+TỐI ƯU HÓA THỜI GIAN THI HÀNH TRUY VẤN TRONG HỆ THỐNG NHÀ KHO DỮ LIỆU VỚI HƯỚNG TIẾP CẬN *VIEW MATERIALIZATION*

Nguyễn Hữu Hòa[1], Nguyễn Nhị Gia Vinh[1],
Nguyễn Minh Trung[1] và Huỳnh Xuân Hiệp[2]

ABSTRACT

Data warehouse systems provide a unified and embraceable view of the company's precious information assets to underpin business management. Nevertheless, the size of data warehouse and the complexity of query are two main factors impacting to user's query performance. Since data warehouse tends to be very large and to grow overtime, retrieving the query results from data warehouse is able to be very time-consuming. One of the effective approaches to improve user's query performance is view materialization technique.

Keywords: data warehouse, query, view, materialization

Title: Optimizing query performance in data warehouse systems using view materialization approach

TÓM TẮT

Hệ thống nhà kho dữ liệu cung cấp một tầm nhìn bao quát và hợp nhất về tài sản thông tin quý giá của công ty để làm nền tảng trong việc quản lý kinh doanh. Tuy nhiên, kích cỡ của nhà kho dữ liệu và độ phức tạp của câu truy vấn là 2 yếu tố chính làm ảnh hưởng đến thời gian thi hành truy vấn của người sử dụng hệ thống nhà kho dữ liệu. Vì nhà kho dữ liệu thường có xu hướng lớn rất nhanh theo thời gian, cho nên việc lấy kết quả từ nhà kho dữ liệu có thể mất rất nhiều thời gian. Một trong những hướng tiếp cận hiệu quả để cải tiến thời gian thi hành truy vấn của người sử dụng hệ thống nhà kho dữ liệu là kỹ thuật "tư liệu hóa khung nhìn" (view materialization).

Từ khóa: Nhà kho dữ liệu, khung nhìn, tư liệu hóa, truy vấn

1 ĐẶT VẤN ĐỀ

Hiện nay, có nhiều kỹ thuật để tăng tốc độ truy vấn trên một cơ sở dữ liệu lớn, chẳng hạn như kỹ thuật *chỉ mục ánh xạ nhị phân* (bit-mapped index) hay *chỉ mục liên kết* (join index). Một trong những kỹ thuật thông dụng và đặc biệt hiệu quả trong hệ thống nhà kho dữ liệu là *tư liệu hóa khung nhìn* (view materialization).

Khung nhìn (view) là một bảng biểu ảo chứa kết quả của một câu truy vấn. Khung nhìn được tư liệu hóa (materialized view) là khung nhìn được tính toán trước và được lưu trữ trong cơ sở dữ liệu (CSDL) như là bảng giản lược (summary table). Thực tế cho thấy, nếu người sử dụng truy vấn dữ liệu trực tiếp từ *bảng sự kiện* (fact table) thì thời gian thi hành truy vấn là rất chậm, vì nó phải dựa trên một núi dữ liệu khổng lồ của *bảng sự kiện*. Do đó, hướng tiếp cận để giải quyết vấn đề này là tạo ra các khung nhìn dựa trên nhu cầu thường xuyên của người sử dụng để lưu trữ dưới dạng bảng giản lược. Khi đó, người sử dụng có thể truy vấn trên các bảng

[1] Bộ Môn Tin Học, Khoa Khoa Học, Đại Học Cần Thơ
[2] Khoa Công nghệ Thông Tin và Truyền Thông, Đại Học Cần Thơ

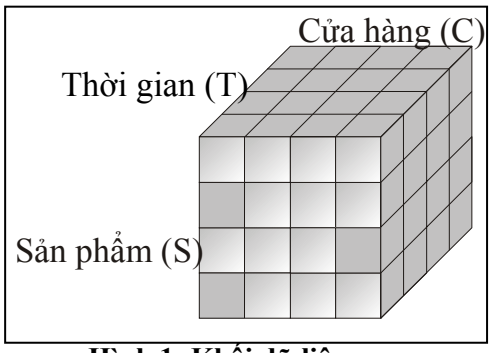
giản lược thay vì trên *bảng sự kiện*. Kỹ thuật này làm giảm thời gian thi hành truy vấn hàng trăm, ngàn lần so với việc truy vấn trực tiếp trên *bảng sự kiện*.

2 NỘI DUNG NGHIÊN CỨU

2.1 Khối dữ liệu

Các nhà quản lý kinh doanh thường có khuynh hướng suy nghĩ theo "*nhiều chiều*" (multidimensionally), chẳng hạn họ có khuynh hướng mô tả việc kinh doanh như "chúng tôi kinh doanh các *sản phẩm* ở nhiều *cửa hàng* khác nhau, và chúng tôi đánh giá hiệu quả kinh doanh theo từng giai đoạn *thời gian*". Suy nghĩ một cách trực giác, việc kinh doanh được xem như là một khối dữ liệu, với các nhãn trên mỗi cạnh của khối (xem Hình 1). Với mô tả kinh doanh ở trên, các cạnh của khối là *Sản phẩm*, *Cửa hàng*, và *Thời gian*. Hầu hết mọi người đều có thể nhanh chóng hiểu và tưởng tượng rằng các ô bên trong khối là các *độ đo hiệu quả kinh doanh* hay *độ đo lợi ích* (measure of interest) mà được kết hợp giữa các giá trị *Sản phẩm*, *Cửa hàng* và *Thời gian*.

