

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

LUẬN VĂN THẠC SĨ KHOA HỌC

**PHƯƠNG PHÁP XỬ LÝ PHÂN TÍCH
TRỰC TUYẾN ÁP DỤNG TRONG XÂY DỰNG
HỆ TRỢ GIÚP QUYẾT ĐỊNH DỰA VÀO DỮ LIỆU**

CHUYÊN NGÀNH: XỬ LÝ THÔNG TIN VÀ TRUYỀN THÔNG

TRẦN ĐÌNH CHIẾN

NGƯỜI HƯỚNG DẪN KHOA HỌC: GS.TS. NGUYỄN THÚC HẢI

HÀ NỘI 2006

MỤC LỤC

Danh mục hình vẽ	5
Danh sách các thuật ngữ và từ viết tắt	6
Lời mở đầu	7
Chương I. Khai thác dữ liệu và xử lý phân tích trực tuyến	10
1.1. Giới thiệu các phương pháp khai thác dữ liệu	10
1.2. Xử lý phân tích trực tuyến (OLAP)	11
1.3. Nguyên tắc của OLAP	12
1.3.1. Khung nhìn đa chiều	12
1.3.2. Tính trong suốt (Transparency)	12
1.3.3. Khả năng truy nhập được	13
1.3.4. Thực hiện việc tạo báo cáo đồng nhất	13
1.3.5. Kiến trúc khách/chủ (Client/Server)	13
1.3.6. Cấu trúc chung cho các chiều (Generic Dimensionality)	13
1.3.7. Làm việc với ma trận	14
1.3.8. Hỗ trợ nhiều người sử dụng	14
1.3.9. Phép toán giữa các chiều không hạn chế	14
1.3.10. Thao tác tập trung vào dữ liệu	14
1.3.11. Tạo báo cáo linh hoạt	15
1.3.12. Không hạn chế số chiều và các mức kết hợp dữ liệu	15
Chương II. Kho dữ liệu (Data Warehouse)	16
2.1. Các thành phần kho dữ liệu	16
2.1.1. Siêu dữ liệu (Metadata)	17
2.1.2. Các nguồn dữ liệu	17
2.1.3. Hệ thống xử lý giao dịch trực tuyến (OLTP)	18
2.1.3.1. Những đặc điểm của hệ thống OLTP	19
2.1.3.2. Các công cụ thu thập, làm sạch và chuyển đổi dữ liệu nguồn	20
2.1.4. Cơ sở dữ liệu của kho dữ liệu	22
2.1.5. Kho dữ liệu	23
2.1.5.1. Định nghĩa	23
2.1.5.2. Đặc điểm dữ liệu trong kho dữ liệu	24
2.1.6. Kho dữ liệu chủ đề (Datamart)	25
2.2. Sử dụng kho dữ liệu	26
2.3. Phương pháp xây dựng kho dữ liệu	28
2.4. Thiết kế CSDL cho kho dữ liệu	29
2.4.1. Giản đồ hình sao (Star)	29
2.4.2. Giản đồ hình tuyết rơi (Snowflake)	32
2.4.3. Giản đồ kết hợp	33
2.4.4. Những vấn đề liên quan tới thiết kế giản đồ hình sao	34
2.4.4.1. Đánh chỉ số	34
2.4.4.2. Chỉ thị về mức	35
2.4.5. Những nhân tố thiết kế cần phải được cân nhắc	35
2.5. Quản trị kho dữ liệu	37

Chương III. Tiếp cận và phân tích đa chiều trong xử lý phân tích trực tuyến	39
3.1. Tiếp cận đa chiều	39
3.2. Phân tích đa chiều	40
3.3. Kiến trúc khối của OLAP (OLAP Cube Architecture)	42
3.3.1. Giới thiệu kiến trúc khối	42
3.3.2. Khối (Cube).....	43
3.3.2.1. Xác định khối	44
3.3.2.2. Xử lý các khối.....	45
3.3.2.3. Khối ảo (Virtual Cube)	46
3.3.3 Chiều (Dimension)	46
3.3.3.1. Xác định các chiều	48
3.3.3.2. Chiều có phân cấp.....	48
3.3.3.3. Phân cấp chiều	49
3.3.3.4. Roll_up và Drill_down dựa trên phân cấp chiều	50
3.3.3.5. Các chiều ảo (Virtual Dimensions).....	50
3.3.4. Các đơn vị đo lường (Measures).....	51
3.3.5. Các phân hoạch (Partitions).....	51
3.3.6. Các phương pháp lưu trữ dữ liệu (MOLAP, ROLAP, HOLAP)	53
3.3.6.1. MOLAP (Multidimensional OLAP).....	53
3.3.6.2. ROLAP (Relational OLAP).....	54
3.3.6.3. HOLAP (Hybrid OLAP).....	55
3.4. Thuật toán chỉ số hoá các khung nhìn trong xử lý phân tích trực tuyến kho dữ liệu	55
3.4.1. Một số khái niệm cơ bản	56
3.4.1.1. Các khối dữ liệu con (Subcubes)	56
3.4.1.2. Câu truy vấn (Queries).....	56
3.4.1.3. Chỉ số (Indexes).....	57
3.4.1.4. Quan hệ tính toán và phụ thuộc	58
3.4.2. Thuật toán chọn View và Index.....	61
3.4.2.1. Ước tính kích thước của mỗi View.....	61
3.4.2.2. Ước tính kích thước của chỉ số Index	61
3.4.2.3. Xác định bài toán	62
3.4.2.4. Giải quyết bài toán.....	63
3.4.3. Kết luận	66
Chương IV. Hệ trợ giúp quyết định dựa vào dữ liệu.....	67
4.1. Hệ trợ giúp quyết định	67
4.1.1. Giới thiệu	67
4.1.2. Hệ trợ giúp quyết định	68
4.1.3. Phân loại các hệ trợ giúp quyết định	69
4.2. Hệ trợ giúp quyết định dựa vào dữ liệu.....	71
4.2.1. Tiếp cận kho dữ liệu và OLAP.....	71
4.2.2. Trợ giúp quyết định dựa vào dữ liệu trên cơ sở kho dữ liệu và OLAP	73
4.2.3. Tiến trình trợ giúp quyết định dựa vào dữ liệu cho bài toán cụ thể	75
4.3. Xây dựng cấu trúc thông tin hỗ trợ việc ra quyết định	77

4.3.1. Vai trò của cấu trúc thông tin	77
4.3.2. Các yếu tố ảnh hưởng	78
4.3.2.1. Các yêu cầu thông tin.....	78
4.3.2.2. Mức độ tích hợp.....	80
4.3.3. Mô hình tổ chức thông tin	81
4.3.3.1. Các yêu cầu thông tin và năng lực của hệ thống thông tin	81
4.3.3.2. Mức độ tích hợp hệ thống	83
4.3.4. Kết luận	84
4.4. Dịch vụ trợ giúp quyết định của Microsoft.....	85
4.4.1. Kho dữ liệu Microsoft	85
4.4.1.1. Microsoft Data Warehousing Framework	86
4.4.1.2. Sự phức tạp của dữ liệu	87
4.4.1.3. Lợi ích đối với việc kinh doanh	88
4.4.1.4. Mô hình dữ liệu.....	88
4.4.1.5. Các hình thức lưu trữ	89
4.4.2. Kiến trúc dịch vụ trợ giúp ra quyết định của Microsoft.....	90
4.4.3. Các vấn đề trong việc triển khai Microsoft DSS.....	91
4.4.3.1. Xây dựng mô hình dữ liệu OLAP cho Microsoft DSS.....	91
4.4.3.2. Lưu trữ mềm dẻo	93
4.4.3.3. Chuyển thông tin tới người sử dụng	97
4.4.3.4. Khả năng của các công cụ OLAP	100
4.5. Hướng nghiên cứu phát triển: Hệ trợ giúp quyết định phân tán	102
Chương V. Xây dựng hệ thống trợ giúp quyết định dựa vào dữ liệu	
bằng công cụ Analysis Services.....	106
5.1. Mục tiêu của hệ thống.....	106
5.2. Yêu cầu về hệ thống.....	106
5.3. Chức năng chính của hệ thống.....	107
5.3.1. Chức năng tạo lập CSDL đa chiều	109
5.3.2. Chức năng phân tích và hiển thị dữ liệu	109
5.4. Giới thiệu hệ thống	110
5.4.1. Khởi động Analysis Manager.....	110
5.4.2. Cài đặt cơ sở dữ liệu và nguồn dữ liệu (Database & Data Source).....	110
5.4.3. Tạo khối.....	111
5.4.4. Lưu trữ và xử lý khối.....	114
5.4.5. Khối ảo tăng cường khả năng xử lý và bảo mật	117
5.4.6. Tạo khối ảo.....	118
5.4.7. Hiển thị dữ liệu khối.....	120
5.4.8. Ví dụ minh họa	121
Phần kết luận	122
Tài liệu tham khảo	124
Tóm tắt luận văn	125

Danh mục hình vẽ

- Hình 1.1. Kho dữ liệu và OLAP
- Hình 2.1. Mô hình kho dữ liệu
- Hình 2.2. Giảm đồ hình sao và hình tuyết rơi
- Hình 3.1. Mô hình dữ liệu đa chiều
- Hình 3.2. Mô hình dữ liệu khối
- Hình 3.3. Giảm đồ khối hình sao
- Hình 3.4. Giảm đồ khối hình tuyết rơi
- Hình 3.5. Sơ đồ mô hình đa khối
- Hình 3.6. Phân cấp chiều Sản_phẩm
- Hình 3.7. Cây phân cấp đối xứng
- Hình 3.8. Roll_up và Drill_down theo phân cấp chiều
- Hình 4.1. Phân loại các Hệ thống tin quản lý
- Hình 4.2. Kho dữ liệu và hệ thống OLAP
- Hình 4.3. Tiến trình trợ giúp quyết định dựa vào dữ liệu cho bài toán cụ thể
- Hình 4.4. Ma trận Yêu cầu/Năng lực
- Hình 5.1. Kiến trúc hệ trợ giúp quyết định dựa vào dữ liệu
- Hình 5.2. Chức năng hệ trợ giúp quyết định dựa vào dữ liệu
- Hình 5.3. Tạo DataSource cho các khối trong Database
- Hình 5.4. Chọn bảng Fact
- Hình 5.5. Chọn đơn vị đo
- Hình 5.6. Tạo chiều
- Hình 5.7. Chọn các mức của chiều
- Hình 5.8. Chọn kiểu lưu trữ
- Hình 5.9. Tăng tốc độ thực hiện
- Hình 5.10. Xử lý khối
- Hình 5.11. Chọn các khối cho khối ảo
- Hình 5.12. Chọn đơn vị đo cho khối ảo
- Hình 5.13. Chọn chiều cho khối ảo
- Hình 5.14. Hiển thị dữ liệu khối

Danh sách các thuật ngữ và từ viết tắt

CSDL		Cơ sở dữ liệu
DBA	DataBase Administrator	Quản trị cơ sở dữ liệu
DM	DataMart	Kho dữ liệu chủ đề
DSS	Decision Support System	Hệ trợ giúp quyết định
HOLAP	Hybrid OLAP	OLAP ghép
ETL	Extract Transformation Load	Trích xuất, chuyển và nạp dữ liệu
LS	Legacy System	Hệ thống đã có sẵn
MIS	Management Information System	Hệ thống tin quản lý
MOLAP	Multidimensional OLAP	OLAP đa chiều
MSS	Management Support System	Hệ hỗ trợ quản lý
OLAP	On-Line Analysis Processing	Xử lý phân tích trực tuyến
OLTP	On-Line Transaction Processing	Xử lý giao dịch trực tuyến
RDBMS	Relational DataBase Management System	Hệ quản trị CSDL quan hệ
ROLAP	Relational OLAP	OLAP quan hệ
SA	Subject Area	Vùng chủ đề

Lời mở đầu

Các hoạt động sản xuất, kinh doanh hiện nay luôn cần có sự đáp ứng nhanh nhạy, tức thời đối với các thay đổi liên tục, vì vậy các nhà quản lý buộc phải thường xuyên ra cùng lúc nhiều quyết định đúng đắn (mà chúng sẽ ảnh hưởng đáng kể đến xu hướng hoạt động và sự cạnh tranh của doanh nghiệp) một cách nhanh chóng. Do đó vấn đề trợ giúp quyết định trở nên rất cần thiết. Người ta cần phải thu thập, tổng hợp và phân tích dữ liệu từ nhiều nguồn khác nhau một cách nhanh và hiệu quả thì mới có thể ra được những quyết định nhanh chóng và phù hợp. Điều này dẫn đến việc cần phát triển những hệ thống tinh thông biết cách làm thế nào để trích chọn và phân tích dữ liệu cho người sử dụng.

Hiện nay có rất nhiều phần mềm cung cấp cho người sử dụng những khả năng truy vấn và lập các báo cáo thông tin, đặc biệt là các hệ quản trị CSDL quan hệ. Tuy nhiên CSDL quan hệ với cấu trúc hai chiều (dòng và cột) không được thiết kế để cung cấp các quan điểm đa chiều trên dữ liệu đầu vào của các phân tích phức tạp. Sử dụng các hệ thống này, chúng ta sẽ gặp rất nhiều khó khăn và bất tiện trong việc tổ chức dữ liệu đa chiều vào các bảng hai chiều, không thể triển khai dữ liệu phân tích với số lượng lớn, công cụ phân tích để tạo ra các dữ liệu quyết định không mạnh, thuận tiện, linh hoạt, nhanh chóng và nhất là không dễ dàng để sử dụng đối với các nhà quản lý, những người ra quyết định.

Như vậy, việc xây dựng một hệ thống mới có khả năng tổ chức dữ liệu đa chiều và có khả năng phân tích dữ liệu linh hoạt để trả lời được các truy vấn đa chiều một cách dễ dàng, nhanh chóng nhằm hỗ trợ cho việc ra quyết định của các nhà quản lý là cần thiết.

Mục đích của đề tài:

Luận văn đề cập đến việc nghiên cứu xây dựng một hệ trợ giúp quyết

định dựa vào dữ liệu, sử dụng phương pháp luận xử lý phân tích trực tuyến (OLAP). Đề tài sẽ tập trung vào hai công việc chính là nghiên cứu vấn đề tổ chức cơ sở dữ liệu đa chiều, phân tích và hiển thị dữ liệu để trợ giúp ra quyết định.

Hệ trợ giúp quyết định theo cách tiếp cận này có thể giúp các nhà quản lý thiết lập một mô hình OLAP cho ứng dụng cụ thể của mình trong việc tổ chức cơ sở dữ liệu đa chiều và dễ dàng điều chỉnh hoạt động phân tích, tìm kiếm thông tin theo những khía cạnh khác nhau của dữ liệu nhằm thu thập được tối đa dữ liệu cần thiết để từ đó đưa được những quyết định tốt nhất một cách nhanh chóng.

Không giống với các hệ trợ giúp quyết định truyền thống thường được xây dựng với mục đích đưa ra giải pháp tối ưu cho một bài toán cụ thể, trong một phạm vi ứng dụng hẹp, hệ trợ giúp quyết định dựa vào dữ liệu hướng đến việc giúp người sử dụng có thể khai thác được tối đa khả năng tiềm ẩn của một khối lượng dữ liệu lớn, nhằm thu được những thông tin tổng hợp ở đủ các khía cạnh khác nhau của dữ liệu, để từ đó có thể ra các quyết định đúng một cách nhanh chóng. Do đặc điểm này, phạm vi ứng dụng của hệ trợ giúp quyết định dựa vào dữ liệu là rộng. Nó có thể được sử dụng để trợ giúp quyết định cho các bài toán khác nhau, trong những lĩnh vực khác nhau.

