng buộc chỉ với những *pilot* với điểm 0; 0 là giá trị ngầm định cho trường số nguyên.

Db4o cũng cung cấp một *shortcut* để tìm kiếm tất cả các dữ liệu của lớp.

```
IObjectSet result = db.QueryByExample(typeof(Pilot));
ListResult(result);
```

Với .NET 2.0 db4o cũng cung cấp một shortcut, sử dụng phương thức truy vấn:

```
ICollection<Pilot> pilots = db.Query<Pilot>(typeof(Pilot));
```

Truy vấn một pilot theo tên:

```
Pilot proto = new Pilot("Michael Schumacher", 0);
IObjectSet result = db.QueryByExample(proto);
ListResult(result); //Truy vấn pilot với một số điểm xác định:
Pilot proto = new Pilot(null, 100);
IObjectSet result = db.QueryByExample(proto);
ListResult(result);
```

3.5.4. Cập nhật các đối tượng

Cập nhật các đối tượng cũng dễ như lưu trữ chúng. Trong thực tế, sử dụng phương thức *Store()* để cập nhật đối tượng.


```

IObjectSet result = db.QueryByExample(new Pilot("Michael Schumacher", 0));
Pilot found = (Pilot)result.Next();
found.AddPoints(11);
db.Store(found);
Console.WriteLine("Added 11 points for {0}", found);
RetrieveAllPilots(db);

```

Một điểm chú ý quan trọng là khi truy vấn tới đối tượng đầu tiên. Khi gọi *Store()* để sửa đổi đối tượng đã được lưu trữ, nếu đối tượng đó là “không biết” (đã lưu trữ trước đây hoặc truy vấn trong phần này), db4o sẽ chèn thêm đối tượng mới. Db4o làm vậy bởi nó không tự động đối sánh được các đối tượng cần lưu trữ với các đối tượng lưu trữ trước đây. Nó cho rằng đó là chèn đối tượng thứ hai, dù đối tượng này có cùng trường giá trị.

3.5.5. Xóa đối tượng

Để xóa đối tượng trong cơ sở dữ liệu sử dụng phương thức *Delete()*.

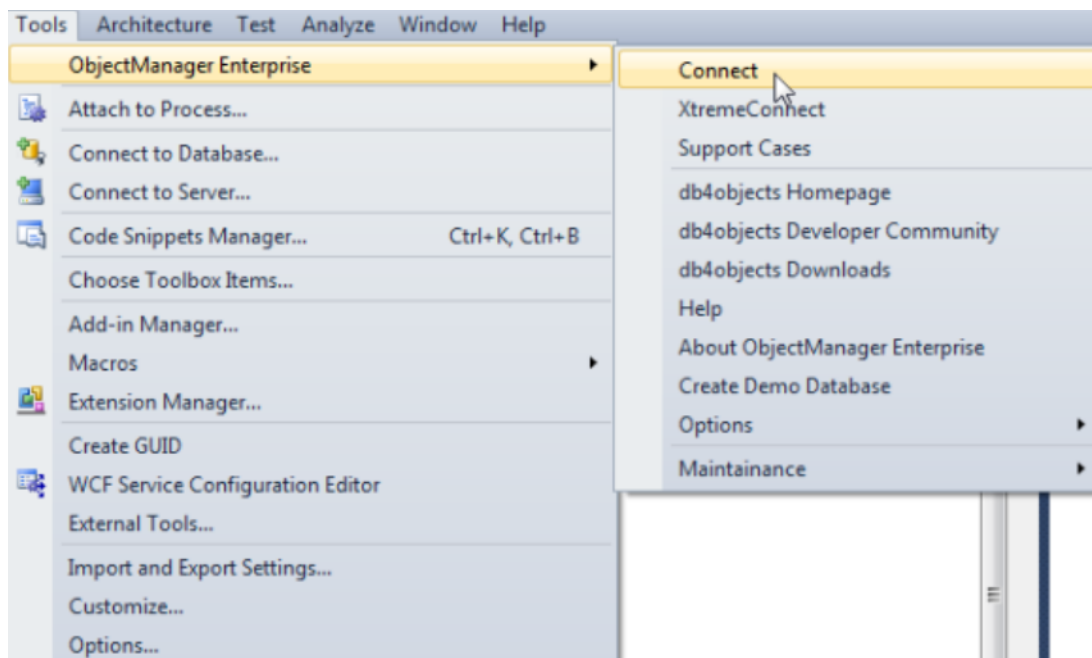
```

IObjectSet result = db.QueryByExample(new Pilot("Michael Schumacher", 0));
Pilot found = (Pilot)result.Next();
db.Delete(found);
Console.WriteLine("Deleted {0}", found);
RetrieveAllPilots(db);

```

3.6. Công cụ quản lý đối tượng

Sau khi cài Object Manager Enterprise (OME), có thể thấy trong menu Tool của Visual Studio như sau:



Hình 3. Object Manager Enterprise trên Visual Studio

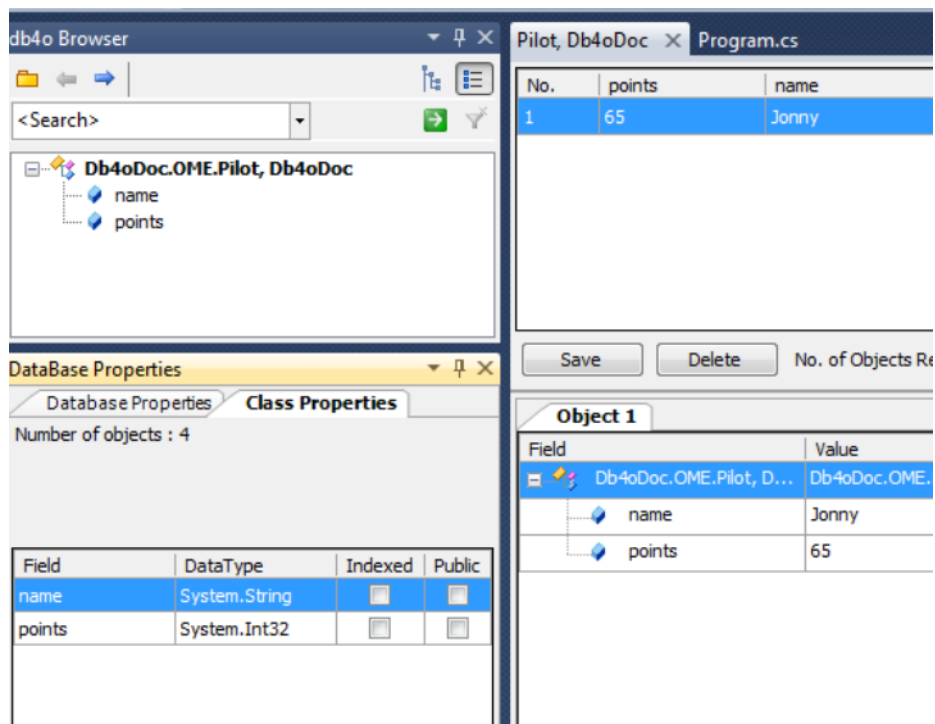
3.6.1. Duyệt cơ sở dữ liệu bằng OME

Ta đã đề cập cách tạo mới cơ sở dữ liệu và lưu trữ các đối tượng. Trước tiên, tạo và lưu trữ một vài đối tượng:

```
IObjectContainer db = Db4oFactory.OpenFile(YapFileName);
try
{
    Pilot pilot1 = new Pilot("Michael Schumacher", 100);
    db.Store(pilot1);
    Console.WriteLine("Stored {0}", pilot1);
    Pilot pilot2 = new Pilot("Rubens Barrichello", 99);
    db.Store(pilot2);
    Console.WriteLine("Stored {0}", pilot2);
}
finally
{
    db.Close();
}
```

Có thể nhìn thấy kết quả trong *Object Manager* bằng cách chọn *Tools* → *Object Manager Enterprise* → *Connect* (hoặc sử dụng *shortcut* từ thanh công cụ) và duyệt file *ome.db4o* trong thư mục *Local Application Data*.