Một khối dữ liệu (data cube) về cơ bản là có thể có nhiều chiều (multi-dimension). Những cạnh của khối được gọi là các chiều (dimension), đó là các mặt hoặc các thực thể ứng với những khía cạnh mà các nhà quản lý kinh doanh muốn ghi nhận. Mỗi chiều có thể kết hợp với một *bảng chiều* (dimension table) nhằm mô tả cho chiều đó. Ví dụ, bảng chiều *Sản phẩm* có thể có những thuộc tính như *Ma_SP*, *Ten_SP*, và *Loai_SP* mà có thể được chỉ ra bởi các nhà phân tích dữ liệu.



Hình 1: Khối dữ liệu

Hơn nữa, một khối dữ liệu trong kho dữ liệu phần lớn được xây dựng để đo *hiệu quả kinh doanh* của công ty. Do đó một mô hình dữ liệu đa chiều đặc thù được tổ chức xung quanh một chủ đề mà được thể hiện bởi một *bảng sự kiện* của nhiều độ đo số học (là các đối tượng của phân tích). Ví dụ, một *bảng sự kiện* có thể chứa *tổng số mặt hàng bán ra, tổng tiền bán hàng*,… vv. Mỗi độ đo số học phụ thuộc vào một tập các chiều cung cấp ngữ cảnh cho độ đo đó. Vì thế, các chiều kết hợp với nhau được xem như xác định duy nhất độ đo, là một giá trị trong không gian đa chiều. Ví dụ như một kết hợp của *Sản phẩm*, *Cửa hàng*, và *Thời gian* là một độ đo duy nhất so với các kết hợp khác. Vì vậy, nếu mỗi chiều chứa nhiều mức trừu tượng, thì dữ liệu có thể được xem từ nhiều khung nhìn linh động khác nhau.

Xét lược đồ cơ sở dữ liệu hình sao (star schema) có 3 *bảng chiều* là "Sản phẩm, Cửa hàng, và Thời gian" và một *bảng sự kiện* chứa độ đo lợi ích là "Tổng tiền bán hàng". Khi đó, ô có địa chỉ (S,C,T) sẽ lưu trữ tổng tiền bán hàng của sản phẩm S, ở cửa hàng C, trong thời gian T (xem Hình 1). Nếu ta thêm vào mỗi *bảng chiều* một giá trị "ALL", thì ô có địa chỉ (S,ALL,T) sẽ lưu trữ tổng tiền bán hàng của sản phẩm S, trong thời gian T. Hay nói cách khác, trong trường hợp này chúng ta muốn biết tổng tiền bán hàng của sản phẩm S, ở tất cả các cửa hàng, trong thời gian T. Ta gọi ô mà địa chỉ của nó có chứa "ALL" là ô phụ thuộc (dependent cell),

vì giá trị của ô này có thể được tính toán từ những ô khác trong khối dữ liệu. Một ô mà địa chỉ của nó không chứa "ALL" gọi là ô độc lập (independent cell) và khi đó giá trị của nó không thể tính toán được từ những ô khác. Thông thường, trong một khối dữ liệu có khoảng 70% ô phụ thuộc (A. Eteleeb, 2005).

Khối dữ liệu thường được lưu trữ ở 2 dạng: (1) hệ cơ sở dữ liệu quan hệ (relational database system), (2) hệ cơ sở dữ liệu đa chiều (multi-dimensional database system). Trong bài viết này, chúng tôi giả sử rằng khối dữ liệu được lưu trữ trong hệ cơ sở dữ liệu quan hệ và khi đó một ô trong khối dữ liệu được xem như là một khung nhìn được tư liệu hóa (materialize). Hay nói cách khác, khung nhìn *v* (bên dưới) được hiểu như là ô có địa chỉ (S,ALL,T) với độ đo lợi ích là Tong_Tien.

v: "Select SanPham_ID, ThoiGian_ID,Sum(Thanh_Tien) as Tong_Tien
From Fact_Table Group by SanPham_ID, ThoiGian_ID"

Vì khung nhìn của tập các ô khác nhau là chỉ khác nhau ở các thuộc tính của mệnh đề *Group-By*, do đó chúng ta sử dụng các thuộc tính ở mệnh đề *Group-By* để nhận dạng một khung nhìn duy nhất. Do đó, việc chọn tập các ô trong CSDL đa chiều để tư liệu hóa sẽ tương đương với việc chọn tập các khung nhìn trong CSDL quan hệ để tư liệu hóa. Trong các phần sau của bài viết này, chúng tôi sẽ sử dụng thuật ngữ "tập khung nhìn" (set of views) thay vì "tập ô trong CSDL đa chiều".

Việc chọn ra các khung nhìn để tư liệu hóa là một quyết định quan trọng trong quá trình thiết kế hệ thống nhà kho dữ liệu. Những quyết định này thường là phức tạp và khó, vì chúng ta phải cân nhắc các yếu tố khác nhau để đạt được sự kết hợp tốt nhất giữa thời gian thi hành truy vấn, chi phí lưu trữ và bảo quản. Có 3 hướng tiếp cận liên quan đến vấn đề này:

- Tư liệu hóa tất cả các khung nhìn. Hướng tiếp cận này là tối ưu về thời gian thi hành truy vấn, nhưng phải trả giá lớn về chi phí lưu trữ và bảo quản.
- Không tư liệu hóa khung nhìn nào cả. Hướng tiếp cận này là tối ưu về chi phí lưu trữ và bảo quản, nhưng phải trả giá lớn về thời gian thi hành truy vấn, vì tất cả các truy vấn của người sử dụng phải được tính toán từ *bảng sự kiện*.
- Chỉ tư liệu hóa các khung nhìn hữu ích (người sử dụng thường xuyên dùng). Chúng tôi cho rằng hướng tiếp cận này hữu hiệu hơn 2 hướng tiếp cận đầu, vì nó giảm chi phí lưu trữ và bảo quản các khung nhìn nhưng vẫn có thể đảm bảo thời gian thi hành truy vấn ở mức độ chấp nhận được (mặc dù không tối ưu). Do đó, trong bài viết chúng tôi chỉ quan tâm đến hướng này.

