

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**
-----❧-----

LUẬN VĂN THẠC SĨ KHOA HỌC

**CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG THỜI GIAN
VÀ
XỬ LÝ TRUY VẤN TRONG CƠ SỞ DỮ LIỆU HƯỚNG
ĐỐI TƯỢNG THỜI GIAN**

**NGÀNH: CÔNG NGHỆ THÔNG TIN
MÃ SỐ:**

NGUYỄN THỊ HỘI

Người hướng dẫn khoa học: TS. NGUYỄN KIM ANH

HÀ NỘI 2006

LỜI CẢM ƠN

Lời đầu tiên của luận văn em xin gửi lời cảm ơn chân thành đến cô giáo, Tiến sỹ Nguyễn Kim Anh người đã trực tiếp hướng dẫn, giúp đỡ và tạo mọi điều kiện thuận lợi cho em từ lúc tìm hiểu, định hướng cũng như tìm kiếm tài liệu trong lĩnh vực Cơ sở dữ liệu hướng đối tượng thời gian cho đến lúc hoàn thành luận văn. Chúc cô mạnh khỏe, công tác tốt và ngày càng có nhiều kết quả mới từ lĩnh vực nghiên cứu mà cô yêu mến.

Em xin được gửi lời cảm ơn đến bạn bè và những người thân trong gia đình đã tạo điều kiện cũng như giúp đỡ em mọi mặt trong quá trình hoàn thành luận văn. Chúc mọi người gặp nhiều may mắn.

Em cũng xin gửi lời cảm ơn đến tất cả các thầy cô giáo của Khoa Công Nghệ Thông Tin và Trung tâm Đào tạo Sau Đại học của Trường Đại học Bách Khoa Hà Nội đã giúp đỡ, chỉ bảo, tạo mọi điều kiện cho em hoàn thành được luận văn này.

Hà nội, mùa thu 2006!

MỞ ĐẦU

Quản lý dữ liệu là một phần không thể thiếu đối với việc phát triển các hệ thống thông tin, quản lý dữ liệu theo thời gian trong các hệ thống thông tin ứng dụng trong các lĩnh vực thay đổi thường xuyên và đòi hỏi nhiều yêu cầu như tài chính, ngân hàng, viễn thông, y tế đã làm cho các hệ quản trị cơ sở dữ liệu quan hệ không quản lý về mặt thời gian bộc lộ những hạn chế. Điều này đòi hỏi các nhà nghiên cứu về cơ sở dữ liệu phải đưa ra những kết quả khả quan hơn, những hệ quản trị mới hơn, các kiểu xử lý khác làm sao để phục vụ được ngày càng tốt hơn nhu cầu mà thực tế đòi hỏi. Vì thế, một hướng nghiên cứu cơ sở dữ liệu mới ra đời đó là xem xét dữ liệu dưới dạng các đối tượng như nó vốn tồn tại trên thực tế và thêm yếu tố thời gian để đáp ứng các yêu cầu cho các ứng dụng quản lý về mặt thời gian.

Cơ sở dữ liệu đối tượng thời gian mới được nghiên cứu vào những năm 90 của thế kỷ 20 và những năm đầu của thế kỷ 21. Một bất ngờ là các nhà khoa học đã thu được những kết quả tốt hơn mong đợi và hứa hẹn một miền ứng dụng rộng lớn sẽ được sử dụng các kết quả này khi đưa vào thực tế. Cơ sở dữ liệu hướng đối tượng thời gian không những thu được các kết quả về cơ sở lý thuyết như các đại số nghiên cứu về cơ sở dữ liệu thời gian (đại số TA, OSAM*/T, ...) [SSH-98] mà còn thu được các kết quả về mặt thực hành là đã xây dựng được các hệ quản trị cơ sở dữ liệu hướng đối tượng thời gian với đầy đủ các tính năng bao gồm mô hình, ngôn ngữ định nghĩa và cả ngôn ngữ truy vấn đầy đủ, toàn vẹn trên dữ liệu hướng đối tượng thời gian (TOOM, TODL, TOQL). Vì lý do như vậy, cùng với sự hướng dẫn của cô giáo Tiến sỹ Nguyễn Kim Anh, em đã chọn tìm hiểu và nghiên cứu về ***“Cơ sở dữ liệu đối tượng thời gian và xử lý truy vấn trong cơ sở dữ liệu đối tượng thời gian”***

Nội dung chính của luận văn là giới thiệu về mô hình cơ sở dữ liệu hướng đối tượng thời gian trong các hệ thống thông tin (TOOBIS). Và để có được các nguyên tắc xử lý bên dưới của ngôn ngữ định nghĩa cũng như ngôn ngữ truy vấn thì

luận văn giới thiệu đại số TA là cơ sở toán học cho xử lý cơ sở dữ liệu hướng đối tượng thời gian trong TODL và TOQL. TOQL được giới thiệu để minh họa lại rõ nét hơn các toán tử cũng như các toán hạng trong cơ sở dữ liệu hướng đối tượng thời gian.

Trong thời gian tìm hiểu và nghiên cứu đề tài, em đã đạt được những kết quả bước đầu về cơ sở dữ liệu hướng đối tượng thời gian, làm cơ sở để hoàn thành cuốn luận văn này. Nội dung của luận văn bao gồm 4 chương :

Chương I: Tổng quan

Chương II: CSDL hướng đối tượng thời gian trong các hệ thống thông tin

Chương III: Cơ sở toán học cho xử lý CSDL hướng đối tượng thời gian

Chương IV: TOQL một ngôn ngữ truy vấn cho CSDL hướng đối tượng thời gian

Chương V: Kết luận của luận văn

Nội dung của luận văn mới chỉ là những vấn đề cơ sở của cơ sở dữ liệu hướng đối tượng thời gian, những vấn đề sâu hơn mong rằng sẽ được trình bày trong một thời gian gần nhất.

CHƯƠNG I – TỔNG QUAN

Giới thiệu

Cơ sở dữ liệu là một thành phần không thể thiếu trong quá trình phát triển các hệ thống thông tin. Trong thời gian gần đây, các ứng dụng thông tin không chỉ dừng lại ở các ứng dụng trong lĩnh vực xử lý dữ liệu kinh doanh, thương mại mà còn phát triển sang các lĩnh vực khác như trí tuệ nhân tạo, thông tin văn phòng, đa phương tiện... Do vậy, cơ sở dữ liệu (CSDL) không còn chỉ có kiểu quan hệ truyền thống mà còn phát triển thành CSDL đối tượng, với các yêu cầu về mặt quản lý thời gian và xử lý dữ liệu theo thời gian. CSDL đối tượng có yêu cầu về mặt quản lý thời gian đã được phát triển nhằm đáp ứng các ứng dụng thông tin một cách đầy đủ hơn.

Trong chương này luận văn trình bày những khái niệm cơ bản của CSDL đối tượng và CSDL thời gian. Phần 1.1 sẽ trình bày về CSDL thời gian cũng như các bài toán liên quan và phần 1.2 sẽ trình bày về CSDL đối tượng.

1.1 Cơ sở dữ liệu thời gian

1.1.1 Khái niệm

Hệ quản trị cơ sở dữ liệu (DataBase Management System - DBMS) [NKA-04] là một hệ thống phần mềm cho phép tạo lập cơ sở dữ liệu (DataBase – CSDL) và điều khiển mọi truy nhập đối với CSDL đó.

Hệ cơ sở dữ liệu có 4 thành phần [NKA-04] :

- CSDL hợp nhất: CSDL của hệ có hai tính chất cơ bản là tối thiểu hóa dư thừa và được chia sẻ.
- Những người sử dụng: Người sử dụng của hệ là bất kỳ một cá nhân nào có nhu cầu truy nhập CSDL, bao gồm những người sử dụng cuối, những người viết chương trình, những người quản trị CSDL.
- Phần mềm DBMS

- Phần cứng: Phần cứng của hệ thống bao gồm các thiết bị nhớ thứ cấp được sử dụng để lưu trữ CSDL.

Sau đây là một số khái niệm về các hệ quản trị CSDL thông thường và thời gian [NN-02] :

- Snapshot DataBases Management Systems (DBMS): các DBMS truyền thống không hỗ trợ cho bất kỳ kiểu thời gian nào.
- Valid Time (Historical) DBMSs: các DBMS chỉ hỗ trợ thời gian hiệu lực.
- Transaction Time (Rollback) DBMSs: các DBMS chỉ hỗ trợ thời gian giao dịch.
- Bi_Temporal DBMSs: các DBMS có hỗ trợ cả hai kiểu thời gian giao dịch và hiệu lực.

1.1.2 Các vấn đề được nghiên cứu trong CSDL thời gian

Trong các hệ quản trị CSDL thời gian thì thuộc tính thời gian được xem là một bộ phận gắn kết với hệ quản trị CSDL thời gian.

Một số vấn đề chính khi nghiên cứu CSDL thời gian là:

- Ngữ nghĩa của dữ liệu thời gian
- Mô hình hóa trình diễn dữ liệu thời gian
- Ngôn ngữ truy vấn trên các hệ quản trị CSDL thời gian
- Thiết kế CSDL thời gian
- Cài đặt và thực thi CSDL thời gian
- Các kỹ thuật tối ưu hóa trong thực thi CSDL thời gian

1.1.3 Ngữ nghĩa của dữ liệu thời gian

Trong CSDL thời gian có hai khái niệm được quan tâm là thời gian hiệu lực và thời gian giao dịch. Khi nói đến ngữ nghĩa của dữ liệu thời gian, chúng ta sẽ nói đến thời gian hiệu lực và thời gian giao dịch.

Thời gian hiệu lực của một sự kiện là tất cả các khoảng thời gian, có thể trong quá khứ, có thể là hiện tại, cũng có thể là tương lai khi sự kiện được xem là đúng.

- Thời gian hiệu lực ghi nhận quá trình thay đổi trạng thái của các sự kiện trong thực tế.
- Thời gian hiệu lực của một sự kiện không phải bao giờ cũng được ghi nhận trong CSDL vì một số lí do như: thời gian hiệu lực của một sự kiện là không thể biết hoặc khoảng thời gian hiệu lực không xác định được một cách chính xác...

Thời gian giao dịch (Transaction Time - TT) của một sự kiện tương đối với CSDL là khoảng thời gian trong đó sự kiện được ghi nhận và tồn tại trong CSDL.

- Thời gian giao dịch ghi nhận sự thay đổi trạng thái của chính bản thân CSDL.
- Các ứng dụng có các yêu cầu hỗ trợ khả năng truy cập đến các trạng thái trước đó của CSDL thì chúng cần sự hỗ trợ thời gian giao dịch.
- Thời gian giao dịch bị chặn hai đầu bởi các thời điểm khi CSDL được khởi tạo và trạng thái hiện thời của chính nó.
- Thời gian giao dịch còn có tính chất là tăng tuần tự và dễ dàng cho việc ghi nhận một cách tự động bởi hệ quản trị CSDL đó.

1.1.4 Mô hình hóa thể hiện dữ liệu thời gian

Để biểu diễn và xây dựng các hệ quản trị CSDL thời gian, người ta mở rộng các hệ quản trị CSDL thông thường (hệ quản trị CSDL quan hệ, hệ quản trị CSDL điểm nổi điểm...) theo một trong hai hoặc cả hai cách sau đây:

- Ghi nhận thông tin thời gian liên quan đến mỗi bộ dữ liệu.
- Ghi nhận thông tin thời gian liên quan đến mỗi thuộc tính trong từng bộ dữ liệu.

Cách thứ nhất, mỗi bộ dữ liệu được thêm 4 trường dùng để lưu các giá trị thời gian bao gồm: $\{TT_B, TT_E, VT_B, VT_E\}$

- TT_B : Thời gian bắt đầu giao dịch
- TT_E : Thời gian kết thúc giao dịch
- VT_B : Thời gian hiệu lực bắt đầu
- VT_E : Thời gian hiệu lực kết thúc

Sau khi thêm 4 trường này, cách biểu diễn của các bộ dữ liệu vẫn ở dạng chuẩn 1 (1NF) và chúng vẫn được hỗ trợ trong hầu hết các hệ quản trị CSDL thông dụng hiện nay. Nhưng nếu các thuộc tính trong cùng một bộ dữ liệu thay đổi không đồng thời với nhau thì cách thể hiện này sẽ vi phạm nguyên tắc dư thừa dữ liệu.

Cách thứ hai biểu diễn thông tin rất phức tạp vì mỗi thuộc tính của mỗi bộ dữ liệu sẽ có các giá trị thời gian riêng tương ứng với nó. Do đó, các hệ quản trị CSDL phải hỗ trợ thêm các phương pháp lưu trữ, xử lý, và truy vấn hiệu quả hơn. Đối với trường hợp này, sử dụng CSDL hướng đối tượng thời gian sẽ có nhiều ưu điểm hơn (sẽ nói chi tiết trong phần 1.2).

1.1.5 Ngôn ngữ truy vấn trên các hệ CSDL thời gian

Để thiết kế xây dựng một ngôn ngữ truy vấn mới, yêu cầu phải tuân theo một số điều kiện như:

- Tương thích với các hệ quản trị CSDL đã có.
- Việc cài đặt ngôn ngữ truy vấn mới không quá khó khăn.
- Ngôn ngữ truy vấn mới phải có cú pháp giống các ngôn ngữ truy vấn đã có.
- Ngôn ngữ truy vấn mới phải có khả năng diễn đạt cao hơn và hỗ trợ nhiều hơn các ngôn ngữ truy vấn đã có.

Với các yêu cầu như vậy, việc thiết kế một ngôn ngữ truy vấn mới là một

công việc rất khó khăn, tốn nhiều thời gian và công sức mà chưa chắc chắn rằng việc phổ dụng nó đã dễ dàng.

Do vậy, giải quyết vấn đề truy vấn trên CSDL thời gian chính là mở rộng các ngôn ngữ truy vấn thông dụng như SQL bằng cách thêm yếu tố thời gian. Đối với CSDL thời gian hướng đối tượng có một số ngôn ngữ truy vấn được xây dựng riêng như TOQL, TOOSQL, OQL/T. Trong chương 4 luận văn sẽ trình bày về TOQL.

1.1.6 Thiết kế CSDL thời gian

Mục đích chính của quá trình thiết kế là khai thác tối đa các khả năng và các tiện ích của các hệ quản trị CSDL để phục vụ cho ứng dụng của hệ thống.

Các ứng dụng yêu cầu hỗ trợ các yếu tố thời gian thường phức tạp hơn các ứng dụng thông thường đồng thời cũng rất khó thiết kế. Các thiết kế CSDL thời gian thường được dựa trên các thiết kế CSDL thông thường, sau đó thêm yếu tố thời gian. Ví dụ:

- Với quá trình thiết kế logic thì đưa thêm yếu tố thời gian và mở rộng các phụ thuộc với yếu tố thời gian. Sau đó, quá trình chuẩn hóa dữ liệu với dữ liệu thời gian được định nghĩa, các thuật toán để chuẩn hoá dữ liệu cũng được thay đổi.
- Với quá trình thiết kế mức khái niệm cũng có những vấn đề tương tự. Tức là mức thiết kế khái niệm của các CSDL thời gian đều dựa trên mô hình thực thể liên kết (quan hệ) và một số dựa trên mô hình đối tượng (sẽ nói rõ trong phần CSDL thời gian hướng đối tượng).

Để chuyển đổi từ mô hình thực thể liên kết thông thường sang mô hình có yếu tố thời gian thường phải mở rộng thêm yếu tố thời gian và phải chuyển đổi một cách thủ công. Do đó, nếu sử dụng mô hình đối tượng và chuyển sang mô hình đối tượng thời gian thông qua các đối tượng, chúng ta chỉ cần thêm yếu tố thời gian vào các thuộc tính của đối tượng và các thao tác (hành vi) được tự động thêm vào các yếu tố thời gian.

1.1.7 Cài đặt các CSDL thời gian

Đối với việc cài đặt CSDL thời gian [NN-02] có hai hướng tiếp cận chính là: Hướng tiếp cận tích hợp (Intergrated) và Hướng tiếp cận sử dụng lại các hệ đã có (Layered).

Hướng tiếp cận tích hợp sử dụng nhân là một hệ thống CSDL sẵn có, sau đó mở rộng và chỉnh sửa thêm các yếu tố thời gian.

Ưu điểm chính của tiếp cận này là:

- Hệ quản trị CSDL mới có thể xử lý triệt để, hiệu quả với dữ liệu thời gian.
- Các ứng dụng CSDL sẽ truy xuất trực tiếp với hệ quản trị CSDL mà không cần qua thành phần trung gian nên hiệu quả hơn.
- Có thể sử dụng các tiện ích của hệ quản trị đối với dữ liệu phi thời gian.

Nhược điểm đáng kể của hướng tiếp cận này:

- Khối lượng công việc cần chỉnh sửa, thêm mới rất lớn và độ phức tạp truy xuất cao.
- Chi phí sản xuất và chi phí thời gian là đáng kể.

Hướng tiếp cận sử dụng các hệ sẵn có với mục đích chính là khắc phục nhược điểm thứ hai của cách tiếp cận tích hợp: giảm chi phí khi cài đặt.

- Chọn một hệ quản trị CSDL sẵn có, giữ nguyên nhân hệ thống này.
- Xây dựng một mức trung gian cung cấp các tính năng hỗ trợ thời gian, thực hiện biến đổi các yêu cầu liên quan đến dữ liệu thời gian thành các yêu cầu thông thường mà hệ quản trị CSDL ở mức dưới có thể hiểu và xử lý được.
- Chuyển đổi các câu truy vấn cho hệ quản trị CSDL ở mức dưới thực hiện.
- Thực hiện một số bước xử lý thích hợp đối với kết quả nhận được.

Ưu điểm chính của hướng tiếp cận này:

- Giảm chi phí cài đặt nên tính khả thi cao hơn.
- Thích hợp cho các mẫu thử sử dụng trong nghiên cứu thực nghiệm.

Nhược điểm chính:

- Xử lý các dữ liệu thời gian không được triệt để do các hệ quản trị CSDL ở mức dưới không tận dụng hết các ưu điểm của dữ liệu thời gian.

1.2 Cơ sở dữ liệu hướng đối tượng

Hiện nay có một lớp các ứng dụng mới trong các lĩnh vực khác nhau như: Các ứng dụng thiết kế có máy tính trợ giúp (computer-aided design), các hệ thống đa phương tiện (multimedia information system), trí tuệ nhân tạo (artificial intelligence), hệ thống tin học văn phòng (office information system)... Đối với những ứng dụng này, các hệ quản trị CSDL đối tượng (Object DBMS) được xem là phù hợp hơn với các đặc trưng [MTO-00] :

- Chúng phải lưu trữ và thao tác với các kiểu dữ liệu trừu tượng và khả năng cho phép người dùng định nghĩa các kiểu mới.
- Biểu diễn các đối tượng ứng dụng có cấu trúc bằng mô hình quan hệ phẳng làm mất đi cấu trúc tự nhiên vốn có đối với các ứng dụng. Mô hình quan hệ tồn tại giữa các đối tượng không dễ diễn đạt nhưng trong mô hình đối tượng lại rất dễ thể hiện.
- Các hệ thống đối tượng, các ứng dụng phức tạp có thể được viết bằng một ngôn ngữ lập trình CSDL đối tượng xác định.

Sau đây, luận văn sẽ trình bày những nét cơ bản về CSDL đối tượng.

1.2.1 Đối tượng và cơ sở dữ liệu đối tượng

Hệ quản trị CSDL đối tượng là một hệ thống sử dụng “đối tượng - object” làm đơn vị cơ bản để mô hình hoá và truy xuất [MTO-00] . Đặc điểm chung của các mô hình dữ liệu đối tượng là:

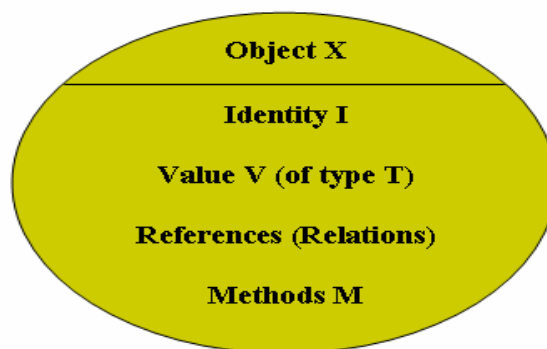
Định danh đối tượng: Mỗi thực thể trong thế giới thực được mô hình hóa vào trong CSDL được gọi là đối tượng. Mỗi một đối tượng được xác định duy nhất bởi định danh đối tượng. Định danh đối tượng do hệ thống sinh ra và được quản lý bởi hệ thống.

Các đối tượng phức hợp: Các mô hình dữ liệu đối tượng cho phép cấu trúc kiểu dữ liệu mới từ các kiểu dữ liệu đã có bởi cấu trúc bộ (tuple) và cấu trúc tập (set).

Sự phân cấp kiểu: Mô hình dữ liệu đối tượng cho phép có các kiểu con với các tính chất đặc biệt riêng của nó.

Một đối tượng có 4 đặc trưng [MV-03] :

- Định danh: Mỗi đối tượng có một định danh (Identifier - ID) duy nhất xác định trong toàn bộ hệ thống.
- Giá trị: Mỗi đối tượng có thể có một giá trị tại một thời điểm trong CSDL.
- Các tham chiếu: Các tham chiếu hay các quan hệ mà đối tượng có thể có.
- Phương thức: Các phương thức mà đối tượng có thể thực hiện.



Hình 1. 1 Mô hình một đối tượng

CSDL đối tượng được xét trên khía cạnh cấp phát, quản lý và huỷ bỏ các đối tượng và các hành vi (cách ứng xử) của chúng. Với mỗi đối tượng có hai vấn

đề cần xem xét đó là Trạng thái (Status) và Hành vi (Behavior).

- Trạng thái được biểu diễn như một tập các biến thể hiện (instance variable) hay thuộc tính (attribute) và chúng là các giá trị.
- Hành vi của đối tượng được biểu diễn thông qua các phương thức (method). Hành vi định nghĩa các hành động được phép trên những đối tượng này và được sử dụng để thao tác với các đối tượng đó.

Trong mô hình dữ liệu đối tượng chúng ta xét đến một số khái niệm liên quan:

- Các kiểu dữ liệu
- Quản lý đối tượng
- Truy vấn trên dữ liệu đối tượng ...

1.2.2 Các kiểu dữ liệu

Kiểu dữ liệu trừu tượng (Abstract Data Type - ADT) [MTO-00]: là một hình thức mô tả cho tất cả các đối tượng thuộc kiểu đó. Một ADT mô tả kiểu của dữ liệu bằng cách cung cấp một miền dữ liệu với cùng cấu trúc và các phương thức áp dụng cho các phần tử thuộc miền đó. Khả năng trừu tượng của ADT được gọi là đóng gói (encapsulation) các chi tiết cài đặt, được viết bằng một ngôn ngữ hình thức.

Thành phần kết hợp (Composition): Là một trong những đặc trưng tiêu biểu nhất trong mô hình dữ liệu đối tượng. Nó cho phép chia sẻ tham chiếu đến đối tượng thông qua các OID của nó.

Lớp (Class): Lớp là một nhóm các thể hiện đối tượng thuộc về một kiểu đã cho. Một lớp thường có dòng tộc của nó, đó là một tập hợp tất cả các đối tượng có kiểu đi kèm với lớp đó

Tập (Collection): Tập là một nhóm do người dùng định nghĩa chứa các đối tượng. Sự khác biệt giữa lớp và tập như sau:

Bảng 1. 1 Bảng so sánh giữa lớp và tập

<i>Lớp</i>	<i>Tập</i>
<i>Lớp có khả năng tạo ra các đối tượng</i>	<i>Tập không có khả năng tạo ra đối tượng</i>
<i>Một đối tượng là phần tử của một lớp duy nhất</i>	<i>Một đối tượng có thể ở trong nhiều tập</i>
<i>Quản lý lớp là không tường minh và do hệ thống tự động duy trì các lớp</i>	<i>Quản lý tập là tường minh</i>
<i>Một lớp bao gồm các mở rộng của một kiểu duy nhất và các mở rộng của các kiểu con của nó.</i>	<i>Một tập là nhóm các đối tượng do người dùng định nghĩa nên không thuần nhất về kiểu. Chúng chứa các đối tượng thuộc các kiểu không có liên quan với nhau</i>

1.2.3 Quản lý đối tượng

Quản lý định danh đối tượng: Định danh đối tượng OID được hệ thống cấp phát và được dùng để xác định một cách duy nhất mỗi đối tượng trong hệ thống.

Điều chế con trỏ: Trong các hệ quản trị CSDL đối tượng, để duyệt từ một đối tượng này đến một đối tượng khác, chúng ta dùng các biểu thức đường dẫn (path expression) chứa các thuộc tính với giá trị của chúng dựa trên đối tượng. Đối tượng này chính là các con trỏ. Có hai loại con trỏ là con trỏ trong bộ nhớ (in memory pointer) và con trỏ đĩa (disk pointer).

Con trỏ đĩa chính là cách sử dụng OID và con trỏ trong bộ nhớ dùng để duyệt từ đối tượng này đến đối tượng khác. Quá trình chuyển đổi con trỏ đĩa thành con trỏ trong bộ nhớ được gọi là điều chế con trỏ (pointer swizzling). Có hai lược đồ: phần cứng và phần mềm được dùng để điều chế con trỏ

Lược đồ phần cứng: Dùng cơ chế khuyết trang (page fault) của hệ điều hành đang sử dụng. Khi một trang được đưa vào bộ nhớ, tất cả các con trỏ trong đó đều được điều chế và chúng chỉ đến các khung nhớ ảo riêng. Các trang dữ liệu tương ứng với các khung ảo này chỉ được tải vào bộ nhớ khi có truy xuất đến chúng. Việc truy xuất trang sẽ sinh ra một khuyết trang của hệ điều hành mà nó phải được ghi nhận và xử lý.

Lược đồ phần mềm: Một bảng đối tượng được dùng cho mục đích điều chế con trỏ. Mỗi con trỏ được điều chế chỉ đến một vị trí trong bảng đối tượng.

1.3 Kết luận chương I

Trong chương này luận văn đã trình bày những khái niệm cơ bản về CSDL thời gian và CSDL đối tượng. Vấn đề đặt ra là khi các yêu cầu của các DBMS đối tượng lại đòi hỏi quản lý về mặt thời gian thì hướng giải quyết sẽ như thế nào? Trong chương 2, luận văn sẽ trình bày chi tiết về CSDL thời gian hướng đối tượng (Temporal Object Oriented DataBase - TOODB) trong các hệ thống thông tin. CSDL thời gian hướng đối tượng được xây dựng nhằm hỗ trợ cho các ứng dụng có yêu cầu quản lý về mặt thời gian đối với dữ liệu.

CHƯƠNG II – CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG THỜI GIAN TRONG CÁC HỆ THỐNG THÔNG TIN

Giới thiệu

Vào hai thập niên trước, nhiều nhà nghiên cứu đã hướng vào dữ liệu thời gian, hay dữ liệu thay đổi theo thời gian. Hầu hết các hệ thống thông tin máy tính hóa hiện nay (và đặc biệt là trong DBMSs) chỉ lưu trữ các thực thể trong thế giới thực với trạng thái hiện thời của nó mà không cung cấp hay không hỗ trợ việc lưu trữ những giá trị trạng thái trong quá khứ hay trong tương lai của các thực thể. Một số hệ thống thông tin cho phép duy trì các trạng thái trong quá khứ cho mỗi thực thể, nhưng những phương tiện này phần lớn phục vụ những mục đích kiểm toán và lưu trữ, cung cấp rất ít khả năng hỗ trợ để lấy lại và phân tích, xử lý thông tin. Về mặt kỹ thuật, thời gian mà mỗi mẫu tin đang đứng trong thế giới thực được gọi là thời gian hiệu lực, trong khi thời gian từ lúc chèn bản ghi vào trong một DBMS đến lúc xóa (logic) nó được gọi là thời gian giao dịch. Phụ thuộc vào việc hỗ trợ kiểu thời gian mà chúng cung cấp cho hai chiều thời gian, DBMSs được phân loại vào bốn phạm trù đã trình bày trong phần 1.1.1

Trong chương này luận văn sẽ trình bày về những vấn đề liên quan đến CSDL đối tượng thời gian. Mục 2.1 sẽ trình bày tổng quan về CSDL hướng đối tượng trong các hệ thống thông tin - Temporal Object Oriented DataBases within Information System (TOOBIS), mục 2.2 trình bày những mục tiêu chung của TOOBIS, mục 2.3 trình bày các kết quả đã đạt được và mục 2.4 trình bày hệ nền của TOOBIS.

2.1 Tổng quan

TOOBIS là một phương pháp luận mô hình hóa khái niệm mới, với cách nắm bắt ngữ nghĩa, phân tích yêu cầu và hình thức hoá thiết kế của các hệ thống thông tin thời gian. Đồng thời có chức năng lưu trữ một hệ quản trị cơ sở dữ

liệu hướng đối tượng (Object Oriented DBMS – OODBMS), với việc lưu kết quả trong một Hệ thống Quản lý Cơ sở dữ liệu Hướng đối tượng Thời gian (TOODBMS), nó cung cấp cơ sở cần thiết cho việc thực thi và bảo trì những ứng dụng thời gian.

Các ứng dụng hiện nay thực hiện theo hướng OODBMS bao gồm :

- Viễn thông: Quản lý mạng, Quản trị, Mô phỏng, Lập kế hoạch, ...
- Công nghiệp hàng không : Dữ liệu kỹ thuật và quản lý tài liệu kỹ thuật.
- Bảo vệ : Chuẩn bị các đặc vụ, Mô phỏng, Tập trận giả, ...
- Ngân hàng và bảo hiểm : Hệ thống nhà thương mại, Hệ thống thông tin khách hàng...
- Dịch vụ : Phân phối điện, Quản lý mạng vận tải, bản đồ học, ...
- Thương mại : (Những ứng dụng DSS về tài chính, ngân hàng, bảo hiểm).
- Sức khỏe : (Mô phỏng, Thao tác y học, Quản lý bệnh, dược lý).

Những lợi ích mà công nghệ TOODBMS mang lại trong quá trình xây dựng các ứng dụng thông tin:

- Tiện ích để phân tích và thiết kế, với các khía cạnh thời gian có hành vi động và tĩnh trên một miền ứng dụng.
- Giảm giá thành cho quá trình thiết kế những ứng dụng thời gian, phát triển, bảo trì và tiến hóa.
- Cải tiến việc thực thi những ứng dụng lưu kết quả.
- Chuẩn bị một khung linh hoạt để lưu dữ liệu thời gian và một ngôn ngữ truy vấn mạnh để thao tác trên chúng.