Sau khi kết nối, màn hình hiện ra như sau:



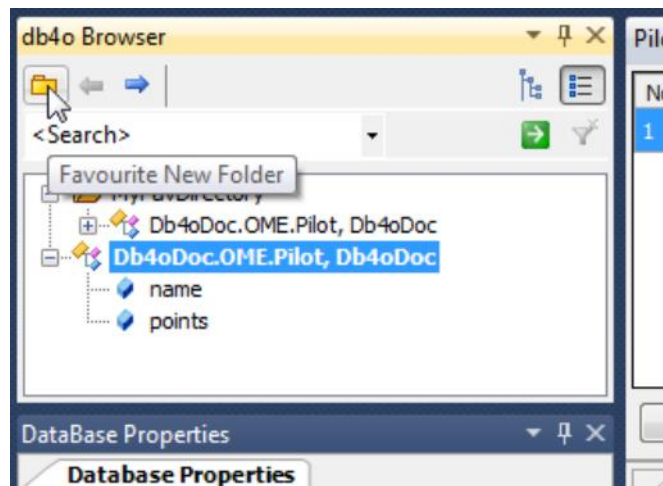
Hình 4. Giao diện duyệt CSDL trên OME

Giao diện gồm:

- *Db4o Browser*: cửa sổ hiển thị chi tiết cơ sở dữ liệu db4o đang được mở
- *Database Properties*: cửa sổ hiển thị các thuộc tính của lớp đang chọn trong cơ sở dữ liệu đang mở
- *Build Query*: cửa sổ cho phép xây dựng truy vấn bằng chức năng kéo và thả
- *Query Results*: cửa sổ hiển thị kết quả khi thi hành truy vấn

Cửa sổ *Db4o Browser* hiển thị lớp và các thuộc tính của lớp. Tại đây ta có thể thay đổi việc đánh chỉ mục bằng cách thay đổi trường “Indexed”.

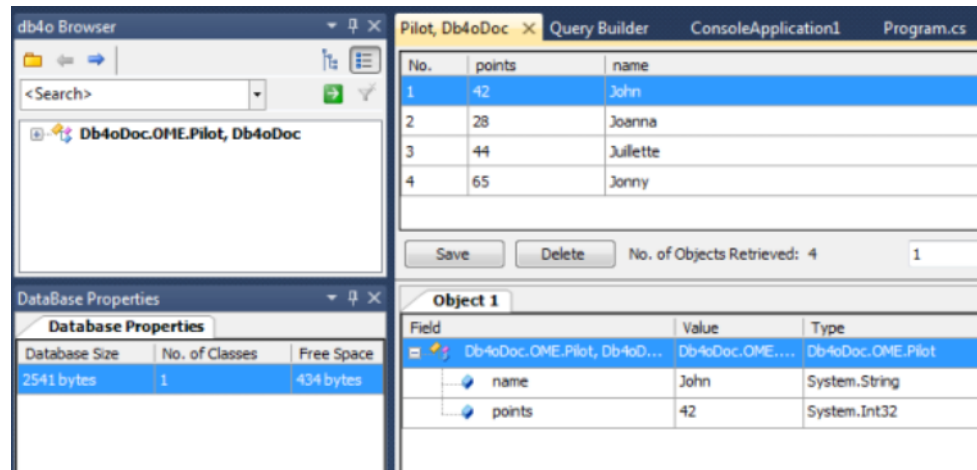
Ta có thể tạo từng thư mục riêng để chứa nhóm các lớp bằng cách nhấn vào biểu tượng thư mục trên thanh công cụ rồi kéo thả lớp vào thư mục đó.



Hình 5. Tạo thư mục quản lý đối tượng trong OME

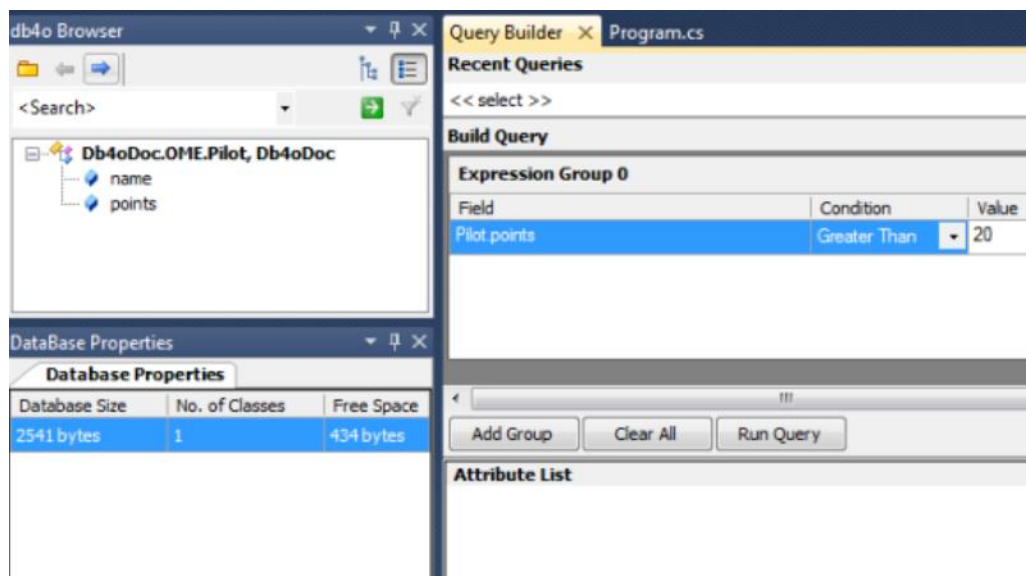
3.6.2. Truy vấn

Để hiển thị toàn bộ dữ liệu trong lớp, nhấn phải chuột vào lớp đó trong cửa sổ *Db4o Browser* rồi chọn “*Show All Objects*”. Danh sách đối tượng sẽ được hiển thị trong khung nhìn *Query Result*. Cửa sổ bên dưới hiển thị chi tiết của từng đối tượng. Ta có thể sửa đổi, xóa đối tượng bằng các nút lệnh trên cửa sổ.



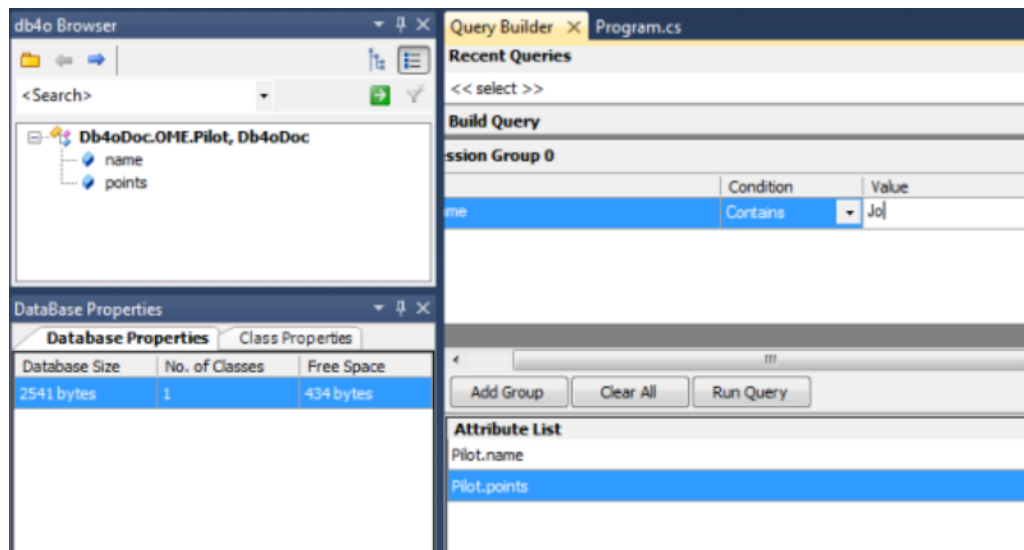
Hình 6. Giao diện truy vấn của OME

Để thực hiện các truy vấn ta dùng cửa sổ *Query Builder*. Kéo trường từ cửa sổ *Db4o Browser* vào cửa sổ *Query Builder*, thiết lập điều kiện và thực thi truy vấn.