2.2 Minh họa

Để minh họa các vấn đề liên quan, chúng tôi sử dụng cơ sở dữ liệu FoodMart[7], với một *bảng sự kiện* là Sales_fact và bốn *bảng chiều* là Product, Store, Customer, và Time. Với bốn *bảng chiều*, thì sẽ có một tập gồm 2^4 khung nhìn khác nhau được liệt kê trong Bảng 1 và biểu đồ lưới của nó được thể hiện như Hình 2.

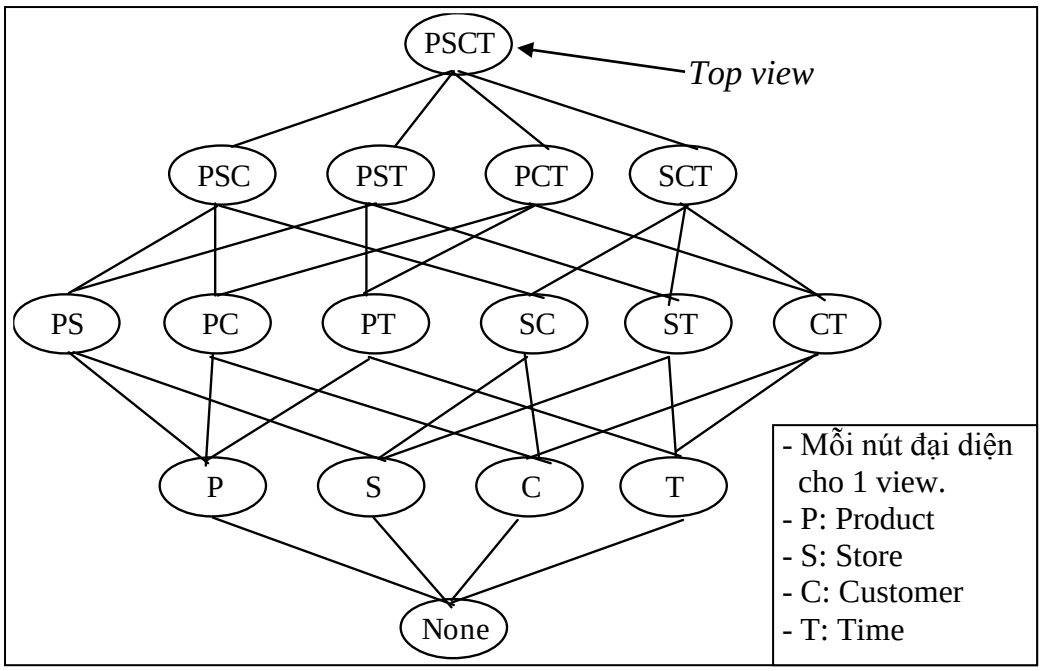
Nhận xét: khung nhìn ở cấp độ thấp hơn có thể được tính toán dựa vào khung nhìn ở cấp độ cao hơn (ví dụ khung nhìn PSC có thể được tính toán từ khung nhìn PSCT). Khung nhìn ở cấp độ cao nhất (PSCT), gọi là *top view*, được tính toán từ

[7] Tập đoàn FoodMart gồm nhiều cửa hàng (siêu thị) kinh doanh tạp hóa ở Mỹ, Mexico, và Cannada

bảng sự kiện. Giả sử để tính "tổng tiền của từng *sản phẩm* được bán ra theo từng *cửa hàng* và theo từng *khách hàng*", nếu sử dụng khung nhìn PSC thì phải duyệt qua 160.000 dòng. Trong khi đó nếu sử dụng khung nhìn PSCT thì cũng chỉ duyệt qua khoảng 160.000 dòng. Mặt khác, khung nhìn PSC có thể tính toán được từ khung nhìn PSCT. Do đó, nếu chúng ta chọn khung nhìn PSC để tư liệu hóa thì hoàn toàn không có lợi. Tương tự, nếu ta chọn khung nhìn PST để tư liệu hóa thì cũng hoàn toàn không có lợi. Từ đó có thể thấy rằng, việc chọn khung nhìn nào để tư liệu hóa là rất quan trọng.

Bảng 1: Tập khung nhìn với 4 *bảng chiều* Product, Customer, Store, Time

View	Các thuộc tính trên mệnh đề *Group-By*	Số dòng trả về của *view* (≅)	*View*	Các thuộc tính trên mệnh đề *Group-By*	Số dòng trả về của *view* (≅)
1	Product, Store, Customer, Time	160.000	9	Store, Customer	9.000
2	Product, Store, Customer	160.000	10	Store, Time	1.500
3	Product, Store, Time	160.000	11	Customer, Time	35.000
4	Product, Customer, Time	35.000	12	Product	1.500
5	Store, Customer, Time	35.000	13	Store	24
6	Product, Store	35.000	14	Customer	8.000
7	Product, Customer	160.000	15	Time	300
8	Product, Time	135.000	16	None	1



"None" có nghĩa là không có thuộc tính nào trên mệnh đề Group-By.