Bố cục của luận văn:

Toàn bộ luận văn được trình bày trong 5 chương:

- Chương 1: Giới thiệu các phương pháp khai thác dữ liệu, các nội dung cơ bản về xử lý phân tích trực tuyến.
- Chương 2: Trình bày các lý thuyết chung về kho dữ liệu và mô hình kho dữ liệu, phương pháp xây dựng và thiết kế CSDL cho kho dữ liệu.
- Chương 3: Trình bày phương pháp tiếp cận và phân tích đa chiều trong xử lý phân tích trực tuyến.

- Chương 4: Giới thiệu Hệ trợ giúp quyết định dựa vào dữ liệu với hai thành phần chính là kho dữ liệu và xử lý phân tích trực tuyến. Tiến trình trợ giúp quyết định dựa vào dữ liệu. Xây dựng cấu trúc thông tin để hỗ trợ việc ra quyết định và giới thiệu về dịch vụ trợ giúp quyết định của Microsoft. Hướng nghiên cứu phát triển.
- Chương 5: Xây dựng hệ thống với chức năng tạo lập cơ sở dữ liệu đa chiều và phân tích hiển thị dữ liệu.

Chương I. Khai thác dữ liệu và xử lý phân tích trực tuyến

1.1. Giới thiệu các phương pháp khai thác dữ liệu

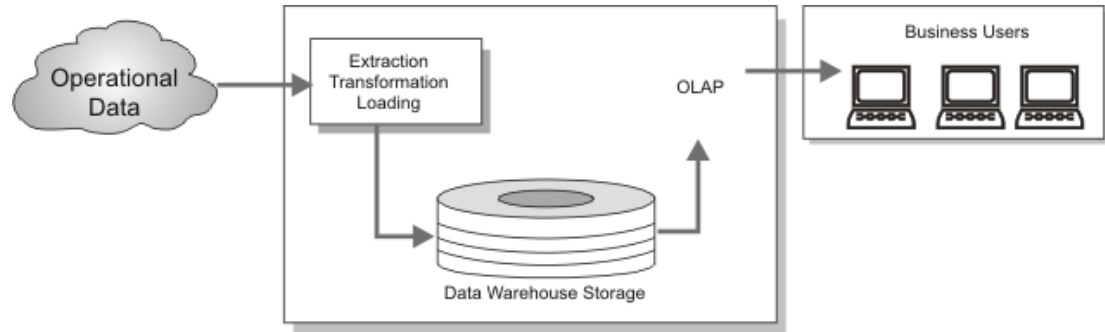
Khai thác dữ liệu là quá trình phát hiện ra những mối quan hệ liên thuộc, các mô hình và các khuynh hướng mới (Patterns & Trends) bằng việc khảo sát một số lượng lớn dữ liệu được lưu trữ trong các kho (Repository) sử dụng các công nghệ về nhận dạng mẫu cũng như các kỹ thuật thống kê và toán học. Khai thác dữ liệu có thể hiểu là kỹ thuật khoan dữ liệu theo chiều sâu và tổng hợp dữ liệu theo chiều ngược lại, là quá trình đào xới xem xét dữ liệu dưới nhiều góc độ nhằm tìm ra các mối liên hệ giữa các thành phần dữ liệu và phát hiện ra những xu hướng, hình mẫu, kinh nghiệm quá khứ tiềm ẩn trong kho dữ liệu. Vì vậy nó rất phù hợp với mục đích phân tích dữ liệu hỗ trợ điều hành và ra quyết định.

Phần lớn các phương pháp khai thác dữ liệu đều dựa trên các lĩnh vực như học máy, thống kê và các công cụ khác. Một số kỹ thuật thường dùng là mạng Nơ-ron (Neuron Network), giải thuật di truyền (Genetic Algorithms) và xử lý phân tích trực tuyến (OLAP).

Xử lý phân tích trực tuyến chính là việc sử dụng kho dữ liệu cho mục đích trợ giúp quyết định. Ý tưởng mô phỏng các chiều trong dữ liệu có thể được mở rộng: một bảng với n thuộc tính có thể được xem như một không gian n chiều. Người quản lý thường đặt những câu hỏi mà có thể phân tích trong những phân tích đa chiều. Các thông tin này không phải dễ phân tích khi bảng được biểu diễn hai chiều và CSDL quan hệ chuẩn không thể đáp ứng tốt công việc này. Trong trường hợp như vậy, sử dụng OLAP tỏ ra thích hợp.

Cũng có một sự khác nhau giữa các công cụ OLAP và khai thác dữ liệu đó là công cụ OLAP không thể học, chúng không tạo nên tri thức mới và không tìm kiếm được giải pháp mới. Như vậy có sự khác nhau cơ bản giữa tri thức đa chiều và kiểu tri thức mà một người có thể lấy ra được từ một CSDL

thông qua khai thác dữ liệu.



Hình 1.1. Kho dữ liệu và OLAP

1.2. Xử lý phân tích trực tuyến (OLAP)

OLAP là một chức năng thông minh trong xử lý nghiệp vụ, làm cho các thông tin có thể hiểu được dễ dàng. OLAP khiến cho người sử dụng đầu cuối (End-User) có thể hiểu được bản chất bên trong thông qua việc truy nhập nhanh, tương tác tới các khung nhìn nhiều dạng của thông tin được chuyển đổi từ các dữ liệu thô để phản ánh sự đa dạng nhiều chiều.

OLAP là một công nghệ phân tích dữ liệu thực hiện những công việc sau:

- Đưa ra một khung nhìn Logic, nhiều chiều của dữ liệu trong kho dữ liệu. Khung nhìn này hoàn toàn không phụ thuộc vào việc dữ liệu được lưu trữ như thế nào (có thể được lưu trữ trong một kho dữ liệu nhiều chiều hay một kho dữ liệu quan hệ).
- Thường liên quan tới những truy vấn phân tích tương tác dữ liệu. Sự tương tác thường là phức tạp, liên quan tới việc khoan sâu xuống những mức dữ liệu chi tiết hơn hoặc cuộn lên mức dữ liệu cao hơn ở mức tổng hợp hoặc kết hợp.
- Cung cấp khả năng thiết lập mô hình phân tích bao gồm tính toán tỉ lệ,

những biến đổi... liên quan tới những đại lượng số hoặc dữ liệu là con số qua nhiều chiều.

- Tạo ra sự tổng hợp và kết hợp, phân cấp và dùng những mức tổng hợp, kết hợp đó cho mỗi phép giao của các bảng theo chiều.
- Hỗ trợ những mô hình chức năng cho việc dự báo, phân tích các xu hướng và phân tích thống kê.
- Lấy và hiển thị dữ liệu theo những bảng 2 chiều hay 3 chiều, theo biểu đồ hay đồ thị, dễ dàng xoay đổi các trục cho nhau. Khả năng xoay là quan trọng vì người sử dụng cần phân tích dữ liệu từ những cách nhìn khác nhau và sự phân tích theo mỗi cách nhìn sẽ dẫn đến một câu hỏi khác, câu hỏi này sẽ được kiểm tra tính đúng đắn dựa trên một cách nhìn khác về dữ liệu đó.
- Đáp ứng những câu trả lời nhanh vì vậy quá trình phân tích không bị cản ngang và thông tin không bị cũ.
- Sử dụng một kho dữ liệu đa chiều, lưu trữ dữ liệu theo các mảng (lưu ý là mảng lưu trữ những phần tử cùng kiểu khác với bản ghi là các phần tử khác kiểu nhau). Những mảng này là sự biểu diễn Logic của các chiều của công việc.

1.3. Nguyên tắc của OLAP

1.3.1. Khung nhìn đa chiều

Đối với người thực hiện thì cách nhìn của họ với công việc là nhiều chiều về bản chất. Vì vậy mô hình OLAP phải là đa chiều về bản chất. Những người sử dụng có thể thao tác dễ dàng trên những mô hình dữ liệu đa chiều như vậy.

1.3.2. Tính trong suốt (Transparency)

Công cụ phân tích cần phải trong suốt với người sử dụng. OLAP nên

tồn tại trong một kiến trúc hệ thống mở, cho phép các công cụ phân tích có thể được nhúng vào bất kỳ nơi nào mà người sử dụng mong muốn mà không có một sự tác động ngược lại nào với các chức năng của công cụ trên máy chủ.

1.3.3. Khả năng truy nhập được

Công cụ OLAP phải ánh xạ được giản đồ Logic của chính nó tới kho dữ liệu vật lý hỗn tạp, truy nhập tới dữ liệu và thực hiện mọi chuyển đổi cần thiết để đưa ra một khung nhìn đơn giản, mạch lạc và đồng nhất cho người sử dụng. Dữ liệu vật lý của hệ thống thuộc kiểu này trở nên trong suốt với người sử dụng và chỉ là mối quan tâm của công cụ.

1.3.4. Thực hiện việc tạo báo cáo đồng nhất

Khi số lượng các chiều tăng thì năng suất báo tạo báo cáo giảm đi.

1.3.5. Kiến trúc khách/chủ (Client/Server)

Thành phần Server của các công cụ OLAP cần phải đủ thông minh đến mức mà nhiều Client có thể được truy nhập tới một cách dễ dàng và có thể lập trình tích hợp. Server thông minh phải có đủ khả năng để ánh xạ và xây dựng dữ liệu từ những cơ sở dữ liệu vật lý và Logic khác hẳn nhau. Điều đó rất cần thiết để đảm bảo tính trong suốt và xây dựng một lược đồ mức khái niệm, Logic, vật lý chung.

1.3.6. Cấu trúc chung cho các chiều (Generic Dimensionality)

Mỗi chiều của dữ liệu phải cân bằng giữa cấu trúc và khả năng thực hiện của nó. Thường chỉ tồn tại một cấu trúc chung cho tất cả các chiều. Mọi chức năng được áp dụng cho một chiều cũng có thể áp dụng cho các chiều khác.

1.3.7. Làm việc với ma trận

Cấu trúc vật lý của OLAP Server cần phải biến đổi cho phù hợp với mô hình phân tích cụ thể được tạo ra và tải vào để việc quản lý các ma trận là tối ưu nhất. Khi làm việc với các ma trận, OLAP Server phải có khả năng suy luận và tìm ra cách lưu trữ dữ liệu hiệu quả nhất. Các phương pháp truy nhập vật lý cũng được thay đổi thường xuyên và cung cấp những cơ chế khác nhau như tính toán trực tiếp, cây nhị phân, kỹ thuật băm hoặc sự kết hợp tốt nhất những kỹ thuật như vậy.

1.3.8. Hỗ trợ nhiều người sử dụng

Những công cụ của OLAP phải cung cấp truy nhập đồng thời (lấy dữ liệu ra và cập nhật), tính toàn vẹn và an toàn để hỗ trợ cho những người sử dụng làm việc đồng thời với cùng một mô hình phân tích hoặc tạo ra những mô hình khác nhau từ cùng một dữ liệu.

1.3.9. Phép toán giữa các chiều không hạn chế

Trong phân tích dữ liệu đa chiều, tất cả các chiều được tạo ra và có vai trò như nhau. Các công cụ OLAP quản lý những tính toán liên quan tới các chiều và không yêu cầu người sử dụng phải định nghĩa những phép toán đó. Việc tính toán đòi hỏi phải định nghĩa các công thức tùy thuộc vào một ngôn ngữ, ngôn ngữ này phải cho phép tính và thao tác với một số lượng chiều bất kỳ mà không bị hạn chế bởi mối quan hệ giữa các phần tử, không liên quan tới số thuộc tính chung của dữ liệu của mỗi phần tử.

1.3.10. Thao tác tập trung vào dữ liệu

Những thao tác như định hướng lại đường dẫn xây dựng dữ liệu hoặc khoan sâu xuống theo các chiều hoặc các hàng được thực hiện bằng hành động trực tiếp trên những phần tử của mô hình phân tích mà không đòi hỏi phải sử dụng những Menu hay ngắt cho giao diện với người sử dụng. Những

chiều được định nghĩa trong mô hình phân tích chứa tất cả thông tin mà người sử dụng cần để thực hiện những hành động cố hữu.

1.3.11. Tạo báo cáo linh hoạt

Với việc sử dụng OLAP Server và các công cụ của nó, một người sử dụng đầu cuối có thể thao tác, phân tích, đồng bộ hoá và xem xét dữ liệu theo bất kỳ cách nào mà người đó mong muốn, bao gồm cả việc tạo ra những nhóm Logic hoặc bố trí những hàng, cột, phần tử cạnh những phần tử khác. Những phương tiện tạo báo cáo cũng phải cung cấp tính linh hoạt và đưa ra những thông tin đã được đồng bộ theo bất kỳ cách nào mà người sử dụng muốn hiển thị chúng.

1.3.12. Không hạn chế số chiều và các mức kết hợp dữ liệu

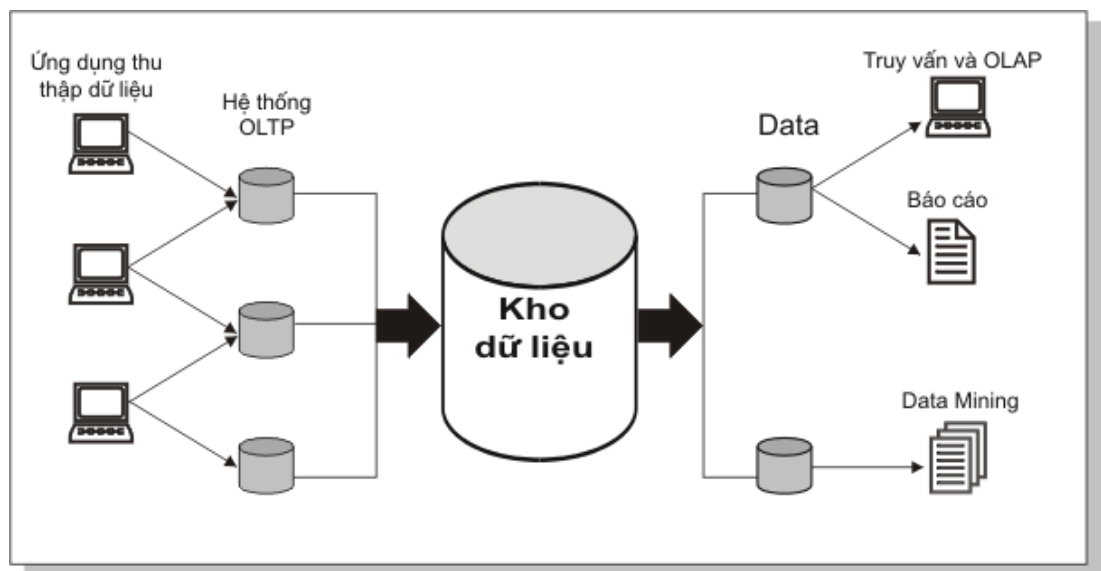
Một OLAP Server có thể chứa được ít nhất là 15 chiều trong một mô hình phân tích thông thường nhất. Mỗi chiều cho phép một số lượng không giới hạn các mức tổng hợp và kết hợp dữ liệu do người sử dụng định nghĩa và đưa ra cách xây dựng các mức đó.

Chương II. Kho dữ liệu (Data Warehouse)

Hiện nay hầu hết các tổ chức đều đang phải đương đầu với sự thay đổi của thị trường. Người ta thấy rằng để có thể đưa ra một quyết định đúng đắn, trước hết phải có khả năng truy nhập tới tất cả các loại thông tin nhanh chóng. Đối với một tổ chức nào đó, để có thể có quyết định đúng đắn, cần nghiên cứu cả những dữ liệu quá khứ, phân tích nhằm định ra toàn bộ các xu hướng có thể. Trong bối cảnh công nghệ thông tin phát triển, dữ liệu được tập trung trong những cơ sở dữ liệu khổng lồ, nhu cầu truy cập vào tất cả các thông tin là cần thiết. Cách có hiệu quả nhất để trợ giúp nhu cầu truy nhập thông tin là tổ chức kho dữ liệu (Data Warehouse).