2.2 Mục tiêu chung của TOOBIS

TOOBIS nhắm vào việc phát triển một Hệ nền cạnh tranh để hỗ trợ vòng đời đầy đủ của các ứng dụng thời gian. Mục tiêu của TOOBIS như sau:

1. Cung cấp một phương pháp phân tích và thiết kế thời gian, bằng việc mở rộng và xây dựng để thích nghi với những phương pháp luận đang tồn tại.
2. Mở rộng chức năng chính bằng việc đề xuất O2 OODBMS, và hỗ trợ cả thời gian hiệu lực lẫn thời gian giao dịch. Những mở rộng này phải tương thích với các chuẩn ODMG và các điều khoản mà các chuẩn SQL/3 sắp định nghĩa .
3. Tuân theo việc phát triển những chuẩn ODBMS, có mặt trong các định nghĩa mà những chuẩn này chịu ảnh hưởng.
4. Làm giàu thêm chức năng của O2 OODBMS, cung cấp mô hình dữ liệu xây dựng để khởi tạo và thao tác trên các đối tượng thời gian.
5. Tăng cường các chuẩn cho ngôn ngữ định nghĩa đối tượng (Object Definition Language - ODL).
6. Mở rộng chuẩn cho ngôn ngữ truy vấn đối tượng ở mức cao (Object Query Language - OQL).
7. Tăng tính hiệu lực với cách tiếp cận và những sản phẩm TOOBIS (TOODBMS và Phương pháp luận) bằng việc phát triển các ứng dụng thực tế...

TOODBMS cung cấp cách thức lưu trữ kiến trúc mô hình dữ liệu để tạo mới và thao tác trên các đối tượng thời gian. Nó kế thừa đối tượng thời gian của O2 OODBMS, ngôn ngữ định nghĩa thời gian (Temporal Object Defined Language - TODL), bằng cách mở rộng ODMG ODL và mở rộng chuẩn ngôn ngữ truy vấn đối tượng mức cao (OQL).

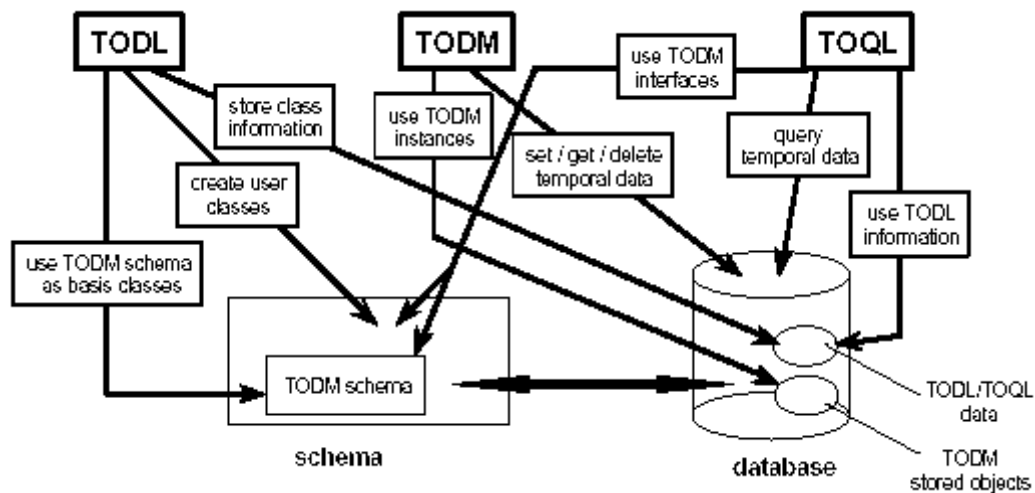
2.3 Kết quả thu được

Dự án TOOBIS đã phát triển một Hệ nền hỗ trợ vòng đời đầy đủ của các hệ thống thông tin thời gian. Hệ nền này bao gồm những thành phần sau :

1. Phương pháp luận thời gian hướng đối tượng (Temporal Object

Oriented Method - TOOM). TOOM được sử dụng trong pha phân tích và thiết kế hệ thống thông tin thời gian. TOOM, ngoài khả năng thực hiện việc nắm bắt và mô hình hóa, còn cung cấp cách thức xây dựng mô hình hoá bao gồm cả khía cạnh cấu trúc lẫn khía cạnh hành vi của hệ thống thông tin.

2. DBMS hướng đối tượng thời gian (TOODBMS). TOODBMS có thể sử dụng để định nghĩa, lưu trữ và truy vấn dữ liệu thời gian, cung cấp cơ sở hạ tầng cần thiết cho việc thực thi và thao tác của các hệ thống thông tin thời gian. TOODBMS gồm ba thành phần phân biệt được minh họa trong Hình 2.1

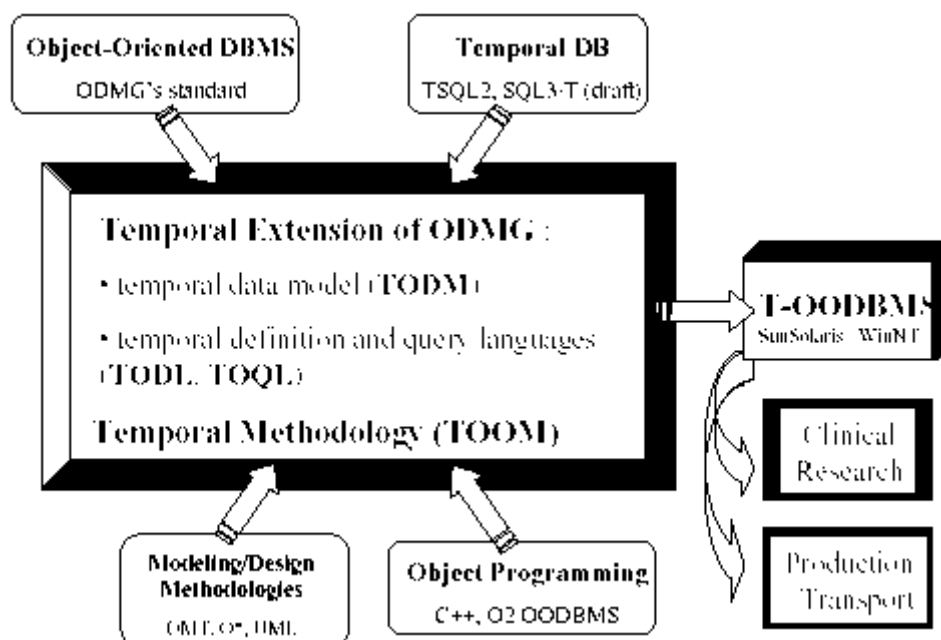


Hình 2. 1 Kiến trúc của TOODBMS

- i) Mô hình Dữ liệu Đối tượng Thời gian (Temporal Object Data Model - TODM). Mô đun này cung cấp các phương tiện cần thiết cho việc tạo thành, lưu trữ và thực thi các thao tác trên những đối tượng phi thời gian và thời gian: thêm mới và xóa bỏ thông tin thời gian, trích chọn thông tin theo giai đoạn thời gian ...
- ii) Ngôn ngữ Định nghĩa Đối tượng Thời gian (Temporal Object Define Language - TODL). Mô đun này cung cấp một phương thức ở mức cao để định nghĩa những đối tượng với đặc trưng thời gian. TODL mở rộng Ngôn ngữ Định nghĩa Đối tượng ODMG (ODL). Hơn nữa TODL còn

lưu trữ một từ điển (meta - data), thông qua người dùng mà họ truy vấn trên những kiểu sẵn có và xác định những thuộc tính của chúng.

- iii) Ngôn ngữ truy vấn Đối tượng Thời gian (Temporal Object Query Language TOQL), ở mức cao ngang mức ngôn ngữ truy vấn để trích chọn thông tin từ những cơ sở dữ liệu Thời gian. TOQL là một mở rộng của ODMG đối tượng với ngôn ngữ truy vấn (OQL), đồng thời cung cấp thêm tính hoạt động đến các ứng dụng có chiều thời gian, tương thích với việc ứng dụng kế thừa ... Kiến trúc hệ thống toàn bộ được minh họa trong Hình 2.2



Hình 2. 2 Kiến trúc tổng quan hệ nền TOOBIS

2.4 Hệ nền TOOBIS

2.4.1 Phương pháp luận hướng đối tượng thời gian

2.4.1.1 Mục tiêu của phương pháp luận

Những vấn đề trọng tâm khi xây dựng phương pháp luận hướng đối tượng là:

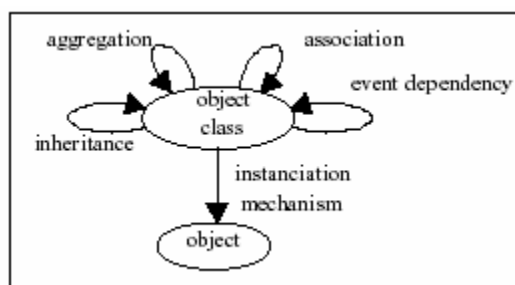
- Đơn vị thời gian dùng để điều khiển tổ chức vòng đời ứng dụng

- Ngữ nghĩa và miền thời gian của mỗi thuộc tính.
- Dữ liệu nào được thể hiện cho thích hợp để lưu trữ và quản lý trong cơ sở dữ liệu.
- Các ràng buộc nào được định nghĩa để đảm bảo việc gắn kết trong quá khứ, tương lai và hiện tại của dữ liệu thời gian.

Đây là những vấn đề then chốt trong việc mở rộng phương pháp luận thời gian, với các khái niệm về độ đo và trình diễn thời gian, các khái niệm cho việc quản lý lịch sử. Những khái niệm này cho phép cho mô hình hóa theo khía cạnh thời gian của ứng dụng ở mức khái niệm.

2.4.1.2 Phương pháp hướng đối tượng

Phương pháp hướng đối tượng cơ bản sử dụng một mô hình duy nhất, được phân phối vào ba nhóm: cấu trúc, hàm và hành vi. Những khái niệm chính của mô hình, được sử dụng để mô hình hoá hiện tượng trong thực tế, cùng với những mối quan hệ nằm bên dưới chúng được cho thấy trong Hình 2.3



Hình 2. 3 Các khái niệm cơ bản của mô hình đối tượng

Trong phần này chúng ta sẽ xem xét theo 2 thuộc tính là: Cấu trúc và Hành vi.

A - Thuộc tính cấu trúc

Hai khái niệm cơ bản của mô hình, là đối tượng và lớp đối tượng, được đưa ra từ sơ đồ hướng đối tượng. Mục đích chính của mô hình là tuân theo khái niệm đối tượng từ pha phân tích cho đến pha thực thi.

- Đối tượng biểu diễn cho một hiện tượng, một thực thể trong thực tế không gian ứng dụng.

- Lớp đối tượng mô tả một nhóm các đối tượng có cùng đặc trưng về hành vi và cấu trúc.
- Thuộc tính là một đặc trưng của một lớp. Tập hợp giá trị của một thuộc tính được định nghĩa thông qua khái niệm miền.

Mô tả cấu trúc của một lớp đối tượng đầy đủ bằng một tập các ràng buộc. Sự ràng buộc có thể được phân loại thành ba phạm trù :

1. Ràng buộc thừa kế, sự phân chia các mô hình, sự phủ và sự tách rời.
2. Ràng buộc thuộc tính, hạn chế các giá trị thuộc tính của một đối tượng, hoặc những giới hạn của các thuộc tính, định nghĩa những đối tượng liên quan hoặc những miền theo một chuẩn nhất định.
3. Ràng buộc duy nhất, được định nghĩa giữa một tập của các thuộc tính và một tập các đối tượng.

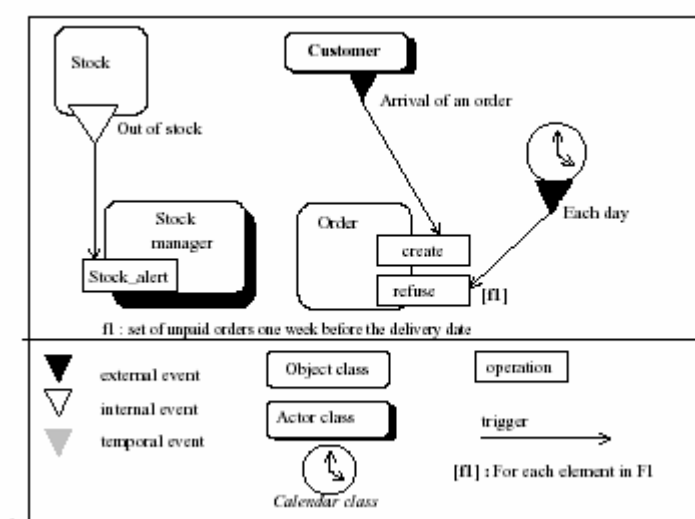
B - Thuộc tính hành vi

Ngoài khía cạnh cấu trúc, mô hình đối tượng cũng xem xét những khái niệm về sự kiện và thao tác theo khía cạnh hành vi của các hệ thống thông tin ở mức khái niệm.

Ba nhóm sự kiện được phân biệt như sau: Sự kiện ngoài, Sự kiện bên trong và Sự kiện thời gian.

- Sự kiện ngoài tương ứng với những sự kiện xuất hiện trong môi trường bên ngoài hệ thống thông tin.
- Sự kiện bên trong tương ứng với trạng thái bên trong thay đổi hoặc tương ứng với những đáp ứng lại của hệ thống do bên ngoài thay đổi.
- Sự kiện thời gian là một trạng thái đặc biệt của đồng hồ đo thời gian.

Sự kiện ngoài gắn liền với một lớp tác nhân, trong khi tất cả các sự kiện thời gian đều gắn liền với lớp lịch biểu. Sự kiện bên trong được gắn với những lớp đối tượng. Hình 2.4 tổng kết sự hợp nhất của các khái niệm khác nhau (Cấu trúc, hàm và Hành vi).



Hình 2. 4 Một lược đồ động

2.4.1.3 Mở rộng thời gian

Mở rộng thời gian tập trung vào trình bày ở mức phân tích, những khái niệm về yêu cầu quản lý thời gian theo khía cạnh của một ứng dụng cơ sở dữ liệu. Mở rộng thời gian của phương pháp luận hướng đối tượng (Temporal Object Oriented Method - TOOM) bao gồm những khía cạnh sau :

- Định nghĩa của một hoặc nhiều lịch biểu cho một ứng dụng,
- Mở rộng của các miền cơ sở đến các miền thời gian,
- Mở rộng lớp đối tượng tới các chiều thời gian,
- Mở rộng của các ràng buộc áp dụng trên những lớp đối tượng,
- Mở rộng thời gian của các sự kiện.

A - Định nghĩa lịch biểu

[EIV-99F] nói rằng một lịch biểu là một hệ thống ma trận được áp dụng với các đường thời gian. OOM đề xướng khái niệm lớp lịch biểu để định nghĩa những sự kiện thời gian của ứng dụng. Trong OOM ban đầu, lịch Gregorian được sử dụng ở mức đối tượng, lớp lịch biểu chỉ có một thể hiện đó là đồng hồ lịch Gregorian.

[EIV-99F] mở rộng định nghĩa của lớp lịch biểu để định nghĩa bất kỳ tiện ích

lịch biểu nào áp dụng cho ứng dụng. Một lịch biểu được mô tả trong [EIV-99F] bao gồm :

- Tên của lịch biểu,
- Gốc hay vị trí bắt đầu của lịch biểu,
- Một đơn vị cơ bản (nguyên tố),
- Một trật tự liệt kê của các thành tố với những chuyển đổi đơn vị lớn hơn và đơn vị nhỏ hơn của chúng,
- Một thao tác chuyển dịch từ một thời điểm đến một giai đoạn,
- Một thao tác chuyển đổi giữa những thành tố của cùng lịch biểu đó,
- Hai thao tác chuyển đổi, từ đâu (from) và tới đâu (to) của lịch biểu Gregorian (Hai thao tác này dễ dàng cho chuyển đổi giữa những lịch biểu với nhau).

B - Mở rộng miền cơ sở tới miền thời gian

Miền thời gian mới được sử dụng để biểu diễn thời gian trong cơ sở dữ liệu bao gồm có ba kiểu: thời điểm (Instant), giai đoạn (Period) và khoảng (Interval).

- Một thời điểm trình bày một điểm trên đường thời gian.
- Một thời kỳ hay giai đoạn trình bày số lượng của thời gian giữa hai thời điểm.
- Một khoảng, là một số lượng không bị hạn chế của thời gian.

Mỗi thể hiện của ba kiểu này được biểu thị trong một đơn thể của một lịch biểu. Những miền thời gian chung bao gồm: (*INSTANT*<*calendar, granule*>, *PERIOD*<*calendar, granule*> và *INTERVAL*<*calendar, granule*>).

Các kiểu cổ điển: ngày tháng, thời gian và ngày tháng, thời gian được tìm thấy trong DBMSs cổ điển. Còn *INSTANT-A*<*calendar, granule*>, *PERIOD-A*<*calendar, granule*> được sử dụng để chỉ miền thời gian tuyệt đối. Miền

thời gian tương đối được trình bày bởi: *INSTANT-R<calendar, granule>* và *PERIOD-R<calendar, granule>*.

C - Mở rộng lớp đối tượng tới các chiều thời gian

Trong mục này luận văn sẽ trình bày các chiều thời gian được mở rộng trong lớp đối tượng bao gồm: Lớp thông thường và lớp thời gian

Lớp thông thường

Một lớp thông thường không hỗ trợ thời gian là một lớp không có yêu cầu về quản lý thời gian. Sau đây là các thuộc tính lớp thông thường không hỗ trợ quản lý mặt thời gian :

- Một thuộc tính của một lớp thông thường có thể là một quan hệ tới lớp khác hoặc một thuộc tính khác, nó được định nghĩa trong lớp đối tượng.
- Một ràng buộc định nghĩa cho một lớp thông thường có thể là một ràng buộc thuộc tính hoặc một ràng buộc thừa kế, nó được mô tả trong lớp đối tượng.
- Phương thức của một lớp thông thường có thể là những truy vấn hoặc những thao tác cơ bản, chúng có thể được áp dụng ở mức đối tượng hoặc mức lớp. Những kiểu thao tác cơ bản như tạo mới, cập nhật, xoá hoặc huỷ bỏ.

Sự kiện bên trong có thể được định nghĩa trong lớp thông thường.

Lớp thời gian

Một lớp thời gian là một lớp với yêu cầu quản lý về mặt thời gian. Lớp thời gian cho phép quản lý toàn vẹn thời gian hiệu lực và/hoặc quản lý thời gian giao dịch trong định nghĩa lớp.

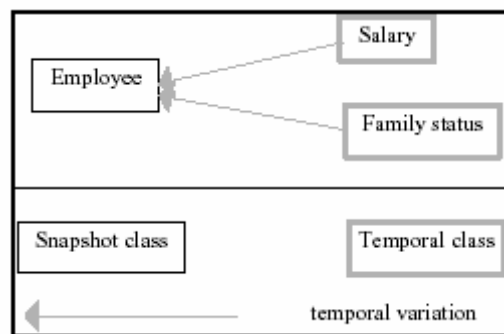
Trạng thái của đối tượng là một tập giá trị ở một thời điểm đã cho của trục thời gian. Một trạng thái có liên quan đến một nhãn thời gian nhất định. Theo [EIV-99F] thì:

- Trạng thái gọi là hiệu lực khi nó có liên quan đến thời gian hiệu lực.

- Trạng thái gọi là cơ sở dữ liệu khi nó liên quan đến thời gian giao dịch.
- Trạng thái thời gian phức hợp là hiệu lực trên thời gian giao dịch của nó.

Trong sơ đồ hướng đối tượng, có ít nhất một thuộc tính không bao giờ thay đổi, nó dùng để nhận biết đối tượng. Như vậy, một lớp thời gian là một sự biến đổi theo thời gian của một lớp thông thường. Mỗi liên kết giữa một lớp thời gian và lớp thông thường của nó được gọi một *<<temporal variation>>*.

Ví dụ, Hình 2.5 miêu tả một nhân viên có hai giá trị biến đổi theo thời gian : tiền lương và tình trạng gia đình



Hình 2. 5 Các lớp thời gian

Việc kết hợp có thể có trong lớp thời gian được tổng kết trong Bảng 2.1

Bảng 2. 1 Định nghĩa quản lý thời gian trong một lớp thời gian

		Thời gian giao dịch		
		Thông thường	Trạng thái	Lịch sử
Thời gian hiệu lực	Thông thường	Không quản lý thời gian (Lớp thông thường)	Trạng thái CSDL cuối cùng của một đối tượng	Lịch sử CSDL của một đối tượng
	Trạng thái	Trạng thái cuối cùng của một đối tượng với thời gian hiệu lực của nó	Trạng thái bitemporal của một đối tượng	Lịch sử CSDL của một đối tượng và trạng thái hiệu lực của nó
	Lịch sử	Lịch sử hiệu lực của một đối tượng	Lịch sử hiệu lực của một đối tượng và trạng thái CSDL cuối cùng của nó	Lịch sử bitemporal của một đối tượng

Để phù hợp với thông tin cần được khôi phục, những đặc trưng của mỗi chiều thời gian cần quản lý phải được định nghĩa một cách phù hợp.

- **Định nghĩa thời gian hiệu lực.** Nếu thời gian hiệu lực được yêu cầu, thì một số đặc trưng của nó phải được xác định như sau:
 - Thời gian hiệu lực trên đơn vị nào? (là thời điểm hoặc giai đoạn),
 - Đơn vị nguyên tố và lịch biểu của thời gian hiệu lực,
 - Kiểu quản lý (trạng thái hoặc lịch sử),
 - Tính đầy đủ của lịch sử (đầy đủ hoặc bộ phận).
- **Định nghĩa thời gian giao dịch.** Nếu thời gian giao dịch được yêu cầu, thì kiểu quản lý (trạng thái hoặc lịch sử) phải được xác định.

D - Mở rộng ràng buộc áp dụng lớp đối tượng tới chiều thời gian

Định nghĩa khung để phân loại ràng buộc

1. Trong OODBMS hoạt động như [ODE], ràng buộc được phân vào hai nhóm: các ràng buộc *intra – object* và các ràng buộc *inter - object*.

- Ràng buộc *intra - object* được định nghĩa cục bộ đến một đối tượng và có thể được kiểm tra vào cuối mỗi phương thức thực hiện.
- Ràng buộc *inter - object* sử dụng một số đối tượng, nó phải được kiểm tra vào cuối của giao dịch cơ sở dữ liệu.

2. Sự phân loại thứ hai từ miền cơ sở dữ liệu thời gian và nhóm các ràng buộc vào hai phạm trù : các ràng buộc *intra - time* và các ràng buộc *inter - time*.

- Ràng buộc *intra - time* tùy thuộc vào thời điểm t. Mỗi thời điểm t cần phải thỏa mãn các ràng buộc, tương ứng với các chiều thời gian sau đây : *intra - VT*, *intra - TT* và *intra - Bi*.
- Ràng buộc *inter - time* được định nghĩa bởi cách dùng thông tin sẵn có hoặc hiệu lực ở những thời điểm khác nhau. Mỗi thời điểm cần phải thỏa mãn các ràng buộc, tương ứng với chiều thời gian xem xét : *inter - VT*, *inter - TT* và *inter - Bi*.

E - Mở rộng thời gian của sự kiện

Sự kiện là khái niệm được sử dụng trong OOM để định nghĩa tính động của ứng dụng. Ba loại sự kiện được [EIV-99F] định nghĩa như sau :

Sự kiện ngoài mô hình hóa thể hiện của một thông báo bởi một tác nhân. Sự kiện ngoài được phân loại vào ba nhóm dựa trên tính hiệu lực của chúng :

- Sự kiện “Đúng lúc” là sự kiện có thời gian hiệu lực thể hiện tại thời gian hiện tại.
- Sự kiện “Tiên nghiệm ” là sự kiện có một thời gian hiệu lực trong tương lai (cho phép xúc tiến các hoạt động trước khi điều đó thực sự xảy ra).
- Sự kiện “Hậu nghiệm ” là sự kiện có thời gian hiệu lực trong quá khứ.

Sự kiện bên trong được xem như một sự thay đổi trạng thái cần ghi nhớ về một đối tượng. Thời gian hiệu lực của một sự kiện bên trong được hạn chế trong TOOM đến “ *Sự kiện đúng lúc* ”.

Một sự kiện dựa vào sự thay đổi trạng thái của một đối tượng được gọi **sự kiện của đối tượng**. Ví dụ, nếu một nhân viên đã tăng thêm tiền lương của anh ấy bốn lần trong ba năm trước, thì ban nhân viên được thông báo. Sự kiện này được gọi sự kiện lịch sử.

Sự kiện thời gian là một trạng thái đặc biệt của việc ghi nhận thời gian. Ba kiểu sự kiện thời gian được định nghĩa như sau: sự kiện có chu kỳ, sự kiện tuyệt đối và sự kiện tương đối. Ví dụ, “ *mỗi 25 tháng 12 hàng năm* ” là một sự kiện có chu kỳ, “ *25/12/1996* ” là một sự kiện tuyệt đối, và “ *1 tháng sau khi hết hạn của tiền vay* ” là một sự kiện tương đối.

2.4.1.4 Kết luận về phương pháp luận TOOBIS

Phương pháp luận TOOBIS hỗ trợ khía cạnh thời gian của các ứng dụng cơ sở dữ liệu và hỗ trợ tương ứng các đặc tả khái niệm đến TOOBIS - TOODBMS với ngôn ngữ định nghĩa TODL của nó.

Mở rộng thời gian giới hạn theo các hướng [EIV-99F] :

- Phép đo thời gian thông qua việc định nghĩa lịch biểu,

- Trình diễn thời gian thông qua những miền thời gian mới,
- Hình thức hóa thời gian, lưu trữ trong lịch sử của dữ liệu với lớp thời gian và các liên kết mới của nó,
- Định nghĩa sự toàn vẹn của dữ liệu thời gian với việc mở rộng các ràng buộc tĩnh.

2.4.2 Hệ quản trị cơ sở dữ liệu thời gian hướng đối tượng

TOODBMS gồm có ba mô đun, bao gồm:

- Mô hình dữ liệu đối tượng thời gian (TODM)
- Ngôn ngữ định nghĩa đối tượng thời gian (TODL)
- Ngôn ngữ truy vấn đối tượng thời gian (TOQL)

Trong phần này luận văn sẽ trình bày hai phần là TODM và TODL, riêng TOQL luận văn sẽ trình bày chi tiết với các ví dụ minh họa rõ ràng trong chương 4.

2.4.2.1 Mô hình dữ liệu đối tượng thời gian (TODM)

TODM là một mở rộng của Mô hình dữ liệu đối tượng (ODM) của ODMG mà các hệ quản trị cơ sở dữ liệu hướng đối tượng (OODBMS) được xây dựng. TODM tập trung cho việc chuyển đổi giữa các OODBMS.

Trước khi giới thiệu dữ liệu thời gian trong khái niệm hướng đối tượng, luận văn sẽ trình bày mô hình dữ liệu đối tượng thời gian TODM. Sau đó, luận văn sẽ trình bày các cấu trúc và giao diện được sử dụng để biểu diễn dữ liệu thời gian.

A - Mô hình và thao tác thời gian

TODM dựa vào một mô hình thời gian gần chuẩn và cổ điển. Nó là một cấu trúc tuyến tính và thời gian được sắp xếp thống kê trong định nghĩa, sử dụng thao tác “*inferior to*”. Trục thời gian có thể được chia thành một số hữu hạn các đoạn nhỏ gọi là những nguyên tố. Nguyên tố nhỏ nhất được gọi là

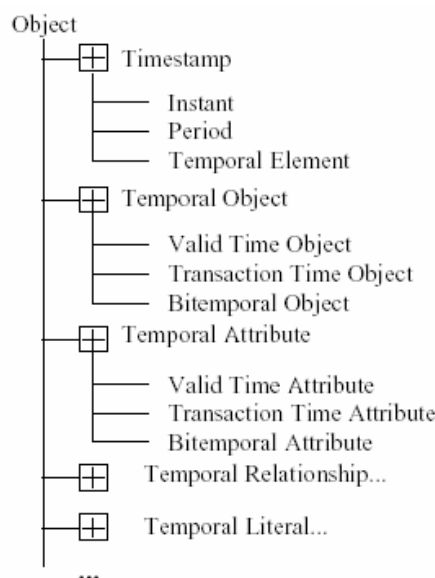
chronon và kích thước của nó là đơn vị nhỏ nhất dùng để thực thi khi xử lý dữ liệu. Thời gian trên số lượng thực thể được [EIV-99F] định nghĩa như sau:

- Một thời điểm là một điểm thời gian trên trục thời gian.
- Một thời kỳ (giai đoạn) là một lượng thời gian giữa hai thời điểm
- Một khoảng là một lượng thời gian với một thuộc tính độ dài thời gian, nhưng không có những ranh giới xác định như giai đoạn.

TODM cung cấp hay hỗ trợ đầy đủ cho lịch Gregorian chuẩn với những nguyên tố chuẩn của nó - năm, tháng, ngày, giờ, phút và giây - cũng như chuẩn bị các lịch biểu hỗ trợ.

B - Dữ liệu thời gian bên trong đối tượng

Cơ sở ban đầu là đối tượng với định danh duy nhất (OID) không thay đổi. Trạng thái của nó được định nghĩa bởi những giá trị mang những thuộc tính tường minh và hành vi của nó được định nghĩa bởi một tập các thao tác. Những đối tượng là những thực thể với các giá trị, và nó có thể tiến triển trong cả thời gian. TODM được thiết kế để lưu trữ những tiến hóa như vậy thông qua thời gian hiệu lực và/hoặc thời gian giao dịch.

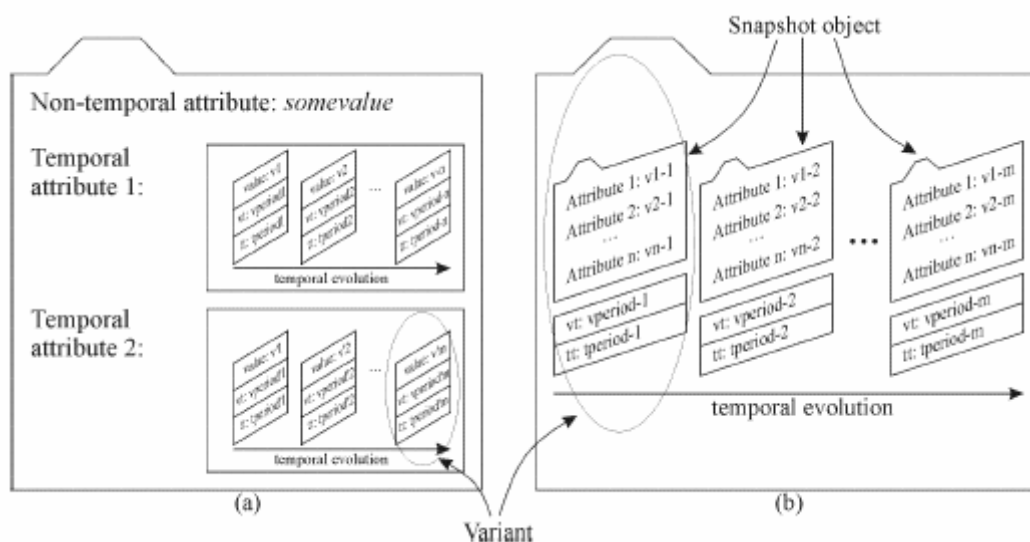


Hình 2. 6 Mở rộng đến kiểu phân cấp của ODMG

Hình 2.6 miêu tả một phần việc mở rộng thứ bậc của kiểu mà được ODMG

giới thiệu bởi TODM.