Hình 2: Biểu đồ lưới biểu diễn quan hệ phụ thuộc trên tập khung nhìn của 4 *bảng chiều*

2.3 Quan hệ phụ thuộc và biểu đồ lưới

2.3.1 Quan hệ phụ thuộc trên tập khung nhìn

Chúng ta có thể tổng quát hóa biểu đồ ở Hình 2 như sau:

Xét 2 khung nhìn v_1 và v_2, ta nói v_1 phụ thuộc v_2 (kí hiệu $v_1 \preceq v_2$) khi và chỉ khi v_1 có thể tính toán được từ v_2. Ví dụ như (Product, Customer) $\preceq$ (Product, Customer, Time). Một cách tương tự, ta nói v_1 không phụ thuộc v_2 (kí hiệu $v_1 \npreceq v_2$) khi và chỉ khi v_1 không thể tính toán được từ v_2. Hai khung nhìn v_1 và $v2$ được xem như là một (kí hiệu $v_1 \equiv v_2$) khi và chỉ khi $v_1 \preceq v_2$ và $v_2 \preceq v_1$.

Khi đó, quan hệ $\preceq$ là một quan hệ phụ thuộc có thứ tự từng phần. Kí hiệu biểu đồ lưới L trên quan hệ phụ thuộc $\preceq$ là $(L, \preceq)$. Khi đó, ta có các định nghĩa sau:

Tổ tiên của khung nhìn v trên $(L, \preceq)$: *Ancestor(v)* = $\{ v_i \mid v \preceq v_i \}$

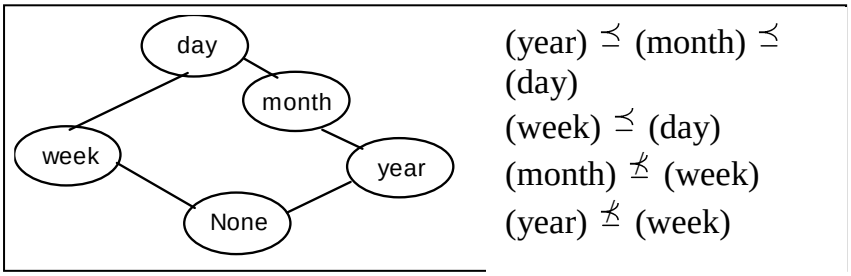
Con cháu của khung nhìn v trên $(L, \preceq)$: *Descendant*(v) = $\{ v_i \mid v_i \preceq v \}$

Cha của khung nhìn v trên $(L, \preceq)$: *Parent*(v) = $\{ v_i \mid v \prec v_i , \nexists v_j, v \prec v_j, v_j \prec v_i \}$

Biểu đồ lưới là một đồ thị vô hướng mà mỗi nút là 1 khung nhìn và có cạnh từ nút v_i đến nút v_j, với *Parent* (v_i)= v_j.

2.3.2 Quan hệ phụ thuộc trên bảng chiều

Trên thực tế, các *bảng chiều* (dimension table) bao gồm nhiều hơn một thuộc tính và được tổ chức theo thứ bậc (hierarchy). Do đó, trên mỗi *bảng chiều* cũng sẽ có các quan hệ phụ thuộc trên tập thuộc tính của nó. Hình 3 minh họa biểu đồ lưới của *bảng chiều* Time với 4 thuộc tính *day, week, month,* và *year*.



Hình 3: Biểu đồ lưới biểu diễn quan hệ phụ thuộc trên tập thuộc tính của *bảng chiều* Time