2.1. Các thành phần kho dữ liệu

Các thành phần cấu thành kho dữ liệu cung cấp một khung cơ bản để trao đổi về kiến trúc, cấu trúc và các chiến lược của kho dữ liệu.



Hình 2.1. Mô hình kho dữ liệu

2.1.1. Siêu dữ liệu (Metadata)

Trong việc tổ chức kho dữ liệu, không chỉ những người dùng đầu cuối mà ngay cả những nhân viên quản trị đều cần truy nhập toàn bộ thông tin trong bảng gồm các đối tượng cũng như các thuộc tính. Do đó họ muốn biết một số vấn đề:

- Có thể tìm thấy dữ liệu ở đâu?
- Tồn tại những loại thông tin, dữ liệu nào?
- Dữ liệu thuộc loại nào, có dạng ra sao?
- Trong các cơ sở dữ liệu khác nhau thì dữ liệu có liên quan với nhau như thế nào?
- Dữ liệu được lấy từ đâu và nó thuộc ai quản lý?

Vì vậy hình thành một dạng cơ sở dữ liệu khác được gọi là Metadata nhằm mô tả cấu trúc nội dung của cơ sở dữ liệu chính. Trong môi trường cơ sở dữ liệu phức hợp, một Metadata phù hợp là không thể thiếu bởi nó định ra cấu trúc cơ sở dữ liệu tác nghiệp và cả cấu trúc kho dữ liệu. Một vấn đề xuất hiện thường xuyên là khả năng giao tiếp với người sử dụng về những thông tin bên trong kho dữ liệu và cách thức chúng được truy nhập. Chính Metadata là cách để người sử dụng và các ứng dụng có thể tiếp cận được với những thông tin được lưu trữ trong kho dữ liệu. Nó có thể định nghĩa tất cả các phần tử dữ liệu và các thuộc tính của chúng.

Metadata cần được thu thập khi kho dữ liệu được thiết kế và xây dựng. Metadata phải có sẵn cho tất cả những người sử dụng kho dữ liệu để hướng dẫn họ dùng kho dữ liệu. Ngoài ra các công cụ trợ giúp cũng được thiết lập và cần được đánh giá.

2.1.2. Các nguồn dữ liệu

Bao gồm các hệ thống trong và ngoài của một tổ chức, rất phong phú

về chủng loại. Các hệ thống nằm trong được coi như các hệ thống nguồn hoặc các hệ thống đã có sẵn.

- Hệ thống đã có sẵn (Legacy System - LS): là một hệ thống tác nghiệp. Hệ thống này đã từng được phát triển, sử dụng các công nghệ có sẵn và vẫn phù hợp với các nhu cầu. Các hệ thống này có thể được thực hiện trong nhiều năm và có lẽ không có hoặc có rất ít minh chứng bằng tài liệu.
- Dữ liệu ngoài: là dữ liệu không nằm trong các hệ thống tác nghiệp của một tổ chức, là những dữ liệu do người sử dụng đầu cuối yêu cầu.

Các LS được phát triển để phục vụ cho các dự án. Các ứng dụng được phát triển cùng với dữ liệu mà các dữ liệu này lại đáp ứng nhiều nhu cầu khác nhau. Cùng là một dữ liệu nhưng lại có tên khác nhau hoặc thuộc các hệ thống đo lường khác nhau. Kết quả cuối cùng là các nguồn dữ liệu cần được đánh giá và các định nghĩa cần được đưa vào Metadata để nhắm tới các vấn đề sau:

- Xác định các nguồn khác nhau, các cấu trúc file khác nhau, các nền (Platform) khác nhau.
- Hiểu được dữ liệu nào có trong các hệ thống nguồn đang tồn tại, các định nghĩa của dữ liệu và bất kỳ các luật nào cho dữ liệu.
- Phát hiện sự giao nhau về thông tin của các hệ thống khác nhau.
- Quyết định dữ liệu tốt nhất trong các hệ thống. Mỗi hệ thống cần được đánh giá để quyết định hệ thống nào có dữ liệu rõ ràng và chính xác hơn.

2.1.3. Hệ thống xử lý giao dịch trực tuyến (OLTP)

Dữ liệu phát sinh từ các hoạt động hàng ngày được thu thập, xử lý để phục vụ công việc cụ thể của một tổ chức thường được gọi là dữ liệu tác nghiệp và hoạt động thu thập xử lý loại dữ liệu này được gọi là xử lý giao

dịch trực tuyến (OLTP).

Dữ liệu tại các CSDL tác nghiệp được lấy từ nhiều nguồn khác nhau nên dễ bị nhiễu, hỗn tạp dẫn đến dữ liệu không sạch, không toàn vẹn. Do đó việc kiểm tra dữ liệu, làm sạch dữ liệu phải được tiến hành ngay tại đây nhằm bảo đảm tính toàn vẹn, tính đúng đắn của dữ liệu để phục vụ cho việc xây dựng kho dữ liệu và trợ giúp ra quyết định sau này.

2.1.3.1. Những đặc điểm của hệ thống OLTP

- Trợ giúp số lượng lớn người sử dụng đồng thời trong việc thêm mới, sửa đổi dữ liệu.
- Diễn tả trạng thái thay đổi bắt buộc của tổ chức nhưng không lưu lại lịch sử của nó.
- Chứa đựng số lượng lớn các dữ liệu, bao gồm dữ liệu tổng quát để kiểm soát thực hiện.
- Được điều chỉnh để đáp ứng nhanh việc thực hiện.
- Cung cấp cơ sở hạ tầng công nghệ để hỗ trợ các thao tác thường ngày của một tổ chức.

Chính từ những đặc điểm này, nếu chúng ta sử dụng OLTP cho phân tích trực tuyến thì thường gặp những khó khăn sau:

- Các yêu cầu phân tích, tổng hợp những khối lượng lớn dữ liệu ảnh hưởng tới khả năng của hệ thống.
- Sự thực hiện của hệ thống khi đáp ứng những yêu cầu phân tích phức tạp có thể chậm hoặc không ổn định, cung cấp sự hỗ trợ không đầy đủ cho người sử dụng trong phân tích trực tuyến.
- Sự thay đổi dữ liệu thường xuyên gây trở ngại cho tính tin cậy của thông tin phân tích.
- An ninh trở nên phức tạp hơn khi phân tích trực tuyến được kết hợp với

xử lý giao dịch trực tuyến.

Kho dữ liệu với nhiệm vụ tổ chức dữ liệu cho mục đích phân tích đã giải quyết được các khó khăn trên bằng việc cung cấp những khóa chính, các kho dữ liệu có thể:

- Kết hợp dữ liệu từ những nguồn dữ liệu hỗn tạp vào trong một cấu trúc đơn thuần nhất.
- Tổ chức dữ liệu trong những cấu trúc đơn giản đáp ứng hiệu quả của các yêu cầu có tính phân tích hơn là cho việc xử lý giao dịch.
- Chứa dữ liệu thay đổi, hợp lệ, chắc chắn và hợp lý hoá trong phân tích.
- Cung cấp dữ liệu ổn định.
- Được cập nhật định kỳ dữ liệu bổ sung hơn là những giao dịch thường xuyên.
- Cung cấp một cơ sở dữ liệu được tổ chức phù hợp cho OLAP hơn là cho OLTP.

2.1.3.2. Các công cụ thu thập, làm sạch và chuyển đổi dữ liệu nguồn

Một yêu cầu quan trọng là sử dụng những dữ liệu đã được tinh chế từ những hệ thống tác nghiệp và đưa chúng vào một khuôn dạng thích hợp cho các ứng dụng thông tin. Những công cụ này thực hiện tất cả các công việc chuyển đổi, tóm tắt những thay đổi quan trọng, những thay đổi về cấu trúc và những cô đọng cần thiết cho sự chuyển đổi dữ liệu riêng rẽ thành thông tin có thể được dùng trong những công cụ hỗ trợ quyết định. Nó sinh ra những chương trình và kiểm soát những câu lệnh Cobol, ngôn ngữ JLC, Unix Script và ngôn ngữ định nghĩa dữ liệu SQL cần thiết để chuyển dữ liệu vào kho dữ liệu từ nhiều hệ thống tác nghiệp khác nhau. Ngoài ra nó cũng duy trì Metadata. Các chức năng chính bao gồm:

- Loại bỏ những dữ liệu không mong muốn từ những cơ sở dữ liệu tác

nghiệp.

- Chuyển đổi thành những tên và những định nghĩa dữ liệu chung.
- Tính toán các tổng và dữ liệu đã được chuyển hóa.
- Thiết lập những mặc định cho các dữ liệu bị mất.
- Làm cho những thay đổi về định nghĩa dữ liệu nguồn trở nên thích hợp.

Những công cụ này có thể tiết kiệm được một cách đáng kể thời gian và sức lực. Tuy nhiên nhiều công cụ có sẵn mới chỉ có ích cho việc tinh chế những dữ liệu đơn giản do đó việc phát triển những thủ tục tinh chế có khả năng tùy biến là cần thiết. Các công đoạn thực hiện bao gồm:

a. Trích lấy dữ liệu

Trích lấy dữ liệu là xử lý để lấy các dữ liệu đã được xác định trước ra khỏi các hệ thống tác nghiệp và các nguồn dữ liệu ngoài. Việc trích lấy dữ liệu nguồn có thể được hoàn thành bởi các công việc: đọc nguồn một cách trực tiếp, đọc một ảnh của nguồn hoặc đọc Log.

Có một số công cụ và các trình tiện ích phục vụ cho quá trình trích lấy dữ liệu. Các vấn đề xung quanh việc trích lấy dữ liệu bao gồm cơ cấu thời gian trong đó dữ liệu được trích lấy và hiệu quả của việc trích lấy dữ liệu đó.

Với mọi phương thức trích chọn dữ liệu, Metadata luôn đóng vai trò quan trọng trong quá trình xử lý. Metadata mẫu bao gồm: các định nghĩa của hệ thống nguồn, các khuôn dạng vật lý, phương thức và bản liệt kê việc trích lấy dữ liệu. Có thể dùng các công cụ hoặc thực hiện bằng tay để thu được Metadata.

Có thể phát hiện ra những thay đổi được thực hiện đối với dữ liệu trong hệ thống LS thông qua việc đọc Log. Những thay đổi đó là các hành động chèn thêm, cập nhật và xoá cũng như thông tin của cột hoặc hàng liên quan. Toàn bộ những thay đổi được ghi lại và sau đó được áp dụng theo trật tự mà các thay đổi đó đã được thực hiện trong hệ thống tác nghiệp.

b. Tinh chế dữ liệu

Dữ liệu sau khi được trích xuất sẽ được tinh chế thông qua các công việc làm sạch (Cleaning), chuyển đổi (Transforming) và tích hợp. Các công cụ đó có thể thực hiện trên một tập các thông số đã được xác định trước, trên Logic mờ hoặc triển khai các thuật toán thông minh. Các thuật toán thông minh Heuristic với tập luật mở rộng mô phỏng suy diễn của con người làm cho việc điều tra tiến hành nhanh hơn.

Trước khi có thể chuyển đổi và tích hợp dữ liệu, nên thiết lập hệ thống đo lường và chuẩn hoá các định/ngữ nghĩa. Mục đích của việc chuyển đổi và tích hợp là chuyển dữ liệu thành thông tin và làm cho chúng dễ hiểu, dễ sử dụng hơn đối với người sử dụng.

Các định nghĩa của dữ liệu phải chính xác, đầy đủ, tin cậy và có giá trị. Nếu dữ liệu đã được đưa vào kho dữ liệu không đúng thì sau đó phải quan tâm tới việc xem xét lại. Việc này liên quan nhiều tới việc tổ chức. Các câu hỏi cần đặt ra trước khi thay đổi cái cũ là: các thay đổi có hợp pháp và đúng quy cách không? Có thể đáp ứng được những thay đổi này không? Thay đổi có phải là lâu dài không? Nếu câu trả lời là có cho cả 3 câu hỏi trên thì thay đổi đó là có thể thực hiện được.

2.1.4. Cơ sở dữ liệu của kho dữ liệu

Cơ sở dữ liệu tập trung là một nền tảng cơ bản của môi trường kho dữ liệu. Cơ sở dữ liệu này hầu hết được cài đặt dựa trên công nghệ của Hệ thống quản trị cơ sở dữ liệu quan hệ (RDBMS). Tuy nhiên việc cài đặt một kho dữ liệu dựa trên kỹ thuật của RDBMS truyền thống bị ràng buộc bởi một thực tế là việc cài đặt RDBMS truyền thống đã được tối ưu hoá đối với việc xử lý cơ sở dữ liệu giao dịch. Những thuộc tính tất yếu của kho dữ liệu như kích cỡ rất lớn, xử lý các truy vấn đặc biệt và sự cần thiết tạo ra những khung nhìn linh hoạt cho người sử dụng bao gồm việc tập hợp, kết hợp nhiều bảng và khoan

sâu (Drill_down) trở thành những định hướng cho các cách tiếp cận khác nhau tới cơ sở dữ liệu của kho dữ liệu. Những cách tiếp cận đó bao gồm:

- Thiết kế CSDL quan hệ song song.
- Một cách tiếp cận mới để làm tăng tốc độ RDBMS truyền thống là cách sử dụng một cấu trúc chỉ số bỏ qua kiểm tra các bảng quan hệ.
- Các cơ sở dữ liệu đa chiều dựa trên công nghệ cơ sở dữ liệu phổ biến hoặc được cài đặt sử dụng trên nền RDBMS quen thuộc. Cơ sở dữ liệu đa chiều được thiết kế để khắc phục những giới hạn tồn tại trong kho dữ liệu gây ra do bản chất của mô hình dữ liệu quan hệ. Cách tiếp cận này gắn liền với các công cụ xử lý phân tích trực tuyến thực hiện như một đối tác của các kho dữ liệu đa chiều. Các công cụ này gộp lại thành một nhóm công cụ truy vấn, tạo báo cáo, phân tích và đào xới dữ liệu.

2.1.5. Kho dữ liệu

2.1.5.1. Định nghĩa

“Kho dữ liệu (Data Warehouse) là tập hợp của các CSDL tích hợp, hướng chủ đề, được thiết kế để hỗ trợ cho chức năng trợ giúp quyết định mà mỗi đơn vị dữ liệu đều liên quan tới một khoảng thời gian cụ thể”.[1]

Kho dữ liệu thường có dung lượng rất lớn, tới hàng trăm Gigabyte hay thậm chí hàng Terabyte dữ liệu được tổ chức, lưu trữ và phân tích phục vụ cho việc cung cấp các dịch vụ thông tin liên quan đến yêu cầu của một tổ chức nào đó. Kho dữ liệu phục vụ cho việc phân tích với kết quả mang tính thông tin cao. Các hệ thống thông tin thu thập, xử lý dữ liệu loại này còn gọi là Hệ xử lý phân tích trực tuyến (OLAP).

Một kho lưu trữ dữ liệu thường được sử dụng như cơ sở cho một hệ thống hỗ trợ quyết định. Nó được thiết kế để khắc phục những vấn đề vấp phải khi một tổ chức cố gắng thực hiện chiến lược phân tích có sử dụng cùng

một cơ sở dữ liệu đã được sử dụng cho xử lý giao dịch trực tuyến.

2.1.5.2. Đặc điểm dữ liệu trong kho dữ liệu

Kho dữ liệu là một tập hợp dữ liệu có những tính chất sau:

a. Dữ liệu có tính tích hợp

Một kho dữ liệu là một khung nhìn thông tin ở mức toàn thể, thống nhất các khung nhìn khác nhau thành một khung nhìn của một chủ đề. Ví dụ, hệ thống OLTP truyền thống được xây dựng trên một vùng phục vụ việc kinh doanh. Một hệ thống bán hàng và Marketing có thể có chung một dạng thông tin về khách hàng, nhưng các vấn đề về tài chính thì lại cần một khung nhìn khác. Một kho dữ liệu sẽ có một khung nhìn toàn thể về một khách hàng, khung nhìn đó bao gồm các phần dữ liệu khác nhau từ tài chính đến Marketing.