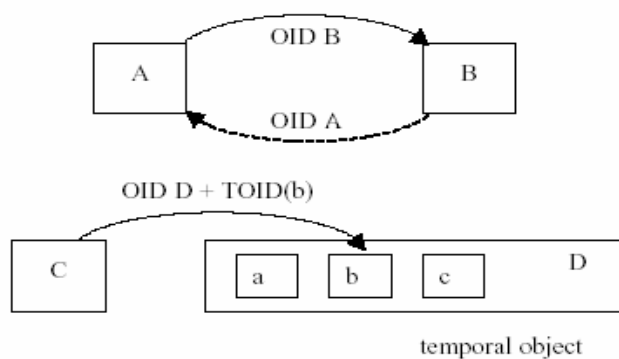
Hình 2.7 minh họa sự khác nhau giữa đối tượng thời gian thể hiện thuộc tính và đối tượng.



Hình 2. 7 Các tính chất thể hiện thời gian và các đối tượng thời gian

TODM giới thiệu một loại quan hệ mới: mối quan hệ trạng thái. Một mối quan hệ trạng thái không chỉ về phía Một đối tượng, mà là về phía Một trạng thái đặc biệt của Một đối tượng thời gian. Trong trường hợp này, OID của đối tượng đích không đủ để mô hình hoá mối quan hệ trạng thái. Thay vào đó, một OID Thời gian (TOID) của nhãn thời gian đối tượng đích liên quan đến một trong số những trạng thái của nó - được sử dụng, cho phép lựa chọn chính xác trạng thái của đối tượng thời gian kéo theo trong trạng thái mối quan hệ.

Hình 2.8. Minh họa những quan hệ và trạng thái truyền thống.



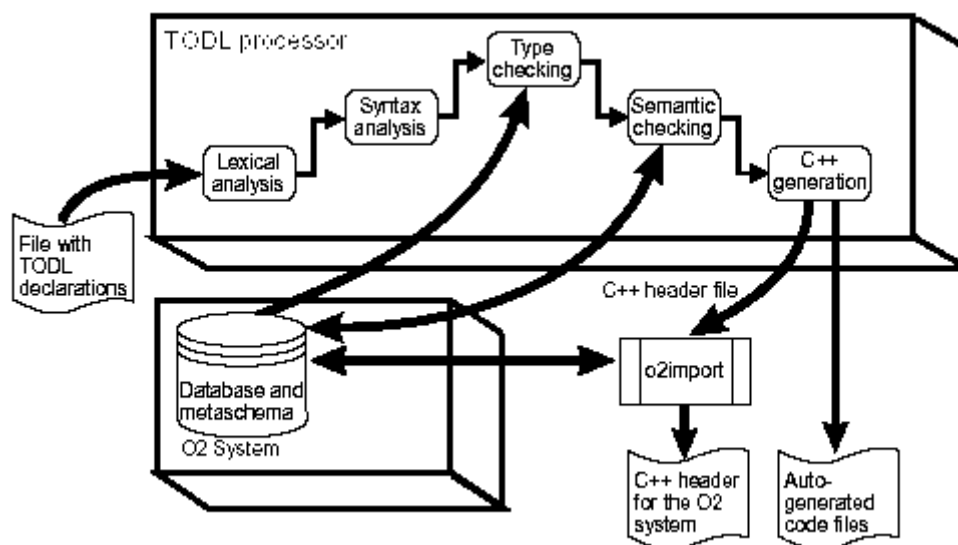
Hình 2. 8 Các quan hệ và các quan hệ trạng thái

2.4.2.2 Ngôn ngữ định nghĩa đối tượng thời gian (TODL)

Ngôn ngữ định nghĩa đối tượng thời gian (TODL) là một ngôn ngữ cho phép người dùng định nghĩa thêm các giao diện thời gian. TODL ngang hàng với những khái niệm ODL của ODMG [EIV-99F] :

- TODL hỗ trợ tất cả ngữ nghĩa xây dựng mô hình hóa dữ liệu bên dưới .
- TODL là một ngôn ngữ định nghĩa cho các đặc tả đối tượng.
- TODL là một ngôn ngữ lập trình độc lập.

TODL liên quan đến định nghĩa các thuộc tính giao diện và thể hiện cho đối tượng thời gian. TODL cho phép định nghĩa dữ liệu thời gian trên cả thuộc tính thể hiện lẫn mức đối tượng, trong khi chính nó được hỗ trợ bởi TODM. Hình 2.9 minh họa cấu trúc bộ xử lý TODL.



Hình 2. 9 Kiến trúc bộ xử lý TODL

2.5 Kết luận chương 2

Trong chương 2, luận văn đã đi vào phân tích chi tiết về cơ sở dữ liệu thời gian hướng đối tượng trong các hệ thống thông tin - TOOBIS. Trong đó trình bày chi tiết về phương pháp luận TOOBIS và Hệ quản trị TOODBMS.

- Phương pháp luận TOOBIS hỗ trợ những người phân tích mô tả những khía cạnh thời gian của các ứng dụng cơ sở dữ liệu và hỗ trợ tương ứng các đặc tả khái niệm đến TOOBIS - TOODBMS với ngôn ngữ định nghĩa TODL.
- Hệ quản trị TOODBMS với ba mô đun TODL, TODM và TOQL. Chương 2 đã trình bày hai mô đun TODM và TODL, với mở rộng dữ liệu thời gian và ngôn ngữ định nghĩa mới, hỗ trợ đầy đủ các tính năng dành cho các ứng dụng quản lý cơ sở liệu có yếu tố thời gian theo phương thức hướng đối tượng.

Chương 3 sẽ giới thiệu một đại số thời gian là cơ sở toán học cho việc xử lý đối với cơ sở dữ liệu thời gian theo phương thức hướng đối tượng.

CHƯƠNG 3 – CƠ SỞ TOÁN HỌC CHO XỬ LÝ CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG THỜI GIAN

Giới thiệu

Mô hình dữ liệu hướng đối tượng có nhiều ưu điểm với các tính chất như kế thừa, định danh đối tượng, đóng kín, đóng gói thông tin, đa hình, các kỹ thuật kiểu/lớp... cho phép các thực thể trong thế giới thực được mô hình hoá trực tiếp, thể hiện đầy đủ để phát triển hệ thống và mô hình hoá dữ liệu. Khung nhìn của một CSDL hướng đối tượng có thể được trình bày bởi một mạng các lớp đối tượng được kết nối trong bằng các kiểu kết hợp khác nhau. Các khung nhìn mở rộng của chúng giống như một mạng của các thể hiện lớp đối tượng được kết nối trong bởi các thể hiện của các kiểu kết nối.

Một hướng đi chính cho đại số TA là các hỗ trợ của nó cho xử lý và đặc tả dựa trên mẫu (hoặc dựa trên đồ thị). Trong một đặc tả truy vấn dựa mẫu, một truy vấn có thể giải quyết hình thức hoá một lớp đối tượng đơn hay các lớp đối tượng phức có cấu trúc tuyến tính, cây hoặc mạng lưới (ví dụ chúng có chứa các vòng lặp). Các nhánh của một cây hoặc một mạng có thể được gán nhãn là các nhánh AND hoặc OR phụ thuộc vào ngữ nghĩa của truy vấn. Ưu điểm của các toán tử đại số tường minh được giới thiệu trong hướng này là:

- Chúng tương thích với các cấu trúc ngôn ngữ mức cao, dễ dàng cho khả năng thiết lập cũng như ánh xạ giữa chúng với nhau.
- Chúng có thể được thực thi trực tiếp với hiệu suất cao hơn việc sử dụng một số các toán tử khác để biểu diễn và thực thi các ngữ nghĩa giống nhau.

Đại số TA được luận văn trình bày trong chương 3 này với các nội dung đã được chấp nhận trong [SSH-98] . Nó có các đặc trưng riêng như sau :

- Sử dụng một sơ đồ dựa trên các mẫu thay vì một sơ đồ dựa vào bảng

các thuộc tính đã sử dụng trong đại số quan hệ và các mở rộng thời gian của nó.

- Các toán tử đại số thao tác trên một hoặc hai tập các mẫu đồng nhất hoặc không đồng nhất của các kết hợp đối tượng để sinh ra một tập các mẫu có thể được xử lý sau này bằng các toán tử khác.
- Trực tiếp hỗ trợ việc xử lý ngữ nghĩa các phi kết hợp, các quan hệ phi thời gian, và ngữ nghĩa các nhánh AND/OR. Nó duy trì kết hợp giữa các đối tượng trong việc duyệt các đường dẫn khác nhau của quá trình xử lý truy vấn thời gian.

Trong chương này, luận văn trình bày về mô hình hoá đối tượng và hình thức hoá truy vấn thời gian, đại số kết hợp thời gian, việc thiết kế ngôn ngữ thời gian và phân tích đại số.

3.1 Mô hình hóa đối tượng thời gian và hình thức hóa truy vấn

3.1.1 Dữ liệu thời gian

Sự khác biệt giữa CSDL phi thời gian và CSDL thời gian chính là các sự kiện, các sự kiện thời gian và việc ghi nhận các sự kiện đó theo một quá trình.

Có 3 sự kiện thời gian tiêu biểu trên các đối tượng đó là: *update*, *delete*, và *insert* (*cập nhật*, *xoá* và *chèn*). Triển khai của một thể hiện trên được xem như là phiên bản lịch sử của các thể hiện và được gọi là *temporal instances* (*các thể hiện thời gian*).

Triển khai của một thể hiện được trình bày bằng một dãy các thể hiện thời gian. Các thể hiện thời gian được sắp xếp theo thứ tự thời gian, còn gọi là trục thời gian hay đường thời gian (time-line).

Trong chương này luận văn trình bày thời gian hiệu lực để mô tả các toán tử đại số. Khoảng hiệu lực của một thể hiện thời gian được mô tả bằng thời gian bắt đầu *ts* và thời gian kết thúc *te*. Nếu một thể hiện trình diễn một số thông tin hiện thời thì khi đó có thể thấy quan hệ: $ts = te = now$, trong đó *now* mô tả

một giá trị đặc biệt cho biết thông tin đó ở trạng thái hiện thời.

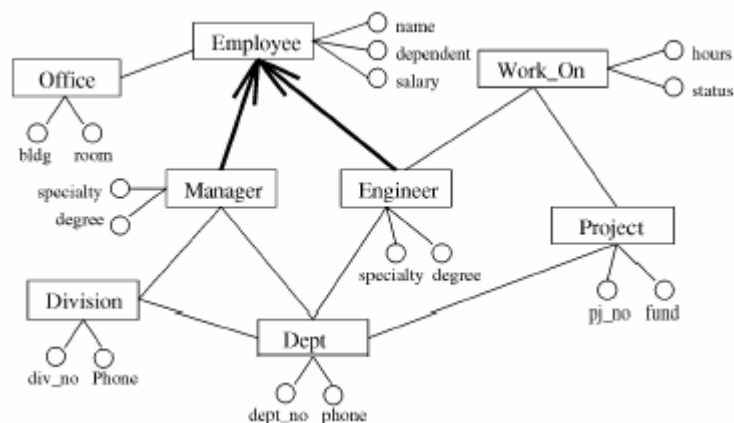
3.1.2 Biểu diễn đồ thị của TOODB

Một trình diễn đồ thị của một CSDL hướng đối tượng đặc tả mối quan hệ giữa các đối tượng dữ liệu thời gian bằng các liên kết kết hợp tường minh [SSH-98]. Trong một mô hình dữ liệu dựa trên đồ thị - graphbased data model, như OSAM*/T [SSH-98] , một CSDL thời gian hướng đối tượng được trình bày trong hai đồ thị: *Schema Graph* (SG - Đồ thị lược đồ) và *Temporal Object Graph* (TOG - Đồ thị đối tượng thời gian) trong đó các đồ thị trình bày tương ứng với CSDL tổng quát và CSDL mở rộng.

3.1.2.1 Đồ thị lược đồ

Trong một đồ thị lược đồ - Schema Graph (SG), [SSH-98] một CSDL được nhìn nhận như một tập các lớp đối tượng thông qua các kiểu kết hợp khác nhau .

Ví dụ SG của một CSDL công ty được trình bày trong hình 3.1.



Hình 3. 1 Đồ thị lược đồ của một CSDL công ty

Các hình chữ nhật biểu diễn các lớp thực thể - *entity classes*, và các hình tròn biểu diễn các thuộc tính được định nghĩa trên các lớp miền – *domain classes*. Các cạnh trong SG trình bày các kết hợp giữa các lớp đối tượng. Mũi tên hai đầu ký hiệu các liên kết tổng quát hoá - *generalization* và các đỉnh đơn giản là ký hiệu các liên kết gộp – *aggregation* giữa các lớp đối tượng. Tất cả các liên kết gộp - *aggregation* đều đặt tên nhưng tên của chúng không đưa ra theo

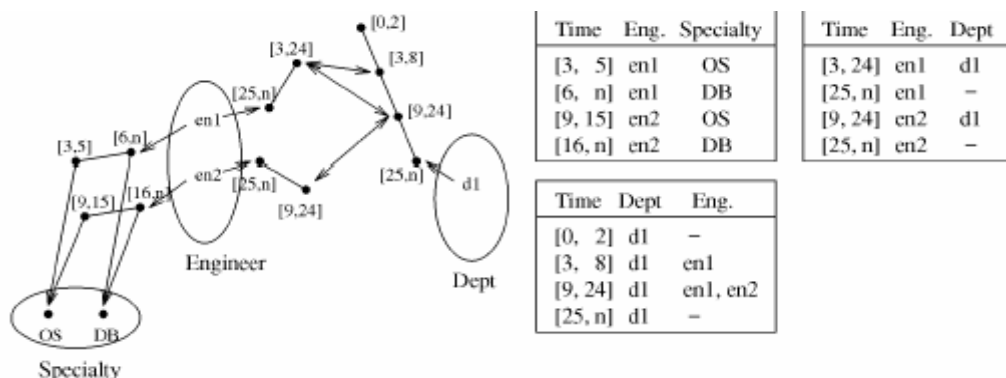
thứ tự.

3.1.2.2 Tổ chức dữ liệu thời gian và đồ thị đối tượng thời gian

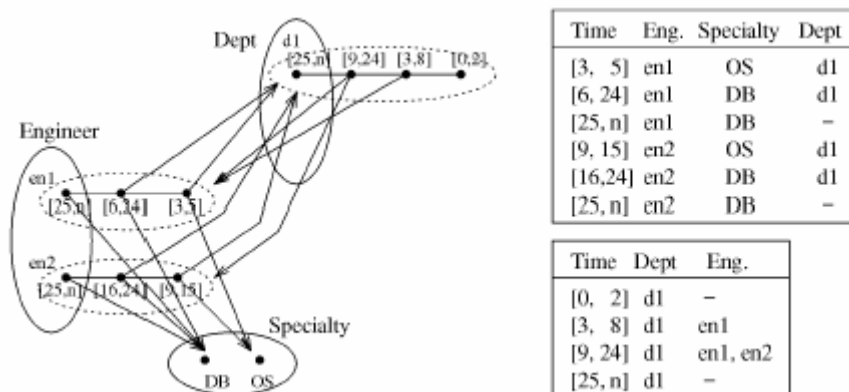
Việc mở rộng của một CSDL thời gian có thể được trình bày bởi một - Temporal Object Graph (TOG). Trong một TOG, các cạnh biểu diễn các thể hiện thời gian. Các kết hợp giữa chúng được biểu diễn bằng các liên kết trong. Trong một TOG, [SSH-98] một CSDL được xem như là một tập của các thể hiện thời gian đã được kết nối trong thông qua các liên kết kết hợp phi kiểu. Một thể hiện thời gian trong một TOG có thể được định danh bằng một TIID, nó được hình thức hoá bằng một chuỗi các sự kiện liên tiếp của một nhân thời gian và một IID.

Các bộ nhân thời gian - Timestamping [SSH-98] và các nhân thời gian thuộc tính – Attribute timestamping [SSH-98] được dùng trong việc mở rộng thời gian đến một mô hình dữ liệu quan hệ. Việc triển khai của các thể hiện đối tượng, các quan hệ thời gian của chúng trong một CSDL hướng đối tượng có thể được ghi lại bằng việc dùng hai phương thức chứa nhân thời gian là: *instance timestamping* và *attribute timestamping*. Một khai triển của một thể hiện, chứa cả dữ liệu mô tả và các tham chiếu kết hợp, được ghi lại trong đường thời gian chung. Các tổ chức dữ liệu thời gian dựa trên hai phương thức nhân thời gian được trình bày bằng phương pháp đồ thị với TOG và các bảng trong hình 3.2.

Các trục thời gian đa chiều của một thể hiện trong thuộc tính phương thức nhân thời gian có thể được chuyển đổi thành đường thời gian đơn bằng các thao tác time-alignment [SSH-98].



Hình 3. 2 Thuộc tính nhân thời gian trong tổ chức dữ liệu thời gian



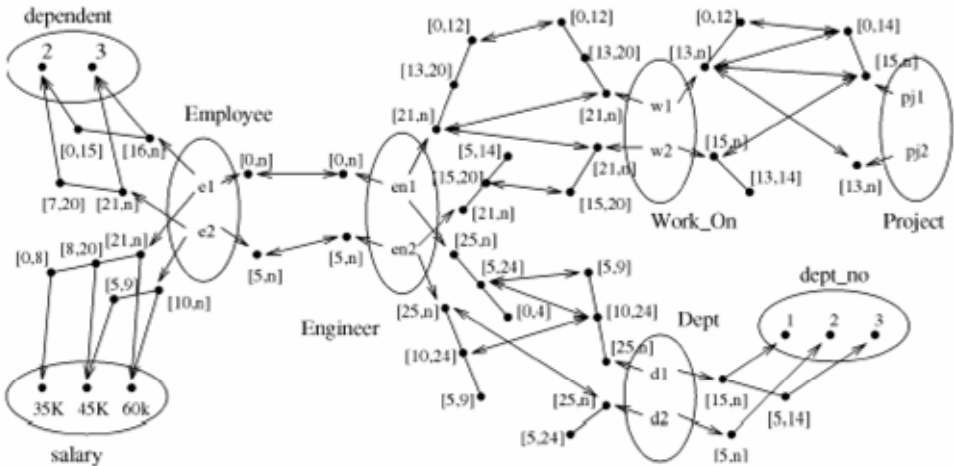
Hình 3. 3 Thể hiện nhân thời gian trong tổ chức dữ liệu thời gian

Hình 3.4 Trình bày TOG trong phương pháp thuộc tính thời gian.

3.1.2.3 Hình thức hoá truy vấn thời gian dựa mẫu

Kết hợp với các mô hình dữ liệu thời gian hướng đối tượng, một số ngôn ngữ truy vấn thời gian đã được thiết kế bằng việc kết hợp các cấu trúc thời gian vào một số ngôn ngữ truy vấn phi thời gian đang tồn tại. Chúng có thể được phân chia thành một số nhóm như [SSH-98] nói tới một số ngôn ngữ: SQL/Quelbased (ví dụ: TOOSQL), PL-based (Ví dụ: OODAPLEX), và association-based (Ví dụ: OQL/T) . Các ngôn ngữ thời gian có 3 thành phần truy vấn cơ bản:

- Các đặc tả thời gian
- Các điều kiện lựa chọn dữ liệu
- Các thao tác với CSDL



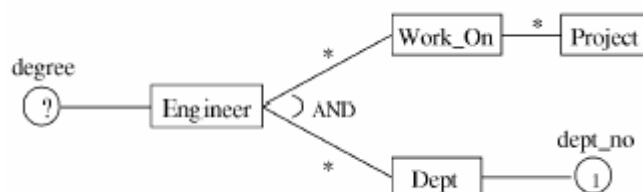
Hình 3. 4 Đồ thị đối tượng thời gian

Ưu điểm của hình thức hóa truy vấn này là đơn giản trong việc xác định các truy vấn phức tạp mà chúng bao hàm các lớp phức [SSH-98]. Mẫu của các kết hợp đối tượng thời gian có thể được đặc tả bằng một biểu thức đại số thời gian có chứa các lớp đối tượng và các toán tử đại số mà chúng đã được kết nối trong một cấu trúc tuyến tính, cây hoặc mạng lưới.

Ví dụ Q1: Tìm tất cả các trình độ của các kỹ sư những người đã tham gia trong các dự án khi đang làm tại phòng ban có số hiệu phong ban dept_no = “1” trong thời gian T [15, 30]. Biểu thức tương ứng của OQL/ T là:

```
CONTEXT Engineer AND (* Work_On * Project, * Dept)
WHEN T [15, 30]
WHERE Dept.dept_no = "1"
RETRIEVE Engineer.degree
```

Một truy vấn có thể được trình bày bằng một *Query Graph* (QG). Một QG là một đồ thị con của SG như trình bày trong Hình 3.4 và trình diễn ngữ nghĩa truy vấn giữa các thực thể/các lớp miền (*entity/domain classes*) được đặc tả trong truy vấn.



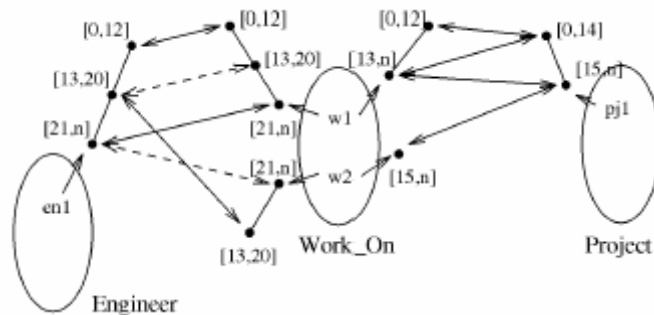
Hình 3. 5 Đồ thi truy vấn của Q1

3.2 Đại số kết hợp thời gian

Các toán tử của đại số TA thao tác trên các mẫu của các kết hợp đối tượng thời gian để sinh ra các mẫu. [SSH-98] định nghĩa các toán tử đại số TA dựa trên lý thuyết tập hợp với phép giao, phép hợp, phép trừ thời gian trên tập các thành viên. Tương tự, các toán tử đại số - TA như T-Join, T-OJoin, và T-Divide (xem phần 3.2.5.2), sử dụng các ngữ nghĩa tương ứng như các toán tử quan hệ cho các thao tác trên các mẫu thời gian của các kết hợp đối tượng (xem phần 3.2.2).

3.2.1 Các mẫu kết hợp thời gian nguyên thủy

Trong việc xử lý truy vấn dựa mẫu, một mẫu kết hợp phức tạp có thể được phân tích dựa vào một tập các mẫu cơ sở và các biểu diễn đại số tương ứng có thể được thao tác sau đó bằng các toán tử đại số. Trong Hình 3.6, các kết nối kết hợp thời gian được trình bày bằng các mũi tên liền nét. Nếu hai thể hiện đối tượng thời gian không được kết hợp trong một số khoảng thời gian, một kết nối bù (được ký hiệu bằng mũi tên chấm chấm) được vẽ giữa chúng. Nếu một truy vấn tường minh yêu cầu các thể hiện đối tượng thì nó không được kết hợp với các cái khác.



Hình 3. 6 Minh họa liên kết kết hợp thời gian

Năm mẫu nguyên thủy được mô tả trong TOG trình bày trong Hình 3.6.

1. *Mẫu thời gian bên trong (Temporal Inner Pattern- TIP)* là một vector đơn trong một TOG, biểu diễn một thể hiện thời gian, có ký hiệu là: $\{[t_s, t_e] a_i\}$.

2. *Mẫu kết nối thời gian trong (Temporal Interpattern-TI)* là một kết nối giữa 2 thể hiện thời gian được kết nối trong, có ký hiệu là: $\{[t_s, t_e]a_i b_j\}$.
3. *Mẫu kết hợp bù thời gian (Temporal Complement Pattern- TCP)* mô tả một quan hệ phi kết hợp giữa hai TIP, có ký hiệu là $\{[t_s, t_e]C(a_i b_j)\}$.
4. *Mẫu kết nối thời gian trong gián tiếp (Temporal Derived Interpattern-TDP)* đặc tả kết hợp của hai vector không liên kề, chúng được kết nối thông qua một đường dẫn của các TI, có ký hiệu là $\{[t_s, t_e]D(a_i b_j)\}$. Nó ký hiệu hai thể hiện được kết hợp không trực tiếp với các cái khác.
5. *Mẫu kết nối trong bù thời gian gián tiếp (Temporal Derived Complement Pattern- TDCP)* ký hiệu là: $\{[t_s, t_e]DC(a_i c_j)\}$. Mô tả phi kết hợp giữa hai vector không liên kề được kết nối thông qua một đường dẫn chứa ít nhất một TCP.

Các mẫu nguyên thủy được trình bày kiểu đồ thị với tính chất đại số trong Hình 3.7. Các mẫu kết hợp phức có thể được trình bày đại số bằng một tập các mẫu kết hợp thời gian nguyên thủy, được minh họa trong Hình 3.8

3.2.2 Thể hiện mẫu thời gian và tập mẫu thời gian

Đại số TA đưa ra một đặc tả truy vấn bằng một mẫu đầy đủ các lớp đối tượng và một đặc tả khoảng thời gian (time-interval). TPI có thể được biểu diễn kiểu đại số như một tập của các mẫu kết hợp thời gian nguyên thủy như đã trình bày ở mục 3.2.1.

Một tập mẫu thời gian -Temporal Pattern Set (TPS) là một tập của tất cả các TPI thỏa mãn đặc tả mẫu đầy đủ và khoảng thời gian của một truy vấn. Do đó, TPS là một đồ thị con của TOG. Một TPS là một tập của các tập mẫu kết hợp thời gian nguyên thủy (các TPI). Điều này được trình bày trong Hình 3.10.

Các toán tử đại số TA thực hiện trên một hoặc hai TPS để sinh ra một TPS.

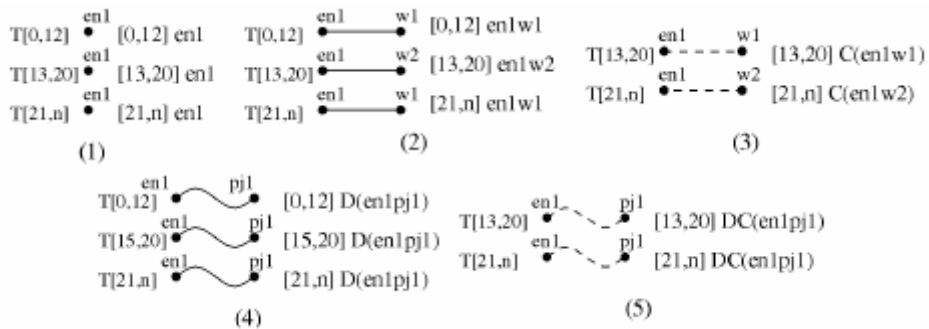
- Một TPS được coi là *đồng nhất* nếu tất cả các TPI của nó đều có mẫu hoàn toàn giống nhau.

- Một TPS là *không đồng nhất* nếu có tồn tại một số TPI mà không có mẫu hoàn toàn giống nhau. Một TPS không đồng nhất có thể là kết quả từ một thao tác đại số giữa hai TPS không có khả năng kết hợp (nonunion-compatible).

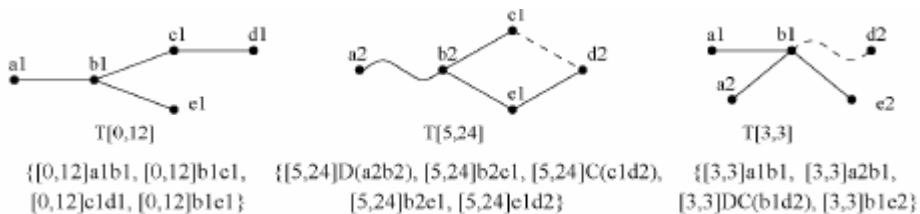
3.2.3 So sánh khoảng thời gian

Các TPI biểu diễn các mẫu kết hợp, chúng có hiệu lực trong các khoảng thời gian khác nhau. Do đó, đại số TA cần phải thao tác trên các khoảng thời gian cũng như trên các mẫu kết hợp để thoả mãn các ngữ nghĩa truy vấn khác nhau. Ý nghĩa so sánh khoảng có thể được đặc tả trong các giới hạn của các từ khoá trong một ngôn ngữ truy vấn ở mức cao (xem mục 3.3.1).

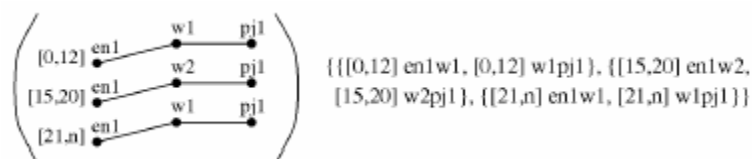
Đại số - TA trình bày các so sánh khoảng khác nhau bằng việc so sánh các thời điểm bắt đầu và thời điểm kết thúc của hai khoảng, nó có thể được định danh bằng hai hàm thời gian tương ứng là $s()$ và $e()$. Ví dụ: “ $e(T1) < s(T2)$ ” biểu diễn “thời điểm kết thúc của T1 nhỏ hơn thời điểm bắt đầu của T2”; và “ $s(T1) = s(T2) \wedge e(T1) > e(T2)$ ” biểu diễn “T1 chứa T2 từ lúc bắt đầu và tiếp tục sau đó cho đến kết thúc T2.”



Hình 3. 7 Các mẫu kết hợp thời gian nguyên thủy



Hình 3. 8 Các mẫu kết hợp thời gian phức tạp



Hình 3. 9 Các TPI và một TPS

3.2.4 Ký hiệu

Các ký hiệu được dùng trong biểu diễn các toán tử đại số TA (Bảng 3.1). Riêng hợp khoảng (interval-union) ($\cup_i$) được dùng để gộp - *coalesce* hai khoảng chồng nhau hay các khoảng liên kề khi các mẫu kết hợp của chúng được định danh.

Ví dụ: Đưa ra hai khoảng liên kề với dữ liệu được xác định như: $\{[1,5]en1\}$, $\{[6, n]en1\}$ hoặc hai khoảng chồng nhau với dữ liệu xác định như: $\{[5, 9]d1\}$, $\{[7, 25]d1\}$, chúng có thể được kết hợp tương ứng thành: $\{[1, n]en1\}$ và $\{[5, 25]d1\}$.