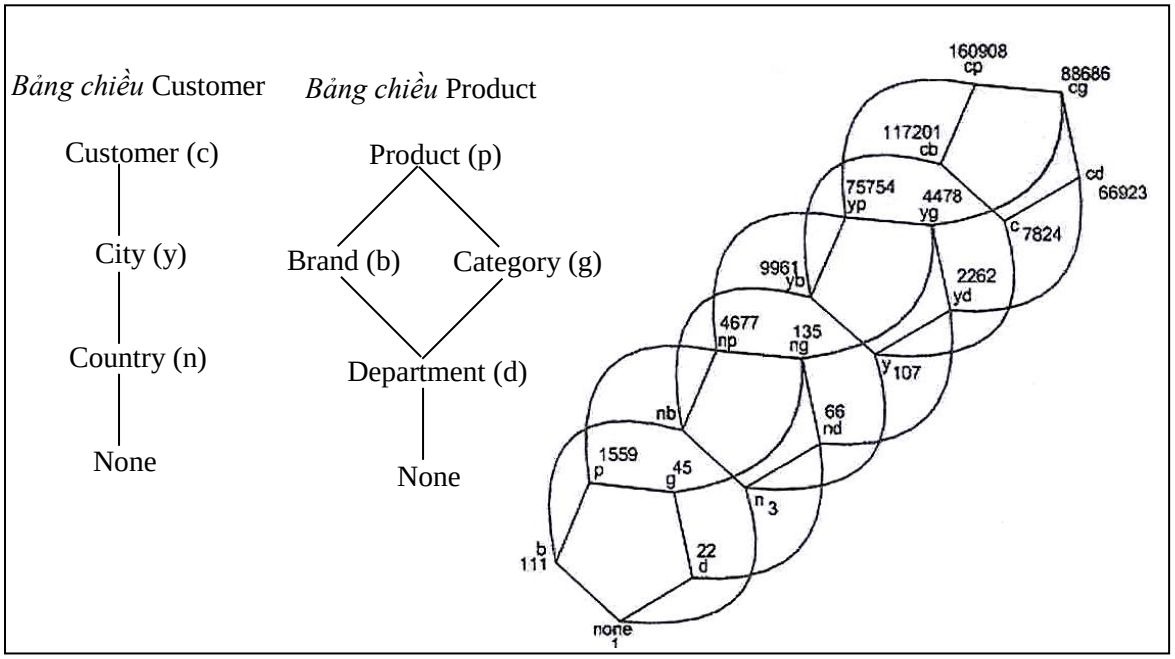
2.3.3 Biểu đồ lưới

Gọi a_i là một thuộc tính trên *bảng chiều* thứ i. Khi đó quan hệ phụ thuộc giữa các khung nhìn trên đồ thị lưới được định nghĩa như sau:

$(a_1, a_{2, \ldots}, a_n) \preceq (b_1, b_{2, \ldots}, b_n)$ với $a_i \preceq b_i$, i=1..n

Biểu đồ lưới (lattice diagram) của CSDL FoodMart với một *bảng sự kiện* Sales_fact và hai *bảng chiều* Product và Customer, được biểu diễn như Hình 4.

Dựa vào biểu đồ lưới ta có thể biết được những khung nhìn nào cần phải được tư liệu hóa trước và những khung nhìn nào nên tư liệu hóa sau. Nếu ta sử dụng các khung nhìn đã được tư liệu hóa trước đó để tạo ra các khung nhìn khác, thì thời gian để tìm tập khung nhìn sẽ giảm đáng kể. Biểu đồ lưới giúp chúng ta hình dung được cấu trúc thứ bậc (hierarchy) của việc truy vấn trên các *bảng chiều*. Trong các công cụ khai thác kho dữ liệu, thao tác *drill-down* (giảm mức độ trừu tượng) được hiểu là "trượt từ nút phía dưới lên nút bên trên trong đồ thị lưới", thao tác *roll-up* (tăng mức độ trừu tượng) được hiểu như "trượt từ nút bên trên xuống nút phía dưới trong đồ thị lưới".



Hình 4: Biểu đồ lưới với 1 *bảng sự kiện* Sales_fact và 2 *bảng chiều* Product và Customer

2.4 Thử nghiệm giải thuật Greedy

Trong phần này, chúng tôi tìm hiểu giải thuật Greedy được đề xuất bởi (V. Harinarayan, 1996). Sau đó, chúng tôi làm thử nghiệm giải thuật trên CSDL FoodMart với biểu đồ lưới Hình 4. Kết quả thử nghiệm được thể hiện trên Bảng 2 và Hình 5.

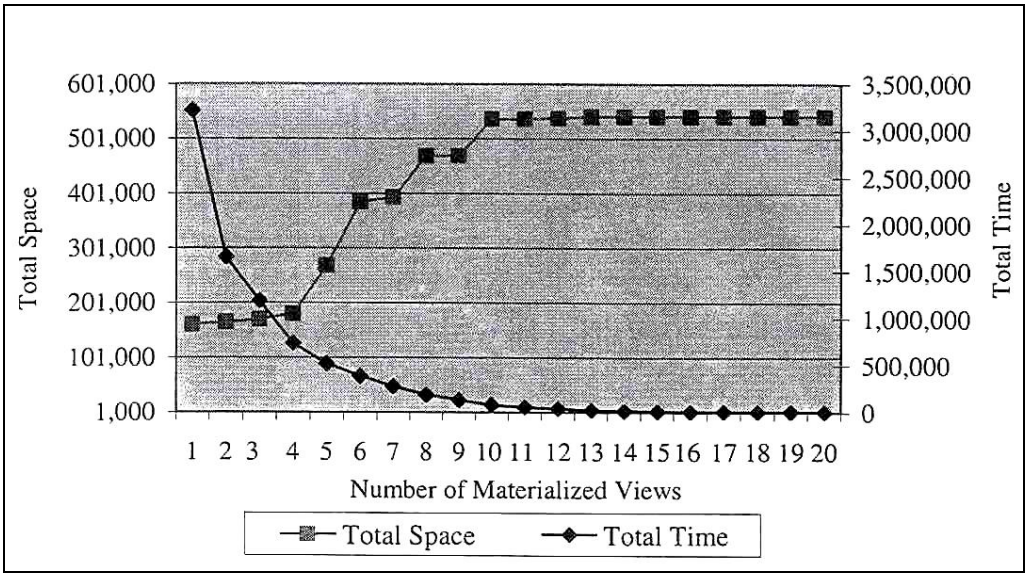
Bảng 2: Kết quả thử nghiệm của giải thuật Greedy trên CSDL FoodMart

No.	View	Total space	Benefit	Total time	No.	View	Total space	Benefit	Total time
1	cp	160.908	3.218.160	3.218.160	11	ng	536.880	27.252	56.103
2	np	165.585	1.562.310	1.655.850	12	p	538.439	15.590	40.513
3	yg	170.063	469.290	1.186.560	13	yd	540.701	15.398	25.115
4	yb	180.024	452.841	733.719	14	y	540.808	9.854	15.261
5	cg	268.710	216.666	517.053	15	g	540.853	4.542	10.719
6	cb	385.911	131.121	385.932	16	b	540.964	4.344	6.375
7	c	393.735	109.377	276.555	17	nd	541.030	534	5.841
8	yp	469.489	85.154	191.401	18	n	541.033	330	5.511
9	nb	469.822	57.768	133.633	19	d	541.055	178	5.333
10	cd	536.745	50.278	83.355	20	none	541.056	110	5.223