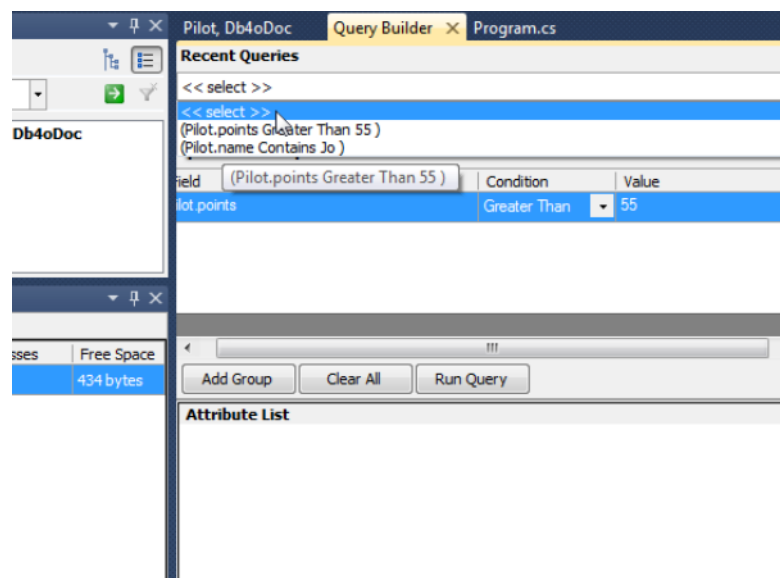
Hình 7. Cửa sổ Query Builder của OME

Nếu chỉ muốn chọn các trường và hiển thị kết quả thì kéo và thả các trường đó từ cửa sổ *Db4o Browser* vào cửa sổ *Attribute List*.



Hình 8. Kết quả Query Builder trên OME

Khi tạo truy vấn mới, truy vấn cũ được chọn lại từ hộp thả xuống phía trên.



Hình 9. Lịch sử truy vấn trên Query Builder

3.7. Các hệ thống truy vấn

Db4o cung cấp 3 hệ thống truy vấn, *Query-By-Example* (QBE), *Native Queries* (NQ), and the SODA API.

- Truy vấn *Query-By-Example* (QBE) thích hợp cho việc lưu trữ và tìm kiếm nhanh đối tượng.
- Truy vấn *Native Queries* (NQ) là giao diện truy vấn chính của db4o và được khuyến cáo sử dụng.

- Truy vấn *SODA* là truy vấn API mức thấp và có thể thích hợp cho các ứng dụng sản sinh tự động các yêu cầu truy vấn.

Ngoài ra, db4o hỗ trợ truy vấn *.NET LINQ*. Để sử dụng *LINQ*, cần tham chiếu đến *Db4objects.Db4o.Linq.dll* và khai báo lớp:

```
using System.Linq;  
using Db4objects.Db4o.Linq;
```

4. Phát triển ứng dụng sử dụng CSDL db4o

4.1. Giới thiệu

Trong phần xây dựng ứng dụng với CSDL HĐT này, nhóm hiện thực một ứng dụng quản lý hồ sơ của một trường học, sử dụng CSDL HĐT db4o nhằm mục đích tìm hiểu cách thức lưu trữ đối tượng, truy xuất dữ liệu. Thực hiện cơ chế xác thực người dùng, cơ chế phân quyền sử dụng người dùng nhằm bảo mật dữ liệu mức ứng dụng.

Chức năng chính của phần mềm:

- Quản lý thông tin cơ bản về giáo viên, sinh viên, môn học, lớp học, điểm sinh viên, ...
- Thực hiện việc xác thực, phân quyền người dùng để bảo mật dữ liệu.

Tuy nhiên, do thời gian và kiến thức có hạn, nên ứng dụng được phát triển ở mức cơ bản, giới hạn ứng dụng chỉ sử dụng trên một máy tính, lược đồ tổ chức đối tượng tương đối đơn giản so với mô hình thực tế. Nhóm cũng xác định rằng, ứng dụng này mang nặng tính học thuật, không có nhiều ý nghĩa trong ứng dụng quản lý thực tế.

4.2. Thiết kế mô hình đối tượng

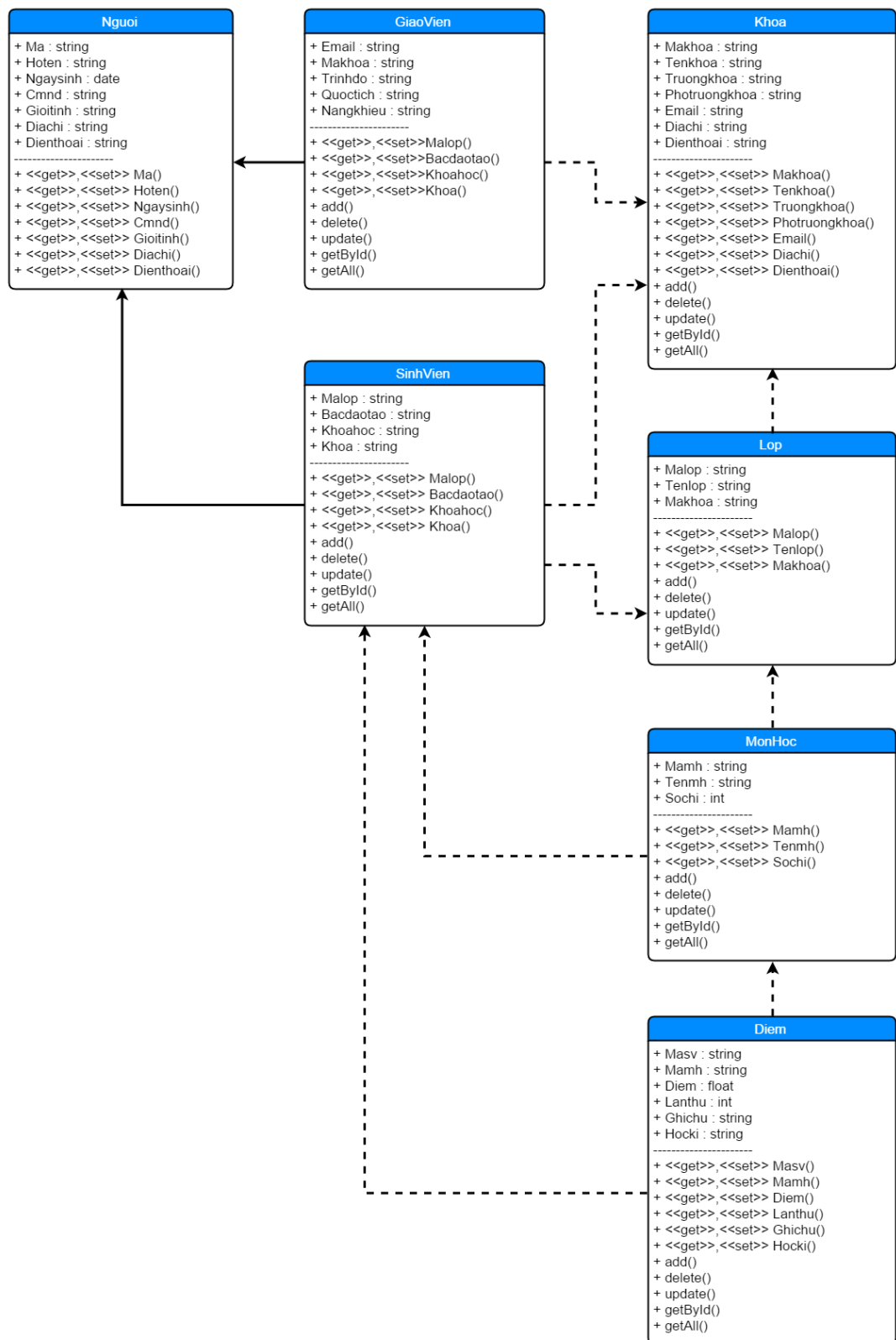
4.2.1. Các đối tượng chính

Như đã phân tích ở phần trên, mô hình ứng dụng cần có các lớp đối tượng:

- Người
- Giáo viên
- Sinh viên
- Khoa
- Môn học
- Điểm
- Lớp

Trong đó lớp *Sinh viên* và *Giáo viên* có những thuộc tính chung kế thừa từ lớp *Người*.

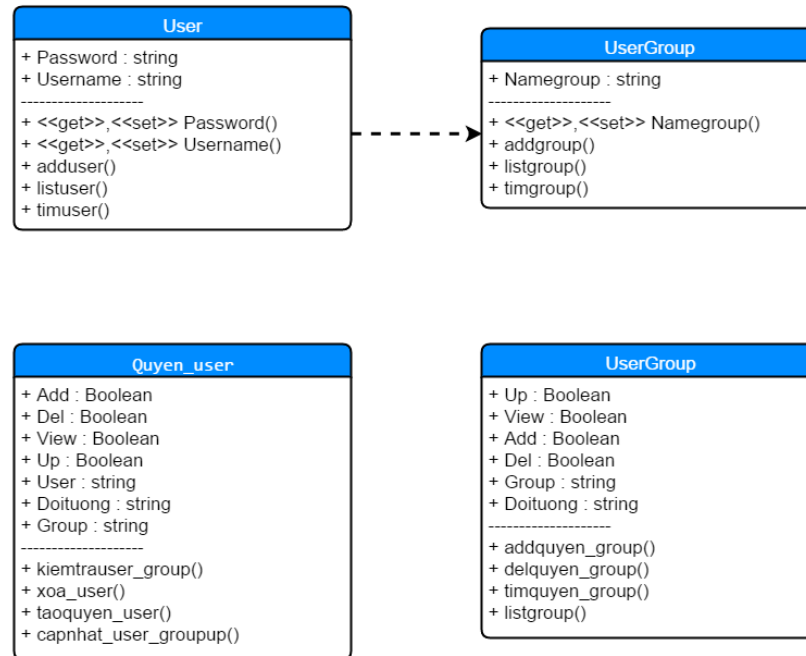
Như đã đề cập, ứng dụng này chủ yếu giúp nhóm hiểu hơn về CSDL HĐT thực tế, hiểu cách hoạt động cũng như cách xây dựng một ứng dụng trên đó, nên có thể có những phần chưa phù hợp với môi trường ứng dụng thực tế.



Hình 10. Sơ đồ các đối tượng chính trong ứng dụng

4.2.2. Các đối tượng trong phân quyền

Để quản lý việc xác thực và phân quyền người dùng, chúng ta cần thêm một số đối tượng phụ khác. Đối tượng *User* và *UserGroup* lưu trữ thông tin người dùng. Các đối tượng *Quyen_user*, *Quyen_group* lưu trữ danh sách các quyền truy cập của từng đối tượng trên.



Hình 11. Các đối tượng lưu trữ phân quyền

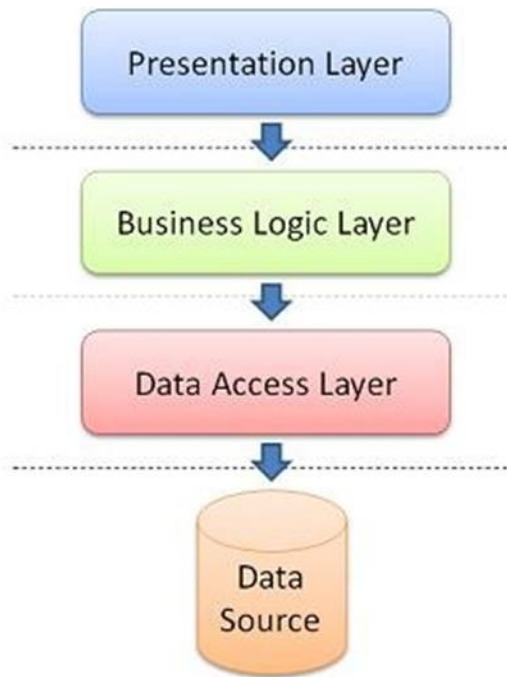
4.3. Hiện thực

4.3.1. Môi trường hiện thực

- Ngôn ngữ C# .NET
- Hệ điều hành Windows Vista hoặc hơn
- Trình biên dịch Visual Studio 2010 hoặc hơn
- Cơ sở dữ liệu hướng đối tượng Db4o

4.3.2. Cấu trúc chương trình

Chương trình được tổ chức gồm 3 tầng chính, được thể hiện như ở hình bên dưới.



Hình 12. Cấu trúc chính của chương trình

Tầng *DataAccess* là tầng gần với cơ sở dữ liệu nhất, trực tiếp giao tiếp với cơ sở dữ liệu. Tầng này nắm giữ thông tin kết nối với cơ sở dữ liệu, đảm nhiệm việc đóng mở kết nối với cơ sở dữ liệu và thực hiện các truy vấn. Các tầng khác muốn truy cập cơ sở dữ liệu đều phải thông qua tầng này.

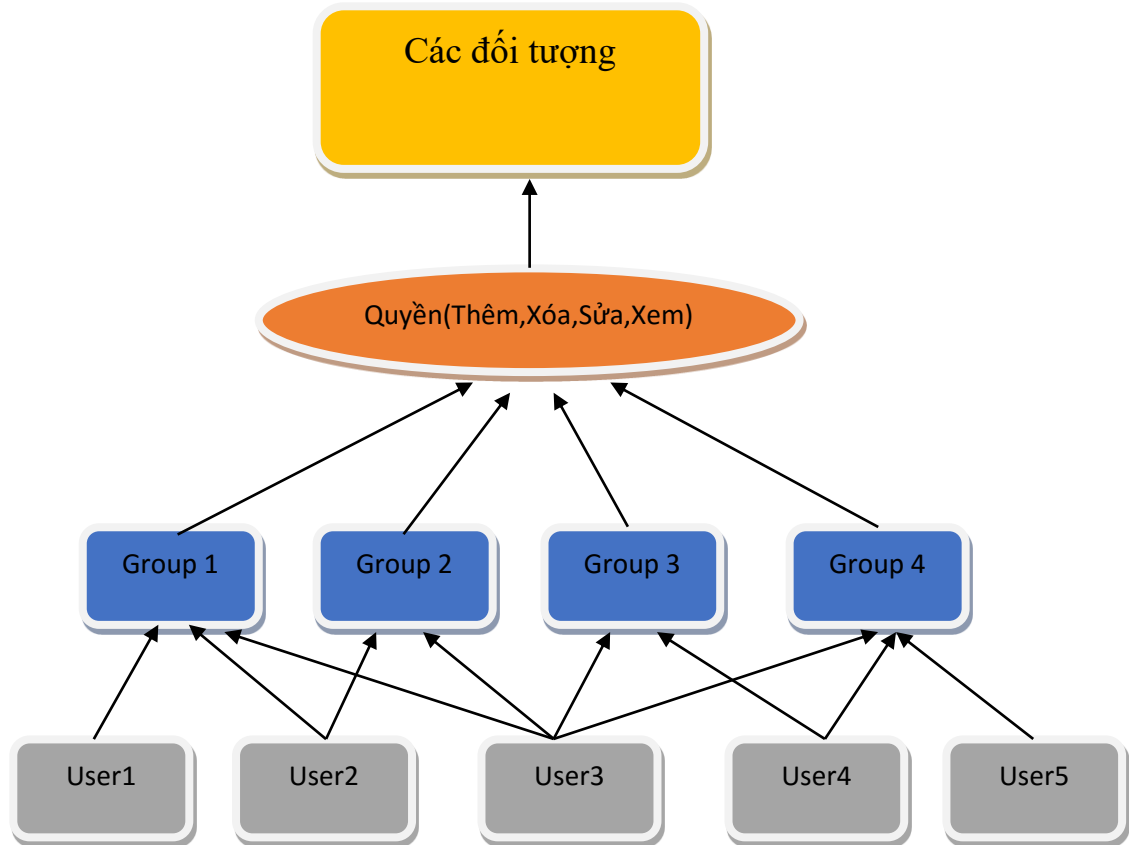
Tầng *BusinessLogic* là tầng trung gian giữa tầng *Presentation* và *DataAccess*. Nhiệm vụ của tầng này:

- Xử lý, tính toán các dữ liệu nhận được từ tầng *DataAccess* và chuyển cho *Presentation* để hiển thị.
- Lấy thông tin từ người dùng ở tầng *Presentation*, xử lý và chuyển lại cho *DataAccess* để cập nhật cơ sở dữ liệu.

Cuối cùng là tầng *Presentation*. Tầng này gồm các form giao diện hiển thị và nhập dữ liệu.

4.3.3. Mô hình phân quyền

Như đã đề cập ở phần trên, cơ sở dữ liệu hướng đối tượng trên db4o nói riêng, cũng như trên các cơ sở dữ liệu hướng đối tượng nói chung không có một DBMS thực thụ làm trung gian. Vì thế việc quản lý phải được thực hiện ở lớp ứng dụng. Ở đây, nhóm muốn phân quyền người dùng, hạn chế các quyền truy cập của từng nhóm người dùng đối với từng đối tượng.

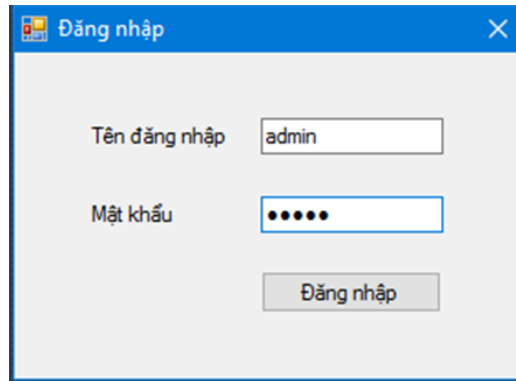


Hình 13. Mô hình phân quyền người dùng đối với các đối tượng

Khi một người dùng đăng nhập vào hệ thống, hệ thống sẽ kiểm tra quyền của người dùng đối với từng thao tác trên từng đối tượng dữ liệu. Một người dùng (*user*) có thể thuộc nhiều nhóm người dùng (*usergroup*). Các quyền của từng thao tác (xem, thêm, sửa, xóa) trên từng đối tượng sẽ được gán cho từng nhóm người dùng. Quyền của một người dùng là quyền kết hợp các nhóm mà nó thuộc vào.

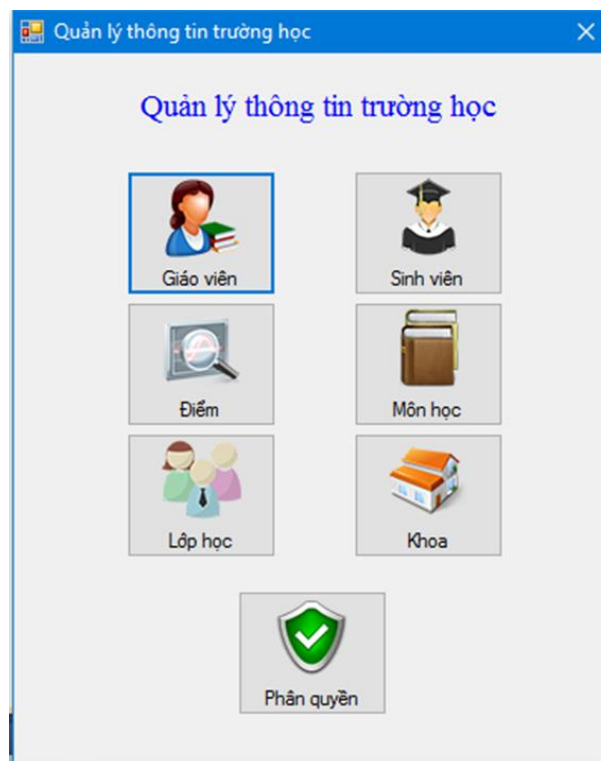
4.3.4. Giao diện chương trình

Giao diện đăng nhập cho phép người dùng đăng nhập vào hệ thống. Chương trình sẽ dựa vào thông tin đăng nhập để quyết định cho phép người dùng có hay không có quyền truy cập vào các phần của hệ thống.



Hình 14. Giao diện đăng nhập

Giao diện chính của chương trình. Khi người dùng đăng nhập thành công, chương trình sẽ hiển thị giao diện này. Đây là đầu mối dẫn đi đến các giao diện khác của hệ thống như: giao diện quản lý sinh viên, giao diện quản lý môn học, ... Tại đây, người dùng cũng được phân quyền để có hoặc không có quyền truy cập vào từng chức năng của hệ thống. Khi đăng nhập thành công sẽ xuất hiện giao diện quản lý thông tin trường học.



Hình 15. Giao diện chính sau khi đăng nhập thành công

Trong giao diện chính có 6 nút dẫn đến 6 form quản lý chính: Giáo viên, Sinh viên, Khoa, Lớp, Môn học, Điểm. Nút phân quyền còn lại để đi đến giao diện quản lý người

dùng. Trong từng giao diện quản lý chính sẽ bao gồm các chức năng cơ bản: Thêm, Sửa, Xóa, Tìm kiếm và Nhập hoặc Xuất danh sách bằng file Microsoft Excel.

Thêm: Chức năng này giúp người dùng thêm dữ liệu về một đối tượng như sinh viên, giáo viên, môn học, khoa, lớp học, điểm. Sẽ có những ô dữ liệu không được điền mà sẽ do hệ thống tự sinh ra theo một quy luật đã quy định sẵn. Đồng thời là những ô chứa các dữ liệu mà người dùng sẽ lựa chọn như khoa, lớp học. Còn lại là những ô mà người dùng sẽ điền dữ liệu, những ô này sẽ có những kiểm tra ràng buộc về cú pháp để loại bỏ dữ liệu rác hoặc dữ liệu nhập không đúng từ người dùng.

Xóa: Đây là chức năng giúp người dùng xóa từng dòng dữ liệu, người dùng chọn dữ liệu cần xóa sau đó chọn chức năng Xóa, hệ thống sẽ xóa dữ liệu mà người dùng đã chọn.

Sửa: Chức năng này giúp người dùng sửa dữ liệu mà người dùng muốn sửa. Người dùng sẽ chọn dữ liệu cần sửa để sửa dữ liệu. Tương tự như việc điền dữ liệu của chức năng thêm dữ liệu, người dùng sẽ điền dữ liệu đúng theo ràng buộc của hệ thống, nếu không hệ thống sẽ báo lỗi tới người dùng.