3.2.5 Toán tử

Các toán tử được chia thành 3 nhóm:

- Hình thức hóa các mẫu nguyên thủy
- Thao tác mẫu kết hợp đối tượng
- Thao tác mẫu không kết hợp đối tượng

3.2.5.1 Hình thức hóa mẫu nguyên thủy

Trong quá trình xử lý một CSDL thời gian hướng đối tượng được trình diễn bằng một đồ thị lược đồ (SG) và một đồ thị đối tượng thời gian (TOG), chúng ta cần xác định và định danh các thể hiện đối tượng trong một cặp lớp đối tượng mà chúng được kết hợp hoặc không kết hợp được với các đối tượng khác thành thể thức của một tập các TI nguyên thủy hoặc TCP (xem mục 3.2.1). Có 3 toán tử cho các mẫu hình thức này là: T- Associate, T- Complement, T-Nonassociate

Bảng 3. 1 Các ký hiệu biểu diễn trong biểu thức đại số TA

A, B	Các lớp đối tượng như <i>Engineer</i> và <i>Project</i>
W, X, Y, Z	Tập các lớp như: $W=\{Engineer, Project\}$
$R(A,B)$	Liên kết giữa hai lớp A và B
$P\{R(A,B)\}$	Mẫu kết hợp của $R(A,B)$
$\alpha, \beta, \gamma, \omega$	Các TPS là một tập của các TPI
$\alpha_i, \beta_j, \gamma_k$	Các TPI của $\alpha, \beta, \gamma, 1 \leq i, j, k \leq n$
$P(\alpha)$	Tập mẫu kết hợp của α
$P(\alpha_i)$	Một thành phần mẫu kết hợp của α
$P(\omega_l)$	Mẫu kết hợp của $\omega, 1 \leq l \leq n$, được mô tả trong các lớp của W
$\tilde{T}$	Một khoảng thời
$\tilde{T}(\alpha_i)$	Một khoảng thời gian của α
$\cup_t, \cap_t, -_t$	Tập các toán tử: Kết hợp, Giao và Trừ của các khoảng thời gian
$\subseteq_t, \supseteq_t$	Các toán tử của tập con

Ba toán tử trong nhóm này được trình bày sau đây: (xem trong hình 3.10)

1. Toán tử kết hợp thời gian - $T - Associate(*_{|R(A,B)|})$. Toán tử này xác định các kết nối trong giữa hai lớp đối tượng A và B thông qua một liên kết kết hợp đặc biệt $R(A,B)$ dựa trên dữ liệu được lưu trong bộ nhớ chính hoặc CSDL cố định trước đó. Thao tác này trả về tập các mẫu nguyên thủy kết nối trong đến các đối tượng thời gian của A và B . Công thức như sau:

Công thức:

$$\alpha *_{|R(A,B)|} \beta = \gamma$$

$$\gamma = \{ \cup_t \gamma_k | P(\gamma_k) = P(\alpha_i) \cup P(\beta_j) \cup a_m b_n, \}$$

$$a_m \subseteq P(\alpha_i), b_n \subseteq P(\beta_j), a_m b_n \in P\{R(A,B)\}, T(\gamma_k) = T(\alpha_i) \cap_t T(\beta_j) \neq \emptyset$$

Trong đó a_m là một IP trong $P(\alpha_i)$, b_n là một IP trong $P(\beta_j)$, γ_k được cấu trúc bằng việc liên kết hai TPI α_i và β_j bằng một IP và $a_m b_n$ có một liên kết kết hợp $R(A, B)$ trong một khoảng thời gian chung $T(\gamma_k)$. Mỗi TPI được biểu diễn bằng một

tập các mẫu thời gian nguyên thủy không chứa bản sao, khi đó γ_k là một tập các mẫu nguyên thủy được cấu trúc bằng việc hợp của α_i , β_j , và $a_m b_n$ trên một khoảng thời gian chung - common time-interval $T(\gamma_k)$ mà các kết quả từ giao của các khoảng interval-intersect ($\cap_i$) giữa các khoảng của α_i và β_j . $\cup_i \gamma_k$ gộp các khoảng liên kế có các mẫu kết hợp đã được định danh.

2. Toán tử bù thời gian - T-Complement ($[R(A,B)]$). T-Complement định danh tất cả các cặp đối tượng trong các lớp A và B mà nó không được kết nối thông qua một liên kết kết hợp đặc biệt $R(A, B)$. Kết quả là một tập TCP. Các phi kết hợp của cặp này cũng được bao gồm trong kết quả.

Công thức:

$$\begin{aligned} \alpha|_{[R(A,B)]} \beta &= \gamma \\ \gamma &= \{\gamma_k | P(\gamma_k) = P(\alpha_i) \cup P(\beta_j) \cup C(a_m b_n), \\ a_m &\subseteq P(\alpha_i), b_n \subseteq P(\beta_j), C(a_m b_n) \in P\{R(A,B)\}, T(\gamma_k) = T(\alpha_i) \cap_i T(\beta_j) \neq \emptyset\} \end{aligned}$$

Trong trường hợp đặc biệt, khi α (hoặc β) là rỗng hoặc không có một TIP (ví dụ: không có thể hiện nào thỏa mãn một điều kiện lựa chọn), tất cả các TIP β (hoặc α) chứa trong kết quả hình thức TPS các liên kết bù thời gian - Temporal Complement với các OID có giá trị *null*.

3. Toán tử - T-Nonassociate ($!_{[R(A,B)]}$). T-Nonassociate định danh tất cả các cặp của các đối tượng trong hai lớp đặc tả A và B . Một đối tượng của một lớp trong một cặp không chứa bất kỳ quan hệ thời gian nào với các đối tượng của các lớp khác thông qua các liên kết kết hợp được mô tả là $R(A, B)$ trong một hay một số khoảng thời gian. Do đó, kết quả là một tập của các TCP.

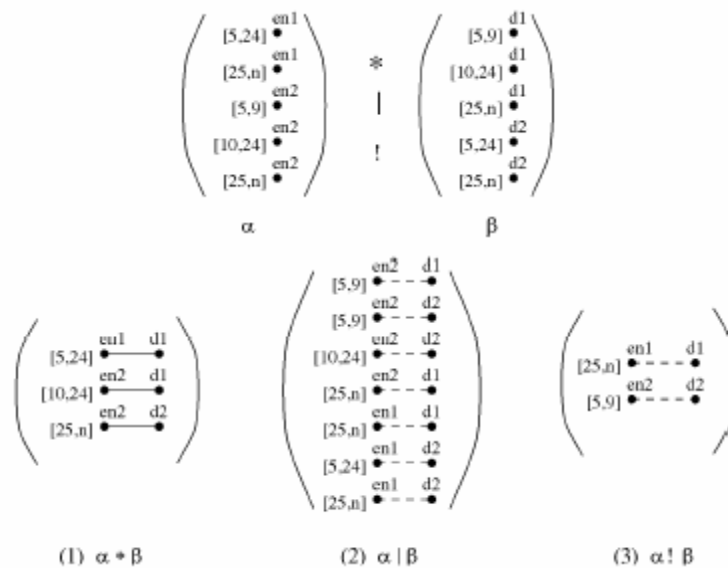
Công thức:

$$\alpha!_{[R(A,B)]} \beta = \{(\alpha_i - \Pi_{[P(\alpha_i)]}(\alpha_i^*_{[R(A,B)]} \beta_j)_{[R(A,B)]} (\beta_j - \Pi_{[P(\beta_j)]}(\alpha_i^*_{[R(A,B)]} \beta_j))\}$$

Biểu thức trên chứa cả toán tử T-Complement($|$) và hai mệnh đề - terms. Mỗi vế chứa một toán tử chiếu T - Project(Π) và một toán tử trừ T - Difference($-$). Trong vế đầu tiên, $\Pi_{[P(\alpha_i)]}(\alpha_i^*_{[R(A,B)]} \beta_j)$ chiếu trên tất cả các TPI của α các kết hợp với bất kỳ TPI nào của β . Sau đó, một toán tử T-Difference($\tilde{-}$) liệt kê các các TPI từ α , do đó chỉ bao hàm các TPI của α và nó không được kết hợp với bất kỳ TPI nào của

β . Thủ tục tương tự đưa ra cho β . Sau đó, toán tử $T\text{-Complement}()$ định danh bù các liên kết giữa các TPI của α và β thông qua $R(A, B)$.

$T\text{-Nonassociate}$ không là nguyên thủy. Tuy nhiên, thao tác này cũng rất đầy đủ cho việc công thức hoá truy vấn và do đó bao hàm cả tập của các toán tử đại số - TA. Ví dụ sau đây trình bày các thao tác của ba toán tử hình thức mẫu và Hình 3.10 minh hoạ chúng trên phương diện đồ thị :



Hình 3. 10 Các toán tử T- Associate, T- Complement và T-Nonassociate

Toán tử $T\text{-Complement}()$ có nghĩa là: “Tất cả các cặp không kết hợp của các TPI của hai lớp”. Tại đó, toán tử $T\text{-Nonassociate}()$ có nghĩa là “Các cặp TPI, mà mỗi một trong chúng không có bất kỳ sự kết hợp nào với các TPI của các lớp khác”. Do đó, các kết quả sau đó là các tập con của kết quả được hình thức trước đó.

3.2.5.2 Thao tác mẫu

Các toán tử thao tác mẫu tương tự các toán tử đại số quan hệ ngoại trừ một số tính chất như sau:

- Một số trong chúng cho phép đặc tả mô hình hoá của các kết hợp đối tượng, để ý nghĩa các thuộc tính không bị mất mát khi xử lý .
- Tất cả các toán tử có thể thực hiện trên cả TPS không đồng nhất và TPS

đồng nhất.

- Thao tác trên các khoảng thời gian là ý nghĩa của các toán tử đó.

1) T-Join ($\langle \rangle_{[W]}$). T-Join ràng buộc một cặp TPI, một trong chúng có hai mô tả toán hạng TPS đặc biệt, nếu hai TPS có một đặc tả mô hình con chung bởi W và các khoảng thời gian của chúng chồng nhau. Thao tác này thực hiện ý nghĩa của một đặc tả nhánh AND trong một đồ thị truy vấn (Xem hình 3.4).

Trong cách thực hiện này, các cấu trúc cây và mạng lưới phức tạp với các nhánh AND biểu diễn trong ngôn ngữ truy vấn dựa mẫu có thể được thực thi.

T-Join tương đương chức năng phép kết nối thời gian trong quan hệ, ngoại trừ rằng T-Join giải quyết các cặp của TPI. Chúng có thể có các cấu trúc rất phức tạp được hình thức hoá bằng nhiều lớp thay vì các cặp của các bộ phẳng của hai quan hệ thời gian.

Công thức:

$$\alpha \langle \rangle_{[W]} \beta = \gamma$$

$$\gamma = \{ \gamma_k | P(\gamma_k) = P(\alpha_i) \cup P(\beta_j), P(\omega_i) \subseteq (P(\alpha_i) \cap P(\beta_j)) \neq \emptyset, T(\gamma_k) = T(\alpha_i) \cap_i T(\beta_j) \neq \emptyset \}$$

Biểu thức $P(\gamma_k) = P(\alpha_i) + P(\beta_j)$ biểu diễn sự kết nhập của các mẫu α_i và β_j . Biểu thức $P(\omega_i) \subseteq (P(\alpha_i) \cap P(\beta_j)) \neq \emptyset$ đặc tả một mẫu con là chung cho tất cả các TPI.

Hình 3.11 trình bày trên phương diện đồ hoạ của thao tác T-Join. T-Join được thay đổi và kết hợp điều kiện được trình bày trong phần tiếp theo.

$$\begin{aligned} \alpha \diamond_{[W]} \beta &= \beta \diamond_{[W]} \alpha && (Comm.) \\ (\alpha \diamond_{[W']} \beta) \diamond_{[W'']} \gamma &= \alpha \diamond_{[W']} (\beta \diamond_{[W'']} \gamma) && (Assn.) \\ & \text{(if } (W' - W'') \cap Z = \emptyset \text{ and } (W'' - W') \cap X = \emptyset \end{aligned}$$

Việc kết hợp không luôn luôn đúng bởi vì có những trường hợp trong nội dung một mẫu của β (trình bày bởi Y) là có lỗi khi đối sánh với bất kỳ mẫu nào trong γ (trình bày bởi Z), có thể thành công bằng phép giao đầu tiên với một mẫu trong α (được trình diễn bởi X) trong thao tác $\langle \rangle_{[W]}$ và sau đó bằng việc giao với một mẫu của γ trong thao tác $\langle \rangle_{[W]}$

2) T-OJoin ($\langle \cdot \rangle_{[W]}$). T-OJoin là tương tự với outer-join trong đại số quan hệ. Nó là T-Joins các TPI của hai TPS mở rộng khi chúng cùng chia sẻ đặc tả mẫu con giống nhau trong W. Hơn nữa, nó cũng bao hàm các TPI của một toán hạng chứa các mẫu con được đặc tả trong W, nhưng không tương ứng với TPI trong toán hạng còn lại, chúng có các mẫu con giống nhau. Thao tác này thực hiện ý nghĩa của đặc tả nhánh OR trong một đồ thị truy vấn.

Công thức:

$$\alpha \langle \cdot \rangle_{[W]} \beta = \gamma$$

$$(1) \gamma_k \in (\alpha \langle \cdot \rangle_{[W]} \beta)$$

$$\gamma = \{\gamma_k \mid (2) P(\omega_l) \subseteq (P(\gamma_k) = P(\alpha_i), T(\gamma_k) \subseteq_t T(\alpha_i))$$

$$(3) P(\omega_l) \subseteq (P(\gamma_k) = P(\beta_j), T(\gamma_k) \subseteq_t T(\beta_j))$$

Biểu thức (1) trình bày một kết quả TPI từ một T-Join. Biểu thức (2) và (3) trình bày các vị trí thời gian tương ứng của α_i và β_j .

Một T-OJoin giữa các kết quả α_i và β_i trong một T-Join TPI và hai TPI điều kiện là chúng không có các vị trí thời gian tương ứng trong các cái khác...

3) T-Select ($\delta_{[\hat{p}]}$). T-Select trích chọn các TPI từ một toán hạng TPS mà nó thỏa mãn biểu thức Boolean của các tính chất được biểu diễn bởi $\hat{p}$. Biểu thức Boolean được ước lượng dựa vào mỗi TPI. Nếu kết quả đánh giá là *true*, thì TPI chứa các kết quả của TPS.

Công thức:

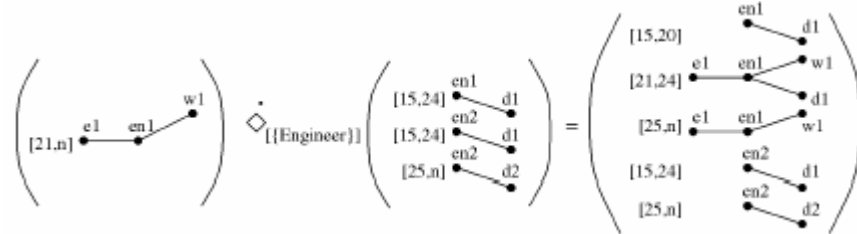
$$\delta_{[\hat{p}]}(\alpha) = \gamma, \gamma = \{\gamma_k \mid \gamma_k = \alpha_k \mid \hat{p}(\alpha_k) = \text{True}\}$$

Ở đây, $\hat{p}(\alpha_k)$ có thể là:

- Một so sánh giữa hai thuộc tính (hoặc giữa một thuộc tính và một giá trị đặc biệt)
- Một sự xác minh của kết nối giữa hai thể hiện (như các kết hợp của chúng - *association(*)* hoặc các phi kết hợp của chúng - *nonassociation(!)*)
- Một so sánh của hai khoảng thời gian $T_1 \subseteq T_2, (T_1 \subseteq T_2) \neq \emptyset, s(T_1) \subseteq s(T_2)$ (đã thảo luận ở mục 3.3.3).



Hình 3. 11 Toán tử T-Join



Hình 3. 12 Toán tử T-OJoin

Thao tác ở trên được trình diễn dạng đồ hoạ trong Hình 3.12. $\hat{p}$ biểu diễn với các phép so sánh khoảng thời gian phức tạp hơn sẽ được trình bày trong mục 3.4.2.

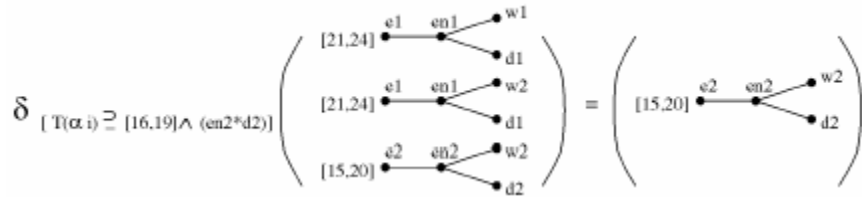
4) T-Project ($\Pi_{[w,\hat{T}]}$). T-Project trích chọn các mẫu con của một khoảng thời gian quan tâm hoặc cả hai từ các TPI của một toán hạng TPS. Các mẫu con và khoảng thời gian được chiếu được mô tả tương ứng bằng W và $\hat{T}$. Trong đại số TA, toán hạng TPS có thể không đồng nhất và thao tác T-Project có thể được dùng để trích chọn các cấu trúc đồng nhất từ các cấu trúc không đồng nhất.

Công thức:

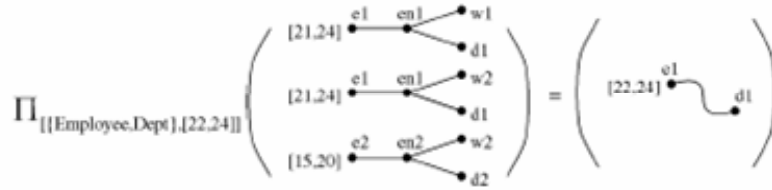
$$\Pi_{[w,\hat{T}]}(\alpha) = \gamma, \gamma = \{ \cup_t \gamma_k | P(\gamma_k) \subseteq P(\alpha_i), P(\gamma_k) \subseteq P(\omega), T(\gamma_k) \subseteq \hat{T} \}$$

Thao tác này sinh ra các TPI có thời gian hiệu lực liên kế hoặc chồng lấp và có các mẫu kết hợp định danh. Với các khoảng thời gian có thể được kết hợp bằng $\cup_t \gamma_k$ mà vẫn duy trì được các ý nghĩa của các khoảng thời gian (xem lại mục 3.3.4).

Hình 3.13, 3.14 trình bày đồ hoạ của các thao tác trên.



Hình 3. 13 Toán tử T- Select



Hình 3. 14 Toán tử T- Project

5) T-Union (+).

Công thức:

$$\alpha + \beta = \gamma, \gamma = \{\cup_t \gamma_k | \gamma_k \in \alpha \text{ or } \gamma_k \in \beta\}$$

T-Union xây dựng một TPS chứa tất cả các TPI mà xuất hiện trong một hoặc cả hai toán hạng TPS. Các TPI trong kết quả TPS, của các mẫu giống nhau và các khoảng thời gian là liền kề hoặc chồng lấp, được kết hợp vào một khoảng thời gian đơn bằng thao tác liên kết khoảng. T-Union dựa trên khái niệm giống như thao tác Union quan hệ được áp dụng vào một CSDL thời gian. Tuy nhiên, các toán hạng không cần phải có khả năng có thể kết hợp - unioncompatible.

Hình 3.14 trình bày dạng đồ hoạ của thao tác trên. Thao tác T-Union trên kết hợp tất cả các TPI của hai toán hạng để sinh ra một TPS không đồng nhất.

6) T-Intersect (•).

Công thức:

$$\alpha \bullet \beta = \gamma,$$

$$\gamma = \{\gamma_k | P(\gamma_k) \in P(\alpha) \text{ and } P(\gamma_k) \in P(\beta), T(\gamma_k) = T(\alpha) \cap T(\beta) \text{ for } P(\alpha_i) = P(\beta_j)\}$$

T-Intersect tạo thành một TPS chứa tất cả các TPI mà chúng xuất hiện trong cả hai mô tả TPS trong các khoảng thời gian hiệu lực. Điều này là trực tiếp từ một thao tác giao tập hợp - set-intersection trên hai tập của các mẫu thời gian .

Hình 3.15 trình bày các toán hạng trên phương diện đồ hoạ.

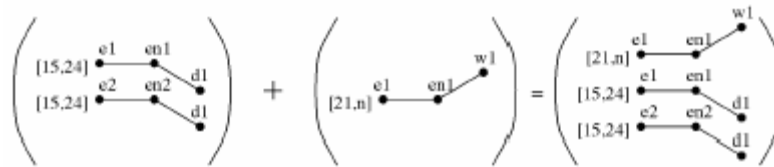
7) T-Difference (-). T-Difference liệt kê tất cả các TPI của toán hạng TPS thứ nhất nếu các mẫu giống nhau xuất hiện trong toán hạng TPS thứ hai trong một số khoảng hiệu lực chung. Giống như T-Union và T-Intersect, thao tác này có thể tính toán trên các toán hạng có các cấu trúc đồng nhất hoặc không đồng nhất và union-compatibility giữa các toán hạng thì không có bất cứ yêu cầu nào.

Công thức:

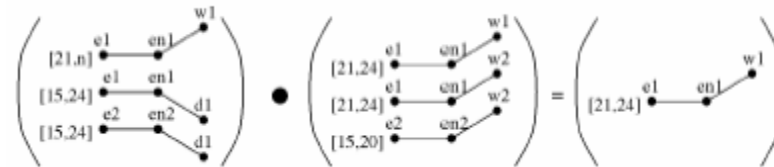
$$\alpha - \beta = \gamma,$$

$$\gamma = \{\gamma_k | P(\gamma_k) = P(\alpha_i), T(\gamma_k) \in P(\beta_j) \text{ for } P(\alpha_i) \neq P(\beta), T(\gamma_k) = T(\alpha_i) - T(\beta_j) \text{ for } P(\alpha_i) = P(\beta)\}$$

Điều kiện trong mệnh đề thứ hai trình bày rằng một mẫu của một TPI trong α không tồn tại trong β , và điều kiện trong mệnh đề thứ ba trình bày rằng một mẫu giống hệt nhau tồn tại trong cả α và β .



Hình 3. 15 Toán tử T- Union



Hình 3. 16 Toán tử T-Intersect

Thao tác được trình bày dạng đồ họa trong hình 3.16.

8) T-Divide ($\div_w$). T-Divide định danh một nhóm của các TPI với một đặc trưng dữ liệu chung chắc chắn (ví dụ như các mẫu con - subpattern) trong toán hạng TPS thứ nhất mà chứa tất cả các TPI trong toán hạng TPS thứ hai. Mẫu con chung được mô tả bằng một tập các lớp được ký hiệu W . Sau đó, các mẫu con chung được trích chọn từ các TPI được định danh bởi một T-Project. Nếu W không được mô tả, thao tác giữ lại tất cả các TPI của toán hạng TPS thứ nhất, mỗi một trong chúng chứa đựng ít nhất là một TPI của β , và chúng cùng chứa tất cả các TPI của β (T-Project không không liên quan đến).

Công thức:

$$\alpha \div_{|W|} \beta = \Pi_{|W|}(\gamma_s)$$

$$\gamma_s = \{ \gamma_s^k | P(\gamma_s^k) = P(\gamma_s), P(\omega_1) \subseteq \alpha_s^k, P(\beta_j) \subseteq P(\alpha_s^k), T(\gamma_s^k) = T(\alpha_s^k) \cap_t T(\beta_j) \neq \emptyset \}$$

α_s là một tập con của các TPI của α , các mẫu con chung của chúng có các lớp được mô tả trong W và cùng chứa tất cả các TPI của β . T-Project trích chọn các mẫu con

Các thao tác trên được trình bày dạng đồ hoạ trong hình 3.17.

3.2.5.3 Thao tác mẫu không tính toán thời gian

Trong việc xử lý truy vấn, các toán tử đại số dựa trên các thể hiện được sử dụng đầy đủ để xử lý các truy vấn thời gian chứa các mẫu thời gian phức tạp, không liên quan đến các sự kiện xảy ra có liên quan đến thời gian.

Ví dụ: “Tìm tất cả các kỹ sư những người vừa *either* làm việc trên dự án trong thời gian $T[0, 20]$ hoặc *or* thuộc về một phòng ban trong khoảng thời gian $T[25, n]$,” không thể xác định ngữ nghĩa của các thao tác giao, hợp, trừ phi biết thời gian tương ứng. Đại số TA nắm bắt tất cả các ngữ nghĩa rõ ràng bằng các toán tử đại số phi thời gian.

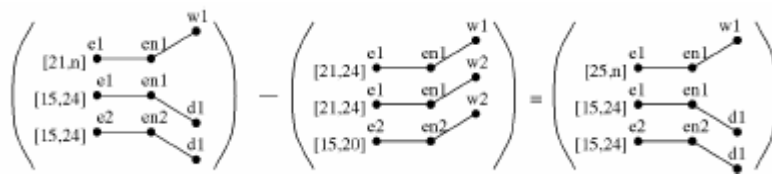
1) NT-Intersect ($\circ_{|W|}$). NT-Intersect xây dựng một TPS chứa tất cả các TPI, mà chúng có một mẫu con chung được mô tả trong W và các mẫu kết hợp của chúng xuất hiện trong cả hai toán hạng TPS, bỏ qua việc kiểm tra các khoảng thời gian của chúng.

Công

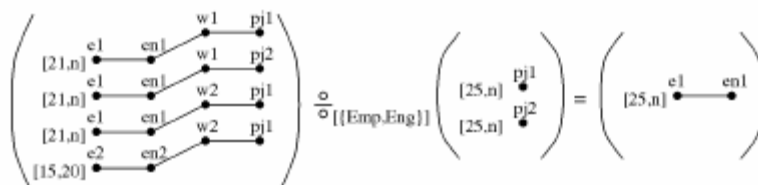
thức:

$$\alpha \circ_{|W|} \beta = \gamma, \gamma = \{ \cup_t \gamma_k | \gamma_k \in \alpha \text{ and } \gamma_k \in \beta, P(\omega_1) \subseteq P(\gamma_k) \}$$

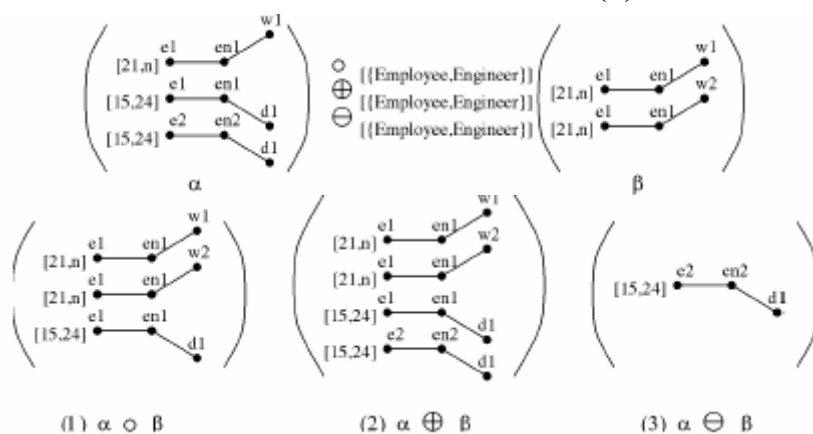
Thông qua các thao tác không đánh giá các đặc tả thời gian của các TPI, các thể hiện được định danh mang trên nó thời gian của chính các đặc tả và các mẫu của chúng.



Hình 3. 17 Toán tử T- Difference



Hình 3. 18 Toán tử T-Divide(+)



Hình 3. 19 Các toán tử NT-Intersect, NT-Union và NT-Difference

Kết quả các TPI với các khoảng liên kề hoặc chồng lấp sau đó được kết hợp với nhau bằng $\cup, \gamma_k$ để duy trì các tính chất ý nghĩa của khoảng được thảo luận trong mục 3.3.4. NT-Intersect có thể được thay thế và kết hợp các điều kiện (tham khảo lại phần T-Join với các điều kiện).

2) NT-Union ($\oplus_{|W|}$). NT-Union xây dựng một TPS chứa tất cả các TPI mà chúng có một mẫu con được mô tả bởi W và chúng có các mẫu kết hợp xuất hiện trong một hoặc cả hai toán hạng TPS, không kiểm tra các khoảng thời gian của chúng.

Công

thức:

$$\alpha \oplus_{|W|} \beta = \gamma, \gamma = \{ \cup_t \gamma_k | \gamma_k \in \alpha \text{ or } \gamma_k \in \beta, P(\omega_1) \subseteq P(\gamma_k) \}$$

Kết quả của các TPI với các khoảng liên kề hoặc chồng lấp sau đó được kết hợp bằng $\cup_i \gamma_k$ để duy trì tính chất ngữ nghĩa khoảng. NT-Union có thể được thay thế và kết hợp các điều kiện.

3) NT-Difference ($\Theta_{[W]}$). NT-Difference xây dựng một TPS chứa tất cả các TPI, mà chúng có một mẫu con chung được mô tả bởi W và các mẫu kết hợp của chúng xuất hiện trong toán hạng đầu tiên nhưng không xuất hiện trong toán hạng thứ hai, bỏ qua các kiểm tra về các khoảng thời gian.

Công

thức:

$$\alpha \Theta_{[W]} \beta = \gamma, \gamma = \{ \cup_i \gamma_k | \gamma_k \in \alpha \text{ and } \gamma_k \notin \beta, P(\omega_1) \subseteq P(\gamma_k) \}$$

Các ví dụ sau đây minh hoạ cho các thao tác của 3 toán tử phi thời gian ở trên và Hình 3.18 minh hoạ chúng theo dạng đồ hoạ. Các quyền ưu tiên của các toán tử nhị phân được đưa ra theo thứ tự sau đây: $*, |, !, \langle \rangle, \langle * \rangle, \bullet, \div, -, +$.

Bảng 3.2 thống kê các tính chất toán học của các toán tử đại số TA

Bảng 3. 2 Các tính chất của các toán tử đại số TA

Operator	Primitiveness	Math. Property
T-Associate	o	Comm.
T-Complement	o	Comm.
T-Nonassociate	x	Comm.
T-Join	o	Comm./{(cond.)Assn.
T-Ojoin	o	Comm./{(cond.)Assn.
T-Select	o	(Unary)
T-Project	o	(Unary)
T-Union	o	Comm./Assn.
T-Intersect	o	Comm./Assn.
T-Difference	o	
T-Divide	o	
NT-Intersect	o	Comm./{(cond.)Assn.
NT-Union	o	Comm./{(cond.)Assn.
NT-Difference	o	

3.3 Thiết kế ngôn ngữ thời gian và phân tích đại số học

Phần này đưa ra một phác hoạ của việc thiết kế ngôn ngữ thời gian và biểu diễn một số ví dụ truy vấn có thể được phân tách và biểu diễn bằng các biểu thức đại số của đại số học TA.