Tính tích hợp thể hiện ở chỗ dữ liệu tập hợp trong kho dữ liệu được thu thập từ nhiều nguồn và trộn ghép với nhau tạo thành một thể thống nhất.

b. Dữ liệu gắn thời gian và có tính lịch sử

Một kho chứa dữ liệu bao hàm một khối lượng lớn dữ liệu mang tính lịch sử. Dữ liệu được lưu trữ thành một loạt các Snapshot, mỗi Snapshot phản ánh những giá trị của dữ liệu tại một thời điểm nhất định thể hiện một khung nhìn của một vùng chủ đề trong một giai đoạn. Do vậy nó cho phép khôi phục lại lịch sử và so sánh một cách chính xác các giai đoạn khác nhau. Yếu tố thời gian đóng vai trò như một phần của khoá để bảo đảm tính đơn nhất và cung cấp đặc trưng về thời gian cho dữ liệu.

c. Dữ liệu chỉ đọc

Dữ liệu trong kho dữ liệu là dữ liệu chỉ đọc, có thể được kiểm tra và không được sửa đổi bởi người sử dụng.

d. Dữ liệu không biến động

Thông tin trong kho dữ liệu được tải vào sau khi dữ liệu trong hệ thống

điều hành được cho là quá cũ. Không biến động thể hiện ở chỗ: dữ liệu được lưu trữ lâu dài trong kho dữ liệu. Mặc dù có thêm dữ liệu mới nhập vào nhưng dữ liệu cũ trong kho vẫn không bị xoá, điều đó cho phép cung cấp thông tin về một khoảng thời gian dài, cung cấp đủ số liệu cần thiết cho các mô hình nghiệp vụ phân tích, dự báo.

e. Dữ liệu tổng hợp và chi tiết

Dữ liệu chi tiết là thông tin mức thấp nhất được lưu trữ trong kho dữ liệu. Dữ liệu tác nghiệp là thông tin mức thấp nhất cho một tổ chức. Dữ liệu tác nghiệp thuần túy không được lưu trữ trong kho dữ liệu. Dữ liệu tổng hợp được tích lại qua nhiều giai đoạn khác nhau.

2.1.6. Kho dữ liệu chủ đề (Datamart)

Kho dữ liệu chủ đề (Datamart - DM) là CSDL có những đặc điểm giống với kho dữ liệu nhưng với quy mô nhỏ hơn và lưu trữ dữ liệu về một lĩnh vực, một chuyên ngành. Các Datamart có thể được hình thành từ một tập con dữ liệu của kho dữ liệu hoặc cũng có thể được xây dựng độc lập và sau khi xây dựng xong các Datamart có thể được kết nối, tích hợp lại với nhau tạo thành kho dữ liệu.

Datamart là một kho dữ liệu thứ cấp gồm các dữ liệu tích hợp của kho dữ liệu. Datamart được hướng tới một phần của dữ liệu, thường được gọi là một vùng chủ đề (SA) được tạo ra dành cho một nhóm người sử dụng. Dữ liệu trong Datamart cho thông tin về một chủ đề xác định, không phải về toàn bộ các hoạt động nghiệp vụ đang diễn ra trong một tổ chức. Thể hiện thường xuyên nhất của Datamart là một kho dữ liệu riêng rẽ theo phương diện vật lý, thường được lưu trữ trên một Server riêng trong một mạng cục bộ phục vụ cho một nhóm người nhất định. Đôi khi Datamart với công nghệ OLAP tạo ra các quan hệ theo dạng hình sao đặc biệt hoặc những siêu khối (Hypercube) dữ liệu cho việc phân tích của một nhóm người có cùng mối quan tâm trên một

phạm vi dữ liệu. Có thể chia Datamart ra làm 2 loại: Datamart độc lập và Datamart phụ thuộc.

Datamart phụ thuộc chứa những dữ liệu được lấy từ kho dữ liệu và những dữ liệu này sẽ được trích lọc, tinh chế, tích hợp lại ở mức cao hơn để phục vụ một chủ đề nhất định.

Datamart độc lập không giống như Datamart phụ thuộc, nó được xây dựng trước kho dữ liệu và dữ liệu được lấy từ các nguồn dữ liệu tác nghiệp. Phương pháp này đơn giản hơn và chi phí thấp hơn nhưng đổi lại có những điểm yếu. Mỗi Datamart độc lập có cách tích hợp riêng do đó dữ liệu từ nhiều Datamart khó đồng nhất với nhau.

Datamart thể hiện hai vấn đề: tính ổn định khi một Datamart nhỏ ban đầu lớn lên nhanh chóng theo nhiều chiều và sự tích hợp dữ liệu. Vì vậy khi thiết kế Datamart phải chú ý tới tính ổn định của hệ thống, sự đồng nhất của dữ liệu và vấn đề về khả năng quản lý.

2.2. Sử dụng kho dữ liệu

Kho dữ liệu được sử dụng theo ba cách chính:

- Theo cách khai thác truyền thống, kho dữ liệu được sử dụng để khai thác các thông tin bằng các công cụ vấn đáp và báo cáo. Tuy nhiên, nhờ có việc xuất ra, tổng hợp và chuyển đổi từ các dữ liệu thô sang dạng các dữ liệu chất lượng cao và có tính ổn định, kho dữ liệu đã giúp nâng cao các kỹ thuật biểu diễn thông tin truyền thống (hỏi đáp và báo cáo). Bằng cách tạo ra một tầng ẩn giữa người dùng và CSDL, các dữ liệu đầu vào của kỹ thuật này được đặt vào một nguồn duy nhất. Việc hợp nhất này loại bỏ được rất nhiều lỗi sinh ra do việc phải thu thập và biểu diễn thông tin từ rất nhiều nguồn khác nhau cũng như giảm bớt được sự chậm trễ do phải lấy các dữ liệu bị phân đoạn trong các CSDL khác nhau, tránh cho người dùng khỏi những câu lệnh phức tạp. Tuy nhiên

đây mới chỉ là cách khai thác với kỹ thuật cao để đưa ra các dữ liệu tinh và chính xác hơn chứ chưa đưa ra được dữ liệu “tri thức”.

- Các kho dữ liệu được sử dụng để hỗ trợ cho phân tích trực tuyến (OLAP). Trong khi ngôn ngữ truy vấn chuẩn SQL và các công cụ làm báo cáo truyền thống chỉ có thể miêu tả những gì có trong CSDL thì phân tích trực tuyến có khả năng phân tích dữ liệu, xác định xem giả thuyết đúng hay sai. Tuy nhiên phân tích trực tuyến lại không có khả năng đưa ra được các giả thuyết. Hơn nữa, kích thước quá lớn và tính chất phức tạp của kho dữ liệu làm cho nó rất khó có thể sử dụng cho những mục đích như đưa ra các giả thuyết từ các thông tin mà chương trình ứng dụng cung cấp (ví dụ như khó có thể đưa ra được giả thuyết giải thích được hành vi của một nhóm khách hàng).
- Trước đây, kỹ thuật học máy thường được sử dụng để tìm ra những giả thuyết từ các thông tin dữ liệu thu thập được. Tuy nhiên thực nghiệm cho thấy chúng thể hiện khả năng rất kém khi áp dụng với các tập dữ liệu lớn trong kho dữ liệu. Phương pháp thống kê tuy ra đời đã lâu nhưng không có gì cải tiến để phù hợp với sự phát triển của dữ liệu. Đây chính là lý do tại sao một khối lượng lớn dữ liệu vẫn chưa được khai thác và thậm chí được lưu chủ yếu trong các kho dữ liệu không trực tuyến (Offline). Điều này đã tạo nên một lỗ hổng lớn trong việc hỗ trợ phân tích và tìm hiểu dữ liệu, tạo ra khoảng cách giữa việc tạo ra và việc khai thác dữ liệu đó. Trong khi đó càng ngày người ta càng nhận thấy rằng nếu được phân tích thông minh thì dữ liệu sẽ là một nguồn tài nguyên quý giá. Từ đó người ta đã đưa ra một phương pháp mới đáp ứng cả nhu cầu trong khoa học cũng như trong hoạt động thực tiễn, đó chính là công nghệ khai phá dữ liệu (Data Mining). Đây chính là ứng dụng chính thứ ba của kho dữ liệu.

2.3. Phương pháp xây dựng kho dữ liệu

Xây dựng kho dữ liệu vừa là một tiến trình công việc và cũng đồng thời là một kiến trúc nhằm thực hiện các nội dung như: lựa chọn, chuyển đổi, lưu chuyển, bảo toàn tính toàn vẹn, tích hợp, làm sạch dữ liệu, đưa dữ liệu từ nhiều nguồn dữ liệu tác nghiệp vào hệ thống quản lý cơ sở dữ liệu để phục vụ các quá trình ra quyết định. Kiến trúc của các kho dữ liệu cung cấp nhiều khả năng mềm dẻo, nhiều khả năng mở rộng để phục vụ cho các ứng dụng hiện có cũng như cho các ứng dụng mới trong tương lai. Kho dữ liệu gồm các thành phần thiết yếu sau:

- Các nguồn dữ liệu tác nghiệp ODS (Operational Data Sources).
- Chuyển đổi và xuất ra dữ liệu (Data Conversion and Extraction).
- Tóm lược và làm giàu dữ liệu (Data Sumaization & Data Enrichment).
- Hệ thống quản lý các CSDL của kho dữ liệu (Database Management System - DBMS).
- Quản lý các siêu dữ liệu.
- Các công cụ (Tools) truy nhập và phân tích.

Quá trình xây dựng kho dữ liệu có thể bắt đầu bằng việc xây dựng các Datamart, có nghĩa là sau khi xây dựng xong các Datamart ta tiến hành kết nối, tích hợp chúng với nhau tạo thành kho dữ liệu. Theo cách này, Datamart chính là mô hình và là bước đầu tiên của quá trình xây dựng kho dữ liệu. Cách thứ hai, ta có thể xây dựng kho dữ liệu trước sau đó tạo ra các Datamart. Mỗi phương pháp đều có thuận lợi và khó khăn của nó, tùy điều kiện cụ thể ta lựa chọn hay kết hợp các phương pháp cho phù hợp.

Phương pháp phân tích, thiết kế và quá trình xây dựng kho dữ liệu có thể được chia thành các giai đoạn, trong mỗi giai đoạn có các bước:

- Giai đoạn khảo sát

Bước 1: Xác định chiến lược và xây dựng kế hoạch

Bước 2: Khảo sát, đánh giá hiện trạng hệ thống

- Giai đoạn phân tích thiết kế

Bước 3: Phân tích, thiết kế hệ thống và xây dựng mẫu thử nghiệm (Prototype)

- Giai đoạn xây dựng, phát triển hệ thống

Bước 4: Triển khai xây dựng hệ thống

Bước 5: Khai thác và duy trì hệ thống

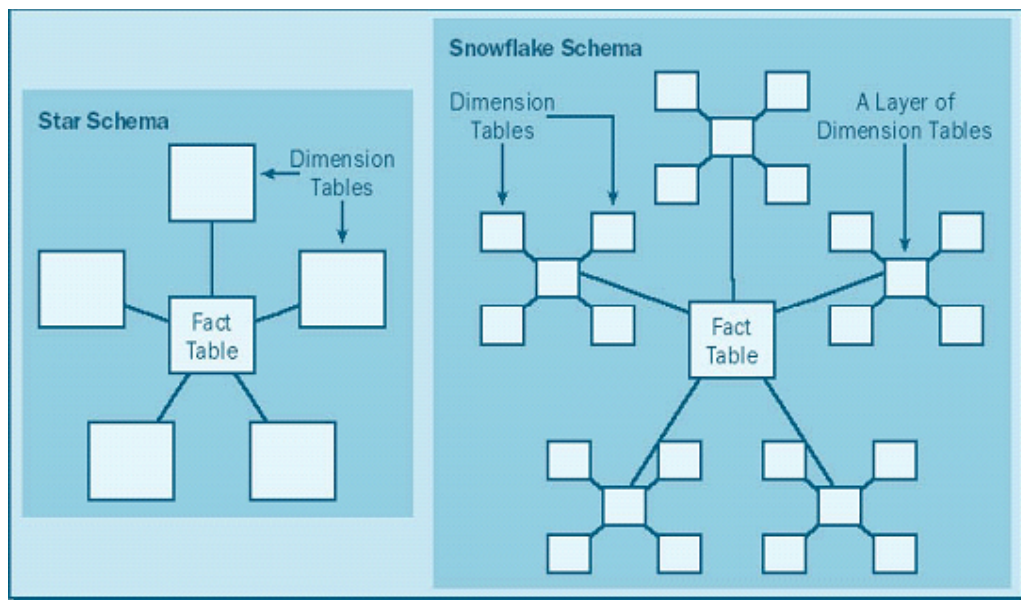
2.4. Thiết kế CSDL cho kho dữ liệu

Một vài phương pháp và công cụ phục vụ tốt cho việc tạo ra các hệ thống tác nghiệp gần như là không phù hợp với những yêu cầu khác nhau của kho dữ liệu. Điều này rất đúng trong các hệ thống quản trị cơ sở dữ liệu. Hệ thống OLTP truyền thống được thiết kế một cách đơn giản không phù hợp với những yêu cầu của phương pháp kho dữ liệu. Những dự án dùng phương pháp kho dữ liệu buộc phải lựa chọn giữa một mô hình dữ liệu và một giản đồ dữ liệu liên quan trực quan cho việc phân tích nhưng nghèo nàn về thể hiện. Một giản đồ - mô hình là cách thực hiện tốt hơn nhưng không phù hợp lắm cho việc phân tích. Khi phương pháp kho dữ liệu được tiếp tục phát triển thì những cách tiếp cận mới cho việc thiết kế giản đồ dữ liệu phù hợp hơn với việc phân tích được hình thành và đó là điều cốt yếu dẫn đến thành công của phương pháp kho dữ liệu. Một giản đồ được chấp nhận sử dụng rộng rãi cho phương pháp kho dữ liệu là giản đồ hình sao.

2.4.1. Giản đồ hình sao (Star)

Việc phân tích, dự báo đòi hỏi những giản đồ CSDL chủ yếu tập trung vào những truy vấn mà bản chất là đa chiều và hướng mảng (Array-oriented). Như vậy, công nghệ CSDL chính của kho dữ liệu là RDBMS. Ta sẽ xem xét việc thiết kế giản đồ dữ liệu khi gắn liền nó với công nghệ CSDL quan hệ.

Giản đồ hình sao được đưa ra lần đầu tiên bởi Raph Kimball như là một lựa chọn thiết kế CSDL cho kho dữ liệu. Trong giản đồ hình sao, dữ liệu được xác định và phân loại theo 2 kiểu: sự kiện (bảng Fact: đối tượng trung tâm) và phạm vi (các bảng Dimension: các bảng liên kết). Trong giản đồ hình sao chỉ có một bảng liên quan trực tiếp tới hầu hết các bảng còn lại đó là bảng Fact và là bảng chứa yếu tố cốt lõi cần được phân tích. Nó được gọi là giản đồ hình sao bởi vì các sự kiện nằm ở trung tâm của mô hình và được bao quanh bởi các phạm vi liên quan, rất giống với các điểm của một ngôi sao. Các sự kiện là các đại lượng số của công việc. Các phạm vi là các bộ lọc hoặc các ràng buộc của những sự kiện này. Ví dụ: thông tin về khách hàng như tên, địa chỉ là một phạm vi, trong khi đó thông tin bán hàng cho khách hàng đó là một sự kiện.