Nhập và xuất dữ liệu bằng file excel: Để thuận lợi cho người dùng, hệ thống sẽ cung cấp thêm chức năng là nhập và xuất dữ liệu file excel. Chức năng này giúp người sử dụng nhập dữ liệu bằng file excel, đây là cách hệ thống mô phỏng lại việc tạo dữ liệu. Hệ thống cũng đồng thời kiểm tra dữ liệu mà họ nhập có trùng lặp hay sai về các ràng buộc mà hệ thống quy định lại không. Thực chất là hệ thống đang mô phỏng chức năng thêm dữ liệu với nhiều dữ liệu cần nhập hay xuất ra từ hệ thống.

Chức năng tìm kiếm: Chức năng này giúp người dùng tìm kiếm thông tin nhanh trong hệ thống. Để sử dụng chức năng này, tùy vào từng giao diện là sinh viên, giáo viên,... mà chức năng tìm kiếm sẽ hiện thực những thông tin nào sẽ được dùng để tìm kiếm.

Quản Lý Thông Tin Sinh Viên

Tìm Kiếm Sinh Viên
 Khoa: Mã Sinh Viên: Lớp Học:

Danh sách

Mã sinh viên	Họ Tên	Mã lớp	Bậc đào tạo	Khóa học	Khoa	Cmnd	Ngày sinh	Giới tính	Địa chỉ	Điện thoại
201600000002	Đặng Hoài Trinh	LOP0008	Đại học	2012	Khoa Cơ Khí	650727817	06/10/1989	Nam	Hà Nội	0911257171
201600000005	Nguyễn Ngọc Nhung	LOP0008	Đại học	2013	Khoa Cơ Khí	970149415	11/05/1985	Nam	Khánh Hòa	0918254684
201600000012	Lê Văn Yến	LOP0006	Cao đẳng	2012	Khoa Cơ Khí	590520175	21/09/1986	Nữ	Hà Nội	0940104271
201600000014	Trần Thanh Nhung	LOP0006	Cao đẳng	2012	Khoa Cơ Khí	227816526	24/05/1986	Nữ	Ninh Thuận	0987400892
201600000019	Lý Văn Yến	LOP0005	Đại học	2013	Khoa Cơ Khí	639890984	12/03/1983	Nữ	Long An	0964325095
201600000020	Lê Hoài Hữu	LOP0008	Đại học	2012	Khoa Cơ Khí	885345574	23/11/1981	Nam	Khánh Hòa	0970204394
201600000025	Lý Hoài Hùng	LOP0006	Cao đẳng	2012	Khoa Cơ Khí	850483136	31/01/1987	Nam	Đà Lạt	0995171745
201600000031	Trần Hoài Hùng	LOP0008	Đại học	2015	Khoa Cơ Khí	163709284	25/01/1989	Nữ	Ninh Thuận	0941978561
201600000039	Trần Ngọc Yến	LOP0008	Đại học	2013	Khoa Cơ Khí	667947489	01/04/1985	Nam	Ninh Thuận	0907518700
201600000042	Lý Ngọc Nhung	LOP0008	Cao đẳng	2013	Khoa Cơ Khí	431394660	03/02/1982	Nam	Long An	0949593580
201600000047	Nguyễn Hoài Nhung	LOP0007	Đại học	2015	Khoa Cơ Khí	694388867	29/09/1982	Nam	Đà Lạt	0960746735

Thông tin chi tiết

Mã sinh viên: Điện thoại: Khóa học:

Họ tên sinh viên: Mã lớp: Khoa:

Ngày sinh: Giới tính: Cmnd:

Địa chỉ: Bậc đào tạo:

Hình 16. Giao diện quản lý thông tin sinh viên

Quản lý giáo viên

Quản Lý Giáo Viên

Tìm Kiếm
 Từ khóa:

Danh sách

Mã	Họ tên	Ngày sinh	Giới tính	Địa chỉ	Điện thoại	Email	Tên Khoa	Trình độ	CMND	Quốc tịch	Năng khiếu
GV0001	Đặng Thanh Hùng	28/02/1959	Nam	Bình Thuận	0989603974	hungdt291@outlook.com	Khoa Điện & Điện Tử	Thạc Sĩ	291740939	Việt Nam	Bóng Đá
GV0002	Lý Hoài Hùng	25/12/1962	Nữ	Kiên Giang	0901571706	hungth285@outlook.com	Khoa Cơ Khí	Thạc Sĩ	285352666	Việt Nam	Bơi
GV0003	Trần Văn Nhung	19/12/1961	Nam	Hà Nội	0964225063	nhungtv219@outlook.com	Khoa Cơ Khí	Thạc Sĩ	219026421	Mỹ	Bóng Đá
GV0004	Đặng Ngọc Trung	21/08/1957	Nam	Bình Thuận	0924656937	trungdn581@yahoo.com	Khoa Xây Dựng	Thạc Sĩ	581557920	Mỹ	Bơi
GV0005	Lê Thanh Yến	30/07/1951	Nam	Sóc Trăng	0958830637	yenik667@yahoo.com	Khoa Cơ Khí	Cử Nhân	667605437	Mỹ	Múa
GV0006	Đặng Thị Trung	16/09/1953	Nam	Ninh Thuận	0974599764	trungdt904@gmail.com	Khoa Khoa Học & Kỹ Thuật Máy Tính	Thạc Sĩ	904282177	Việt Nam	Hát
GV0007	Đặng Hoài Hữu	25/06/1963	Nữ	Ninh Thuận	0976569504	huudh392@outlook.com	Khoa Điện & Điện Tử	Thạc Sĩ	392006779	Việt Nam	Bơi
GV0008	Nguyễn Thị Trung	24/08/1950	Nữ	Sóc Trăng	0997420789	trungnt198@yahoo.com	Khoa Xây Dựng	Thạc Sĩ	198468780	Mỹ	Hát
GV0009	Đặng Ngọc Trinh	09/01/1950	Nam	Ninh Thuận	0944649471	trinhdn390@outlook.com	Khoa Cơ Khí	Tiến Sĩ	390195460	Việt Nam	Bơi
GV0010	Trần Hoài Hùng	28/08/1959	Nam	Long An	0920140845	hungth829@outlook.com	Khoa Điện & Điện Tử	Kỹ Sư	829460369	Mỹ	Hát
GV0011	Trần Thị Hữu	13/12/1959	Nam	Sóc Trăng	0932314125	huutt205@gmail.com	Khoa Cơ Khí	Cử Nhân	205605017	Việt Nam	Bóng Đá

Mã: Email:

Họ tên: Khoa:

Ngày sinh: Giới tính: Trình độ:

Địa chỉ: CMND:

Điện thoại: Quốc tịch: Năng khiếu:

Hình 17. Giao diện quản lý thông tin giáo viên

Quản lý Điểm Sinh Viên

Tìm kiếm
 Từ khóa

Danh sách

Mã SV	Mã MH	Điểm	Lần thử	Ghi chú	Học kì
201600000016	MH000029	5	3		2/2015
201600000034	MH000016	9	4		1/2012
201600000062	MH000008	9	4		2/2014
201600000009	MH000022	10	3		1/2014
201600000062	MH000002	10	3		2/2014
201600000096	MH000020	6	4		1/2015
201600000049	MH000003	10	4		1/2015
201600000049	MH000026	6	3		2/2012
201600000080	MH000004	9	2		2/2013
201600000063	MH000013	7	3		1/2015
201600000029	MH000021	9	2		2/2014