3.3.1 Các hàm thời gian và các thao tác so sánh khoảng

Các ngôn ngữ truy vấn thời gian mức cao hầu hết cung cấp một số các hàm

thời gian được xây dựng bên trong nó và các thao tác so sánh khoảng để biểu diễn các ngữ nghĩa truy vấn thời gian khác nhau. Những hàm và các thao tác này có thể được dùng cùng với các toán tử đại số trong quá trình phân tích đại số của các truy vấn thời gian mức cao. Trong phần này, luận văn mô tả ngắn gọn về các hàm thời gian và các thao tác định nghĩa trong OQL/T [SSH-98]. Một số trong chúng được sử dụng trong các truy vấn đưa ra trong mục 3.3.2.

1) Các hàm đặc tả khoảng (Interval Specification Functions). Các hàm trong danh mục này bao gồm: INTERVAL, TIME(NOW +/-). Chúng trả về các thời gian hiệu lực của các thể hiện thời gian.

- Hàm INTERVAL định nghĩa không tường minh các khoảng thời gian của truy vấn.
- Hàm TIME(NOW +/-) định danh các điểm thời gian (thời điểm) của các thể hiện mô hình liên quan đến thời gian hiện tại.

2) Các toán tử so sánh khoảng (Interval Comparison Operations). Các thao tác so sánh khoảng có thể được sử dụng để định danh các quan hệ thời gian giữa hai khoảng thời gian tường minh hoặc không tường minh. Cho hai khoảng thời gian: $T1$ và $T2$, OQL/T [SSH-98] mô tả 7 thao tác so sánh giữa hai khoảng như sau:

- $T1$ BEFORE $T2$ hoặc $T2$ AFTER $T1$;
- $T1$ PRECEDE $T2$ hoặc $T2$ FOLLOW $T1$;
- $T1$ P-CROSS $T2$ hoặc $T2$ F-CROSS $T1$;
- $T1$ EQUAL $T2$;
- $T1$ L-CONTAIN $T2$ hoặc $T2$ L-WITHIN $T1$;
- $T1$ O-CONTAIN $T2$ hoặc $T2$ I-WITHIN $T1$;
- $T1$ R-CONTAIN $T2$ hoặc $T2$ R-WITHIN $T1$.

Ở đây, P thay cho PRECEDE, F thay cho FOLLOW, L thay cho LEFT, R thay cho RIGHT, O thay cho OUTER, và I thay cho INNER.

Hình thức hoá của các thao tác so sánh này dựa trên hai hàm thời gian chính

là $s()$ và $e()$ được tóm tắt trong mục 3.2.3.

3) Các hàm xác định thứ tự (Ordering Functions). Các hàm trong danh mục này bao gồm: FIRST, LAST, N-TH, B-FIRST (trong đó B thay cho “Backward”), B-LAST, B-NTH, FORMER, và NEXT. Chúng trả về các thể hiện thời gian của các trục thời gian đã được sắp xếp dựa trên các đặc tả khoảng thời gian.

4) Các hàm kết hợp (Aggregate Functions).

- Hàm GROUP_BY nhóm các TPI có cùng đặc tả các mẫu con chung.
- Hàm COUNT đếm số lượng các khai triển một thể hiện mẫu được kết hợp.
- TC-SUM trả về tổng của các giai đoạn liên tiếp nhau của các thể hiện mẫu được kết hợp.
- T-SUM trả về tổng các giai đoạn.
- T-MIN trả về khoảng thời gian ngắn nhất trong một đường thời gian.

5) Các thao tác với cửa sổ thời gian (Time Window Operations). OQL/T định nghĩa hai thao tác trong danh sách này là ANY và EVERY. Chúng thực thi hành động di chuyển cửa sổ (moving-window). Một cửa sổ thời gian [SSH-98] là một giai đoạn thời gian được di chuyển tại một bước cố định từ giới hạn thấp nhất đi về phía giới hạn cao nhất của một khoảng thời gian. Trong một ứng dụng di chuyển cửa sổ, các điều kiện trong một truy vấn thời gian được đánh giá như là phép dịch cửa sổ trong khoảng thời gian.

3.3.2 Một số ví dụ phân tích đại số học

Trong mục này luận văn trình bày một số ví dụ cho các phân tích đại số của các truy vấn mức cao biểu diễn trong phiên bản mới của OQL/T, có sử dụng các thao tác đại số của đại số TA và các hàm/thao tác thời gian. Có thể biểu diễn các biểu thức đại số xen kẽ nhau cho một truy vấn.

Q2: Tìm mức lương của Jonh khi anh ta đang làm việc cho phòng ban có mã

số dept_no = “1” trước khi lần đầu tiên Mary trở thành một quản lý.

Biểu thức đại số TA:

$$\begin{aligned}
 \sigma_1 &= \delta_{[name=John]}(Employee) \\
 \sigma_2 &= \delta_{[dept_no=1]}(Dept) \\
 \sigma_3 &= (\sigma_1 * Engineer) \diamond_{[[Engineer]]} (Engineer * \sigma_2) \\
 \hat{T}_1 &= INTERVAL(\sigma_3) \\
 \sigma_4 &= \delta_{[name=Mary]}(Employee) \\
 \sigma_5 &= \sigma_4 * Manager \\
 \hat{T}_2 &= FIRST(INTERVAL(\sigma_5)) \\
 \sigma_6 &= \delta_{[e(\hat{T}_1) < (\hat{T}_2)]}(\sigma_1) \\
 \sigma_7 &= \sigma_6 * salary \\
 \sigma_8 &= \Pi_{[[salary], \hat{T}_1]}(\sigma_7)
 \end{aligned}$$

Các khoảng $\hat{T}_1, \hat{T}_2$ được xác định bằng hai hàm INTERVAL. Trong σ_6 , chúng được so sánh với định danh các khoảng thời gian của John được kết hợp với phòng ban “1” trong đó phải thỏa mãn là before - trước khi khoảng thời gian Mary trở thành người quản lý đầu tiên và sau đó các khoảng định danh được sử dụng để lựa chọn các thể hiện của John. Mỗi liên hệ thời gian của mức lương của John được định danh trong σ_7 và các giá trị của lương được trích chọn bằng một phép chiếu T- Project trong σ_8 .

Q3: Tìm tất cả các phòng ban có người quản lý phục vụ trong thời gian ngắn nhất?

Biểu thức đại số TA:

$$\begin{aligned}
 \sigma_1 &= \Pi_{[[0, n]]}(Dept) \\
 \sigma_2 &= \Pi_{[[0, n]]}(Manager) \\
 \sigma_3 &= \sigma_1 * \sigma_2 \\
 \sigma_4 &= T-MIN(\sigma_3) \\
 \sigma_5 &= \Pi_{[[Dept]]}(\sigma_4)
 \end{aligned}$$

Tất cả các thể hiện mẫu Dept–Manager được lựa chọn trong σ_1 và σ_2 được tạo thành bằng T-Associate(*) và được gán cho σ_3 . Sau đó, T-MIN(σ_3) định danh các TPI(s) mà chúng có thời gian hiệu lực ngắn nhất. Dãy con T-Project(Π) sinh tương ứng các phòng ban chính là kết quả của truy vấn.

Q4: Tìm tất cả các kỹ sư tham gia vào dự án trong khoảng T[5, 20] và đã làm việc trong phòng ban có số hiệu dept_no = “8” trong khoảng T[30, 40].

Biểu thức đại số TA:

$$\begin{aligned}
 \sigma_1 &= \Pi_{[5, 20]}(\text{Engineer}) \\
 \sigma_2 &= \Pi_{[5, 20]}(\text{Work_On}) \\
 \sigma_3 &= \Pi_{[5, 20]}(\text{Project}) \\
 \sigma_4 &= (\sigma_1 * \sigma_2) \hat{\cap}_{[\text{Work_On}]} (\sigma_2 * \sigma_3) \\
 \sigma_5 &= \Pi_{[30, 40]}(\text{Engineer}) \\
 \sigma_6 &= \delta_{[T(\text{dt}) \supseteq [30, 40] \wedge \text{dept_no}=8]}(\text{Dept}) \\
 \sigma_7 &= \Pi_{[30, 40]}(\sigma_6) \\
 \sigma_8 &= \sigma_5 * \sigma_7 \\
 \sigma_9 &= \sigma_4 \circ_{[\text{Engineer}]} \sigma_8 \\
 \sigma_{10} &= \Pi_{[\text{Engineer}]}(\sigma_9)
 \end{aligned}$$

Ví dụ này minh họa một phép phân tích đại số của thao tác phi thời gian và các kết quả chứa tất cả các thể hiện kỹ sư thỏa mãn hai mô hình độc lập thời gian.

3.4 Kết luận chương 3

Trong chương III, luận văn đã trình bày một đại số kết hợp thời gian, và cung cấp một tập các toán tử đại số được hỗ trợ trực tiếp cho việc xử lý các mẫu phi kết hợp hoặc kết hợp đối tượng thời gian. Đại số TA dùng một sơ đồ dựa mẫu để trình bày, xử lý các đối tượng và các kết hợp AND/OR của chúng trong việc tạo thành các TPS cùng TPI. Các toán tử của nó có khả năng thực hiện trên các cấu trúc không đồng nhất và các cấu trúc đồng nhất

Các toán tử thực hiện trên các mẫu để sản sinh các mẫu. Do vậy, tính chất đóng - closure-property của đại số học chắc chắn rằng các kết quả của một thao tác đại số thời gian có thể thực hiện trên các toán tử khác và nhiều thao tác hơn. Bằng cách trực tiếp hay gián tiếp các kết hợp đối tượng thời gian hay phi thời gian được trình bày trong các TPS và các TPI. Chúng được trình bày rõ ràng hơn các quan hệ ngữ nghĩa giữa các đối tượng thời gian hơn là việc trình bày dựa trên các bảng quan hệ.

CHƯƠNG 4 – MỘT NGÔN NGỮ TRUY VẤN CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG THỜI GIAN

Giới thiệu

Truy vấn CSDL là bài toán không thể thiếu trong các mô hình CSDL. Truy vấn trong CSDL thời gian đã trình bày trong mục 1.1.5. Trong chương này luận văn tập trung trình bày việc đặc tả và thiết kế ngôn ngữ truy vấn thực thi trên CSDL thời gian hướng đối tượng đó là TOQL. TOQL là một trong 3 mô đun đã được luận văn đề cập đến trong mục 2.4.2.

4.1 Đặc tả TOQL

4.1.1 Giới thiệu

TOQL là một mở rộng thích hợp trên cơ sở OQL. Mỗi phát biểu có hiệu lực trong OQL 5.1.2 [EIV-99D] cũng có hiệu lực trong TOQL. Như vậy, khi cho phép những ứng dụng không hợp nhất ngữ nghĩa thời gian tới các chức năng của DBMS mở rộng. TOQL cũng cung cấp những mở rộng cho quản lý dữ liệu thời gian. Những mở rộng này phải gắn liền với toàn bộ cú pháp OQL.

TOQL cung cấp chức năng thời gian đầy đủ với ngôn ngữ mở rộng, các chức năng thích hợp để hỗ trợ các yêu cầu của ứng dụng. Các nhãn thời gian định nghĩa những thao tác thời gian, cùng với việc chọn lọc thời gian và nhãn thời gian cũng phải được cung cấp đầy đủ trong TOQL.

Mục tiêu của TOQL là có cú pháp truy vấn và ngữ nghĩa thời gian cần phải sáng sủa, rõ ràng. Đánh giá câu truy vấn phải giữ được khả năng cập nhật những kết quả tới phạm vi lớn nhất có thể được.

4.1.2 Các kiểu dữ liệu cho trình diễn thời gian

TOQL cung cấp các phương tiện để thao tác trên các kiểu dữ liệu được sử dụng cho trình diễn thời gian, tức là INTERVAL, INSTANT, PERIOD và PERIOD_SET. Một câu truy vấn TOQL có thể chứa những thành tố kiểu

cũng như hàm, vị từ và toán tử được áp dụng trên các kiểu dữ liệu này. Các mục sau mô tả các đặc tính của TOQL .

4.1.2.1 Các thành tố

A - Thời điểm

Một thành tố kiểu Instant được xác định bằng từ khoá Instant, theo sau là chuỗi ký tự chứa một giá trị thực. Trả về chuỗi ký tự là một đặc tả nguyên tố. Đặc tả nguyên tố có thể được đưa ra một đặc tả lịch biểu.

- Khuôn dạng của chuỗi ký tự sau từ khóa Instant là phụ thuộc lịch biểu. Trong lịch Gregorian, chuỗi ký tự một đặc tả Instant là *YYYY - MM - DD HH:MM:SS*.
- Chỉ có *YYYY - năm* là bắt buộc có , còn tất cả các phần khác đều để tùy chọn. Những thành phần bị khuyết sẽ sử dụng những giá trị mặc định (trong lịch Gregorian, mặc định tháng và ngày là 1, giờ, phút và giây có giá trị mặc định là 0).
- Chuỗi ký tự có thể chứa một trong số từ ngữ BEGINNING, FOREVER và NOW, chỉ định tương ứng các nhãn thời gian cực tiểu, cực đại và hiện thời.

Bảng 4. 1 Một số kiểu thời điểm

<i>Kí tự</i>	<i>Giá trị</i>
<i>INSTANT '1990' YEAR CALENDAR Gregorian</i>	<i>Một thời điểm năm 1990 của lịch Gregorian.</i>
<i>INSTANT 'NOW' MINUTE</i>	<i>Thể hiện 'Now' nhưng lưu đơn vị Phút nên sẽ bỏ qua các giá trị dư thừa</i>
<i>Instant 'Spring 1996' SEMESTER CALENDAR Academic</i>	<i>Thời điểm là Mùa xuân năm 1996 với lịch một niên học. Và đây sử dụng đơn vị là một “Học Kỳ”</i>

B - Khoảng

Một kí tự kiểu INTERVAL có thể được xác định bằng từ khoá INTERVAL, theo sau là một chuỗi ký tự chứa giá trị thực. Chuỗi ký tự được trả về, sau đó

là chuỗi đặc tả lịch biểu. Nếu đặc tả nào bị bỏ qua thì giá trị ghi nhận là giá trị mặc định giống như Instant.

Bảng 4. 2 Một số kiểu khoảng

<i>Thành tố</i>	<i>Giá trị</i>
<i>INTERVAL '10' DAY CALENDAR Gregorian</i>	<i>Một khoảng 10 ngày trong lịch Gregorian</i>
<i>INTERVAL '5' YEAR</i>	<i>Một khoảng 5 năm trong Gregorian lịch biểu</i>
<i>INTERVAL '2' SEMESTER CALENDAR Academic</i>	<i>Một khoảng hai học kỳ trong lịch năm học</i>

C - Giai đoạn hay thời kỳ

Kiểu PERIOD được xác định bằng từ khoá PERIOD theo sau bởi một chuỗi ký tự chứa giá trị thực.

- Chuỗi ký tự phải có một giá trị bắt đầu với một dấu móc vuông trái ('['), kết thúc với một dấu ngoặc đơn phải (')') và chứa hai kí tự phân cách chúng là dấu phẩy.
- Duy nhất ' Giá trị chuỗi - 'string value' ' phân chia các kiểu Instant cần phải xác định. Một giá trị kiểu PERIOD được xem bao gồm tất cả các thời điểm từ t_i lên đến t_j nhưng không chứa t_j .

Bảng 4. 3 Một số kiểu giai đoạn

<i>Thành tố</i>	<i>Giá trị</i>
<i>PERIOD '[1990, 1991)' YEAR CALENDAR Gregorian</i>	<i>Năm 1990 của lịch Gregorian</i>
<i>PERIOD '[NOW, 2000-01-01)' DAY</i>	<i>Một giai đoạn bắt đầu từ hiện thời và kết thúc ở ngày đầu tiên năm 2000.</i>
<i>PERIOD '[Winter 1996, Spring 1997)' CALENDAR Academic</i>	<i>Giai đoạn '[Mùa đông 1996, Mùa xuân 1997)' Lưu với lịch niên học từ mùa đông năm trước đến mùa xuân năm sau</i>

D - Tập giai đoạn

Tập hợp giai đoạn – PERIOD SET được xác định bằng một cấu trúc, trong đó

đặt những kí tự có thể được định nghĩa trong OQL. Cấu trúc có tên `period_set`, nó cho phép xác định một giá trị nguyên tố và một ký hiệu lịch biểu. Cấu trúc chấp nhận một danh sách các giá trị của kiểu `PERIOD` mà những phần tử là những tập hợp `PERIOD`.

Tuy nhiên, bộ xử lý TOQL chấp nhận những đặc tả giai đoạn đầy đủ cho bất kỳ tham số nào tới cấu trúc .

Bảng 4. 4 Một số kiểu tập giai đoạn

<i>Thành tố</i>	<i>Giá trị</i>
<code>period_set('[1990,1991)', '[1992, 1994)') YEAR</code>	Một tập giai đoạn chứa hai khoảng. Lịch Gregorian được sử dụng cho tập hợp giai đoạn.
<code>period_set('[Winter 1996, Spring 1997)') SEMESTER CALENDAR Academic</code>	Học kỳ lưu bởi một tập giai đoạn với một giai đoạn đơn , biểu thị trong lịch niên học

4.1.2.2 Hàm

TOQL giới thiệu những hàm mới, có thể ứng dụng trên dữ liệu `Instant`, `Period` và `Period_set`.

Những ký hiệu của các hàm mới, cùng với mô tả ngắn gọn chức năng của chúng được giới thiệu trong những phần sau.

Bảng 4. 5 Danh sách một số hàm mới của TOQL

<i>Cú pháp</i>	<i>Ví dụ</i>
<i>PERIOD period</i> (in <i>Instant I1</i> , in <i>Instant I2</i>)	<code>period(INSTANT '1994-01' MONTH, INSTANT '1995-01' MONTH) = PERIOD '[1994-01, 1995-01)' MONTH</code> <code>period(INSTANT '1995-01' MONTH, INSTANT '1994-01' MONTH) = NIL</code>
<i>PERIOD intersection</i> (in <i>PERIOD P1</i> , in <i>PERIOD P2</i>) <i>PERIOD_SET intersection</i> (in <i>PERIOD_SET PS1</i> , in <i>PERIOD_SET PS2</i>)	<code>intersection ('[1994-01, 1995-01)', '[1995-06, 1996-01)') = NIL</code> <code>intersection(period_set('[1994-01, 1995-01)', '[1996-01, 1997-01)') MONTH, period_set('[1994-06, 1995-06)') MONTH) = {'[1994-06, 1995-01)'}</code>
<i>PERIOD merge</i> (in <i>PERIOD P1</i> , in <i>PERIOD P2</i>);	<code>merge('[1994-01, 1995-01)', '[1995-06, 1996-01)') = NIL</code> <code>merge(period_set('[1994-01, 1995-01)') MONTH,</code>

PERIOD_SET merge (in PERIOD_SET P1, in PERIOD_SET P2)	<code>period_set('[1994-06, 1995-06]') MONTH = {'[1994-01, 1995-06]'}</code>
short year (in Instant I); short day (in Instant I); short hour (in Instant I); short minute (in Instant I);	<code>year(INSTANT '1994-01' MONTH) = 1994</code> <code>day(INSTANT '1994-01-08 04:59:01' SECOND) = 8</code> <code>hour(INSTANT '1994-01-08 04:59:01' SECOND) = 4</code> <code>minute(INSTANT '1994-01-08 04:59:01' SECOND) = 59</code>
INSTANT begin (in PERIOD P); INSTANT end (in PERIOD_SET PS)	<code>begin(PERIOD '[1994-01, 1995-01]' MONTH) = '1994-01'</code> <code>end(PERIOD '[1994-01, 1995-01]' MONTH) = '1995-01'.</code> <code>end(period_set('[1994-01, 1995-01]', '[1995-06, 1997-01]') MONTH) = '1997-01'</code>
INTERVAL duration (in PERIOD P): INTERVAL duration (in PERIOD_SET PS):	<code>duration(PERIOD '[1994-01, 1995-01]' MONTH) = INTERVAL '12' MONTH</code> <code>duration(period_set('[1994, 1995]', '[1996, 1997]') YEAR) = 2</code>
PERIOD first (in PERIOD_SET PS): PERIOD last (in PERIOD_SET PS):	<code>first(period_set('[1994, 1995]', '[1996, 1997]') YEAR) = '1994, 1995'</code> <code>last(period_set('[1994, 1995]', '[1996, 1997]') YEAR) = '1996, 1997'</code>

Ngoài những hàm giới thiệu ở trên, TOQL cho một cú pháp cấu trúc dễ dàng chuyển đổi giữa những kiểu trình bày thời gian đã sử dụng quen thuộc. Để chuyển đổi một phần tử dữ liệu kiểu INSTANT, INTERVAL, PERIOD hoặc PERIOD SET tới một đơn vị khác. Cú pháp cấu trúc sử dụng như sau :

CAST datum TO granularity_specification

trong đó *granularity_specification* là một đơn vị sẽ chuyển đến.

4.1.2.3 Vị từ

TOQL hỗ trợ tất cả các toán tử so sánh được định nghĩa trong OQL và giới thiệu những vị từ mới dễ dàng cho việc định vị tương đối của Instant, những giá trị tập hợp PERIOD và PERIOD. Những vị từ mới được mô tả trong bảng 4.6:

i1 và i2 biểu thị những giá trị Instant; p1 và p2 biểu thị những giá trị giai đoạn;

Bảng 4. 6 Các vị từ mới của TOQL

<i>Cú pháp</i>	<i>Ví dụ</i>
<i>p1 overlaps p2</i> <i>ps1 overlaps ps2</i>	<i>PERIOD '[1994-01, 1995-01]' MONTH overlaps PERIOD '[1994-06, 1995-06]' MONTH trả về True</i> <i>PERIOD '[1994-01, 1995-01]' MONTH overlaps PERIOD '[1995-06, 1996-01]' MONTH trả về False</i>
<i>i1 precedes i2</i> <i>p1 precedes p2</i> <i>ps1 precedes ps2</i>	<i>INSTANT '1990-01' MONTH precedes INSTANT '1991-01' MONTH trả về True</i> <i>PERIOD '[1994-01, 1995-01]' MONTH precedes PERIOD '[1994-06, 1995-06]' MONTH trả về False</i>
<i>p1 contains p2</i> <i>ps1 contains ps2</i>	<i>PERIOD '[1994-01, 1996-01]' MONTH contains PERIOD '[1995-01, 1995-06]' MONTH trả về True</i> <i>PERIOD '[1994-01, 1995-01]' MONTH contains PERIOD '[1994-06, 1995-06]' MONTH trả về False</i>
<i>p1 meets p2</i>	<i>'[1994-01, 1995-01]' meets '[1995-01, 1995-06]' trả về True</i> <i>'[1994-01, 1995-01]' meets '[1994-06, 1995-06]' trả về False</i>

4.1.2.4 Toán tử

Số học và tập hợp chuẩn lý thuyết các toán tử có thể được sử dụng để thực hiện những tính toán trên các kiểu dữ liệu sử dụng thời gian được trình bày sau đây:

Bảng 4. 7 Thao tác trên khoảng

Toán hạng trái	Toán tử	Toán hạng phải	Kết quả
	-	<i>interval</i>	<i>interval</i>
	+	<i>interval</i>	<i>interval</i>
<i>interval</i>	+	<i>interval</i>	<i>interval</i>
<i>interval</i>	-	<i>interval</i>	<i>interval</i>
<i>interval</i>	*	<i>number</i>	<i>interval</i>
<i>number</i>	*	<i>interval</i>	<i>interval</i>
<i>interval</i>	/	<i>number</i>	<i>interval</i>
<i>interval</i>	/	<i>interval</i>	<i>number</i>

Bảng 4. 8 Thao tác trên tập giai đoạn

Toán hạng trái	Toán tử	Toán hạng phải	Kết quả
<i>period_set</i>	<i>union(+)</i>	<i>period_set</i>	<i>period_set</i>
<i>period_set</i>	<i>except(-)</i>	<i>period_set</i>	<i>period_set</i>
<i>period_set</i>	<i>intersect(*)</i>	<i>period_set</i>	<i>period_set</i>
<i>period_set</i>	+	<i>interval</i>	<i>period_set</i>
<i>period_set</i>	-	<i>interval</i>	<i>period_set</i>

Bảng 4. 9 Thao tác trên thời điểm

Toán hạng trái	Toán tử	Toán hạng phải	Kết quả
<i>instant</i>	+	<i>interval</i>	<i>instant</i>
<i>interval</i>	+	<i>instant</i>	<i>instant</i>
<i>instant</i>	-	<i>interval</i>	<i>instant</i>
<i>instant</i>	-	<i>instant</i>	<i>interval</i>

Bảng 4. 10 Thao tác trên giai đoạn

Toán hạng trái	Toán tử	Toán hạng phải	Kết quả
<i>period</i>	+	<i>interval</i>	<i>period</i>
<i>period</i>	-	<i>interval</i>	<i>period</i>
<i>interval</i>	+	<i>period</i>	<i>period</i>
<i>period</i>	>>	<i>interval</i>	<i>period</i>
<i>period</i>	<<	<i>interval</i>	<i>period</i>
<i>period</i>	<i>union (+)</i>	<i>period</i>	<i>period</i>
<i>period</i>	<i>except (-)</i>	<i>period</i>	<i>period</i>
<i>period</i>	<i>intersect(*)</i>	<i>period</i>	<i>period</i>

4.1.2.5 Nguyên tắc tự động chuyển đổi kiểu

Trong các ngữ cảnh của một trong các hàm, vị từ và toán tử đã mô tả tại các mục 4.1.2.2, đến 4.1.2.4 sử dụng những tham số và bộ xử lý TOQL đưa ra những chuyển đổi kiểu tự động, để có thể đưa ra một hàm với tham số thích hợp.

Nếu không phù hợp, bộ xử lý TOQL cố gắng chuyển đổi tự động để liên kết các toán hạng về cùng đơn vị của chúng. Các mục sau đây giới thiệu các tính chất này của TOQL.

A - Chuyển đổi kiểu

- Thời điểm là đẳng cấu (isomorphic) tới những Period thông thường, do đó, nếu một thời điểm (Instant) được sử dụng trong một Period được yêu cầu, thì thời điểm I1 của đơn vị G1 được chuyển đổi tới giai đoạn, bắt đầu là I1 và kết thúc bằng $I1 + INTERVAL \text{ '1' } G1$ (G1 là đơn vị chỉ định).
- Period là đẳng cấu tới những Period_set đơn, do đó, nếu giai đoạn P1 với đơn vị G1 sử dụng Period_set được yêu cầu, thì Period được chuyển đổi tới một Period_set có giá trị bằng $period_set(P1) G1$.

Kết hợp hai quy tắc chuyển đổi giới thiệu ở trên, thời điểm phải đẳng cấu tới tập giai đoạn; do đó, nếu một Instant được sử dụng một Period_set, thì Instant đầu tiên được chuyển đổi tới một Period và sau đó tới một Period_set, theo quy tắc chuyển đổi thích hợp.

Trong sự chuyển đổi này, đơn vị giây và lịch Gregorian được sử dụng.

Ví dụ:

G1 là đơn vị của instant1, period1.

period1 meets instant1 chuyển đổi thành

period1 meets period(instant1, instant1 + interval '1' G1)

B - Chuyển đổi đơn vị

TOQL nhận đơn vị của toán hạng bên trái và chuyển đổi đơn vị về đơn vị của toán hạng này.

Ví dụ với thao tác sau:

INSTANT '1990-01-31' DAY + INTERVAL '1' MONTH

4.1.3 Cơ sở dữ liệu mẫu

Với phần dữ liệu thời gian, [EIV-99D] đưa ra một CSDL mẫu, luận văn sẽ sử dụng cho các ví dụ ở mục 4.1.4 và sử dụng CSDL mẫu này trong cài đặt Demo ứng dụng. CSDL này chứa thông tin về nhật ký của các sản phẩm sữa và các nhà máy mà chúng được sản xuất. Các lược đồ sau mô tả thông qua

các câu lệnh của TODL :

```

interface      Product
(extent Products
  key   ProductName)
{ // Các tính chất thể hiện
    attribute String ProductName;
    attribute String Description;
    attribute float AvgSales valid state overlaps
                granularity month;
    attribute List<String> Ingredients transaction;
    attribute Interval LifeTime valid state granularity day;
    relationship Set<Factory> manufactured valid state
granularity month transaction
inverse Factory::manufactures;
}

interface      Factory
(extent Factories
  key   Owner, Location)
{ // Các tính chất thể hiện
    attribute String Owner;
    attribute String Location;
    attribute String Manager valid state granularity month;
    attribute Long Turnover valid state granularity month;
    relationship Set<Product> manufactures valid state
granularity month transaction
inverse Product::manufactured;
}

```

Giả sử chúng chứa các nội dung sau:

Obj001	ProductName: 'Life Orange Juice' Description: 'Orange juice. Sold in 330ml, 1 lt and 2 lt packages' AvgSales: {(value: 1000, VT: [1995-01, 1996-01]), (value: 1300, VT: [1995-06, 1996-06]), (value: 1400, VT: [1996-01, 1997-01]), (value: 1600, VT: [1996-06, 1997-06])}, Ingredients: {(value: {'Concentrated Orange Juice', 'Water', 'Sugar', 'E210'}, TT: [1995-01-01, 1995-06-01]), (value: {'Concentrated Orange Juice', 'Water', 'Sugar'}, TT: [1995-06-01, 1996-01-01]), (value: {'Concentrated Orange Juice', 'Water', 'E210'}, TT: [1995-06-01, UC])}, LifeTime: {(value: 18, VT: [1995-01-01, 1995-06-01]), (value: 9, [1995-06-01, 1996-06-01]), (value: 12, [1996-06-01, forever])}, Manufactured: {(value: {Ref<Obj005>}, VT: [1995-01-01, 1996-06-01], TT: [1994-12-01, UC]), (value: {Ref<Obj005>}, VT: [1996-06-01, 1997-01-01], TT: [1994-12-01, 1995-11-16]), (value: {Ref<Obj005>, Ref<Obj006>}, VT: [1996-06-01, 1997-01-01], TT: [1995-11-16, UC]), (value: {Ref<Obj006>}, VT: [1996-07-01, 1997-06-01], TT: [1996-04-10, UC])}
--------	--

Obj002
ProductName: 'Champion' Description: 'Chocolate milk with honey and malt. Sold in packages of 330 ml.' AvgSales: {(value: 600, VT: [1994-01, 1994-06]), (value: 500, VT: [1994-06, 1996 -09]), (value: 450, VT: [1994-06, 1995-01]) } Ingredients: {(value: {'Cow''s Milk', 'Honey', 'Chocolate', 'Malt', 'Color'}, TT: [1994-01, 1994-05]), (value: {'Cow''s Milk', 'Honey', 'Chocolate', 'Malt'}, TT: [1994-05, 1995-02])} LifeTime: {(value: 6, VT: [1994-01-01, 1994-06-01]), (value: 9, [1994-06-01, 1995-01-01]), (value: 12, [1995-06-01, forever])} Manufactured: {(value: {Ref<Obj005>}, VT: [1994-01-01, 1996-01-01], TT: [1993-11-07, 1994-10-10]), (value: {Ref<Obj005>}, VT: [1994-06-01, 1995-01-01], TT: [1994-10-10, UC])}

Obj005
Owner: 'Delta Dairy S.A.' Location: 'St. Stefanos' Manager: {(value: 'Stefanou', VT: [1994-01, 1995-01]), (value: 'Nikou', VT: [1995-01, 1996-09])} Turnover: {(value: 8000, VT: [1995-01, 1996-01]), (value: 9000, [1996-01, 1996-06])} Manufactures: {(value: {Ref<Obj002>}, VT: [1994-01-01, 1996-01-01], TT: [1993-11-07, 1994-10-10]), (value: {Ref<Obj002>}, VT: [1994-06-01, 1995-01-01], TT: [1994-10-10, UC]), (value: {Ref<Obj001>}, VT: [1995-01-01, 1997-01-01], TT: [1994-12-01, UC])}

Obj006
Owner: '3E Canning Co.' Location: 'Korinthos' Manager: {(value: 'Andreou', VT: [1993-06, 1995-04]), (value: 'Ioannou', VT: [1995-04, 1997-01])} Turnover: {(value: 5000, VT: [1994-01, 1994-09]), (value: 6000, VT: [1995-01, 1995-07])} Manufactures: {(value: {Ref<Obj001>}, VT: [1996-06-01, 1997-01-01], TT: [1995-11-16, 1996-04-10]), (value: {Ref<Obj001>}, VT: [1996-06-01, 1997-06-01], TT: [1996-04-10, UC])}

4.1.4 Truy vấn trên dữ liệu thời gian

4.1.4.1 Các kiểu biểu thức hỗ trợ xử lý truy vấn của TOQL

Truy vấn đơn giản nhất là trả về tất cả các đối tượng trong một phạm vi (Ví dụ: trong phạm vi *Products*). Kết quả chứa tất cả các thể hiện sản phẩm đã được ghi vào CSDL .