Giải thích Bảng 2:

- No. (cột 1) là thứ tự "trước – sau" theo quy trình lựa chọn khung nhìn.
- View (cột 2) là các khung nhìn khác nhau, được phân biệt bởi các thuộc tính trên mệnh đề *Group-By*. Các kí tự c, y, và n là tên tắt của các thuộc tính Customer (c), City (y), và Country (n) trong *bảng chiều* Customer. Các kí tự p, b, g, và d là tên tắt của các thuộc tính Product (p), Brand (b), Category (g), và Department (d) trong *bảng chiều* Product (xem Hình 4).
- Đơn vị tính của các yếu tố total space, benefit, và total time là số dòng của các khung nhìn.
- Total space (cột 3) là tổng không gian lưu trữ (số dòng) của một khung nhìn.

- Benefit (cột 4) là giá trị lợi ích của khung nhìn.
- Total time (cột 5) là tổng thời gian để duyệt qua số dòng của các khung nhìn. Ví dụ như để tìm được khung nhìn 3 (yg) thì mất một chi phí thời gian để duyệt qua 1.186.560 dòng.



Hình 5: Đồ thị biểu diễn kết quả của giải thuật Greedy trên CSDL FoodMart

Nhận xét: Đối với vài khung nhìn đầu (5 khung nhìn đầu), giải thuật chọn ra các khung nhìn với *tổng không gian lưu trữ* (total space) tối thiểu và *tổng thời gian* (total time) giảm đáng kể. Tuy nhiên, sau 5 khung nhìn đầu thì *tổng thời gian* không được cải thiện nhiều, trong khi đó *tổng không gian lưu trữ* tăng đáng kể. Từ đó có thể thấy rằng, ranh giới của việc khi nào nên dừng chọn khung nhìn. Hay nói cách khác, nếu chúng ta chọn tập S gồm 5 khung nhìn đầu để tư liệu hóa, thì chi phí sẽ là tối ưu. (A. Eteleeb, 2005) đã chỉ ra giải thuật Greedy mất nhiều thời gian để tìm tập khung nhìn, vì độ phức tạp lớn - $O(k.n^2)$, với k là tổng số khung nhìn được chọn để tư liệu hóa và n là số đỉnh trên biểu đồ lưới. Chúng tôi đề xuất một giải thuật khác, gọi là MVP (materialized views picker), với mục tiêu là giảm thời gian tìm tập khung nhìn.

2.5 Giải thuật MVP

Giải thuật mà chúng tôi đề xuất với hướng tiếp cận *heuristic* để tìm tập khung nhìn theo thứ tự tăng dần của *không gian lưu trữ* (space) của khung nhìn. Ở đây, *không gian lưu trữ* của khung nhìn được hiểu là kích cỡ hay số dòng của khung nhìn. Giải thuật được mô tả như sau:

S = {top view}; W = {tất cả các khung nhìn trên biểu đồ lưới};
While space > 0 Do
Begin
    Chọn khung nhìn v từ tập W, sao cho space của v là nhỏ nhất;
    If (space > |v|) [and benefit > B_v] Then
        Begin
            Space = space - |v|;
            S = S ∪{v}; W = W – {v};
        End ;
End;

- S là tập khung nhìn cần tìm;
- B_v là giá trị *lợi ích* (benefit) của khung nhìn *v*, được tính bởi công thức:
- $B_v = (N_{topview} - N_v)*(N_{children} + 1)/N_v$
- $N_{topview}$ là số dòng của khung nhìn ở cấp độ cao nhất (top view).
- N_v là số dòng của khung nhìn *v*.
- $N_{children}$ là số dòng của các khung nhìn con của *v* trên biểu đồ lưới.
- Space là hằng số ràng buộc để giới hạn tổng số dòng của tất cả khung nhìn.
- Benefit là hằng số ràng buộc để giới hạn giá trị lợi ích của từng khung nhìn.
- Điều kiện [and benefit > B_v] là tùy chọn.

Nhận xét:

Giải thuật MVP bắt đầu tìm khung nhìn có kích cỡ nhỏ nhất và tăng dần cho đến khi đạt tới giới hạn nhất định. Cách tính giá trị *lợi ích* của giải thuật MVP là đơn giản hơn so với Greedy (V. Harinarayan, 1996). MVP có độ phức tạp O(n log n), trong khi Greedy có độ phức tạp là $O(k.n^2)$; với k là số lượng khung nhìn được chọn để tư liệu hóa và n là số đỉnh trên biểu đồ lưới. Giải thuật MVP vẫn có thể đảm bảo hầu hết các yếu tố như đã được đề cập trong giải thuật Greedy. Tuy nhiên, Greedy hoạt động theo nguyên lý "gần như vét cạn theo chiều sâu (A. Eteleeb, 2005)", trong khi MVP mang tính chất *heuristics*, vì thế kết quả của Greedy gần với kết quả tối ưu hơn MVP.