Hình 2.2. Giản đồ hình sao và hình tuyết rơi

Với giản đồ hình sao, người thiết kế có thể dễ dàng mô phỏng những chức năng của CSDL đa chiều. Sự phi chuẩn hóa có thể coi là sự tiền kết nối

(Pre-joining) các bảng để cho các ứng dụng không phải thực hiện công việc kết nối, làm giảm thời gian thực hiện.

Giản đồ hình sao được thiết kế là để khắc phục những hạn chế của mô hình quan hệ hai chiều. Với cơ sở dữ liệu được thiết kế theo giản đồ hình sao, những truy vấn với những câu hỏi phức tạp liên quan tới nhiều bảng và số liệu tổng cộng trở nên đơn giản hơn và số lượng công việc cần thực hiện để đưa được ra câu trả lời là ít nhất so với một mô hình quan hệ chuẩn. Giản đồ hình sao cải thiện đáng kể thời gian truy vấn và cho phép thực hiện một số tính năng đa phạm vi. Giản đồ này rất trực quan, dễ sử dụng, thể hiện khung nhìn đa chiều của dữ liệu dùng ngữ nghĩa của CSDL quan hệ. Khóa của bảng Fact được tạo bởi những khóa của các bảng chứa thông tin theo từng phạm vi (bảng Dimension). Tất cả các khóa đều được xác định với cùng một chuẩn đặt tên.

Ví dụ, để lấy được thông tin thành phố của khách hàng cụ thể, cần phải kết hợp khóa chỉ khách hàng đó trong bảng sự kiện (bảng Fact) với khóa của khách hàng đó trong bảng phạm vi (bảng Dimension) và đặt thuộc tính thành phố của khách hàng đó là thành phố mà họ quan tâm.

Bảng Fact có chứa khóa của các bảng Dimension, có thể là với tên khác đi để đảm bảo tính duy nhất của mỗi hàng. Các bảng Dimension thường có định danh duy nhất và chứa đựng những thông tin về chiều (Dimension) của bảng đó.

Vì bảng Fact được tổng hợp từ trước và được kết hợp theo nhiều chiều nên xu hướng có rất nhiều hàng và tăng trưởng một cách nhanh chóng trong khi đó các bảng Dimension không có nhiều hàng và sự tăng trưởng là tĩnh. Bảng Fact có thể bao gồm hàng chục triệu hàng. Bảng Dimension chứa đựng các thuộc tính có thể được sử dụng như các tiêu chí tìm kiếm và thường có kích thước nhỏ hơn nhiều, rất quen thuộc với người sử dụng từ trước. Khoá

của nó không là khoá ghép như bảng Fact. Nếu một bảng Dimension bắt đầu có sự tương đồng với bảng Fact thì nó cần được tiếp tục chia ra thành các bảng Dimension nữa. Nếu một bảng Dimension được chia thành Dimension chính và Dimension phụ thì cấu trúc thu được gọi là một giản đồ tuyết rơi hoặc một cấu trúc sao mở rộng.

Một giản đồ hình sao đơn giản chỉ gồm một bảng Fact và một vài bảng Dimension. Một giản đồ hình sao phức tạp bao gồm hàng trăm bảng Fact và bảng Dimension. Một vài kỹ thuật để cải thiện hiệu suất của các truy vấn trong giản đồ hình sao bao gồm:

- Xác định sự kết hợp các bảng Fact đang tồn tại hay tạo ra một sự kết hợp mới các bảng Fact.
- Phân chia bảng Fact đến mức mà hầu hết các truy vấn chỉ truy nhập tới phần đó.
- Tạo ra các bảng Fact riêng rẽ.
- Tạo ra những tệp chỉ số đơn duy nhất hoặc các kỹ thuật khác để cải thiện năng suất kết hợp.

Cả bảng Fact và các bảng Dimension đều không bắt buộc ở dạng chuẩn như đối với phương pháp thiết kế truyền thống tức là có dư thừa dữ liệu. Loại giản đồ này cho phép lưu trữ dư thừa dữ liệu, đổi lại khả năng truy nhập nhanh hơn phù hợp với những câu hỏi phân tích nhiều chiều, phức tạp. Về bản chất bảng Fact thuộc dạng chuẩn 1 với mức độ dư thừa dữ liệu rất lớn.

Có thể nói giản đồ hình sao là một CSDL chỉ đọc, việc cập nhật dữ liệu là rất khó nếu không muốn nói là không thể được. Một vài bảng Dimension chứa dữ liệu có thể được thêm vào bằng các truy vấn có kết nối, một vài bảng khác lại không chứa dữ liệu gì ngoài việc phục vụ đánh chỉ số cho dữ liệu.

2.4.2. Giản đồ hình tuyết rơi (Snowflake)

Giản đồ hình tuyết rơi là một sự mở rộng của giản đồ hình sao, tại đó

mỗi cánh sao không phải là một bảng Dimension mà là nhiều bảng. Trong dạng giản đồ này, mỗi bảng theo chiều của giản đồ hình sao được chuẩn hóa hơn. Giản đồ hình tuyết rơi cải thiện năng suất truy vấn, tối thiểu không gian đĩa cần thiết để lưu trữ dữ liệu và cải thiện năng suất nhờ việc chỉ phải kết hợp những bảng có kích thước nhỏ hơn thay vì phải kết hợp những bảng có kích thước lớn lại không chuẩn hóa. Nó cũng làm tăng tính linh hoạt của các ứng dụng bởi sự chuẩn hóa và ít mang bản chất theo chiều hơn. Nó làm tăng số lượng các bảng và làm tăng tính phức tạp của một vài truy vấn cần có sự tham chiếu tới nhiều bảng. Một vài công cụ đã che giấu người sử dụng giản đồ CSDL vật lý và cho phép họ có thể làm việc ở mức khái niệm. Những công cụ này đã ánh xạ những truy vấn của người sử dụng tới sơ đồ vật lý. Họ cần một bộ quản trị CSDL để thực hiện công việc này một lần đầu tiên khi công cụ này được cài đặt.

2.4.3 Giản đồ kết hợp

Là kết hợp giữa giản đồ hình sao dựa trên bảng Fact và những bảng Dimension không chuẩn hóa theo các chuẩn 1, 2, 3 và giản đồ hình tuyết rơi trong đó tất cả các bảng Dimension đều đã được chuẩn hóa. Trong giản đồ loại này chỉ những bảng Dimension lớn là được chuẩn hóa còn những bảng khác chứa một khối lượng lớn các cột dữ liệu chưa được chuẩn hóa.

Một vài CSDL và các công cụ truy vấn của người sử dụng, nhất là các công cụ xử lý phân tích trực tuyến (OLAP) đòi hỏi mô hình dữ liệu phải là giản đồ hình sao bởi vì nó là một mô hình dữ liệu quan hệ nhưng lại được thiết kế để hỗ trợ mô hình dữ liệu đa chiều, là điểm cốt lõi của OLAP. Các cơ sở dữ liệu và công cụ này được điều chỉnh cho phù hợp để thực hiện được các yêu cầu truy vấn đối với mô hình này.

2.4.4. Những vấn đề liên quan tới thiết kế giản đồ hình sao

Mặc dù hầu hết các chuyên gia đều đồng ý rằng giản đồ hình sao thích hợp cho phương pháp thiết lập mô hình cho phương pháp kho dữ liệu nhưng vẫn còn một số vấn đề của hệ quản trị cơ sở dữ liệu quan hệ liên quan tới việc cài đặt giản đồ hình sao.

2.4.4.1. Đánh chỉ số

Sử dụng việc đánh chỉ số có thể đảm bảo sự duy nhất của các khóa và có thể cải thiện năng suất đọc. Vì các bảng trong thiết kế hình sao điển hình chứa sự phân cấp tổng thể của các thuộc tính, cách thức này được chấp nhận cho những thiết kế bình thường nhưng nó cũng thể hiện một vài vấn đề trong mô hình giản đồ hình sao đó là:

- Nó đòi hỏi sự định nghĩa Metadata phức tạp (một cho mỗi thành phần khóa) để xác định một mối quan hệ đơn (một bảng). Điều này làm cho thiết kế thêm phức tạp và hiệu suất kém đi nhiều.
- Vì bảng Fact phải chứa tất cả các khóa thành phần như một phần của khóa chính nên việc thêm vào hay xóa bỏ một mức trong sơ đồ phân cấp sẽ đòi hỏi sự thay đổi vật lý ở các bảng liên quan mất nhiều thời gian và hạn chế tính linh hoạt.
- Việc chứa tất cả các đoạn khóa của mỗi Dimension trong bảng Fact làm tăng kích thước của bảng chỉ số và tác động mạnh tới hiệu suất và sự ổn định.

Một phương pháp đối với khóa ghép như trên là cắt khóa ra thành các khóa đơn. Cách này giải quyết được 2 vấn đề đầu nhưng kích thước của bảng chỉ số vẫn là một vấn đề. Cách tốt nhất là thay những khóa có ý nghĩa bằng việc sử dụng một khóa do mình tạo ra là một khóa nhỏ nhất có thể mà vẫn bảo đảm tính duy nhất của mỗi bản ghi. Những khóa có nghĩa được thay thế như

nói ở trên không cần thiết phải hủy bỏ, đơn giản chúng có thể được chuyển đến một thuộc tính không phải là khóa. Kết quả thiết kế theo mô hình hình sao bao gồm một bảng Fact với một khóa chính có đúng một cột khóa cho mỗi chiều, tại đó mỗi khóa là khóa được tạo ra. Phương pháp này cho khả năng linh hoạt ở mức cao nhất, việc bảo trì là ít nhất và cho hiệu suất cao nhất có thể.

2.4.4.2. Chỉ thị về mức

Để định hướng các chiều một cách thành công, việc thiết kế các bảng Dimension thường bao gồm một mức chỉ dẫn phân cấp cho mỗi bản ghi. Mỗi truy vấn lấy dữ liệu từ các bản ghi chi tiết của một bảng lưu trữ chi tiết và những dữ liệu kết hợp phải sử dụng chỉ dẫn này như một ràng buộc thêm để thu được kết quả đúng. Mức này là một công cụ có ích cho các môi trường được kiểm soát chặt chẽ bởi các DBA và trong môi trường đó một vài truy vấn đặc biệt được cho phép sử dụng. Nếu người sử dụng không quan tâm tới chỉ thị về mức hoặc giá trị của nó không đúng thì mặc dù quá trình truy vấn là đúng vẫn có thể đưa ra kết quả không hợp lệ.

Sự lựa chọn tốt nhất cho việc dùng chỉ thị về mức là sử dụng giản đồ hình tuyết rơi. Trong giản đồ loại này, các bảng Fact kết hợp được tạo ra một cách riêng biệt từ những bảng chứa dữ liệu chi tiết. Thêm vào với các bảng Fact chính, giản đồ hình tuyết rơi còn chứa các bảng Fact riêng rẽ cho mỗi mức kết hợp, vì vậy không mắc lỗi trong việc lựa chọn các bản ghi chi tiết. Tuy nhiên giản đồ hình tuyết rơi phức tạp hơn giản đồ hình sao và thường đòi hỏi những câu lệnh SQL phức tạp hơn để nhận được câu trả lời.

2.4.5. Những nhân tố thiết kế cần phải được cân nhắc

Thiết kế cấu trúc kho dữ liệu có thể làm ảnh hưởng đến tính dễ dàng trong việc thiết kế và xây dựng các khối (Cube). Microsoft SQL Server OLAP

Services dựa vào dữ liệu được cung cấp bởi kho dữ liệu có tính chính xác, ổn định và toàn vẹn. Khi tạo ra một kho dữ liệu sử dụng với OLAP, những nhân tố thiết kế cần phải được cân nhắc là:

- Sử dụng sơ đồ hình sao hoặc bảng phẳng chính (Flat) nếu có thể. Nếu một sơ đồ dạng hình tuyết rơi là cần thiết thì giảm thiểu số bảng Dimension vượt ra ngoài mức thứ nhất từ bảng chính.
- Thiết kế các bảng Dimension cho người dùng. Các bảng Dimension cần có thông tin ý nghĩa về thực thể mà người dùng muốn tìm hiểu.
- Áp dụng việc chuẩn hoá thông thường vào thiết kế bảng Dimension. Không nên kết hợp dữ liệu không quan hệ vào bảng Dimension đơn và không nên lặp lại dữ liệu trong các bảng Dimension. Ví dụ: tạo Dimension khách hàng riêng biệt thay vì lặp lại thông tin khách hàng trong nhiều bảng Dimension.
- Không tổng hợp thừa trong bảng chính. Giữ lại mức tinh tế cần thiết cho người dùng truy cập và giữ lại tất cả các bản ghi của bảng chính trong cùng một mức độ chi tiết. OLAP Services được thiết kế để tạo ra và quản lý dữ liệu tổng hợp từ các kho lưu trữ dữ liệu hạt nhân mức cao để không làm tăng thời gian trả lời yêu cầu.
- Sử dụng cấu trúc chung cho bảng chính (Fact) cho dữ liệu cùng loại. Dữ liệu sử dụng trong một khối có thể được lưu trữ trong các bảng chính đa chiều nhưng những bảng này phải có cùng cấu trúc.
- Không tạo các bảng phụ cho dữ liệu tổng. OLAP Services tính toán trước các tổng theo cấu trúc mà được thiết kế cho việc truy vấn có hiệu quả. Các bảng tổng phụ không được sử dụng.
- Tạo chỉ số cho các trường khoá. Với mỗi bảng Dimension tạo ra một chỉ số trên cột khoá của nó, với mỗi bảng Fact tạo ra một chỉ số đơn trên tổ hợp các cột mà nó chứa các khoá ngoại của bảng Dimension

được kết hợp với bảng Fact. OLAP Services sử dụng những chỉ số này khi chúng Load các cấu trúc dữ liệu đa chiều và các tính toán dữ liệu tổng. Những chỉ số này cải tiến đáng kể quá trình xử lý.

- Bảo đảm tính toàn vẹn. Đây là điều quan trọng vì các bảng Fact được biểu diễn theo các bảng Dimension. Các bảng Fact mà không có khoá tương ứng trong bảng Dimension có thể gây lỗi hoặc các hàng trong bảng Fact bị bỏ đi nếu các bảng Fact và bảng Dimension được dùng trong cùng một khối. Các bảng Dimension chứa thông tin không được biểu diễn trong bảng Fact có thể gây ra các ô trống trong các khối. Những ô trống này có thể gây trở ngại cho một số kết quả tính toán phân tích.
- Thiết kế một chiến lược cập nhật dữ liệu. Khi dữ liệu được thêm vào hoặc thay đổi trong kho lưu trữ dữ liệu, các khối được xây dựng từ dữ liệu trước phải được cập nhật trước khi dữ liệu mới được cung cấp cho người dùng. Việc sát nhập dữ liệu bổ sung trong các khối đòi hỏi thời gian ít hơn việc xây dựng các khối khi dữ liệu tồn tại thay đổi.

2.5. Quản trị kho dữ liệu

Kho dữ liệu có độ lớn gấp khoảng nhiều lần một kho dữ liệu tác nghiệp tổng thể. Nó không được đồng bộ với dữ liệu tác nghiệp liên quan trong thời gian thực nhưng có thể được cập nhật thường xuyên nếu như ứng dụng yêu cầu đến nó.