Products ->Kiểu trả về là một *Bag<Product>*.

Các đối tượng thời gian (các đối tượng thời gian hiệu lực, giao dịch và đa chiều) cũng có thể được thực hiện giải quyết như tập lựa chọn các chỉ mục - *indexed collections*, ban đầu đến các danh sách, các tập, các túi và các mảng được hỗ trợ bởi OQL. Một số kiểu biểu thức hỗ trợ như sau:

Bảng 4. 11 Danh sách các biểu thức hỗ trợ của TOQL

Tên biểu thức và cú pháp(nếu có)	Một số biểu thức
----------------------------------	------------------

Biểu thức cho đối tượng trạng thái hiệu lực không cho phép nhân chồng lấp và đối tượng sự kiện hiệu lực : Set<struct(Value: T, VT: timestampType)>	count(valid_obj);first(valid_obj); last(valid_obj);valid_obj[number]; valid_obj[n1:n2];valid_obj[instant]; valid_obj[period]
Biểu thức cho đối tượng trạng thái hiệu lực cho phép nhân chồng lấp: Set<struct(Value: T, VT: period)>	count(valid_obj);first(valid_obj); last(valid_obj);valid_obj[number]; valid_obj[n1:n2];valid_obj[instant];
Biểu thức cho đối tượng thời gian giao dịch Set<struct(Value: T, TT: period)>	count(trans_obj);first(trans_obj); last(trans_obj);trans_obj[number];
Biểu thức cho đối tượng bitemporal không cho phép các nhân thời gian hiệu lực và các đối tượng sự kiện bitemporal chồng lên nhau	count(bitemp_obj);bitemp_obj[valid at instant]; bitemp_obj[current at instant]; bitemp_obj[valid at instant1, current at instant2];
Biểu thức cho những đối tượng bitemporal mà cho phép nhân thời gian hiệu lực chồng lên nhau	count(bitemp_obj); bitemp_obj[valid at instant]; bitemp_obj[current at instant]; bitemp_obj[valid at instant1; current at instant2]

4.1.4.2 Các ví dụ

Những truy vấn sau có thể được trả lời bằng TOQL, sử dụng những biểu thức mô tả tại mục 4.1.4.1 :

Truy vấn: Tìm thời gian vòng đời sản phẩm “Life Orange Juice” khi nó lần đầu tiên được giới thiệu?

```
select first(valid p->LifePeriod)
from Products as p
where p->ProductName = "Life Orange Juice"
```

Biểu thức(valid p->LifePeriod)[1] có thể được dùng thay vì sử dụng first(valid p->LifePeriod)

Truy vấn: Tìm thời gian vòng đời sản phẩm “Life Orange Juice” Khi nó lần đầu tiên được giới thiệu và khi vòng đời của nó được thay đổi?

```
select first(valid p->LifePeriod) as LifePeriod,
end(valid(first(valid p->LifePeriod))) as ChangePoint
from Products as p
where p->ProductName = "Life Orange Juice"
```

Hàm giao dịch cũng được hỗ trợ và có thể được áp dụng vào những biến thể trích chọn từ thời gian giao dịch hoặc những đối tượng bitemporal.

Truy vấn: Tìm các vòng đời của các sản phẩm “Life Orange Juice” từ 1994-01-01?

```
select (valid p->LifePeriod) [INSTANT '1994-01-01' DAY]
from Products as p
where p->ProductName = "Life Orange Juice"
```

Truy vấn: Tìm vòng đời của sữa “Champion” khi nó được đưa ra trên thị trường?

```
select last(valid p->LifePeriod)
from Products as p
where p->ProductName = "Champion"
```

Truy vấn: Những thành phần nào được dùng cho sữa “Champion” trong vòng đời của nó?

```
select transaction p->Ingredients
from Products as p
where p->ProductName = "Champion"
```

4.1.4.3 Kiểu kết quả của các toán tử kịch bản con

Tương tự các biểu thức được mô tả tại mục 4.1.4.1, nếu một toán tử kịch bản con lựa chọn hơn một phương án áp dụng trên một đối tượng thời gian, kết quả của toán tử là một tập các cấu trúc, với mỗi cấu trúc chứa phương án đánh giá cùng một hoặc hai nhãn thời gian, phụ thuộc vào toán tử kịch bản con và các chiều thời gian được hỗ trợ bởi đối tượng thời gian.

Nếu các toán tử kịch bản con trả lại các đối tượng thời gian, thì có thể người dùng công thức hóa truy vấn lựa chọn một đối tượng thời gian giao dịch mà không bao gồm thông tin hiện thời.

Đối tượng này có thể đang gán tới một thể hiện thời gian mà thuộc tính sử dụng, ngôn ngữ lập trình của chính nó, hoặc thông qua một phương pháp cập nhật trong một truy vấn, như được thấy trong ví dụ sau :

```
select p->set_ingredient_history((transaction p->Ingredients)
[period '[1990-01-01, 1991-01-01)' day])
from Products as p
```

Phương pháp ***set_ingredient_history*** được sử dụng trong tập ví dụ mà toàn bộ lịch sử thuộc tính Ingredient của đối tượng nhận được bằng lịch sử có trong tham số.

4.1.4.4 Hỗ trợ các giá trị bị hủy bỏ và tiến triển

Theo mặc định, tất cả các các toán tử kịch bản con bị bỏ qua với các giá trị vị trí định vị đã bị xóa [EIV-99D] và chỉ quan tâm đến các giá trị có vị trí định vị như “evolved” hoặc “current”. Tuy nhiên, những truy nhập tới phương án đã bị xóa được cung cấp tới biểu thức kịch bản con có một trong những từ khóa như evolved, deleted hoặc cả hai. Từ khóa evolved là mặc định, như vậy sự có mặt của nó không thay đổi kiểu kết quả hoặc giá trị mà được mô tả trong mục 4.1.4.1.

Tuy nhiên nếu biểu thức đi theo bằng từ khóa deleted hoặc tất cả từ khóa, thì kiểu và giá trị kết quả được sửa đổi như sau :

set<struct(value: T, VT: timestampType, TT: period, flag: variant_type)>.

T là kiểu nhãn thời gian, timestampType là hoặc thời điểm hoặc giai đoạn, phụ thuộc vào thời gian hiệu lực của đối tượng thời gian được xác định tương ứng như sự kiện hoặc trạng thái và variant_type là kiểu liệt kê (current, evolved, deleted).

Một trong số các thành phần VT và TT có thể không xuất hiện trong mô hình kết quả, nếu kết quả của các toán tử tương ứng không chứa các thành phần tương ứng, như được mô tả tại mục 4.1.4.1.

Mô tả tương ứng truy vấn:

```
select (valid p->LifePeriod) [INSTANT '1994-01-01' DAY] all
from Products as p
where p->ProductName = 'Life Orange Juice'
```

sẽ trả lại một cái bag chứa một phần tử đơn.

Phần tử này sẽ là một tập các cấu trúc, mỗi cấu trúc trong tập kết quả có một giá trị kiểu khoảng, một nhãn thời gian kiểu giai đoạn và một cờ hiệu kiểu variant_type. Cấu trúc này tương ứng tới tất cả các phương án trong lịch sử hiệu lực “Life Orange Juice” giai đoạn vòng đời của sản phẩm, các nhãn thời gian hiệu lực chồng lên nhau với thời điểm đã gán tên (‘1994 -01-01’). Trạng thái của mỗi phương án (evolved hoặc deleted) được báo cáo trong thành

phần flag.

4.1.4.5 Trích chọn các trạng thái đối tượng

Nếu một lớp đã được khai báo có ngữ nghĩa thời gian hiệu lực và/hoặc ngữ nghĩa thời gian giao dịch, những biểu thức được mô tả tại mục 4.1.4.1 có thể được sử dụng với những biểu thức có giá trị là một thể hiện của lớp.

Ví dụ: Nếu khai báo lớp Factory như sau:

```
interface Factory
  (extent Factories
   key Owner, Location)
//Đơn vị trạng thái hiệu lực giao dịch ngày đơn vị;
// Các thuộc tính
  attribute String Owner;
  attribute String Location;
  attribute String Manager;
  relationship Product manufactures
  inverse Product::manufactured;
```

Sau đó các truy vấn

```
select (bitemporal state f)[valid at INSTANT '1995-01-01' DAY,
current at INSTANT 'NOW' DAY]
from Factories as f
```

và

```
select (bitemporal state f)[current at INSTANT 'NOW' DAY,
valid at PERIOD '[1995-01-01, 1995-06-01)' DAY]
from Factories as f
```

có thể sử dụng các trạng thái của các nhà máy, như được ghi vào trạng thái cơ sở dữ liệu hiện thời cho thời điểm ngày 01/01/1995 và giai đoạn là năm tháng đầu tiên của năm 1995. Như vậy, cú pháp để đưa ra một thao tác trích chọn trạng thái trên một đối tượng là:

(time_dimension state temporal_object)extraction_operation

trong đó *time_dimension* có thể là một trong những từ khóa valid, transaction và bitemporal, và *extraction_operation* phải là một toán tử kịch bản con có khả năng áp dụng trên các đối tượng được xác định các chiều thời gian.

Từ khóa valid có thể được sử dụng cho những đối tượng thời gian hiệu lực và bitemporal; từ khóa transaction thích hợp với những đối tượng thời gian giao

dịch, những đối tượng sự kiện bitemporal và đối tượng trạng thái bitemporal không chồng nhau của các nhãn thời gian hiệu lực; từ khóa bitemporal chỉ cho phép trên những đối tượng bitemporal.

Kiểu kết quả của một thao tác trích chọn trạng thái sẽ là một trạng thái đối tượng, hoặc một tập các cấu trúc, với mỗi cấu trúc giữ một trạng thái đối tượng và một nhãn thời gian (Hoặc hai nhãn thời gian), phụ thuộc vào số lượng trạng thái của biểu thức.

Những quy tắc để xác định số lượng các nhãn thời gian tương tự với những quy tắc mô tả tại mục 4.1.4.1 .

Những bổ nghĩa như: trạng thái time_dimension có thể đặt trước những đối tượng thời gian khi chúng được sử dụng như một tham số tới một trong số các hàm count, first và last.

Trong những trường hợp này, những hàm này trả lại tương ứng số lượng trạng thái trong đối tượng, trạng thái với nhãn thời gian nhỏ nhất và trạng thái với nhãn thời gian lớn nhất. Những hàm first và last có thể không được sử dụng khi time_dimension là tập bitemporal.

4.1.4.6 Giải quyết vấn đề ngữ nghĩa nhập nhằng

Trong các biểu thức thì việc xuất hiện sự nhập nhằng giữa các đối tượng thời gian có thể xuất hiện trong cách dùng các toán tử là thành viên anh em (. và ->), những bổ nghĩa thời gian (valid, transaction và bitemporal), các toán tử kịch bản con với những hàm đặc biệt (count, first, last), các toán tử có thể có ý nghĩa cho cả hai chiều thời gian và thông thường trong tham số.

Ví dụ với truy vấn

```
select count (p->Ingredients)
from Products as p
where p->ProductName = "Life Orange Juice"
```

có thể được giải thích là “ Tìm số lượng những thành phần hiện thời được sử dụng cho sản phẩm Life Orange Juice” (Hàm áp dụng tới chiều thông thường của thuộc tính Ingredients) Hoặc “ Tìm số lần mà các thành phần của sản

phẩm Life Orange Juice được chuyển đổi” (Hàm áp dụng tới chiều thời gian của thuộc tính Ingredients).

Sự nhập nhằng này được giải quyết bởi việc cho một quy tắc ngữ nghĩa, bắt đầu theo thứ tự cho hàm hoặc các toán tử để áp dụng với chiều thời gian của tham số của nó, tham số phải được đặt trước với một từ bỏ nghĩa thời gian; cách khác, hàm áp dụng đến chiều thông thường.

Quy tắc ngữ nghĩa này đảm bảo tính thông thường, một khi ngữ nghĩa của những truy vấn thông thường không chuyển đổi nếu mô hình cơ sở dữ liệu đã được làm giàu thêm bằng ngữ nghĩa thời gian.

Như vậy, để thực hiện liệt kê danh sách đầu tiên của các ingredients cho sản phẩm “Life Orange Juice” đã được ghi vào trong cơ sở dữ liệu, truy vấn phải được phát biểu như sau:

```
select (transaction p->Ingredients)[1]
from Products as p
where p->productName = "Life Orange Juice"
```

Nếu dấu ngoặc bị bỏ qua từ truy vấn này, lựa chọn danh sách đọc *transaction p->Ingredients[1]* thuộc tính Ingredients được chuyển đổi tới giá trị thông thường của nó, các toán tử kịch bản con sau đó được áp dụng.

Cuối cùng, thực hiện ước lượng từ bỏ nghĩa giao dịch trên kết quả chuỗi của các toán tử kịch bản con và như vậy kết quả có một lỗi ngữ nghĩa, một khi những từ bỏ nghĩa thời gian không thích hợp trên chuỗi.

4.1.5 Tham chiếu đến các đối tượng thay đổi

Khi một biến được định nghĩa dưới dạng một đối tượng thời gian, nó có nhiều biến thể hiện thời gian được lưu trữ trong đối tượng thời gian. Tất cả các phương án đều được xem xét, cho dù nhãn thời gian của chúng là hiệu lực hay giao dịch.

Định nghĩa biến được sử dụng như tham số trong các hàm hiệu lực và giao dịch để trả lại nhãn thời gian tương ứng. Một đối tượng thời gian có thể được định nghĩa trong mệnh đề với cú pháp:

temporal_object as identifier.

Trong các cấu trúc định lượng phổ biến và các cấu trúc tồn tại, một biến được định nghĩa sử dụng trong từ khóa. Vì thế, một truy vấn định lượng tồn tại kéo theo một biến định nghĩa trên một đối tượng thời gian có thể thức như sau:

exists identifier in temporal_object: query

Trong khi mà một truy vấn định lượng phổ thông kéo theo một biến định nghĩa trên một đối tượng thời gian được viết :

for all identifier in temporal_object: query

Những đối tượng thời gian có thể được sử dụng trong không gian của một tập cho phép một truy vấn thành viên kiểm thử (một thành phần trong tập hợp).

Ví dụ

1. Truy vấn: Lựa chọn những sản phẩm đã có một vòng đời bằng 15 ngày?

```
select p from Products as p
where exists l in valid p->LifeTime: l = INTERVAL '15' DAY
```

Truy vấn này có thể được công thức hóa sử dụng thành phần kiểm tra như sau:

```
select p from Products as p
where INTERVAL '15' DAY in valid p->LifeTime
```

Truy vấn này có thể được phát biểu sử dụng bằng một vị từ phức như sau:

```
select p from Products as p
where INTERVAL '15' DAY = any valid p->LifeTime
```

2. Truy vấn: Cho mỗi sản phẩm, danh sách thời gian tên và vòng đời của nó, cùng với giai đoạn tương ứng

```
select p->ProductName as ProductName,
l as LifeTime,
valid(l) as LifeTimePeriod
from Products as p, valid p->LifeTime as l
```

3. Truy vấn: Cho mỗi sản phẩm, đưa ra danh sách tên và những thành phần của nó, nếu những thành phần chứa ' E210 '.

```
select p->ProductName as ProductName,
p->Ingredients as Ingredients,
transaction(Ingredients) as TT
from Products as p, transaction p->Ingredients as Ingredients
```

where exists Ingredient in Ingredients: Ingredient = 'E210'

4.1.6 Chuyển đổi giữa giá trị thời gian và thông thường

4.1.6.1 Bổ nghĩa thông thường

TOQL cung cấp khả năng chuyển đổi những đối tượng thời gian tới một tập các giá trị thông thường, sử dụng từ bổ nghĩa thông thường. Từ bổ nghĩa thông thường có thể đặt trước bất kỳ truy vấn nào trả về một đối tượng thời gian và có hiệu quả trên tất cả nhãn thời gian mà chúng trả về các giá trị đơn.

Ví dụ, cho truy vấn:

```
select p->ProductName as ProductName,  
max(snapshot p->LifeTime) as MaxLifeTime  
from Products as p
```

Truy vấn trên trả về một tập các bộ, mỗi bộ dữ liệu là tên của một sản phẩm, và vòng đời dài nhất mà nó từng có.

4.1.5.2 Bổ nghĩa hiệu lực

Từ bổ nghĩa hiệu lực sử dụng để chuyển đổi những tập tới những đối tượng thời gian hiệu lực. Cú pháp bổ nghĩa hiệu lực là:

```
valid valid_type [overlaps] [granularity] [calendar]  
'[query : expression]'
```

trong đó truy vấn là bất kỳ truy vấn nào trả về tập hợp và biểu thức là một truy vấn cung cấp các nhãn thời gian.

Ví dụ với truy vấn:

```
valid state [select Ingredient  
from transaction aProduct->Ingredients as Ingredient :  
transaction(Ingredient)]
```

Sinh một đối tượng thời gian hiệu lực, với mỗi phương án tương ứng với một giá trị thông thường của lịch sử thời gian giao dịch của thuộc tính thành phần.

Những quy tắc để xác định kiểu kết quả như sau : nếu *valid_type* là trạng thái, thì sự chỉ định các nhãn chồng lấp bị bỏ qua và đơn vị là *G* và lịch biểu *C* được mô tả, sau đó kiểu kết quả là *<type>* đơn vị trạng thái hiệu lực *G* lịch biểu *C*.

Một lỗi sẽ nảy sinh nếu:

- Biểu thức sinh ra các giai đoạn chồng lên nhau.
- Biểu thức sinh ra các giai đoạn đồng nhất.
- Biểu thức sinh ra những thời điểm đồng nhất.

Trong mọi trường hợp, đặc tả lịch biểu có thể bị bỏ qua, nếu đặc tả nào vắng mặt tới lịch Gregorian. Cuối cùng, < Type> chưa được phép có những đặc trưng thời gian, trong bất kỳ trường hợp nào.

4.1.6.3 Bổ nghĩa giao dịch

Từ bổ nghĩa giao dịch sử dụng để chuyển đổi những đối tượng thông thường tới những đối tượng thời gian giao dịch. Cú pháp là:

transaction '[' query ']

Trong đó truy vấn là bất kỳ truy vấn nào trả về một đối tượng đơn.

Kết quả chứa một phương án đơn, giá trị của chúng là tập hợp kết quả của truy vấn, trong khi timestamp của nó là tập hợp [NOW, UC).

Mỗi phần tử của tập hợp là một đối tượng thời gian giao dịch. Kiểu của kết quả với từ bổ nghĩa giao dịch phụ thuộc vào kiểu trả về bởi truy vấn được bao trong dấu móc vuông.

4.1.6.4 Bổ nghĩa bitemporal

Từ bổ nghĩa bitemporal sử dụng để chuyển đổi những tập hợp tới những đối tượng bitemporal. Cú pháp của từ bổ nghĩa bitemporal là:

bitemporal valid_type [overlaps] [granularity] [calendar]

'[query : expression']'

chức năng của từ bổ nghĩa này tương tự từ bổ nghĩa hiệu lực, ngoại trừ nó sinh ra những đối tượng bitemporal, thay vì đối tượng thời gian hiệu lực. Nhân thời gian giao dịch của mỗi phương án là tập hợp [NOW, UC).

4.1.7 Kết nối thời gian

Kết nối thời gian là cần thiết khi thông tin lưu giữ trong hai (hoặc hơn) đối tượng thời gian phải được kết hợp.

Xem xét truy vấn ' Với mỗi nhà máy, hãy liệt kê chủ nhân của nó, vị trí địa lý và tiền doanh thu, cùng với tên của giám đốc tương ứng với các giai đoạn thời gian '. Thông tin về tiền doanh thu của nhà máy được ghi trong thuộc tính thể hiện tiền doanh thu, trong khi thông tin về những giám đốc của nhà máy được lưu giữ trong thuộc tính thể hiện Giám đốc. Cả hai thuộc tính thể hiện này có ngữ nghĩa thời gian hiệu lực.

Truy vấn có thể được trả lời trong TOQL sử dụng việc tham chiếu đối tượng thay đổi, được minh họa ở bên dưới

```
select f->Owner as Owner, f->Location as Location,  
t as TurnOver, m as Manager,  
intersection(valid(t), valid(m)) as time  
from Factories as f, valid f->Turnover as t, valid f->Manager as m  
where valid(t) overlaps valid(m)
```

Truy vấn ở trên có thể xuất hiện nhiều phần tử cho mỗi nhà máy, một cho mỗi kết hợp tiền doanh thu và giám đốc với những giai đoạn thời gian hiệu lực chồng lên nhau. Mô hình kết quả có thể được chuyển đổi để sản sinh một phần tử cho mỗi nhà máy, gồm có ba thành phần, hai thể hiện đầu là chủ nhân và vị trí của nhà máy và thể hiện thứ ba là đối tượng thời gian hiệu lực của kiểu *struct(Turnover: integer, Manager: string)* trạng thái hiệu lực có đơn vị là tháng.

Điều này có thể đạt được nhờ sử dụng từ bỏ nghĩa hiệu lực như sau

```
select f->Owner as Owner, f->Location as Location,  
Info: valid state [select t as Turnover, m as Manager  
from valid f->Turnover as t,  
valid f->Manager as m  
where valid(t) overlaps valid(m)] :  
intersection(valid(t), valid(m))]  
from Factories as f
```

Sử dụng toán tử tstruct, truy vấn ở trên có thể được trình bày như sau:

```
select f->Owner as Owner, f->Location as Location,  
tstruct(Turnover: valid f->Turnover,  
Manager: valid f->Manager) as Info
```

from Factories as f

Như vậy, cú pháp cho toán tử *tstruct* tương tự như cú pháp của toán tử *struct* chuẩn. Ví dụ: *tstruct(identifier: query {, identifier: query})* trong đó mỗi truy vấn ước lượng tới một đối tượng thời gian.

4.1.7.1 Kiểu kết quả của kết nối thời gian

Kiểu kết quả của toán tử *tstruct* phụ thuộc vào các kiểu tham số của nó. Người dùng được phép tính kết quả của thời gian kết nối tới một lớp thời gian đã được khai báo trong cơ sở dữ liệu thông qua TODL, sử dụng ký hiệu OQL chuẩn cho sự chuyển đổi kiểu.

Cả hai tham số đến toán tử *tstruct* phải sử dụng cùng lịch biểu cho nhãn timestamps hiệu lực của chúng. Tuy nhiên, đơn vị không cần phải tương tự và, trong trường hợp mà các đơn vị của những tham số không đối sánh được, đơn vị của kết quả là thời gian hiệu lực được đặt làm đơn vị của tham số bên trái thời gian hiệu lực.

- Một đối tượng trạng thái hiệu lực không chồng lên nhau, chứa những biến thể khác nhau với nhãn thời gian chồng lấp.
- Một đối tượng trạng thái hiệu lực cho phép chồng lên nhau, chứa những biến thể khác nhau với nhãn thời gian đồng nhất.
- Một đối tượng sự kiện hiệu lực, chứa những biến thể khác nhau với nhãn thời gian đồng nhất. Trong tất cả những trường hợp này, một ngoại lệ được phát sinh để thông báo lỗi.

Quy tắc để xác định kiểu kết quả của một kết nối thời gian được phân phối từ ngữ nghĩa phân biệt của kiểu đối tượng thời gian hiệu lực:

- Những đối tượng trạng thái hiệu lực không cho phép chồng lấp lưu giữ thông tin được áp dụng đến mức giai đoạn cũng như mức thời điểm.
- Những đối tượng trạng thái hiệu lực cho phép thông tin lưu giữ chồng lấp được áp dụng ở mức giai đoạn.

- Những đối tượng sự kiện hiệu lực chỉ lưu giữ thông tin được áp dụng ở mức thời điểm.

Trong mọi trường hợp, thứ tự kết quả của một kết nối thời gian hợp nhất ở một số mức thông tin (Period hoặc Instant), mức này của thông tin phải được bao gồm cả hai toán hạng.

4.1.7.2 Giá trị kết quả của một kết nối thời gian

Các quy tắc áp dụng khi hai đối tượng thời gian được kết nối thông qua toán tử *tstruct*.

Nếu VO1 và VO2 là hai đối tượng tùy theo kiểu kết nối:

- Kiểu trạng thái hiệu lực $\langle T1 \rangle$ và trạng thái hiệu lực $\langle T2 \rangle$, $Tstruct(id1: VO1, id2: VO2)$ sinh ra một đối tượng trạng thái hiệu lực không cho phép chồng lấp các nhãn thời gian hiệu lực và có kiểu hiệu lực $state\langle struct(id1: T1, id2: T2) \rangle$.
- Kiểu trạng thái hiệu lực $overlaps\langle T1 \rangle$ và trạng thái hiệu lực $overlaps\langle T2 \rangle$, $struct(id1: VO1, id2: VO2)$ sinh ra một đối tượng thời gian hiệu lực cho phép chồng lấp của các nhãn thời gian hiệu lực và có kiểu trạng thái hiệu lực $overlaps\langle struct(id1: T1, id2: T2) \rangle$.
- Kiểu hiệu lực $event\langle T1 \rangle$ và hiệu lực $event\langle T2 \rangle$, $Tstruct(id1: VO1, id2: VO2)$ sinh ra một đối tượng sự kiện hiệu lực của kiểu hiệu lực $event\langle struct(id1: T1, id2: T2) \rangle$.
- Kiểu hiệu lực $state\langle T1 \rangle$ và kiểu trạng thái hiệu lực $overlaps\langle T2 \rangle$, $Tstruct(id1: VO1, id2: VO2)$ sinh ra một đối tượng trạng thái hiệu lực của kiểu trạng thái hiệu lực $overlaps\langle struct(id1: T1, id2: T2) \rangle$.
- Kiểu hiệu lực $state\langle T1 \rangle$ và kiểu hiệu lực $event\langle T2 \rangle$, $Tstruct(id1: VO1, id2: VO2)$ sinh ra một đối tượng sự kiện hiệu lực của kiểu hiệu lực $event\langle struct(id1: T1, id2: T2) \rangle$.

Bảng 4. 12 Điều kiện của các kiểu kết nối thời gian

Tên kiểu kết nối	Điều kiện
	Một phương án v với giá trị <i>struct(id1: x1, id2: x2)</i> và nhãn thời gian hiệu lực <i>validv</i> thuộc về đối tượng kết quả nếu những điều kiện sau thoả mãn :
Hai đối tượng trạng thái hiệu lực không chồng lên nhau	- Phương án v1 với giá trị x1 và timestamp hiệu lực <i>validv1</i> thuộc về VO1. - Phương án v2 với giá trị x2 và timestamp hiệu lực <i>validv2</i> thuộc về VO2. - <i>intersection(validv1, validv2) = validv</i>
Hai đối tượng trạng thái hiệu lực cho phép chồng lên nhau	- Phương án v1 với giá trị x1 và timestamp hiệu lực <i>validv1</i> thuộc về VO1. - Phương án v2 với giá trị x2 và timestamp hiệu lực <i>validv2</i> thuộc về VO2. - <i>validv1 = validv</i> và <i>validv2 = validv</i>
Hai đối tượng sự kiện hiệu lực	- Phương án v1 với giá trị x1 và timestamp hiệu lực <i>validv1</i> thuộc về VO1. - Phương án v2 với giá trị x2 và timestamp hiệu lực <i>validv2</i> thuộc về VO2. - <i>validv1 = validv</i> và <i>validv2 = validv</i> .
Một đối tượng trạng thái hiệu lực không chồng lấp và một đối tượng trạng thái hiệu lực chồng lấp	- Phương án v1 với giá trị x1 và timestamp hiệu lực <i>validv1</i> thuộc về VO1. - Phương án v2 với giá trị x2 và timestamp hiệu lực <i>validv2</i> thuộc về VO2. - (<i>validv1</i> chứa <i>validv</i> hoặc <i>validv1 = validv</i>) và <i>validv2 = validv</i>
Một đối tượng trạng thái hiệu lực không chồng lấp và một đối tượng sự kiện hiệu lực	- Phương án v1 với giá trị x1 và timestamp hiệu lực <i>validv1</i> thuộc về VO1. - Phương án v2 với giá trị x2 và timestamp hiệu lực <i>validv2</i> thuộc về VO2. - (<i>validv1</i> chứa <i>validv</i> hoặc <i>validv1 = validv</i>) và <i>validv2 = validv</i> .

4.1.7.3 Kết nối thời gian trên nhiều đối tượng

Toán tử *tstruct* có thể được áp dụng tới bất kỳ số lượng của các đối tượng thời gian *to1, to2,... ton* với các kiểu $\langle T1 \rangle vtc1, \langle T2 \rangle vtc2,... \langle Tn \rangle vtcn$, tương ứng (*Ti* biểu diễn kiểu nhãn thời gian, trong khi *vtci* đại diện những đặc trưng thời gian hiệu lực của những đối tượng tương ứng). Trong trường hợp này, kiểu đối tượng kết quả được xác định như sau :

- Kiểu nhãn thời gian là *struct(id1: T1, id2: T2,..., idn: Tn)*, trong đó *idi* là những định danh được sử dụng trong toán tử *tstruct*.
- Đơn vị của timestamps hiệu lực là tập đơn vị của các timestamps hiệu lực của đối tượng bên trái nhất.
- Ngữ nghĩa của thời gian hiệu lực của kết quả là tập những đặc trưng

thời gian hiệu lực với mức ưu tiên cao nhất, trong số những cái liên kết.