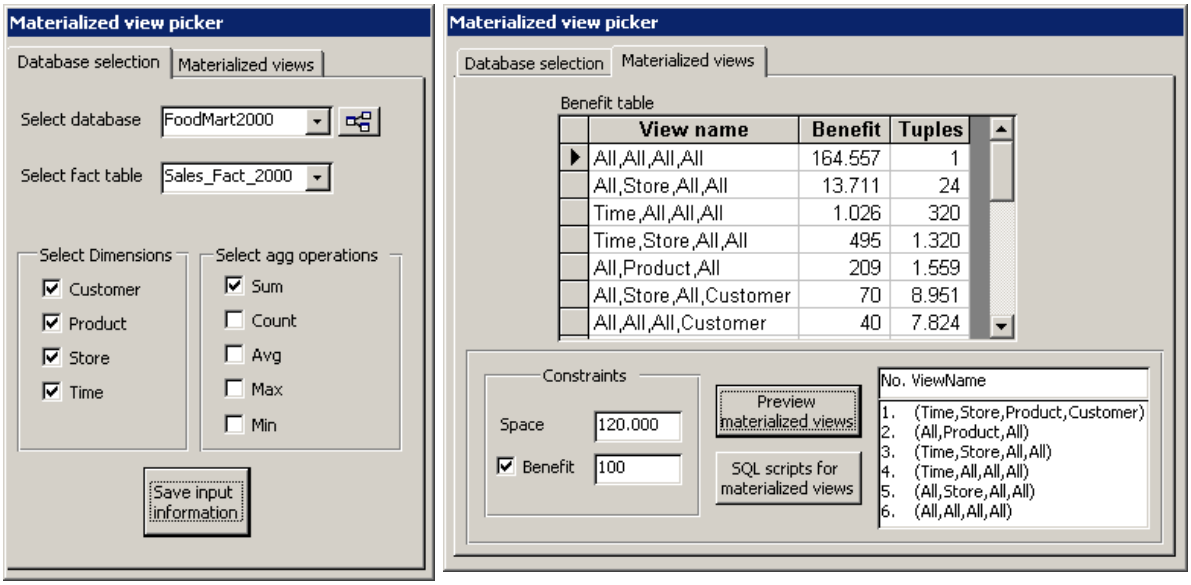
Đối với những kho dữ liệu lớn thì MVP sẽ thích hợp hơn so với Greedy, vì thời gian tìm tập khung nhìn nhanh hơn nhiều so với Greedy. Chúng tôi đã làm thử nghiệm trên CSDL FoodMart với 4 *bảng chiều* và 1 *bảng sự kiện* thì thời gian tìm tập khung nhìn là 12 giây.

3 KẾT QUẢ THỬ NGHIỆM

Để thử nghiệm giải thuật MVP, chúng tôi đã xây dựng công cụ MVP (Hình 6). Chức năng chính của công cụ MVP là tìm tập khung nhìn thỏa ràng buộc về *không gian lưu trữ* (space) và ràng buộc về giá trị *lợi ích* (benefit) từ một lược đồ CSDL hình sao (star schema), rồi sau đó tạo ra SQL script để tư liệu hóa các khung nhìn.

3.1 Mô tả công cụ MVP

Trong sự thử nghiệm này, trên mỗi bảng chiều chúng tôi chỉ quan tâm đến một thuộc tính nào đó của nó. Ví dụ như khung nhìn với các thuộc tính trên mệnh đề Group-By (Time, Store, Product, Customer) có thể hiểu như (Month, Store_Name, Product_Name, Customer_Name). Hay nói cách khác, công cụ MVP chưa xem xét đến sự phân cấp của nhiều thuộc tính trên các bảng chiều. Một hạn chế khác nữa là công cụ MVP chỉ xử lý trên một bảng sự kiện. Hình 6 mô tả dữ liệu vào (input) và dữ liệu ra (output) của công cụ MVP.



Hình 6: Dữ liệu vào và dữ liệu ra của công cụ MVP

Dữ liệu vào bao gồm:

- *Select database*: chọn một CSDL từ một hệ quản trị CSDL (RDBMS)
- *Select fact table*: chọn một *bảng sự kiện* từ CSDL.
- *Select dimensions*: chọn các *bảng chiều* liên quan đến *bảng sự kiện*.
- *Select operations*: chọn phép toán để tính *độ đo* (measure) trên *bảng sự kiện*.
- *Contraints*: chọn sự giới hạn về 2 yếu tố *không gian lưu trữ* và *lợi ích* của các khung nhìn. Người quản trị CSDL (DBA) sẽ cân nhắc các yếu tố khác nhau để chọn sự kết hợp tốt nhất giữa 2 yếu tố này.
- *Space*: là hằng số ràng buộc *không gian lưu trữ* để giới hạn tổng số dòng của tất cả khung nhìn được chọn.
- *Benefit*: là hằng số ràng buộc *lợi ích* (tùy chọn) để giới hạn giá trị *lợi ích* của từng khung nhìn.