Hầu hết các sản phẩm của kho dữ liệu bao gồm các công để truy nhập tới các nguồn dữ liệu phức tạp mà không phải viết lại các phần mềm chuyển đổi, dịch và sử dụng dữ liệu. Trong một môi trường kho dữ liệu hỗn tạp, rất nhiều các CSDL khác nhau nằm trên những hệ thống riêng rẽ vì thế đòi hỏi các công cụ làm việc trao đổi giữa các mạng. Điều đó dẫn đến sự cần thiết phải quản trị các thành phần hạ tầng. Quản trị kho dữ liệu bao gồm:

- Quản trị về an toàn, bảo mật và độ ưu tiên
- Quản trị cập nhật từ nhiều nguồn khác nhau
- Kiểm tra chất lượng dữ liệu
- Quản trị và cập nhật Metadata
- Kiểm toán, lập báo cáo về việc sử dụng và trạng thái của kho dữ liệu
- Làm sạch dữ liệu
- Tái tạo dữ liệu, chia nhỏ dữ liệu thành những tập con và phân tán dữ liệu
- Sao lưu và phục hồi dữ liệu
- Quản trị các kho dữ liệu

Chương III. Tiếp cận và phân tích đa chiều trong xử lý phân tích trực tuyến

3.1. Tiếp cận đa chiều

OLAP là hoạt động xử lý tạo lập, quản lý dữ liệu đa chiều trong thực tế, giúp người sử dụng dễ dàng trong việc phân tích, tham khảo dữ liệu, nhằm hiểu được các thông tin tiềm ẩn mà dữ liệu đang chứa đựng. Các yêu cầu chính yếu của OLAP là:

- Truy xuất, tính toán nhanh.
- Có khả năng phân tích mạnh.
- Linh hoạt (phân tích linh hoạt, giao diện linh hoạt, hiển thị dữ liệu linh hoạt).
- Hỗ trợ nhiều người sử dụng.

Vấn đề đặt ra là phải chọn tiếp cận tổ chức dữ liệu nào để đáp ứng được những yêu cầu chức năng này của OLAP và mô hình dữ liệu đa chiều thực tế. Nhiều người đã cố tìm cách sử dụng bảng tính hay SQL để áp dụng OLAP vào nhưng điều này rất khó khăn, nhiều hạn chế và điều quan trọng là không thể hiện được những đặc trưng của OLAP, không đáp ứng được với những yêu cầu chức năng của OLAP và mô hình đa chiều. Lý do chủ yếu dẫn đến việc bảng tính bị hạn chế khi cố gắng tạo lập mô hình dữ liệu đa chiều đó là vì bảng tính không tách cấu trúc của mô hình ra khỏi những thể hiện của mô hình đó. Như vậy nó chỉ có thể được áp dụng đối với một bài toán đơn giản, trên một số lượng nhỏ dữ liệu được tổ chức dưới dạng bảng hai chiều. SQL cho chúng ta phương tiện truy vấn dựa trên các cột của dữ liệu nhưng không áp dụng được cho tất cả các trường hợp phân tích và cho việc so sánh trên các dòng. Cả hai tiếp cận này đều không làm cho chúng ta truy vấn dễ dàng khỏi lượng dữ liệu lớn được tổ chức một cách phức tạp. Tiếp cận tốt nhất để cung cấp xử lý hướng đến quyết định dựa trên phân tích và phù hợp với những yêu

cầu của OLAP là tiếp cận đa chiều. Các mô hình doanh nghiệp yêu cầu khả năng gộp dữ liệu ở nhiều mức khác nhau trong các chiều. Người phân tích cần có khả năng lướt nhanh dữ liệu thông qua việc thay đổi cấu hình hiển thị của dữ liệu trên màn hình. Họ cần có khả năng phân tích dữ liệu, chủ yếu là dựa vào việc tổng hợp và so sánh dữ liệu trên các chiều. Tiếp cận đa chiều có nhiều ưu điểm rõ ràng hơn tiếp cận bảng tính (Spreadsheet) hay SQL trên cả hai công việc định nghĩa và sử dụng các mô hình như vậy.

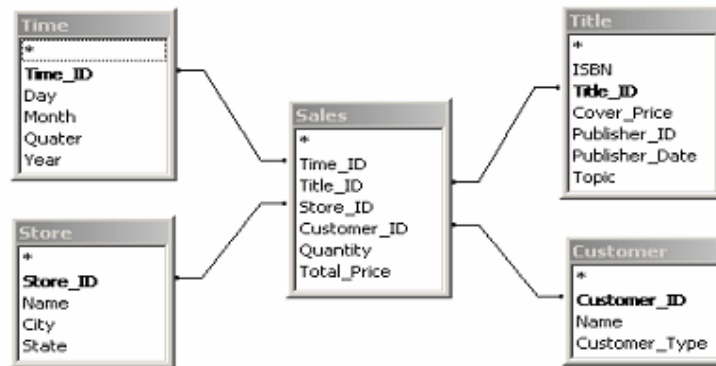
Sự tách riêng cấu trúc dữ liệu (được định nghĩa trong các chiều) ra khỏi biểu diễn của dữ liệu là một thuận lợi lớn của tiếp cận đa chiều. Nó làm tối thiểu sự cần thiết lập lại các thông tin về cấu trúc và cung cấp sự hỗ trợ trực tiếp cho việc làm thay đổi dễ dàng các yêu cầu hiển thị. Ngoài ra sự hỗ trợ trực tiếp của các chiều đa mức và khả năng gán các công thức trên trục (Axis-based) thay vì các công thức trên ô (Cell-based) làm việc định nghĩa các phép gộp đa mức và các tính toán đa chiều dễ dàng.

OLAP là công cụ phân tích trực tuyến. Bản chất cốt lõi của OLAP là dữ liệu được lấy ra từ kho dữ liệu hoặc Datamart sau đó được chuyển thành mô hình đa chiều và được lưu trữ trong một kho dữ liệu đa chiều (dữ liệu được lưu trữ theo mảng thay vì bản ghi như mô hình quan hệ). Các dịch vụ (hay công cụ) OLAP lấy dữ liệu trong kho dữ liệu để thực hiện các công việc phân tích đặc biệt theo nhiều chiều, phức tạp hỗ trợ cho việc ra quyết định. Giản đồ hình sao được dùng để thiết kế mô hình dữ liệu trong kho dữ liệu hoặc Datamart là mô hình dữ liệu quan hệ nhưng lại mang những thuộc tính nhiều chiều có rất nhiều thuận lợi cho việc cài đặt OLAP.

3.2. Phân tích đa chiều

Tất cả những dữ liệu có quan hệ với nhau đều cần được phân tích. Trong xử lý phân tích thì trọng tâm là phân tích dữ liệu, đặc biệt là phân tích đa chiều. Trong phân tích đa chiều, dữ liệu được miêu tả thành các chiều

(Dimensions) chẳng hạn như ‘Sản phẩm’, ‘Khu vực’ và ‘Khách hàng’. Các chiều thường liên quan tới những sự phân cấp ví dụ như ‘Thành phố’, ‘Vùng’ và ‘Nước’. Chiều thời gian là một chiều chuẩn với sự phân cấp của riêng nó là ‘Ngày’, ‘Tuần’, ‘Tháng’, ‘Quý’ và ‘Năm’.



Hình 3.1. Mô hình dữ liệu đa chiều

Để giải quyết sự phân tích phức tạp, phân tích nhiều chiều thể hiện một khung nhìn dữ liệu gần gũi với người sử dụng. Chẳng hạn, một người sử dụng có thể truy nhập tới ngân khố theo từng phòng ban và lưu trữ 4 quý cuối cho một tập các sản phẩm. Kết quả có thể được xoay để thay đổi vị trí các trục và khung nhìn. Thêm nữa người sử dụng có thể xem các chiều bằng cách khoan sâu (Drill-down) hay cuốn lên (Roll-up) theo các thành phần của mỗi chiều. Việc khoan sâu trên các chiều có thể tạo ra các khung nhìn khác. Phạm vi của xử lý thông tin thường đơn giản hơn (chỉ gồm 2 hoặc 3 chiều). Phân tích những dữ liệu lịch sử để hiểu được quá khứ là sự phân tích tĩnh. Xử lý phân tích có thể được dùng cho những phân tích lịch sử phức tạp với thao tác mở rộng hay gọi là sự phân tích động: lên kế hoạch và dự báo tiếp quá khứ như là phần mở đầu cho tương lai.

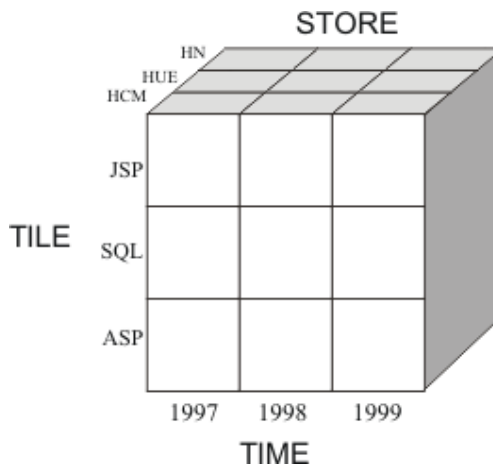
Trong kho dữ liệu, dữ liệu được lưu trữ cho việc truy vấn, phân tích và

các mục đích khác như OLTP, khi đó dữ liệu được thu thập và lưu trữ cho các hoạt động tác nghiệp và các mục đích kiểm soát.

3.3. Kiến trúc khối của OLAP (OLAP Cube Architecture)

3.3.1. Giới thiệu kiến trúc khối

Cơ sở dữ liệu OLAP sử dụng hình khối dữ liệu làm căn bản. Để hiểu hình khối OLAP như thế nào, chúng ta thử hình dung xem dữ liệu được chuyển vào CSDL OLAP xuất phát từ việc truy vấn dữ liệu từ bảng dữ liệu Fact và những bảng Dimensions. Nói cách khác, báo cáo cuối cùng của việc phân tích dữ liệu được kết xuất từ các loại bảng dữ liệu trên cùng với việc ứng dụng một số hàm tính toán.

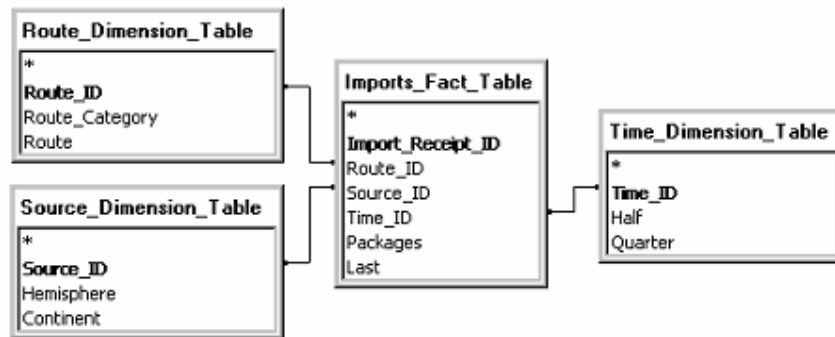


Hình 3.2. Mô hình dữ liệu khối

Để mô tả dữ liệu hình khối, chúng ta thử tưởng tượng dữ liệu trong bảng Fact được phân bố như sau: Đối tượng chính của OLAP là khối, một sự biểu diễn đa chiều của dữ liệu chi tiết và tổng thể. Một khối bao gồm một bảng sự kiện (Fact), một hoặc nhiều bảng chiều (Dimensions), các đơn vị đo (Measures) và các phân hoạch (Partitions). Ta có thể thiết kế các khối dựa trên cơ sở các yêu cầu phân tích của người sử dụng. Một kho dữ liệu có thể hỗ

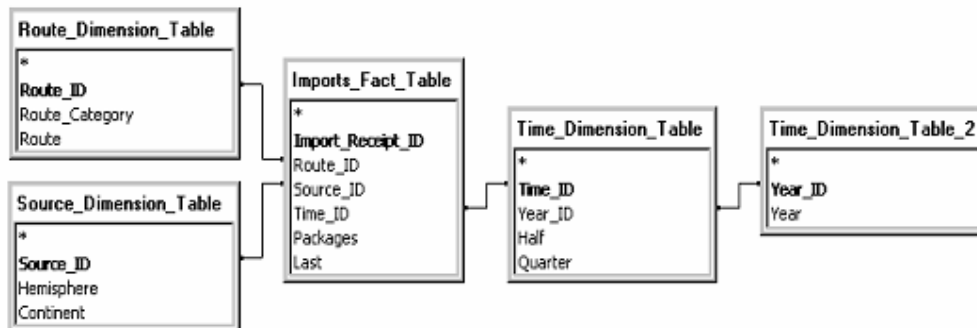
trợ nhiều khối khác nhau: khối về lương, khối về hàng tồn kho...

Ví dụ một giản đồ khối hình sao có dạng như sau:



Hình 3.3. Giản đồ khối hình sao

Ở đây nếu muốn, ta có thể mở rộng khối theo nhiều năm bằng cách thêm cột ‘Year_ID’ vào ‘Time_Dimension_Table’ và tạo thêm một bảng Dimension là ‘Time_Dimension_Table_2’ chứa hai cột ‘Year_ID’ và ‘Year’. Lúc này ta có được một giản đồ khối hình tuyết rơi như sau:



Hình 3.4. Giản đồ khối hình tuyết rơi

3.3.2. Khối (Cube)

Khối là phần tử chính trong xử lý phân tích trực tuyến, một công nghệ

cung cấp sự truy cập nhanh tới dữ liệu trong kho dữ liệu. Các khối cung cấp cơ chế truy vấn dữ liệu với thời gian trả lời nhanh và không phụ thuộc vào số lượng dữ liệu trong khối hoặc sự phức tạp của truy vấn.

Khối là tập con (Subset) dữ liệu từ kho dữ liệu, được tổ chức và tổng hợp trong các cấu trúc đa chiều. Bản tóm tắt của dữ liệu được tính toán trước để thời gian đáp ứng các yêu cầu phức tạp là nhanh và không đổi.

3.3.2.1. Xác định khối

Xác định khối là bước đầu tiên trong ba bước tạo khối. Các bước khác là các bước chỉ ra kế hoạch tóm tắt bằng việc thiết kế các khối tập hợp (các thành phần dữ liệu được tính toán trước) và Load khối bằng việc xử lý nó.

Để xác định một khối, ta chọn một bảng Fact và các đơn vị đo lường đồng nhất (các cột số theo sự quan tâm của người dùng khối) trong bảng Fact. Sau đó chọn các chiều, mỗi chiều gồm một hay nhiều cột từ bảng liên quan khác. Các chiều cung cấp mô tả rõ ràng bởi các đơn vị đo lường được chia ra của người dùng khối. Ví dụ: một khối cho phân tích bán hàng bao gồm các đơn vị đo lường 'Item_Sale_Price' và 'Item_Cost' từ bảng 'Sales_Fact' và các chiều 'Store_Location', 'Product_Line' và 'Fiscal_Year'. Khối này cho phép người dùng phân chia 'Item_Sale_Price' và 'Item_Cost' thành các loại khác nhau bởi 'Store_Location', 'Product_Line' và 'Fiscal_Year'.

Mỗi chiều có thể chứa một hệ thống các cấp độ để chỉ sự phân chia rõ ràng của người dùng. Ví dụ: Chiều 'Store_Location' có thể gồm hệ thống các cấp độ 'Continent', 'Country', 'Region', 'State_Province', 'City' và 'Store_Number'. Mỗi cấp độ trong chiều lại chi tiết hơn mức cha của nó. Ví dụ: 'Continent' chứa 'Country', 'State_Province' chứa 'City'. Tương tự, hệ thống chiều thời gian 'Time' có thể gồm có các cấp độ 'Year', 'Quarter', 'Month' và 'Day'.