4.1.8 *Toán tử cấu trúc lại*

Toán tử cấu trúc lại là tiện ích cho hình thức hoá các biểu diễn tương đương khác nhau của dữ liệu thời gian. Hai toán tử cấu trúc lại được cung cấp, toán tử thứ nhất chuyển đổi *period-timestamped* tới các *instant-timestamped*, trong khi cấu trúc toán tử thứ hai cho phép chọn trực thời gian trên các nhãn thời gian cực đại được sinh ra. Các toán tử này được giới thiệu sau đây:

Chuyển đổi *period-timestamped* tới *instant-timestamped*

Cú pháp đưa ra theo toán tử tổ chức lại là:

temporal_object(partition time_axis as instant)

Trong đó *time_axis* là trực thời gian hoặc là *valid* hoặc *transaction*. Áp dụng toán tử cấu trúc lại (phân hoạch hiệu lực như thời điểm) trên một đối tượng của kiểu *T* trạng thái hiệu lực *G1* lịch biểu *C1* (*G1* đơn vị của timestamps hiệu lực) sinh ra một kết quả kiểu *set<struct(Value: T, VT: Instant)>*, với những thời điểm có đơn vị *G1* và lịch biểu *C1*.

Một phần tử (*vr*, *vtr*) thuộc về tập kết quả nếu một phương án (*vvar*, *vtvar*) xuất hiện trong đối tượng thời gian hiệu lực và những điều kiện sau được thoả mãn :

vr = *vvar*
vtvar chứa *vtr*

Tương tự, áp dụng toán tử Phân hoạch giao dịch như thời điểm trên một đối tượng kiểu giao dịch *T*, kết quả trong một *set<struct(Value: T, TT: Instant)>*.

Một phần tử (*vr*, *ttr*) thuộc về tập kết quả nếu một phương án (*vvar*, *ttvar*) xuất hiện trong đối tượng thời gian giao dịch và những điều kiện sau được thoả :

vr = *vvar*
ttvar chứa *ttr*

Nếu cùng toán tử đó áp dụng trên một đối tượng *bitemporal*, thì mô hình kết quả mở rộng đến những phần tử của tập kết quả bao gồm một thành phần mở rộng với tên *VT*.

Ví dụ, truy vấn: Cho nhà máy sở hữu “Delta Dairy S.A.” tại “St.Stefanos”,
Liệt kê giám đốc của mỗi tháng:

```
select MonthInfo
from Factories as f,
(valid f->Manager) (partition valid as instant) as MonthInfo
where f->Owner = "Delta Dairy S.A"
and f->Location = "St.Stefanos"
```

Từ giá trị của thành phần Manager cho đối tượng chứa thông tin về nhà máy
sở hữu “Delta Dairy S.A.” tại “ St.Stefanos” là đối tượng trạng thái hiệu lực

```
{(value: 'Stefanou', VT: [1994-01, 1995-01]),
(value: 'Nikou', VT: [1995-01, 1996-09])}
```

(xem mục 4.1.3), kết quả của truy vấn này có thể tương ứng như sau:

```
{(Value: 'Stefanou', VT: 1994-01), (Value: 'Stefanou', VT: 1994-
02), ..., (Value: 'Stefanou', VT: 1994-12),
```

Sinh ra nhãn thời gian cực đại.

Mô hình dữ liệu có thể chọn bất kỳ kiểu biểu diễn nào để lưu giữ thông tin
thời gian vào trong cơ sở dữ liệu, nhưng những truy vấn khác nhau có thể cần
có sự trình bày khác nhau, để sinh ra các kết quả đúng.

Toán tử cấu trúc thứ hai, có cú pháp là:

temporal_object(partition time_axis as period)

trong đó time_axis có thể hoặc hiệu lực hoặc giao dịch.

Khi các toán tử được áp dụng tới một đối tượng của kiểu T trạng thái hiệu lực
G1 và lịch biểu C1 (G1 là đơn vị của timestamps hiệu lực), nó sửa đổi những
biến thể của đối tượng thời gian để sinh số lần cực đại trên trục xác định và
sản sinh một kết quả kiểu *set<struct(Value: T, VT: period, TT: Period)>*.

Một phần tử (vres, vtres, ttres) xuất hiện trong tập kết quả, nếu một phương án
với cùng giá trị và bằng timestamps xuất hiện trong thể thức cấu trúc lại của
đối tượng thời gian.

Cú pháp chuyển đổi được giữ tối thiểu giữa thể thức ban đầu và thể thức cuối
cùng của truy vấn.

Trên các giới hạn của biến nào tồn tại mệnh đề được định nghĩa đã được sửa

đổi từ T trạng thái hiệu lực G1 lịch biểu C1 giao dịch *set<struct(Value: T, VT: period, TT: period)>* .

4.1.9 Tập hợp

OQL 5.1.2 giới thiệu mệnh đề **group by**, phân chia kết quả của một truy vấn **select/from/where** vào các tập hợp con, mỗi một phần tử chứa những giá trị đồng nhất trong một hoặc nhiều thuộc tính.

Ví dụ đối với truy vấn:

```
select p
from Products as p
group by p->Description like '*milk*' as IsMilk
```

Lựa chọn tất cả các sản phẩm từ Products và tách chúng thành hai tập hợp con, phụ thuộc vào mô tả sản phẩm chứa sữa từ không chứa sữa.

Những phần tử của tập con đầu tiên của thuộc tính *expression p->Description like '*milk*' được mô tả trong mệnh đề **group by** đánh giá trả về True (đúng) cho mỗi một trong số chúng, trong khi cùng biểu thức đó ước lượng trả về False (sai) cho mỗi thành viên của tập con thứ hai.*

Mô hình kết quả của truy vấn là *bag<struct(IsMilk: boolean, partition: bag<Product>)>*, Trong đó việc phân hoạch vùng lưu trữ tất cả các phần tử của tập hợp con, trong khi IsMilk lưu giữ giá trị chung của biểu thức group-by cho tất cả các thành viên của tập con.

Ví dụ với truy vấn:

```
select IsMilk, partition as TheProducts,
count(partition) as NumProds
from Products as p
group by p->Description like '*milk*' as IsMilk
```

Trong khi IsMilk lần nữa phân chia những đối tượng trong Product vào trong hai tập hợp, phụ thuộc vào mô tả của chúng có chứa sữa hay không, nhưng đồng thời tính toán số lượng của các sản phẩm trong mỗi tập con và trả về bản số tính toán được vào trong thành phần NumProducts của mô hình kết quả (Mô hình kết quả bây giờ *bag<struct(IsMilk: boolean, TheProducts:*

bag<Product>, NumProducts: integer)>).

Cuối cùng, sau khi công thức hoá tập con rõ ràng có xảy ra, một số tập con có thể được lọc ra khỏi kết quả cuối cùng, phụ thuộc vào việc chúng có thỏa mãn một điều kiện hoặc không, khi có sử dụng mệnh đề. Điều kiện trong mệnh đề *having* được đánh giá đối lập với những phần tử của mỗi tập con (các thành viên của thành phần phân hoạch mỗi cấu trúc kết quả) và cấu trúc kết quả xuất hiện trong kết quả chỉ khi điều kiện ước lượng trả về *true* (đúng).

Do vậy truy vấn:

```
select IsMilk, partition as TheProducts,
count(partition) as NumProds
from Products as p
group by p->Description like '*milk*' as IsMilk
having avg(select p1->LifeTime
from partition as p1) > INTERVAL '12' DAY
```

phân chia đối tượng trong phạm vi *Products* như được mô tả ở trên, nhưng chứa mỗi tập con trong kết quả cuối cùng chỉ khi thời gian sống hiện thời trung bình của các đối tượng trong nó vượt hơn 12 Ngày.

TOQL cung cấp hai kiểu phân hoạch thời gian.

- Kiểu thứ nhất cho phép phân hoạch từng phần một đối tượng thời gian đơn vào trong các tập con khác nhau, mỗi cái gắn với một phần đặc biệt của trục thời gian.
- Phương pháp phân hoạch thứ hai cho phép kết hợp những biến thể của nhiều đối tượng thời gian gắn với một phần đặc biệt của trục thời gian vào trong một tập hợp đơn.

Hai kiểu phân hoạch, cùng với phương pháp cho việc trích chọn phân hoạch và tính toán giá trị tổng thể được mô tả tại các mục sau.

4.1.9.1 Phân hoạch một đối tượng thời gian đơn

Kiểu phân hoạch đầu tiên tách một đối tượng thời gian đơn thành những tập hợp của những phương án, với mỗi tập hợp chứa những phương án gắn liền với một giai đoạn đặc biệt. Phân hoạch có thể được thực hiện trên trục thời

gian hiệu lực hoặc trên trục thời gian giao dịch.

Cú pháp của thể thức phân hoạch này là:

***TemporalObject(partition time_axis as interval_query
[leading interval_query] [trailing interval_query] [calendar])***

Với truy vấn “ Cho tất cả các sản phẩm, hãy đưa ra tên và quá trình sống của chúng mỗi năm trong lịch calendric “

```
select p->ProductName as Name,  
       (valid p->LifeTime) (partition valid as INTERVAL '1' YEAR calendar)  
as YearlyLifeTime  
from Products as p
```

Thành phần YearlyLifeTime trong mô hình kết quả của truy vấn này là một tập các cấu trúc, với mỗi cấu trúc tương ứng với một phân hoạch của trục thời gian hiệu lực với khoảng thời gian bằng một năm.

Mỗi cấu trúc chứa hai thành phần thành viên.

- Thành viên thứ nhất có tên TimeSlice, kiểu của nó là Period và chứa phần của trục thời gian mà thông tin của cấu trúc này liên quan. Đơn vị và lịch biểu của thành phần TimeSlice được lấy từ các đặc trưng tương ứng của nhãn thời gian trong trục thời gian phân hoạch.
- Thành viên thứ hai là một tập có tên Partition, và chứa giá trị và nhãn thời gian hiệu lực của những biến thể liên quan đến thành phần trục thời gian hiệu lực lưu giữ trong thành phần TimeSlice tương ứng.

Trong trường hợp tổng quát, nếu là một đối tượng thời gian, mô hình thành viên Partition do một thao tác phân hoạch là một cấu trúc, thì những thành phần được xác định như sau :

Cấu trúc luôn luôn chứa một thành phần có tên là Value - Giá trị mà có kiểu đồng nhất với nhãn thời gian của kiểu đối tượng thời gian .

Nếu TO có ngữ nghĩa thời gian hiệu lực, cấu trúc chứa một thành phần có tên VT lưu giữ thời gian hiệu lực của phương án. Kiểu của thành phần VT là Period, nếu TO có ngữ nghĩa trạng thái hiệu lực. Trong tất cả các trường hợp,

đơn vị và lịch biểu của thành phần VT đồng nhất với các đặc trưng tương ứng của TO có nhãn thời gian hiệu lực.

Nếu TO có ngữ nghĩa thời gian giao dịch, cấu trúc chứa một thành phần có tên TT lưu giữ phương án có thời gian giao dịch. Kiểu thành phần TT luôn luôn là Period với lịch biểu và đơn vị mặc định sử dụng cho chiều hệ thống thời gian giao dịch .

Một số đặc trưng bổ sung của các toán tử phân hoạch được biểu diễn.

- Đặc trưng thứ nhất là giá trị đặc biệt *forever* trong nhãn thời gian hiệu lực là tình huống con bởi giá trị của nhãn thời gian hiện thời cho những mục đích phân hoạch.
- Đặc trưng thứ hai của thủ tục phân hoạch biểu diễn trong ví dụ trên là việc điều khiển các biến thể *period-timestamped*, những nhãn thời gian chưa hoàn toàn được chứa trong một đoạn đơn trên trục thời gian.

Cuối cùng, đơn vị phân hoạch cơ sở, cũng như những biểu thức khoảng trong những mệnh đề tùy ý và kéo theo không cần phải chuyển đổi cho tất cả các đối tượng.

4.1.9.2 Kết hợp những biến thể từ nhiều đối tượng thời gian

Kiểu phân hoạch con thứ hai cho phép kết hợp những biến thể từ nhiều đối tượng thời gian vào trong các nhóm, với mỗi nhóm gắn liền với một phần đặc biệt hoặc trục thời gian hiệu lực hoặc trục thời gian giao dịch.

Việc lọc nhóm và tính toán giá trị tổng thể được thực hiện bằng việc dùng những cơ chế OQL chuẩn, ví dụ tương ứng mệnh đề *where* và các hàm liên hợp. Cú pháp thay thế của mệnh đề *group by* là:

**group by time_axis interval_query [leading interval_query]
[trailing interval_query] as identifier**

với *time_axis* hoặc hiệu lực hoặc giao dịch.

Chiều thời gian chỉ định trong mệnh đề *group by* phải xuất hiện trong những

đối tượng từ truy vấn *select/from/where*.

Đơn vị và các lịch biểu của tất cả các truy vấn khoảng phải đồng nhất; đơn vị tất yếu không đối sánh được với những đặc trưng của đối tượng thời gian có phân chia trực thời gian.

4.2 Bộ xử lý TOQL

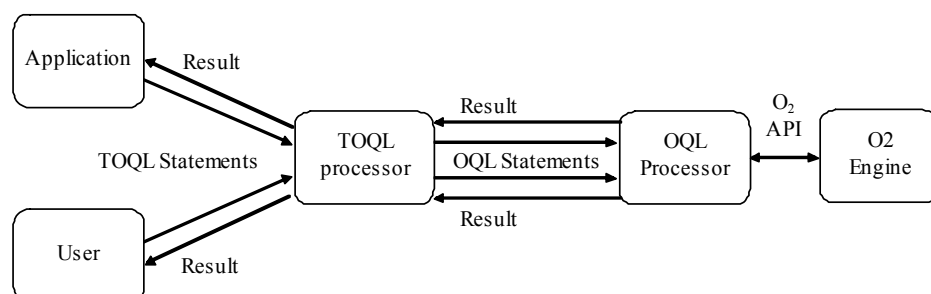
4.2.1 Giới thiệu

Trong mục này, luận văn giới thiệu bộ xử lý truy vấn TOQL. Mục đích đề xuất và giải quyết trong thiết kế bộ xử lý [EIV-99D] :

- Chức năng thời gian đầy đủ cần phải điều phối
- Tính tương thích đầy đủ với DBMS thông thường
- Hiệu năng thực hiện
- Tính khả chuyên

4.2.2 Kiến trúc bộ xử lý TOQL

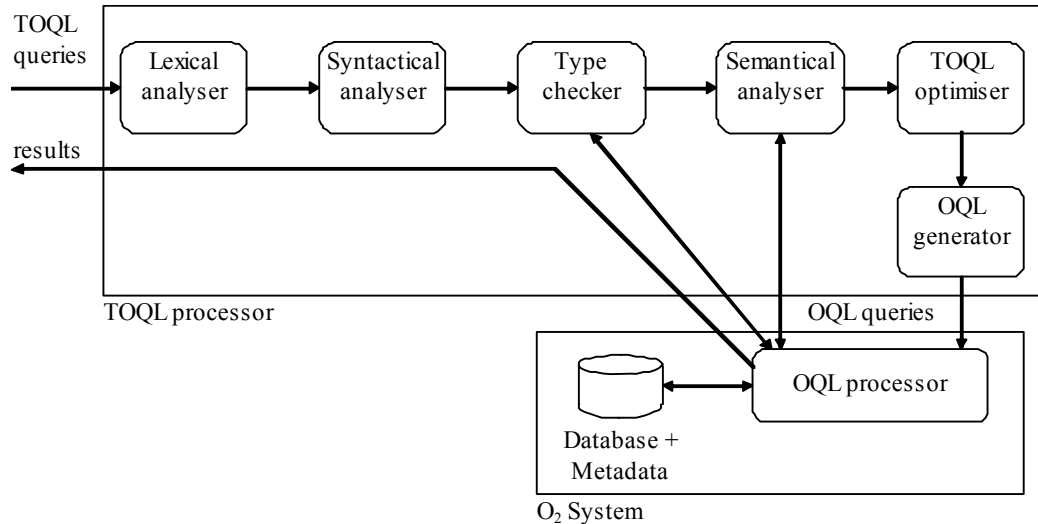
Bộ xử lý TOQL được thực hiện như chức năng mô đun phần mềm ở trên bộ xử lý OQL của O2. Những truy vấn TOQL sẽ được giữ lại và chuyển đổi tới những truy vấn OQL, chúng sẽ trình bày tới bộ xử lý OQL cho việc đánh giá. Những kết quả trở lại bởi bộ xử lý OQL được chuyển tới cho người dùng, hoặc ứng dụng mà đã đưa ra truy vấn TOQL. Sơ đồ này được minh họa trong Hình 4.2.



Hình 4. 1 Kiến trúc bộ xử lý TOQL

Chính bộ xử lý TOQL chi tiết hơn đó là được chia nhỏ ra thành những mô

đơn phần mềm nhỏ hơn, với mỗi trong số chúng có trách nhiệm với một giai đoạn của việc xử lý truy vấn, như được minh họa trong Hình 4.3. Những mô đơn này được mô tả tại các mục sau.



Hình 4. 2 Mô đơn bộ xử lý TOQL

- Phân tích từ vựng (Lexical analyser) xử lý những truy vấn đang trình bày tới bộ xử lý TOQL và chia nó thành từng phần với dấu hiệu từ vựng, với mỗi trong số chúng tương ứng tới một phần tử của TOQL.
- Kiểm tra cú pháp (syntax checker) đọc dòng từ vựng có các dấu hiệu do những người phân tích và kiểm tra từ vựng đã tạo thành một truy vấn TOQL hiệu lực.
- Phân tích cú pháp (Syntactical analyser) phát sinh một cây cú pháp, mô tả cấu trúc của truy vấn TOQL. Kiểm tra kiểu (type checker) duyệt qua cây cú pháp, việc kiểm tra thao tác xác định trong truy vấn tính hợp lệ đối với các kiểu tham số của chúng.
- Mô đơn phân tích ngữ nghĩa (semantical analyser module) xác định những thao tác thực tế phải được kéo theo, để ước lượng truy vấn).

TOQL tối ưu hóa sắp xếp bằng việc loại bỏ những thao tác thừa hoặc làm gọn những thao tác có thể có ở những nơi có thể.

Mô đơn khái quát OQL xử lý những cây cú pháp và cấu trúc dữ liệu Troduced

bởi việc kiểm tra kiểu, mô đun phân tích ngữ nghĩa và công thức hóa một truy vấn OQL, mà sẽ thu được những kết quả mong muốn.

4.2.3 Xử lý truy vấn

Như được trình bày trong các mục 4.2.2, bộ xử lý TOQL phân tích những truy vấn biểu diễn đến nó và chuyển đổi chúng thành những truy vấn OQL tương đương, mà sau này được cho vào trong bộ xử lý OQL của hệ thống O2. Trong những mục sau, những biến đổi cú pháp thực hiện bởi bộ xử lý TOQL và những phương pháp phụ trợ được sử dụng trong cách viết lại truy vấn được đưa ra.

4.2.3.1 Biến đổi cú pháp

Tất cả các đặc tả trong phần đặc tả đều được biến đổi phù hợp sang C++ khi thiết kế TOQL. Có thể tham khảo chi tiết trong [EIV-99D]

4.2.3.2 Thực thi TOQL ở trên OQL 5.1.1 tương thích OODBMS

TOQL, như được mô tả trong mục 1.2 tới 1.9 là một mở rộng chắc chắn đến chuẩn gần đây hơn của OQL, tức là là phiên bản 1.2 ([EIV-99D]) Trong phiên bản 1.2, một số cấu trúc cú pháp của phiên bản 1.1 ([EIV-99D]) không hiệu lực, một khi chúng đã được thay thế bởi SQL - 92 giống với các cấu trúc cú pháp. Mặc dầu cú pháp chuyển đổi trong OQL 5.1.1 là substantial, thiết kế mô đun bộ xử lý TOQL kể cả chuyển bộ xử lý TOQL ở trên mọi OQL 5.1.1 OO - DBMS tương thích hoặc ở trên một OQL 5.1.2 OO - DBMS tương thích với hiệu năng cực tiểu. Đặc biệt, chỉ mô đun Generator OQL cần sửa đổi như một cổng giao tiếp, như vậy trong khi phát sinh ra mã chuyên biệt về phiên bản cho cấu trúc cú pháp thì không đồng nhất trong cả hai đặc tả ngôn ngữ. Đặc tính duy nhất của OQL 5.1.2 không trực tiếp được hỗ trợ ở trên một OQL 5.1.1 conformant OO - DBMS là sắp xếp chung với thứ tự sắp xếp tùy ý.

4.3 Cài đặt chương trình

Trong phần này luận văn giới thiệu chương trình cài đặt bản demo dựa trên

CSDL mẫu đã giới thiệu trong mục 4.1.3

Do TOQL là một ngôn ngữ truy vấn vừa mới được xây dựng và các kết quả của nó đang trong quá trình thử nghiệm nên trong phần cài đặt này luận văn sử dụng một cơ sở dữ liệu hướng đối tượng đó là Ozone với phiên bản 1.2.1 và cài đặt mở rộng thêm các lớp đối tượng đã nói đến trong các mục từ 4.1.3 cho đến 4.2

Ozone là một hệ quản trị cơ sở dữ liệu đối tượng hỗ trợ ngôn ngữ lập trình Java. Chương trình demo dựa trên các truy vấn và xây dựng một số hàm đã trình bày trong nội dung của chương 4.

Với CSDL đã được luận văn giới thiệu ở mục 4.1.3 thì mỗi đối tượng được cài đặt dưới dạng một cây mà mỗi cành là một thuộc tính, nếu muốn thêm một thuộc tính mới hoặc chỉnh sửa một thuộc tính đã có chúng ta dựa vào tính chất của cây để thực hiện.

Với các giá trị của Obj001 tại mục 4.1.3 ta có nội dung hiển thị trong demo theo các nội dung sau:

Một số chức năng:

Run thực thi truy vấn demo dựa trên các giá trị và thuộc tính lựa chọn theo:

Object: là nhóm tên các loại đối tượng trong CSDL

Name: Tên của một đối tượng cụ thể trong CSDL

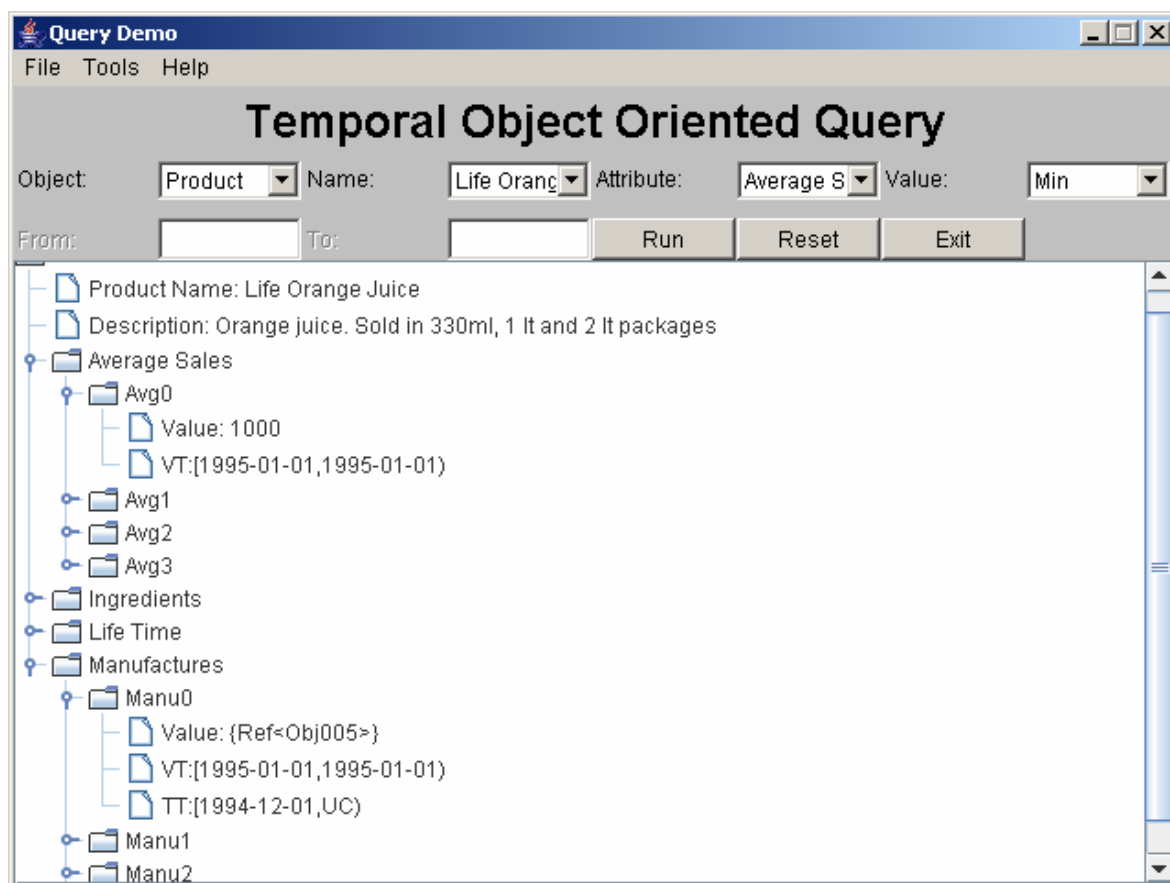
Attribute: Các thuộc tính của đối tượng Name trong CSDL

Value: Giá trị tại một thời điểm của các thuộc tính của đối tượng Name

ReSet: trả về các thuộc tính mặc định và thực thi truy vấn mặc định

Exit: Thoát khỏi chương trình

File: Một số thao tác với tệp câu lệnh truy vấn



Hình 4. 3 Biểu diễn đối tượng Obj001

Ví dụ: Với Obj001

Object: Product

Name: Life Orange Juice

Attribute: AvgSales, Ingredients, LifeTime, Manufactured

Trong đó giá trị của các thuộc tính như sau:

ProductName: Tên sản phẩm

Description: Mô tả thêm thể tích, mẫu mã sản phẩm

AvgSales: Sản lượng bán trung bình trong một khoảng thời gian

Ingredients: Các thành phần của sản phẩm

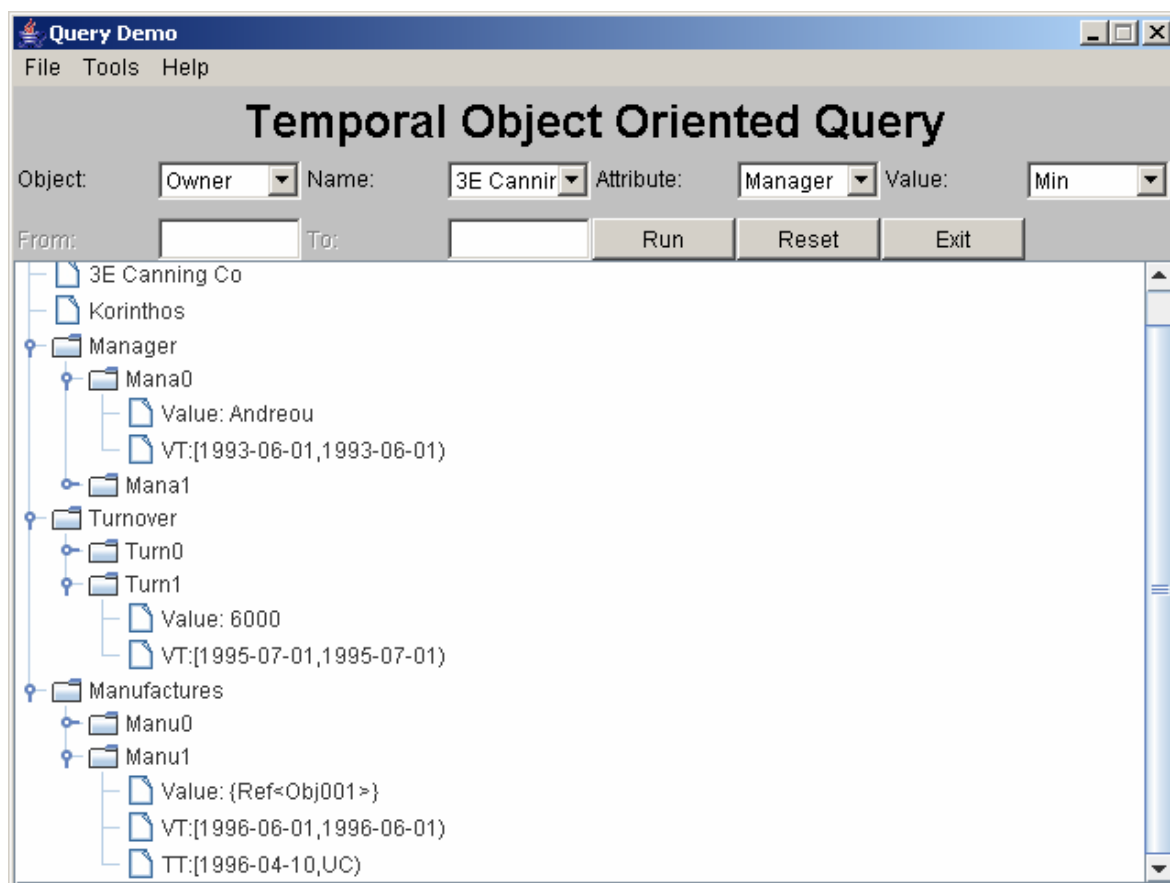
LifeTime: Vòng đời của sản phẩm

Manufactured: Nơi sản xuất

VT: ValidTime (Khoảng thời gian hiệu lực của thuộc tính)

TT: Transaction (Khoảng thời gian giao dịch của thuộc tính)

Giữa Obj001 và Obj006 có sự tham chiếu thông qua thuộc tính Manufactured. Nhìn vào hình 4.4 ta thấy thuộc tính Manufactured có Value là <Obj001> như vậy, đối tượng Obj006 tham chiếu đến Obj001 nhưng Obj001 lại tham chiếu đến Obj005 ... đó là sự khác biệt của CSDL đối tượng.



Hình 4. 4 Biểu diễn đối tượng Obj006

Dựa vào các tính chất của đối tượng và cách biểu diễn này chúng ta có thể truy vấn dễ dàng dựa trên các giá trị thuộc tính và các hàm đã xây dựng dựa theo ý tưởng thiết kế trong mục 4.1.

4.4 Kết luận chương 4

Trong chương 4, luận văn đã trình bày đặc tả và bộ xử lý TOQL mở rộng từ OQL 5. 1.2. Mở rộng này phù hợp với OQL, trong đó TOQL vẫn lưu giữ cú pháp và ngữ nghĩa của OQL. Bộ xử lý TOQL có đầy đủ các tính năng như: tính khả chuyển qua bất kỳ các ODMG DBMS tương thích, xử lý được truy vấn thông thường và truy vấn thời gian. Đánh giá truy vấn dựa vào việc viết lại truy vấn và phương pháp kích hoạt, cho phép khai thác những kỹ thuật tối ưu bên dưới OODBMS.