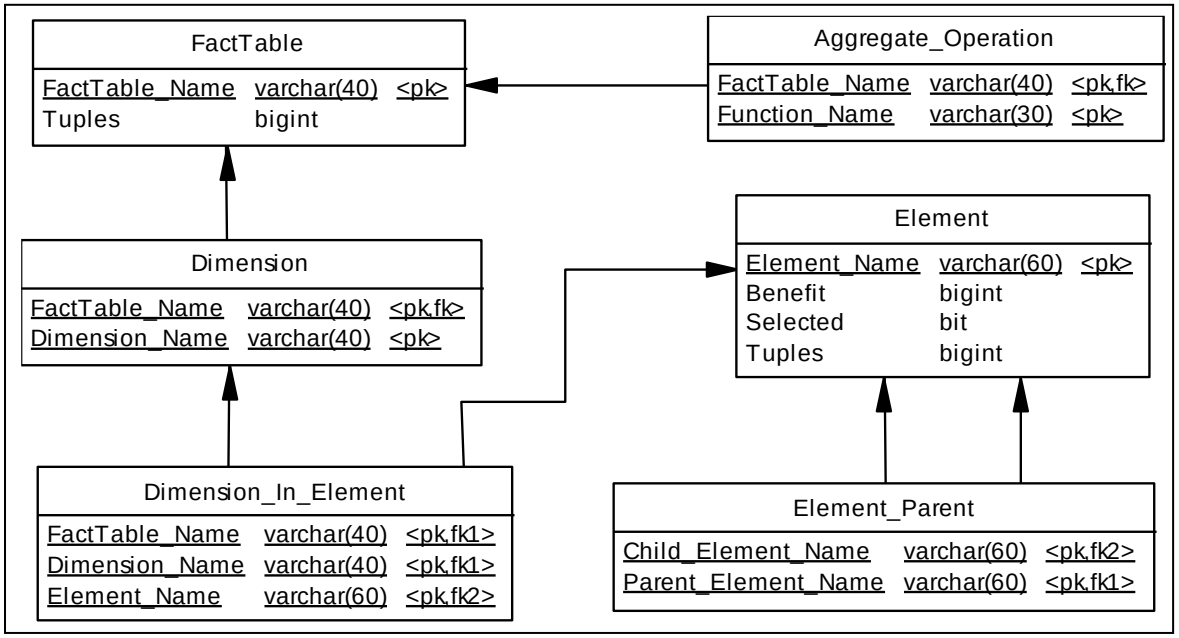
Dữ liệu ra bao gồm:

- *Benefit table*: chứa thông tin của tập khung nhìn, bao gồm :
  (i) Cột View name chứa tên các thuộc tính của các bảng chiều trên mệnh đề Group-By của khung nhìn.
  (ii) Cột Benefit là giá trị lợi ích trong giải thuật MVP.
  (iii) Cột Tuples là không gian lưu trữ (số dòng) của từng khung nhìn.
- Nút lệnh *Previews materialied views* là để xem tập khung nhìn thỏa ràng buộc *không gian lưu trữ* và ràng buộc *lợi ích*. Có 6 khung nhìn thỏa cả hai ràng buộc này (xem Hình 6).
- Nút lệnh *SQL script for materialed views* để tạo ra các câu lệnh SQL để tư liệu hóa các khung nhìn.

3.2 Mô hình cơ sở dữ liệu của công cụ MVP

Công cụ MVP có một kho chứa dữ liệu (repository) riêng biệt. Siêu dữ liệu (metadata) trong kho chứa dữ liệu sẽ mô tả các thông tin về bảng sự kiện, bảng chiều, phép toán tập hợp, khung nhìn (view hay element), và các dữ kiện thống kê như số dòng của bảng sự

kiện, số dòng của khung nhìn, lợi ích của khung nhìn, …vv. Hình 7 mô tả mô hình CSDL của công cụ MVP.



Hình 7: Mô hình cơ sở dữ liệu của công cụ MVP

4 KẾT LUẬN

Hệ thống nhà kho dữ liệu đã trở thành một trong những trọng tâm chính trong "công nghiệp cơ sở dữ liệu". Nhiều nghiên cứu trong lĩnh vực này nhằm vào các vấn đề then chốt như đa dạng hóa thông tin, cải tiến tốc độ xử lý thông tin, …vv. Trong bài viết này, chúng tôi đã trình bày khái quát kỹ thuật *tư liệu hóa khung nhìn* (view materialization). Thêm vào đó chúng tôi đã đề xuất giải thuật MVP để cải tiến thời gian tìm tập khung nhìn. Sau cùng, chúng tôi đã đưa ra kết quả thử nghiệm bằng cách thiết kế và xây dựng công cụ MVP để tìm tập khung nhìn hữu ích nhằm làm tăng tốc độ xử lý hệ thống nhà kho dữ liệu. Trong tương lai, chúng tôi sẽ tiếp tục cải tiến giải thuật và công cụ MVP để giải quyết tiếp các vấn đề liên quan đến nhiều *bảng sự kiện* (fact table) và sự phân cấp của nhiều thuộc tính trên *bảng chiều* (dimension table).

TÀI LIỆU THAM KHẢO

A. Eteleeb, G.D. Bakema, Aspects of Aggregates in Data Warehouses and Multidimensional Databases, HAN University, The Netherlands, June 2005

A. Tsois, T. Sell, The Generalized Pre-Grouping Transformation: Aggregate-Query Optimization in the Presence of Dependencies, Proceedings. The 29th VLDB Conference, Berlin, 2003

P. Ponniah, Data Warehousing Fundamentals, A Comprehensive Guide for IT professionals, John Wiley & Sons, Inc. 2001

R. Kimball, Aggregate Navigation With (Almost) No Metadata, August 1996

R. Kimball, M. Ross, The Data Warehouse Toolkit, 2nd edition, John Wiley & Sons, Inc. 2001

V. Harinarayan, A. Rajaraman, J.D. Ullman, Implementing Data Cubes Efficiently, Proceedings. ACM SIGMOD International Conference On Management of Data, 205-227, 1996

Y. Kotidis, Aggregate View Management in Data Warehouse, Handbook of massive data sets, Kluwer Academic Publishers, Norwell, MA, USA, 2002