Các cấp độ chiều là một công cụ mẫu dữ liệu mạnh bởi vì chúng cho

phép người dùng đưa ra yêu cầu về các vấn đề ở mức độ cao và sau đó mở rộng ra một hệ thống chiều để phát hiện thêm chi tiết. Ví dụ: một nhà phân tích có thể bắt đầu bằng việc yêu cầu xem các giá trị 'Fiscal_Year' của các kết quả ba năm tài chính về trước. Việc phân tích có thể thông báo chi phí năm này cao hơn so với các năm khác. Mở rộng chiều 'Fiscal_Year' tới mức 'Month' thì sự phân tích cho thấy chi phí sản phẩm đặc biệt cao trong tháng nào đó. Sau đó nhà phân tích có thể khảo sát kỹ các cấp độ của chiều 'Store_Location' để thấy một lĩnh vực đặc biệt góp phần đáng kể làm chi phí sản phẩm cao hoặc mở rộng chiều 'Product_Line' để thấy 'Item_Cost' cao đối với một nhóm sản phẩm hoặc một sản phẩm đặc biệt. Kiểu khảo sát này được biết đến như là Drill_down và nó phổ biến trong các ứng dụng OLAP.

Mặc dù khối vừa đề xuất có ba chiều nhưng một khối có thể có tới 64 chiều. Dữ liệu khối và các liên kết (Aggregation) có thể được lưu trữ dưới nhiều phương thức. Các liên kết là các bản dữ liệu sơ lược được tính toán trước, nó cung cấp cơ chế cho việc đáp ứng nhanh yêu cầu trong các hệ thống OLAP.

Các khối có thể đòi hỏi không gian lưu trữ đáng kể để chứa dữ liệu và thông tin sơ lược được tính toán trước trong các cấu trúc đa chiều. Nhân tố tác động đến các yêu cầu lưu trữ là không đáng kể (số lượng các ô trống trong một khối). Ví dụ: nếu một chiều có chứa các mô tả việc bán hàng và một chiều khác chứa các miền, các ô tại điểm giao nhau giữa biểu diễn bán hàng miền Bắc và miền Nam có thể là rỗng.

Các lựa chọn lưu trữ cho phép ta chọn các phương thức và các vị trí lưu trữ thích hợp cho dữ liệu khối. Ta có thể tạo một chiến lược lưu trữ OLAP đáp ứng theo các nhu cầu của ta.

3.3.2.2. Xử lý các khối

Khi ta xử lý một khối thì các khối liên kết đã thiết kế của nó được tính

toán và được Load cùng với khối và dữ liệu. Quá trình xử lý một khối bao gồm việc đọc các bảng Dimensions để xác định các cấp độ dữ liệu hiện tại, đọc bảng Fact, tính toán các liên kết đặc biệt và lưu trữ các kết quả trong khối. Sau khi một khối được xử lý, nó được cung cấp cho yêu cầu của người dùng.

Xử lý là thuật ngữ được dùng chỉ sự tải trọn vẹn dữ liệu của khối. Tất cả các chiều, dữ liệu bảng Fact được đọc và tất cả các khối liên kết đặc biệt được tính toán. Ta phải xử lý một khối khi cấu trúc của nó còn mới hoặc các chiều của nó hay các đơn vị đo lường đã được chọn lọc. Việc xử lý một khối có thể lấy đi một số thời gian thực nếu có một bảng Fact lớn, có nhiều chiều với nhiều cấp độ và nhiều khoản mục trong mỗi cấp độ. Việc tải thông tin chiều là không cần thiết nếu ta dùng các chiều dùng chung đã được xử lý trong các khối.

Các thay đổi trong sơ đồ kho chứa dữ liệu mà ảnh hưởng đến cấu trúc các khối đòi hỏi các khối này có sự thay đổi cấu trúc và sau đó được xử lý. Các thay đổi hoặc các bổ sung vào dữ liệu trong kho chứa dữ liệu không đòi hỏi các khối phải được xử lý hoàn toàn. Như vậy những sự thay đổi có thể được kết hợp trong các khối hiện có sử dụng các lựa chọn xử lý cập nhật gia tăng hoặc làm tươi dữ liệu, phụ thuộc vào cách thay đổi dữ liệu.

3.3.2.3. Khối ảo (Virtual Cube)

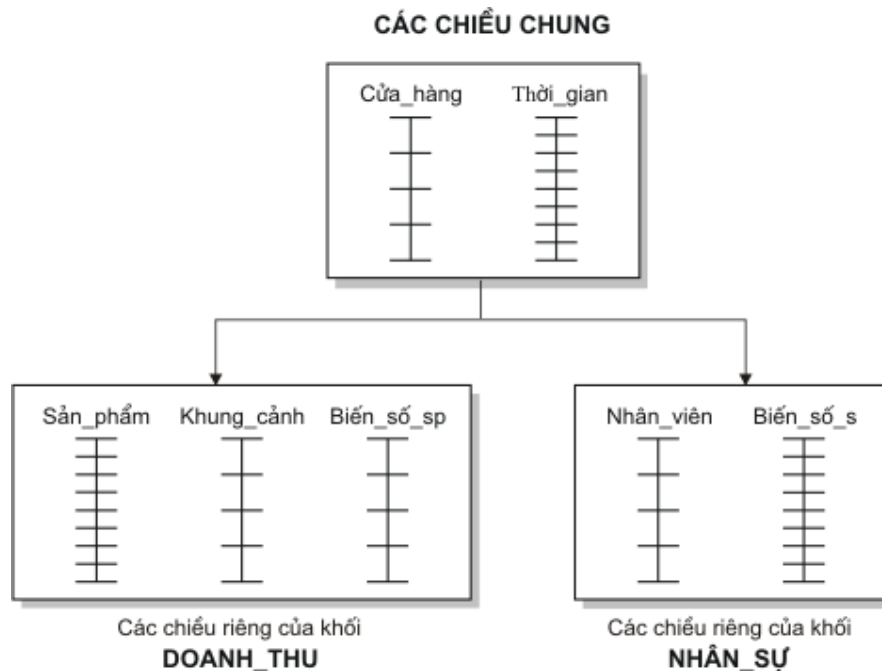
Ta có thể liên kết các khối trong khối ảo giống như các bảng có thể được liên kết với các khung nhìn trong một cơ sở dữ liệu quan hệ. Một khối ảo cung cấp truy cập tới dữ liệu trong các khối kết hợp mà không đòi hỏi xây dựng một khối mới, nó cho phép ta duy trì thiết kế tốt nhất cho mỗi khối riêng biệt.

3.3.3 Chiều (Dimension)

Các chiều là cách mô tả chủng loại mà theo đó các dữ liệu số trong khối

được phân chia để phân tích. Ví dụ: nếu một đơn vị đo lường của khối là tổng số sản phẩm (Production Count) và các chiều của nó là thời gian, nơi sản xuất, sản phẩm (Time, Factory Location, Product) thì người dùng khối có thể phân chia tổng số sản phẩm theo thời gian, nơi sản xuất, sản phẩm (Time, Factory Location, Product).

Một chiều có thể được dùng bởi nhiều khối khác và được gọi là một chiều dùng chung. Nói chung, các khối cần chia sẻ một hay nhiều hơn các chiều. Ví dụ như ta có hai khối: 'DOANH_THU' và 'NHÂN_SỰ'. Hai khối này chia sẻ hai chiều chung: 'Cửa_hàng' và 'Thời_gian'. Ngoài ra khối 'DOANH_THU' có thêm các chiều: 'Sản_phẩm', 'Khung_cảnh' và 'Biến_số_sp'. Khối 'NHÂN_SỰ' có thêm các chiều: 'Nhân_viên' và 'Biến_số_s'.



Hình 3.5. Sơ đồ mô hình đa khối

Các chiều chia sẻ có thể được dùng trong bất cứ khối nào của cơ sở dữ liệu. Bằng việc tạo ra các chiều chia sẻ và dùng chúng trong đa khối, ta tránh

được việc tạo ra các chiều cục bộ giống hệt nhau trong mỗi chiều thuộc các khối.

Các chiều chia sẻ cũng cho phép tiêu chuẩn hoá trong số các khối. Ví dụ: các khối chia sẻ chuẩn cho thời gian và vị trí địa lý đảm bảo rằng dữ liệu được phân tích từ các khối khác nhau sẽ được tổ chức tương tự nhau. Ta cũng có thể tạo một loại chiều khác được biết đến như là chiều ảo.

3.3.3.1. Xác định các chiều

Khi xác định một chiều, ta chọn một hoặc nhiều cột của một trong các bảng liên kết (bảng chiều). Nếu ta chọn các cột phức tạp thì tất cả cần có quan hệ với nhau, chẳng hạn các giá trị của chúng có thể được tổ chức theo hệ thống phân cấp đơn. Để xác định hệ thống phân cấp, sắp xếp các cột từ chung nhất tới cụ thể nhất. Ví dụ: một chiều ‘Thời gian’ (Time) được tạo ra từ các cột ‘Năm’, ‘Quý’, ‘Tháng’, ‘Ngày’ (Year, Quarter, Month, Day).

Mỗi cột trong chiều góp phần vào một cấp độ cho chiều. Các cấp độ được sắp đặt theo nét riêng biệt và được tổ chức trong hệ thống cấp bậc mà nó thừa nhận các cách hợp Logic cho việc đào sâu (Drill_down). Ví dụ: chiều ‘Thời gian’ được miêu tả ở trên cho phép người dùng khối đào sâu (Drill_down) từ ‘Năm’ tới ‘Quý’, từ ‘Quý’ tới ‘Tháng’ và từ ‘Tháng’ tới ‘Ngày’. Mỗi Drill_down cung cấp nét đặc trưng hơn.

Mỗi cấp độ có chứa các thành phần. Các thành phần là các giá trị trong cột xác định cấp độ. Ví dụ: cấp độ ‘Quý’ có thể gồm 4 thành phần: ‘Quý I’, ‘Quý II’, ‘Quý III’ và ‘Quý IV’. Tuy nhiên, nếu dữ liệu trong bảng kéo dài hơn một năm, ví dụ cấp độ ‘Năm’ chứa 3 giá trị khác nhau: ‘1996’, ‘1997’ và ‘1998’ thì cấp độ ‘Quý’ sẽ gồm 12 thành phần.

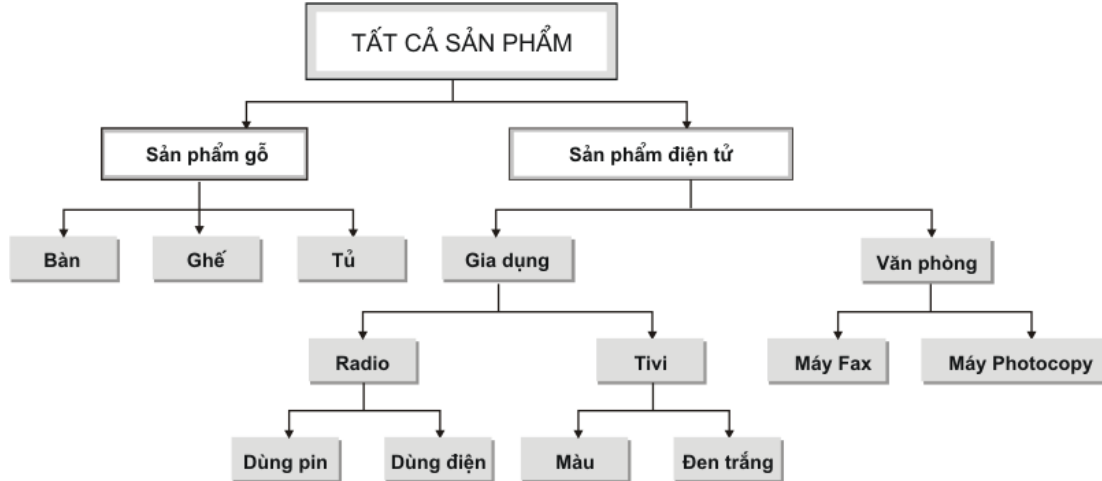
3.3.3.2. Chiều có phân cấp

Phân cấp là cột sống của việc gộp dữ liệu hay nói một cách khác là dựa

vào các phân cấp mà việc gộp dữ liệu mới có thể thực hiện được. Phần lớn các chiều đều có một cấu trúc đa mức hay phân cấp. Nếu chúng ta làm những quyết định về giá sản phẩm để tối đa doanh thu thì chúng ta cần quan sát ở những dữ liệu về doanh thu sản phẩm được gộp theo giá sản phẩm, tức là chúng ta đã thực hiện một cách gộp. Khi cần làm những quyết định khác thì chúng ta cần thực hiện những phép gộp tương ứng khác. Như vậy có thể có quá nhiều tiến trình gộp nên các tiến trình gộp này cần phải được thực hiện một cách rất dễ dàng, linh hoạt để có thể hỗ trợ những phân tích không hoạch định trước. Điều này có thể được giải quyết trên cơ sở có sự trợ giúp của những phân cấp rộng và sâu.

3.3.3.3. Phân cấp chiều

Xét ví dụ về một phân cấp chiều qua hình vẽ sau:

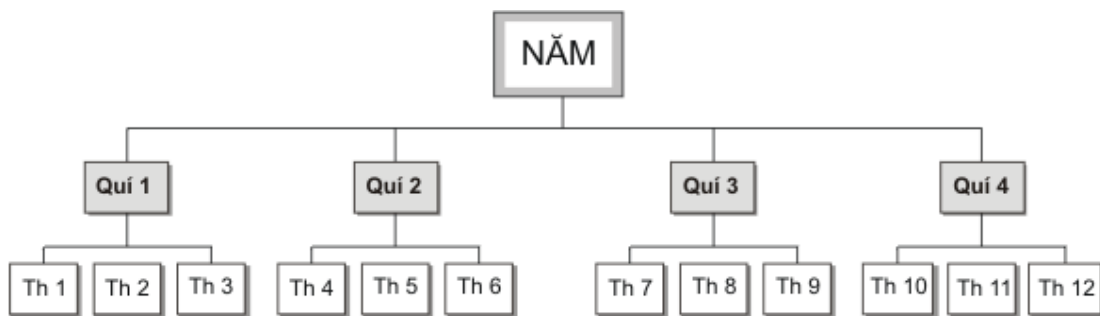


Hình 3.6. Phân cấp chiều Sản_phẩm

Các tham chiếu đến các phần tử trong các ứng dụng đa chiều thường liên quan đến một vài phần tử khác. Tham chiếu liên quan trong một cấu trúc phân cấp thì phức tạp hơn tham chiếu liên quan trong cấu trúc dòng và cột.

Cấu trúc phân cấp thường quan tâm đến hướng mà chúng ta đếm. Ví dụ như khi chúng ta muốn tham chiếu đến tất cả các phần tử có cùng cự ly với ‘Gia dụng’ khi đếm từ gốc thì tập các phần tử này sẽ gồm: ‘Bàn’, ‘Ghế’, ‘Tủ’, ‘Gia dụng’, ‘Văn phòng’ (cùng là hai mức đếm từ gốc).

Phân cấp chiều như trên gọi là phân cấp bất đối xứng. Phân cấp như trong hình sau gọi là phân cấp đối xứng:



Hình 3.7. Cây phân cấp đối xứng

Trong phân cấp đối xứng chúng ta có thể tham khảo đến các phần tử theo mức của nó. Như vậy các ‘Quý’ là một tập hợp các phần tử một mức từ dưới lên và một mức từ trên xuống.