Mã Môn học Ghi chú
 Mã Sinh viên Lần thử
 Điểm Học kì

Hình 18. Giao diện quản lý thông tin điểm sinh viên

Quản lý môn học

Tìm kiếm
 Từ khóa

Danh sách

Mã MH	Tên MH	Số TC
MH000001	Cơ lưu chất	2
MH000002	Hệ cơ sở dữ liệu	3
MH000003	Đại Số	4
MH000004	Toán Cao Cấp 1	3
MH000005	Toán Cao Cấp 2	2
MH000006	Vật lý	3
MH000007	Trắc Địa Địa Cường	3
MH000008	Xác suất thống kê	3
MH000009	Vẽ Kỹ Thuật	3
MH000010	Cơ kết cấu	2
MH000011	Kỹ Thuật vẽ	1

Mã Môn học
 Tên Môn học
 Số chỉ

Hình 19. Giao diện quản lý thông tin môn học

Quản lý lớp học

Tìm kiếm
 Từ khóa

Danh sách

Mã lớp	Tên lớp	Tên Khoa
LOP0001	Điện tử 1	Khoa Điện & Điện Tử
LOP0002	Điện tử 2	Khoa Điện & Điện Tử
LOP0003	Điện tử 3	Khoa Điện & Điện Tử
LOP0004	Điện tử 4	Khoa Điện & Điện Tử
LOP0005	Cơ Khí 1	Khoa Cơ Khí
LOP0006	Cơ Khí 2	Khoa Cơ Khí
LOP0007	Cơ Khí 3	Khoa Cơ Khí
LOP0008	Cơ Khí 4	Khoa Cơ Khí
LOP0009	Máy Tính 1	Khoa Khoa Học & Kỹ Thuật Máy Tính
LOP0010	Máy Tính 2	Khoa Khoa Học & Kỹ Thuật Máy Tính
LOP0011	Máy Tính 3	Khoa Khoa Học & Kỹ Thuật Máy Tính
LOP0012	Máy Tính 4	Khoa Khoa Học & Kỹ Thuật Máy Tính

Mã lớp học Khoa
 Tên lớp học

Hình 20. Giao diện quản lý thông tin lớp học

Quản lý Khoa

Tìm Kiếm
Từ khóa:

Danh sách

Mã Khoa	Tên Khoa	Trưởng Khoa	P. Trưởng Khoa	Email	Địa chỉ	Điện thoại
KHCK	Khoa Cơ Khí	Nguyễn Phùng Tấn	Nguyễn Quốc Văn	khoack@school.edu.vn	10 Trường Chinh	08821236026
KHDT	Khoa Điện & Điện Tử	Vũ Trí Xương	Nguyễn Văn Khua	khoadt@school.edu.vn	10 Trường Chinh	08821236022
KHMT	Khoa Khoa Học & Kỹ Thuật Máy Tính	Trương Quang Trung	Trần Thanh Trang	khoamt@school.edu.vn	10 Trường Chinh	08382123601
KH XD	Khoa Xây Dựng	Phạm Văn Thành	Lê Quang Huy	khoaxd@school.edu.vn	10 Trường Chinh	08381212121

Mã Khoa: Tên Khoa: Trưởng Khoa: P. Trưởng Khoa:
Email: Địa chỉ: Điện thoại:

Hình 21. Giao diện quản lý thông tin khoa

Như đã trình bày ở trên, quyền của người dùng được xác định thông qua quyền của nhóm người dùng. Vì thế các bước để một người dùng được gán quyền sẽ thông qua các bước:

- Tạo nhóm người dùng mới nếu chưa tồn tại
- Gán quyền cho nhóm người dùng (từng quyền trên từng đối tượng)
- Tạo người dùng mới nếu chưa tồn tại
- Gán người dùng vào nhóm người dùng

Phân quyền người dùng

Phân quyền cho người dùng

Tim User
User:

User	Group	Password	Doituong
admin	admin	*****	sinhvien
admin	admin	*****	diemsinhv
admin	admin	*****	giaovien

Tạo user và thêm user vào group
Tên User:
PassWord:
Tên user:
Tên Group:

Tim Group
Group:

Group	Đối tượng	Sửa	Xem
admin	sinhvien	True	True
admin	diemsinhv	True	True
admin	giaovien	True	True

Tạo Group Thêm Hoặc Hủy Quyền Group
Tên Group:
Gán quyền
Tên Group: Đối tượng:
☐ Thêm ☐ Xóa
☐ Sửa ☐ Xem

Hình 22. Giao diện phân quyền người dùng

5. Kết quả

Qua quá trình tìm hiểu về CSDL HĐT và hiện thực ứng dụng trên db4o, nhóm đã thu được kết quả sau:

- Nắm bắt được cách thức vận hành của một CSDL HĐT cụ thể, db4o.
- Hiểu cơ chế lưu trữ dữ liệu là các đối tượng, các cách thức truy vấn các đối tượng dùng LINQ.
- Vận dụng và hiện thực cơ chế xác thực và phân quyền trên ứng dụng để đảm bảo cấp quyền truy cập, xử lý dữ liệu đúng quyền hạn của người dùng.

6. Kết luận

Sau khi nghiên cứu cơ sở dữ liệu hướng đối tượng trên db4o, nhóm đã thấy rõ hơn những đặc điểm chung của một cơ sở dữ liệu hướng đối tượng mà nhóm đã tìm hiểu và trình bày trong phần thuyết trình trên lớp.

Nhóm đã xây dựng được ứng dụng quản lý hồ sơ trường học đơn giản áp dụng db4o. Ứng dụng đã hiện thực một số chức năng cơ bản của một trình quản lý trường học. Tuy nhiên ứng dụng còn nhiều hạn chế: chưa thật sự đầy đủ các tính năng để có thể áp dụng ngay trong thực tế; chỉ chạy mô phỏng trên một máy tính; chưa thử nghiệm với dữ liệu thật sự lớn; cơ chế phân quyền còn nhiều bất cập, chưa thật sự có thể thay thế chức năng bảo mật của một hệ cơ sở dữ liệu.

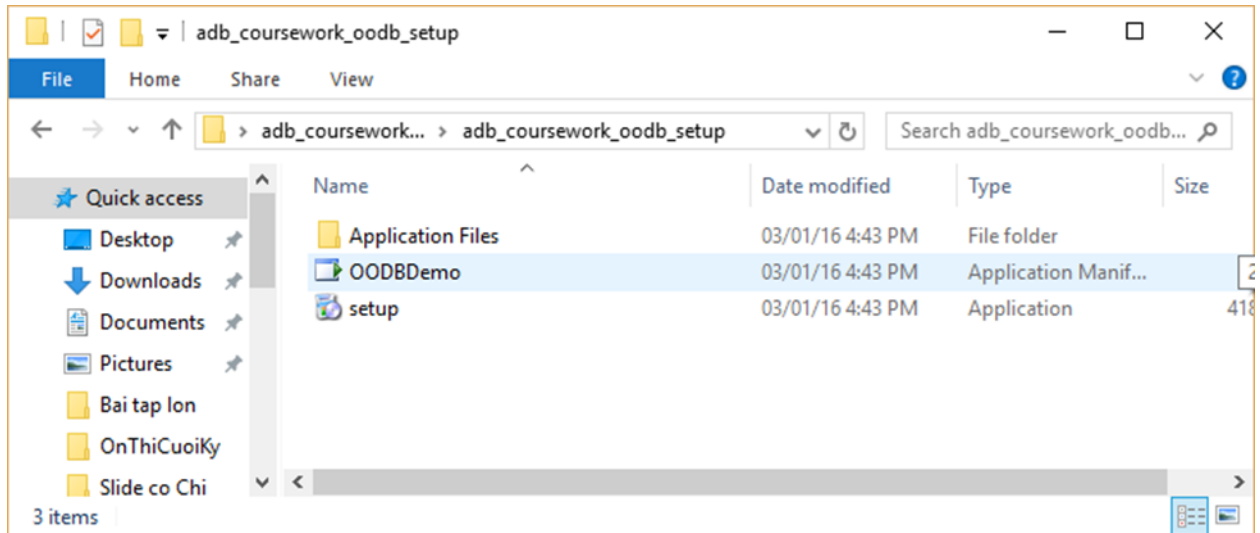
Trong quá trình tìm hiểu và xây dựng ứng dụng, nhóm nhận thấy rằng db4o thể hiện hầu hết các đặc điểm của một hệ cơ sở dữ liệu hướng đối tượng: định danh đối tượng, bao đóng, thừa kế... Db4o đã cung cấp những tính năng trên nền API hỗ trợ cho các lập trình viên phát triển các ứng dụng của họ. Db4o không cung cấp một hệ quản trị cơ sở dữ liệu, đây vừa là điểm mạnh và điểm yếu như đã trình bày ở trên.