CHƯƠNG V: KẾT LUẬN

Cơ sở dữ liệu thời gian với đặc trưng không mất mát thông tin có ý nghĩa rất lớn đối với các ứng dụng thực tế. Quản lý cơ sở dữ liệu trong các ứng dụng thông tin có yếu tố thời gian, có yêu cầu lưu trữ, tìm kiếm, xử lý các dữ liệu liên quan đến khái niệm thời gian là rất quan trọng trong việc phát triển các hệ thống thông tin hiện nay. Hầu hết các ứng dụng quản lý cơ sở dữ liệu hiện nay đều sử dụng các hệ quản trị CSDL quan hệ thông thường. Cho nên nếu có yêu cầu về mặt thời gian thì phải mở rộng thêm các trường trong các bảng quan hệ. Như vậy, vừa phải yêu cầu thêm về bộ nhớ lưu trữ vừa phải yêu cầu thêm về thời gian truy xuất thông tin. Điều này làm hạn chế hiệu quả của hệ thống và đó cũng là điều không một ai trong nhóm những người phát triển hệ thống và người sử dụng mong muốn.

Hệ quản trị CSDL hướng đối tượng thời gian ra đời không những tiết kiệm về mặt bộ nhớ cho hệ thống mà còn tiết kiệm về thời gian truy xuất và hơn thế, các phương pháp truy xuất dữ liệu đơn giản và không khác gì so với cách truy xuất trong các hệ quản trị CSDL quan hệ. Đó là một thành công của các nhà khoa học nghiên cứu về CSDL hướng đối tượng thời gian.

Trong luận văn đã trình bày được cơ sở lý thuyết của cơ sở dữ liệu hướng đối tượng thời gian trong các hệ thống thông tin (TOOBIS) cũng như trình bày cơ sở toán học (Đại số TA) cho việc xử lý cơ sở dữ liệu hướng đối tượng thời gian.

Đại số TA là cơ sở toán học cho việc xử lý CSDL hướng đối tượng thời gian thông qua cách biểu diễn đồ thị và lược đồ. Nó cung cấp đầy đủ các phép toán như đối với CSDL quan hệ và cung cấp các phép toán dành cho việc xử lý dữ liệu đối tượng thời gian. Dựa vào cơ sở toán học này mà TOOBIS đã xây dựng ngôn ngữ TOQL để truy vấn dữ liệu thời gian hướng đối tượng.

TOOBIS đã dựa trên CSDL đối tượng và mở rộng thêm mặt thời gian để xây dựng nên phương pháp luận hướng đối tượng thời gian TOOM. TOOM được

sử dụng trong pha phân tích và thiết kế vòng đời của các hệ thống thông tin có yêu cầu về mặt thời gian. TOOM, ngoài khả năng thực hiện việc nắm bắt và mô hình hóa nó còn cung cấp cách thức xây dựng mô hình bao gồm cả khía cạnh cấu trúc lẫn khía cạnh hành vi của hệ thống thông tin. Từ phương pháp luận TOOM, TOOBIS xây dựng nên ngôn ngữ định nghĩa đối tượng thời gian TODL thông qua mô hình định nghĩa hướng đối tượng thời gian TODM. TODL hỗ trợ tất cả ngữ nghĩa xây dựng mô hình hóa dữ liệu bên dưới (TODM), là ngôn ngữ định nghĩa cho các đặc tả đối tượng và là một ngôn ngữ lập trình độc lập. Thông qua TODL, TOOBIS xây dựng nên ngôn ngữ truy vấn TOQL.

Về mặt ứng dụng, luận văn đã trình bày đặc tả ngôn ngữ truy vấn TOQL cho CSDL hướng đối tượng thời gian với các tính năng đầy đủ cho một ngôn ngữ truy vấn độc lập và tuân theo các chuẩn của ODMG. TOQL có đầy đủ các chức năng thời gian được yêu cầu và tương thích với DBMS thông thường. Do đó hiệu năng thực hiện được đánh giá cao và khả năng chuyển đổi giữa các ứng dụng là đáng tin cậy.

Về mặt hạn chế của cơ sở dữ liệu hướng đối tượng thời gian là việc quản lý các đối tượng như thế nào cho hợp lý vì trên thực tế các đối tượng là các con trỏ và điều phối con trỏ là một vấn đề đang được nghiên cứu tiếp theo của cơ sở dữ liệu đối tượng đặc biệt là cơ sở dữ liệu đối tượng quản lý về mặt thời gian.

Hơn nữa, vấn đề biểu diễn dữ liệu dưới dạng đối tượng cũng gây nên sự khó hiểu, nhập nhằng giữa lập trình hướng đối tượng và cơ sở dữ liệu hướng đối tượng. Bản chất của việc biểu diễn dữ liệu dưới dạng đối tượng thì gần với bản chất vật mang tin thực tế nhưng lại gây khó hiểu đối với người lập trình và người phát triển ứng dụng vì đã quen nhìn nhận các bảng biểu trên CSDL quan hệ thông thường.

Việc quản lý CSDL đối tượng có yếu tố thời gian đối với những nhà phát triển ứng dụng yêu cầu sự trong suốt đối với người sử dụng sẽ gặp khó khăn

nhất là trong các ứng dụng phân tán, các ứng dụng mạng và việc nhân bản các CSDL đối tượng thời gian, có yêu cầu trong suốt về mặt thời gian...

Những hạn chế này của hướng nghiên cứu trong đề tài cũng sẽ là những vấn đề em mong muốn sẽ nghiên cứu tiếp trong thời gian gần nhất để thu được những kết quả toàn vẹn và đầy đủ hơn.

Luận văn được hoàn thành trong một thời gian hạn hẹp nên không thể tránh những thiếu sót, em rất mong muốn nhận được những ý kiến đóng góp của các thầy cô, các bạn để cùng nghiên cứu và hoàn thiện đầy đủ hơn. Em xin chân thành cảm ơn.

DANH SÁCH TÀI LIỆU THAM KHẢO

TÀI LIỆU TIẾNG VIỆT

1. [NKA-04] Nguyễn Kim Anh (2004), *Nguyên lý của các hệ cơ sở dữ liệu*, Nhà xuất bản Đại học Quốc gia Hà Nội, tr. 52 – 57, tr. 177 – 182.
2. [NN-02] Nguyễn Kim Anh, Nguyễn Thái Linh (2002), *Các kỹ thuật Index đối với cơ sở dữ liệu thời gian*, Luận văn thạc sỹ khoa học Công nghệ Thông tin, Trường Đại học Bách Khoa Hà Nội, tr. 2 – 14.
3. [MTO-00] M.Tamer Ozsü (2000), *Nguyên lý các hệ cơ sở dữ liệu phân tán*, Nhà xuất bản Thống kê, Hà Nội, tr. 487 – 557

TÀI LIỆU TIẾNG ANH

1. [EIV-99F] European Community IV Frame work, Esprit Project 20671 (1999), Final Project Report - Deliverable T12R.3, *TOOBIS - Temporal Object-Oriented Databases within Information Systems*.
2. [EIV-99D] European Community IV Frame work, Esprit Project 20671 (1999), Deliverable T33TR.1 *TOOBIS - Temporal Object Query Language Specifications and Design*.
3. [SSH-98] Stanley Y.W.Su (Senior member, IEEE), Soon J.Hyun, Hsin – Hsing M.Chen (1998), *Temporal Association Algebra: A Mathematical Foundation for Processing Object – Oriented Temporal Databases*
4. [LR-96] Leonidas Fegaras, Ramez Elmasri (1996), *A Temporal Object Query Language*
5. [MV-03] Marios Vitos (2003), Rollnr: 73934, Master of Science in Computer Science, Dr. Meersmans, Course: Advanced Databases, *Object Oriented Databases*.
6. [O298] O2 Technology (1998), *ODMG C++ Binding Guide*, (Release 5.0)- April 1998

PHỤ LỤC LUẬN VĂN

DANH SÁCH BẢNG BIỂU

Bảng 1. 1 Bảng so sánh giữa lớp và tập	- 13 -
Bảng 2. 1 Định nghĩa quản lý thời gian trong một lớp thời gian	- 25 -
Bảng 3. 1 Các ký hiệu biểu diễn trong biểu thức đại số TA	- 43 -
Bảng 3. 2 Các tính chất của các toán tử đại số TA	- 53 -
Bảng 4. 1 Một số kiểu thời điểm	- 59 -
Bảng 4. 2 Một số kiểu khoảng	- 60 -
Bảng 4. 3 Một số kiểu giai đoạn	- 60 -
Bảng 4. 4 Một số kiểu tập giai đoạn	- 61 -
Bảng 4. 5 Danh sách một số hàm mới của TOQL	- 61 -
Bảng 4. 6 Các vị từ mới của TOQL	- 63 -
Bảng 4. 7 Thao tác trên khoảng	- 64 -
Bảng 4. 8 Thao tác trên tập giai đoạn	- 64 -
Bảng 4. 9 Thao tác trên thời điểm	- 64 -
Bảng 4. 10 Thao tác trên giai đoạn	- 64 -
Bảng 4. 11 Danh sách các biểu thức hỗ trợ của TOQL	- 67 -
Bảng 4. 12 Điều kiện của các kiểu kết nối thời gian	- 80 -

DANH SÁCH HÌNH VẼ

Hình 1. 1 Mô hình một đối tượng	- 11 -
Hình 2. 1 Kiến trúc của TOODBMS	- 18 -
Hình 2. 2 Kiến trúc tổng quan hệ nền TOOBIS	- 19 -
Hình 2. 3 Các khái niệm cơ bản của mô hình đối tượng	- 20 -
Hình 2. 4 Một lược đồ động	- 22 -
Hình 2. 5 Các lớp thời gian	- 25 -
Hình 2. 6 Mở rộng đến kiểu phân cấp của ODMG	- 29 -
Hình 2. 7 Các tính chất thể hiện thời gian và các đối tượng thời gian	- 30 -
Hình 2. 8 Các quan hệ và các quan hệ trạng thái	- 31 -
Hình 2. 9 Kiến trúc bộ xử lý TODL	- 32 -
Hình 3. 1 Đồ thị lược đồ của một CSDL công ty	- 35 -
Hình 3. 2 Thuộc tính nhãn thời gian trong tổ chức dữ liệu thời gian	- 37 -
Hình 3. 3 Thể hiện nhãn thời gian trong tổ chức dữ liệu thời gian	- 37 -
Hình 3. 4 Đồ thị đối tượng thời gian	- 38 -
Hình 3. 5 Đồ thị truy vấn của Q1	- 38 -
Hình 3. 6 Minh họa liên kết kết hợp thời gian	- 39 -

Hình 3. 7 Các mẫu kết hợp thời gian nguyên thủy	- 41 -
Hình 3. 8 Các mẫu kết hợp thời gian phức tạp	- 41 -
Hình 3. 9 Các TPI và một TPS.....	- 42 -
Hình 3. 10 Các toán tử T- Associate, T- Complement và T-Nonassociate	- 45 -
Hình 3. 11 Toán tử T- Join.....	- 48 -
Hình 3. 12 Toán tử T-OJoin.....	- 48 -
Hình 3. 13 Toán tử T- Select.....	- 49 -
Hình 3. 14 Toán tử T- Project	- 49 -
Hình 3. 15 Toán tử T- Union	- 50 -
Hình 3. 16 Toán tử T-Intersect.....	- 50 -
Hình 3. 17 Toán tử T- Difference	- 52 -
Hình 3. 18 Toán tử T-Divide(+).....	- 52 -
Hình 3. 19 Các toán tử NT-Intersect, NT-Union và NT-Difference	- 52 -
Hình 4. 1 Kiến trúc bộ xử lý TOQL.....	- 87 -
Hình 4. 2 Mô đun bộ xử lý TOQL	- 88 -
Hình 4. 3 Biểu diễn đối tượng Obj001.....	- 91 -
Hình 4. 4 Biểu diễn đối tượng Obj006.....	- 92 -

PHỤ LỤC A: DANH SÁCH CÁC TỪ - NGỮ VIẾT TẮT

STT	Viết tắt	Nghĩa từ - ngữ đầy đủ
1.	CSDL	Cơ sở dữ liệu (DataBase)
2.	DBMS	DataBase Management System
3.	ODMG	Object Definition Management Group
4.	OO	Object Oriented
5.	OOM	Object Oriented Method
6.	OQL	Object Query Language
7.	SQL	Structures Query Language
8.	TA	Temporal Algebra
9.	TODL	Temporal Object Definition Language
10.	TODM	Temporal Object Definition Method
11.	TOOA	Temporal Object Oriented Algebra
12.	TOOBIS	Temporal Object Oriented DataBases within Information System
13.	TOQL	Temporal Object Query Language

PHỤ LỤC B: DANH SÁCH CÁC TỪ KHÓA

Keyword indexs:

valid timestamp, transaction timestamp, object- oriented temporal database, temporal algebra, temporal object query, processing query.

Các chỉ mục từ khóa:

thời gian hiệu lực, thời gian giao dịch, cơ sở dữ liệu thời gian hướng đối tượng, đại số thời gian, truy vấn đối tượng thời gian, xử lý truy vấn đối tượng thời gian.

PHỤ LỤC C: TOQL ĐỐI VỚI CÁC YÊU CẦU NGƯỜI DÙNG

TOQL hỗ trợ các yêu cầu người dùng được liệt kê trong bảng sau: [EIV-99D]

Các yêu cầu của người dùng	Hỗ trợ	Chú thích
Chiều thời gian		
Thời gian hiệu lực	X	
Thời gian giao dịch	X	
Các kiểu đối tượng		
Đối tượng thông thường	X	
Đối tượng hiệu lực	X	
Đối tượng giao dịch	X	
Đối tượng Bitemporal	X	
Cấu trúc thời gian		
Điểm thời gian	X	Các thao tác cung cấp trên thời điểm
Giai đoạn	X	Các thao tác cung cấp trên giai đoạn
Tập các điểm thời gian	X	Chuẩn OQL các toán tử Túi/Tập
Tập các giai đoạn	X	Period_set
Đơn vị và Lịch biểu		
Lịch biểu	X	Gregorian. Các lịch tùy ý do người dùng định nghĩa
Các đơn vị	X	Năm, Tháng, Ngày, Giờ, Phút, Giây trong lịch Gregorian. Các đơn vị do người dùng định nghĩa
Truy vấn thời gian		
Các so sánh về ngày tháng	X	Toán tử trên thời điểm
So sánh và cấu trúc các Giai đoạn	X	Toán tử trên Giai đoạn và các hàm cấu trúc
Ánh xạ ngày tháng đến các Khoảng	X	Toán tử trên thời điểm, các chuyển đổi tương minh trên các Giai đoạn
Tính toán trên các Khoảng	X	Thông qua các toán tử và các hàm cấu trúc
Thao tác trên thời gian quan hệ	X	Các nhãn thời gian quan hệ
Chuyển đổi các đối tượng thông thường đến các đối tượng thời gian khác	X	Hiệu lực, giao dịch và bitemporal
Trích chọn thời gian hiệu lực hoặc giao dịch của một đối tượng	X	Các hàm hiệu lực và giao dịch
Cấu trúc quan hệ hiệu lực đến một đối tượng kết quả	X	Thông qua bổ nghĩa hiệu lực và bổ nghĩa <i>bitemporal</i>

Lựa chọn các giá trị tương ứng một giai đoạn	X	Các toán tử kịch bản con các tham chiếu các đối tượng khác
Lựa chọn các giai đoạn thỏa mãn một điều kiện	X	Tham chiếu đến biến thể đối tượng
Lựa chọn một đối tượng hiệu lực từ một đối tượng hiệu lực	X	Các toán tử kịch bản con
Nhóm thời gian	X	Các đối tượng thời gian được nhóm theo thời gian
Lựa chọn thứ tự thời gian	X	Sắp xếp theo thời điểm và giai đoạn
Các hàm kết hợp	X	Chuẩn kết hợp OQL và các thể thức thời gian kết hợp
Các truy vấn thời gian		
Chuyển đổi các đối tượng hiệu lực tới thông thường, giao dịch và đa chiều	X	Bỏ nghĩa thông thường, giao dịch và đa chiều
Chuyển đổi các đối tượng giao dịch tới thông thường, hiệu lực và đa chiều	X	Bỏ nghĩa thông thường, hiệu lực và đa chiều
Chuyển đổi các đối tượng đa chiều tới thông thường, hiệu lực và giao dịch	X	Bỏ nghĩa thông thường, hiệu lực và giao dịch

PHỤ LỤC D: BẢN TÓM TẮT TIẾNG VIỆT

Cơ sở dữ liệu là một phần không thể tách rời của các hệ thống thông tin, thiết kế và xây dựng cơ sở dữ liệu là một bài toán không mới nhưng cũng không hề cũ đối với các nhà thiết kế hệ thống. Cơ sở dữ liệu với yêu cầu quản lý về mặt thời gian trong các ứng dụng đang ngày càng phổ biến. Hơn thế nữa, các ứng dụng đa phương tiện, tích hợp các hệ thống thông minh, các ứng dụng văn phòng tích hợp đang ngày càng yêu cầu việc phân tích và xây dựng cơ sở dữ liệu trong các hệ thống cao hơn.

Luận văn với đề tài “Cơ sở dữ liệu hướng đối tượng thời gian và xử lý truy vấn trong cơ sở dữ liệu hướng đối tượng thời gian” trình bày các nội dung như sau:

Chương 1. Tổng quan: Trình bày các nội dung cơ bản về cơ sở dữ liệu đối tượng và cơ sở dữ liệu thời gian.

Cơ sở dữ liệu thời gian chính là các hệ thống ứng dụng đòi hỏi việc quản lý theo các khía cạnh thời gian khác nhau. Các khía cạnh thời gian được quan tâm nhiều nhất là thời gian hiệu lực (Valid Time) và thời gian giao dịch (Transaction Time). Có những ứng dụng yêu cầu quản lý dữ liệu theo cả hai chiều thời gian thì gọi là thời gian phức hợp (Bi_Temporal).

Cơ sở dữ liệu đối tượng thì xem và quản lý dữ liệu dưới khía cạnh các đối tượng. Một đối tượng bao gồm 4 thành phần: Một định danh đối tượng (Identifier - ID) xác định trong toàn bộ hệ thống. Giá trị của mỗi đối tượng là giá trị tại một thời điểm trong CSDL. Các tham chiếu hay các quan hệ mà đối tượng có thể có và Phương thức mà đối tượng có thể thực hiện.

Chương 2: Cơ sở dữ liệu hướng đối tượng thời gian trong các hệ thống thông tin. Trong chương này trình bày các nội dung về cơ sở dữ liệu hướng đối tượng thời gian bao gồm cơ sở lý thuyết và các kết quả cơ bản trong TOOBIS (Temporal Object Oriented DataBase with Information System). TOOBIS bao gồm một phương pháp luận (TOOM) dành riêng và một hệ quản trị cơ sở dữ

liệu hướng đối tượng thời gian (TOODBMS)

Chương 3: Đại số TA – Cơ sở khoa học cho xử lý dữ liệu hướng đối tượng thời gian. Đại số TA nhìn nhận dữ liệu đối tượng theo thời gian trên các lược đồ (Schema), việc xử lý truy vấn dựa trên các mẫu của các đồ thị (Graph) là cơ sở dữ liệu của hệ thống. Đại số TA không những có đầy đủ các tính chất của các đại số thông thường khác mà còn xây dựng các phép toán dành riêng cho CSDL đối tượng thời gian.

Chương 4: Một ngôn ngữ truy vấn trên cơ sở dữ liệu hướng đối tượng thời gian. Trong chương này trình bày về ngôn ngữ truy vấn TOQL (Temporal Object Query Language). TOQL được mở rộng từ OQL theo chuẩn của ODMG, là một ngôn ngữ truy vấn độc lập không những dành riêng cho cơ sở dữ liệu đối tượng thời gian mà còn hỗ trợ đầy đủ các tính năng truy vấn cho cơ sở dữ liệu đối tượng thông thường.

PHỤ LỤC E: BẢN TÓM TẮT TIẾNG ANH

In the last two decades, researchers have addressed the issue of *temporal data*, i.e. data varying over time. In the real world, entities evolve over time; thus their respective representation in information systems must change. Most computerised information systems (and DBMSs, in particular) nowadays retain only the *current state* of real-world entities, providing no support for keeping past values or storing predicted future values. Some information systems allow for retaining a log of past states for each entity, but these facilities mostly serve auditing and archiving purposes, providing only minimal support for information retrieval and analysis. An additional time-related requirement for information systems is to store the time that each piece of information was stored in the information repository and the time it was (logically) removed. Technically, the time that each piece of information is true in the real world is termed *valid time*, while the time from the insertion of a piece of information in a DBMS to its (logical) deletion is termed *transaction time*.

Temporal information systems are inherently more complex than *snapshot* information systems (i.e. information systems storing only the current state of the

modelled universe), since an extra dimension (the time dimension) is added. In this respect, along with the tools to support the storage and manipulation of time-varying information, a methodology is also needed to enable analysts and designers to capture, represent and analyse the semantics and requirements of temporal information systems.

This thesis describes an object-oriented temporal association algebra (called TA-algebra) which is intended to serve as a formal foundation for supporting a pattern-based query specification and processing paradigm. Different from the traditional tableand- attribute-based paradigm, the pattern-based paradigm views the intension of an object-oriented temporal database as a network of

object classes interconnected by different association types and its extension as a network of associated temporal object instances. Consistent with this view, queries can be specified in terms of patterns of temporal object associations or nonassociations (i.e., linear, tree and network structures of object classes/objects with logical AND and OR branches). TA-algebra provides a set of algebraic operators for processing these patterns and allows the direct and/or indirect associations and/or nonassociations among temporal object instances to be more explicitly represented and maintained during processing than the traditional tabular representation of temporary or final query results. TA-algebra operators are based on time-interval and valid-time semantics and they preserve the closure property. The algebra is capable of operating on heterogeneous as well as homogeneous patterns of object associations. Both homogeneous and heterogeneous patterns are decomposed into a set of primitive temporal pattern instances for uniform treatment. This paper formally defines the TA-algebra operators and their mathematical properties. The applications of these operators in query decomposition and processing are illustrated by examples.

The last, thesis is present of TOQL is a temporal extension to the OQL, the ODMG standard for querying object databases. The design objectives of TOQL were to formulate a temporal extension of OQL, delivering full temporal functionality, but yet remaining simple and treating uniformly temporal and non-temporal data. Another goal pursued in TOQL's design phase was the provision of the maximum degree of compatibility for legacy applications, i.e. applications that were written with a non-temporal schema in mind.

MỤC LỤC

LỜI CẢM ƠN	- 1 -
MỞ ĐẦU	- 2 -
CHƯƠNG I – TỔNG QUAN	- 4 -
Giới thiệu.....	- 4 -
1.1 Cơ sở dữ liệu thời gian.....	- 4 -
1.1.1 Khái niệm	- 4 -
1.1.2 Các vấn đề được nghiên cứu trong CSDL thời gian.....	- 5 -
1.1.3 Ngữ nghĩa của dữ liệu thời gian.....	- 5 -
1.1.4 Mô hình hóa thể hiện dữ liệu thời gian	- 6 -
1.1.5 Ngôn ngữ truy vấn trên các hệ CSDL thời gian	- 7 -
1.1.6 Thiết kế CSDL thời gian	- 8 -
1.1.7 Cài đặt các CSDL thời gian	- 9 -
1.2 Cơ sở dữ liệu hướng đối tượng	- 10 -
1.2.1 Đối tượng và cơ sở dữ liệu đối tượng	- 10 -
1.2.2 Các kiểu dữ liệu	- 12 -
1.2.3 Quản lý đối tượng	- 13 -
1.3 Kết luận chương I.....	- 14 -
CHƯƠNG II – CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG THỜI GIAN TRONG CÁC HỆ THỐNG THÔNG TIN.....	- 15 -
Giới thiệu.....	- 15 -
2.1 Tổng quan.....	- 15 -
2.2 Mục tiêu chung của TOOBIS.....	- 16 -
2.3 Kết quả thu được	- 17 -
2.4 Hệ nền TOOBIS.....	- 19 -
2.4.1 Phương pháp luận hướng đối tượng thời gian	- 19 -
2.4.1.1 Mục tiêu của phương pháp luận.....	- 19 -
2.4.1.2 Phương pháp hướng đối tượng.....	- 20 -
A - Thuộc tính cấu trúc	- 20 -
B - Thuộc tính hành vi	- 21 -
2.4.1.3 Mở rộng thời gian	- 22 -
A - Định nghĩa lịch biểu.....	- 22 -
B - Mở rộng miền cơ sở tới miền thời gian	- 23 -
C - Mở rộng lớp đối tượng tới các chiều thời gian	- 24 -
D - Mở rộng ràng buộc áp dụng lớp đối tượng tới chiều thời gian	- 26 -
E - Mở rộng thời gian của sự kiện	- 26 -
2.4.1.4 Kết luận về phương pháp luận TOOBIS.....	- 27 -
2.4.2 Hệ quản trị cơ sở dữ liệu thời gian hướng đối tượng.....	- 28 -
2.4.2.1 Mô hình dữ liệu đối tượng thời gian (TODM)	- 28 -

A - Mô hình và thao tác thời gian	- 28 -
B - Dữ liệu thời gian bên trong đối tượng.....	- 29 -
2.4.2.2 Ngôn ngữ định nghĩa đối tượng thời gian (TODL)	- 31 -
2.5 Kết luận chương 2	- 32 -
CHƯƠNG 3 – CƠ SỞ TOÁN HỌC CHO XỬ LÝ CƠ SỞ DỮ LIỆU	
HƯỚNG ĐỐI TƯỢNG THỜI GIAN	- 33 -
Giới thiệu.....	- 33 -
3.1 Mô hình hóa đối tượng thời gian và hình thức hóa truy vấn	- 34 -
3.1.1 Dữ liệu thời gian	- 34 -
3.1.2 Biểu diễn đồ thị của TOODB.....	- 35 -
3.1.2.1 Đồ thị lược đồ	- 35 -
3.1.2.2 Tổ chức dữ liệu thời gian và đồ thị đối tượng thời gian	- 36 -
3.1.2.3 Hình thức hoá truy vấn thời gian dựa mẫu	- 37 -
3.2 Đại số kết hợp thời gian	- 39 -
3.2.1 Các mẫu kết hợp thời gian nguyên thủy	- 39 -
3.2.2 Thể hiện mẫu thời gian và tập mẫu thời gian.....	- 40 -
3.2.3 So sánh khoảng thời gian	- 41 -
3.2.4 Ký hiệu	- 42 -
3.2.5 Toán tử	- 42 -
3.2.5.1 Hình thức hóa mẫu nguyên thủy	- 42 -
3.2.5.2 Thao tác mẫu	- 45 -
3.2.5.3 Thao tác mẫu không tính toán thời gian	- 51 -
3.3 Thiết kế ngôn ngữ thời gian và phân tích đại số học	- 53 -
3.3.1 Các hàm thời gian và các thao tác so sánh khoảng.....	- 53 -
3.3.2 Một số ví dụ phân tích đại số học	- 55 -
3.4 Kết luận chương 3	- 57 -
CHƯƠNG 4 – MỘT NGÔN NGỮ TRUY VẤN CƠ SỞ DỮ LIỆU HƯỚNG	
ĐỐI TƯỢNG THỜI GIAN	- 58 -
Giới thiệu.....	- 58 -
4.1 Đặc tả TOQL.....	- 58 -
4.1.1 Giới thiệu.....	- 58 -
4.1.2 Các kiểu dữ liệu cho trình diễn thời gian.....	- 58 -
4.1.2.1 Các thành tố.....	- 59 -
A - Thời điểm.....	- 59 -
B - Khoảng	- 59 -
C - Giai đoạn hay thời kỳ.....	- 60 -
D - Tập giai đoạn.....	- 60 -
4.1.2.2 Hàm	- 61 -
4.1.2.3 Vị từ.....	- 62 -
4.1.2.4 Toán tử	- 63 -
4.1.2.5 Nguyên tắc tự động chuyển đổi kiểu	- 64 -

A - Chuyển đổi kiểu	- 64 -
B - Chuyển đổi đơn vị	- 65 -
4.1.3 Cơ sở dữ liệu mẫu	- 65 -
4.1.4 Truy vấn trên dữ liệu thời gian	- 67 -
4.1.4.1 Các kiểu biểu thức hỗ trợ xử lý truy vấn của TOQL	- 67 -
4.1.4.2 Các ví dụ	- 68 -
4.1.4.3 Kiểu kết quả của các toán tử kịch bản con.....	- 69 -
4.1.4.4 Hỗ trợ các giá trị bị hủy bỏ và tiến triển	- 70 -
4.1.4.5 Trích chọn các trạng thái đối tượng	- 71 -
4.1.4.6 Giải quyết vấn đề ngữ nghĩa nhập nhằng.....	- 72 -
4.1.5 Tham chiếu đến các đối tượng thay đổi	- 73 -
4.1.6 Chuyển đổi giữa giá trị thời gian và thông thường	- 75 -
4.1.6.1 Bỏ nghĩa thông thường.....	- 75 -
4.1.5.2 Bỏ nghĩa hiệu lực	- 75 -
4.1.6.3 Bỏ nghĩa giao dịch	- 76 -
4.1.6.4 Bỏ nghĩa bitemporal	- 76 -
4.1.7 Kết nối thời gian.....	- 76 -
4.1.7.1 Kiểu kết quả của kết nối thời gian	- 78 -
4.1.7.2 Giá trị kết quả của một kết nối thời gian.....	- 79 -
4.1.7.3 Kết nối thời gian trên nhiều đối tượng	- 80 -
4.1.8 Toán tử cấu trúc lại	- 81 -
4.1.9 Tập hợp.....	- 83 -
4.1.9.1 Phân hoạch một đối tượng thời gian đơn.....	- 84 -
4.1.9.2 Kết hợp những biến thể từ nhiều đối tượng thời gian.....	- 86 -
4.2 Bộ xử lý TOQL	- 87 -
4.2.1 Giới thiệu.....	- 87 -
4.2.2 Kiến trúc bộ xử lý TOQL.....	- 87 -
4.2.3 Xử lý truy vấn	- 89 -
4.2.3.1 Biến đổi cú pháp	- 89 -
4.2.3.2 Thực thi TOQL ở trên OQL 5.1.1 tương thích OODBMS ...	- 89 -
4.3 Cài đặt chương trình.....	- 89 -
4.4 Kết luận chương 4	- 92 -
CHƯƠNG V: KẾT LUẬN.....	- 93 -
DANH SÁCH TÀI LIỆU THAM KHẢO	- 96 -
TÀI LIỆU TIẾNG VIỆT	- 96 -
TÀI LIỆU TIẾNG ANH	- 96 -