Một điểm yếu khác của các ứng dụng xây dựng với cơ sở dữ liệu hướng đối tượng là khả năng bảo trì. Khi thay đổi mô hình dữ liệu của ứng dụng, ta phải sửa mã nguồn và biên dịch lại toàn bộ hệ thống ứng dụng. Điều này nhóm có thể thấy rõ trong quá trình làm ứng dụng quản lý ứng trường học trong bài tập lớn này. Đây được xem là điểm yếu chính khiến cơ sở dữ liệu hướng đối tượng kém hấp dẫn và chưa được phổ biến trong cộng đồng người lập trình.

7. Phụ lục

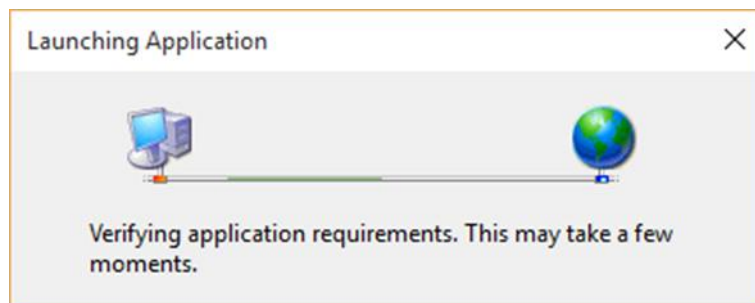
7.1. Hướng dẫn cài đặt ứng dụng

Giải nén tập tin *adb_coursework_oodb_setup.zip* đính kèm theo báo cáo này, ta được thư mục như Hình 23 sau:



Hình 23. Kết quả giải nén thư mục cài đặt

Nhấp đôi chuột vào file *setup.exe*, quá trình cài đặt bắt đầu và sẽ hoàn thành trong vài giây.



Hình 24. Tiến trình cài đặt phần mềm

Sau khi cài đặt xong, cửa sổ đăng nhập sẽ hiện ra như Hình 14. Nhập thông tin đăng nhập như sau:

- Tên đăng nhập: admin
- Mật khẩu: admin

Sau khi đăng nhập thành công, giao diện chính của chương trình sẽ hiện ra như Hình 15.

7.2. Hướng dẫn cài đặt môi trường phát triển ứng dụng

7.2.1. Cài đặt môi trường phát triển

Để biên dịch mã nguồn chương trình trong bài tập lớn này cần cài đặt:

- *Visual Studio 2010* hoặc *Visual Studio 2012* hoặc *Visual Studio 2013*.

Tải về tại <https://www.visualstudio.com/>

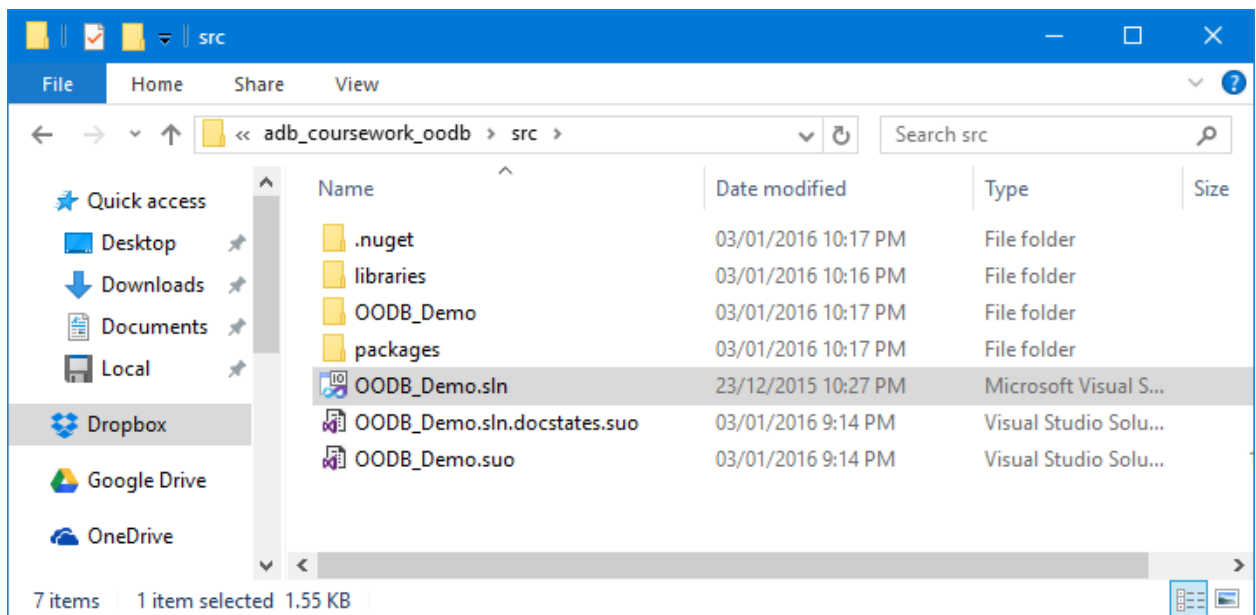
- *.NET Framework 4.0*.

Tải về tại <https://www.microsoft.com/en-us/download/details.aspx?id=17851>

Các thư viện phụ thuộc khác đã được chèn sẵn trong bộ mã nguồn.

7.2.2. Biên dịch mã nguồn

Giải nén tập tin *adb_coursework_oodb_src.zip* đính kèm theo báo cáo này, ta được thư mục như Hình 25 sau



Hình 25. Kết quả giải nén thư mục mã nguồn

Nhấp đôi chuột vào file *OODB_Demo.sln* hoặc sử dụng chức năng *File* → *Open* → *Project/Solution* của *Visual Studio* để mở file *OODB_Demo.sln*.

Vào *Build* → *Build Solution* (hoặc nhấn F7) để biên dịch chương trình.

Tham khảo

- [1] R. Elmasri, S.B. Navathe: *Fundamentals of Database Systems*, 6th Edition, Pearson Addison-Wesley, 2011
- [2] Malcolm Atkinson, Francois Bancilhon, David DeWitt, Klaus Dittrich, David Maier, Stanley Zdonik: *The Object-Oriented Database System Manifesto*, DOOD. Vol. 89. 1989
- [3] Michael Stonebraker, Lawrence A. Rowe, Bruce Lindsay, James Gray, Michael Carey, Michael Brodie, Philip Bernstein, David Beech: “*Third-Generation Database Manifesto*”, ACM SIGMOD Record, 19(3): 31-44, 1990
- [4] C.S.R. Prabhu: *Object-Oriented Database Systems, Approaches and Architectures*, 2001
- [5] Cliff, Layton: *Object-Oriented Database, Emerging Technologies*, Tech 3023: April 24, 2003
- [6] Hoàng Bảo Hùng, *Cơ sở dữ liệu hướng đối tượng*, NXB: Thông tin và truyền thông, 2015
- [7] Actian Support Services, <http://supportservices.actian.com/versant/default.html>