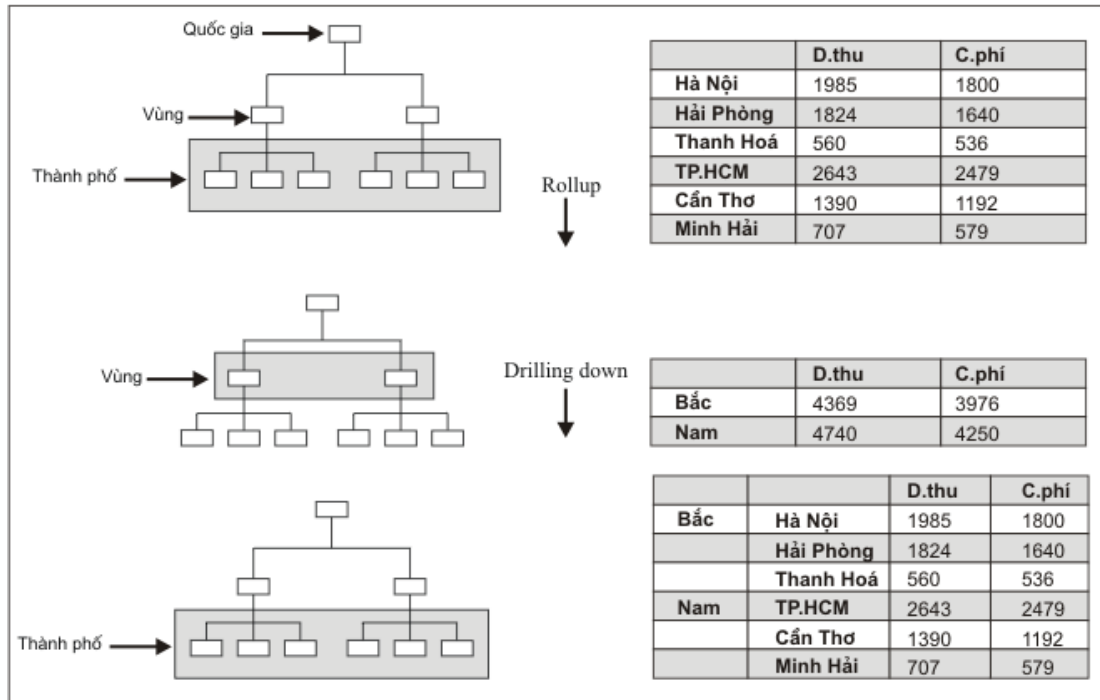
3.3.3.4. *Roll_up* và *Drill_down* dựa trên phân cấp chiều

Dựa trên phân cấp theo chiều, từ một mức dưới chúng ta có thể cuộn lên (*Roll_up*) các mức trên, thực hiện một phép gộp để có được kết quả tổng hợp hơn và từ một mức trên có thể khoan sâu xuống (*Drill_down*) các mức dưới để có các kết quả chi tiết hơn (xem ví dụ hình 3.8).

3.3.3.5. *Các chiều ảo (Virtual Dimensions)*

Chiều ảo là một kiểu đặc biệt, nó ánh xạ các thuộc tính của các thành phần trong các chiều khác vào trong một chiều mà sau đó có thể được dùng ở trong

các khối. Các chiều ảo và thuộc tính thành phần được đánh giá là cần thiết cho các yêu cầu và chúng không đòi hỏi lưu trữ khối vật lý.



Hình 3.8. Roll_up và Drill_down theo phân cấp chiều

3.3.4. Các đơn vị đo lường (Measures)

Các đơn vị đo của khối là các cột trong bảng Fact. Các đơn vị đo lường xác định những giá trị số từ bảng Fact được tổng hợp phân tích như định giá, trị giá hoặc số lượng.

3.3.5. Các phân hoạch (Partitions)

Tất cả các khối đều có tối thiểu một phân hoạch để chứa dữ liệu của nó. Một phân hoạch đơn được tự động tạo ra khi khối được định nghĩa. Khi ta tạo một phân hoạch mới cho một khối, phân hoạch mới này được thêm vào trong tập hợp các phân hoạch đã tồn tại đối với khối. Khối phản ánh dữ liệu đã được

kết nối có trong tất cả các phân hoạch của nó. Một bảng phân hoạch của khối là vô hình đối với người dùng.

Các phân hoạch tiêu biểu cho một công cụ mạnh, mềm dẻo cho việc quản trị các khối OLAP, đặc biệt các khối lớn. Ví dụ: một khối chứa thông tin thương mại có thể chứa trong một hoặc nhiều phân hoạch cho dữ liệu của những năm trước và các phân hoạch cho mỗi quý của năm hiện tại. Cuối năm các bảng phân hoạch của bốn quý có thể được hợp nhất trong một phân hoạch đơn cho năm đó. Các bảng phân hoạch có thể được lưu trữ với các sự lựa chọn kết hợp khác nhau theo phương thức lưu trữ, định vị dữ liệu nguồn và thiết kế kết hợp. Tính mềm dẻo này cho phép ta thiết kế các chiến lược lưu trữ khối thích hợp với các yêu cầu.

Các bảng phân hoạch phải được thiết kế và quản lý phù hợp để tránh các kết quả mâu thuẫn hay sai lệch. Tính toàn vẹn của dữ liệu khối dựa vào dữ liệu được phân bố giữa các phân hoạch của khối vì thế dữ liệu không bị lặp lại giữa các phân hoạch. Khi dữ liệu được tổng kết từ các bảng phân hoạch, bất kỳ một thành phần dữ liệu nào có trong một phân hoạch sẽ được tổng kết như thể chúng là các thành phần dữ liệu khác nhau. Điều này có thể đưa ra các bản tổng kết không chính xác và dữ liệu sai cho người dùng. Ví dụ, nếu công việc kinh doanh thương mại cho sản phẩm X được lặp lại trong các bảng Fact cho hai phân hoạch, các tổng kết của việc mua bán sản phẩm X có thể bao gồm việc tính toán hai lần.

Các phân hoạch có thể được hợp nhất, ta có thể dùng tính năng này trong toàn bộ chiến lược lưu trữ và cập nhật dữ liệu. Các phân hoạch chỉ được hợp nhất nếu chúng có cùng chế độ lưu trữ và các khối tập hợp. Để tạo các phân hoạch dành cho việc hợp nhất về sau, ta có thể lựa chọn chế độ lưu trữ và sao chép các khối kết hợp từ một phân hoạch khác khi ta tạo phân hoạch. Ta cũng có thể sửa đổi một phân hoạch sau khi nó được tạo ra và sao chép các

khối kết hợp từ phân hoạch khác. Việc hợp nhất các phân hoạch cũng phải được thực hiện một cách cẩn thận để tránh sự lặp lại của dữ liệu trong phân hoạch kết hợp, nó có thể làm cho dữ liệu khối bị lỗi.

Khi đang tạo hoặc hợp nhất các phân hoạch, cần thực hiện các thao tác bằng tay hoặc tạo các bộ lọc thích hợp để đảm bảo các phân hoạch của khối luôn luôn chứa dữ liệu chính xác.

3.3.6. Các phương pháp lưu trữ dữ liệu (MOLAP, ROLAP, HOLAP)

3.3.6.1. MOLAP (Multidimensional OLAP)

Dữ liệu cơ bản của khối được lưu trữ cùng với dữ liệu kết hợp (Aggregation) trong cấu trúc đa chiều hiệu suất cao. Cách tiếp cận này kết hợp kho dữ liệu đa chiều và các dịch vụ của OLAP trên cùng một Server. MOLAP là một cấu trúc tối ưu cho việc lưu trữ các sự kiện đã phân loại và cùng với nó là các chiều. Dữ liệu được tổ chức theo khung nhìn dữ liệu và được lưu trữ trong một biểu mẫu được kết hợp và tổng hợp. Tập Index nhỏ hơn khiến cho việc trả lời những truy vấn phức tạp rất nhanh. Vì dữ liệu được lưu trữ trong các mảng, việc cập nhật các giá trị không ảnh hưởng nhiều tới tập chỉ số. Điều này khiến cho việc cài đặt những ứng dụng cập nhật hoặc đọc-ghi như dự báo và điều chỉnh trở nên dễ dàng.

MOLAP là sự lựa chọn tốt nhất cho những ứng dụng có đặc điểm:

- Yêu cầu tốc độ truy vấn cao.
- Có khả năng phân tích dữ liệu phức hợp. MOLAP cung cấp môi trường phân tích mạnh hơn ROLAP.
- Dễ sử dụng: bởi dữ liệu đã được tổng hợp từ trước và được lưu trong kho dữ liệu đa chiều. Tất cả những gì người sử dụng cần làm là xác định các chiều và các nhóm nằm trong các chiều đó. Trong khi đó ROLAP lại yêu cầu người sử dụng phải hiểu được sự ánh xạ tới các

CSDL tác nghiệp.

3.3.6.2. ROLAP (Relational OLAP)

Dữ liệu cơ bản của khối được lưu trữ cùng với dữ liệu kết hợp (Aggregation) trong cơ sở dữ liệu quan hệ. Phương pháp tiếp cận này bao gồm các dịch vụ của OLAP và cơ sở dữ liệu quan hệ. Các dữ liệu được lưu trữ trong những bảng quan hệ và có thể có kích thước hàng trăm Gigabyte. Những hệ ROLAP cung cấp các Engine truy vấn cực kỳ linh động bằng việc “chuẩn bị sẵn sàng” tất cả dữ liệu tác nghiệp cho người sử dụng đầu cuối, dễ dàng trích và tổng hợp dữ liệu theo yêu cầu. Những công cụ ROLAP có thể trích dữ liệu từ rất nhiều nguồn CSDL quan hệ khác nhau.

ROLAP là sự lựa chọn cho kho dữ liệu có những đặc điểm sau:

- Dữ liệu thường xuyên thay đổi: trong một kho dữ liệu hay biến động và người sử dụng lại đòi hỏi những tổng hợp gần như tức thời, ROLAP sẽ là sự lựa chọn duy nhất. MOLAP phải trích lấy và tổng hợp dữ liệu ngoại tuyến (Offline), hơn nữa hầu hết các cơ sở dữ liệu đa chiều đều yêu cầu tính toán lại toàn bộ CSDL khi một chiều được thêm vào, khi một lược đồ tổng hợp thay đổi hoặc khi dữ liệu mới được thêm vào. Những đặc điểm này khiến cho MOLAP không thích hợp với những hệ hỗ trợ quyết định mà nguồn dữ liệu thường xuyên biến động.
- Khối lượng dữ liệu lớn: Đối với những kho dữ liệu có độ lớn cỡ Terabyte, MOLAP đòi hỏi việc tính toán trước dữ liệu với hàng trăm Terabyte không gian lưu trữ.
- Các dạng truy vấn không được biết trước: ROLAP cho phép truy vấn và tổng hợp từ bất kỳ nguồn dữ liệu tác nghiệp nào. Tuy nhiên khả năng này lại dẫn tới sự phức tạp khi sử dụng, trong việc ánh xạ tới các nguồn dữ liệu tác nghiệp.

3.3.6.3. *HOLAP (Hybrid OLAP)*

Là kết hợp hai phương pháp MOLAP và ROLAP. Dữ liệu cơ bản của khối được lưu trữ trong cơ sở dữ liệu quan hệ và dữ liệu kết hợp (Aggregation) được lưu trữ trong cấu trúc đa chiều hiệu suất cao. Lưu trữ HOLAP đưa ra những lợi ích của MOLAP cho việc liên kết mà không cần thiết một bản sao chính xác từ dữ liệu chi tiết.

3.4. Thuật toán chỉ số hoá các khung nhìn trong xử lý phân tích trực tuyến kho dữ liệu

Có hai cách thường được sử dụng để truy nhập trực tiếp vào kho dữ liệu. Cách thứ nhất thông qua các khung nhìn (View) nhiều chiều và thể hiện nó như là cấu trúc nhiều chiều phục vụ cho việc phân tích và lập báo cáo ở các trạm làm việc. Để thực hiện hiệu quả xử lý phân tích trực tuyến trên các khung nhìn dữ liệu, người ta thường tập trung xây dựng các thuật toán để chọn tự động các bảng tổng hợp và chỉ số hóa các khung nhìn. Cách thứ hai là phân tích trực tiếp các khối dữ liệu nhiều chiều được tạo lập từ các kho dữ liệu và tạo ra khả năng tổng hợp, gộp chung, hỗ trợ cho việc ra quyết định về dự báo, phân tích xu thế phát triển và phân tích thống kê.

Trong luận văn này tôi xin giới thiệu thuật toán chọn tự động các Subcubes và các chỉ số tương ứng để xử lý trước sao cho hợp lý nhất.

Xét ví dụ (1), khi quan sát kho dữ liệu quản lý các thông tin kinh doanh từ các cửa hàng của một tổng công ty, người ta nhận thấy những câu hỏi cần xử lý OLAP thường có dạng:

- Số các Mặt_hàng bán ra hàng tuần của mỗi Cửa_hàng?
- Số lượng bán ra của từng Mặt_hàng là bao nhiêu?

Để trả lời cho được những câu hỏi trên thì các chương trình ứng dụng OLAP phải nhìn vào kho dữ liệu theo nhiều chiều (phương diện) khác nhau.

Ở ví dụ trên, các thuộc tính xác định chiều là Cửa_hàng và Mặt_hàng. Đơn vị của chiều mà chúng ta quan tâm nhiều nhất ở đây là: số hàng bán ra. Hệ thống xử lý OLAP cần biểu diễn dữ liệu cho người sử dụng các View nhiều chiều ở dạng hình khối (Data Cube). Trong ví dụ trên, Data Cube sẽ bao gồm 4 Subcube như sau:

- Số lượng bán ra của mỗi Mặt_hàng ở từng cửa_hàng,
- Số lượng bán ra của mỗi Mặt_hàng ở tất cả các Cửa_hàng,
- Số lượng bán ra các Mặt_hàng trong từng Cửa_hàng,
- Số lượng bán ra các Mặt_hàng ở tất cả các Cửa_hàng.

3.4.1. Một số khái niệm cơ bản

3.4.1.1. Các khối dữ liệu con (Subcubes)

Subcube là một bộ phận của khối dữ liệu (Data Cube). Nói cách khác, mỗi phần tử của tập các tập con của các chiều kho dữ liệu sẽ là một Subcube. Xét tiếp ví dụ (1) ở trên, mỗi cặp {Mặt_hàng, Khách_hàng} sẽ tương ứng với một Subcube chứa Mặt_hàng bán ra cho từng Khách_hàng. Trong SQL các Subcube chỉ khác nhau bởi câu lệnh gộp (Groupby Clause). Ở đây chúng ta cũng cho Subcube tương ứng với một tập các thuộc tính có thể gộp được với nhau. Như vậy {Mặt_hàng, Khách_hàng} sẽ tương ứng với một Subcube được xác định bởi câu lệnh trong SQL như sau:

```
SELECT Mặt_hàng, Khách_hàng, SUM(Hàng_bán) AS TotalSales
FROM R
GROUP BY Mặt_hàng, Khách_hàng
```

3.4.1.2. Câu truy vấn (Queries)

Mỗi câu truy vấn có thể sử dụng chiều như là thuộc tính để lựa chọn (trong SQL chiều là thuộc tính trong Groupby Clause - câu lệnh gộp lại hoặc tương ứng với Where Clause - câu lệnh mà ở đó thỏa mãn điều kiện nào đó).

Sử dụng cách viết rút gọn của mô hình, ta có thể viết câu truy vấn Q dưới dạng: $\gamma\delta p s$ trong đó γ xác định những thuộc tính gộp lại (Groupby); δ xác định các thuộc tính chọn để tập hợp lại (Selection) của từng câu hỏi; c: Khách_hàng (customer); p: Mặt_hàng (part) và s: Hàng_bán (sales). Tất nhiên thứ tự các thuộc tính là không quan trọng, câu truy vấn $\gamma p \delta s c$ cũng hoàn toàn giống như $\gamma p \delta c s$.

Mỗi câu truy vấn dạng $\gamma c(\delta p = \text{constant}(\mathbf{R}))$ là yêu cầu về lát cắt thông qua Subcube(customer, part). Ta qui định câu truy vấn tổng quát: $\gamma c \delta p$ và gọi nó là câu truy vấn về lát cắt (Slice Query) đối với Subcube(customer, part). Dạng tổng quát $\gamma G_1, \dots, G_k \delta$ cho Subcube($G_1, \dots, G_k, S_1, \dots, S_l$) là những Subcube nhỏ nhất tham gia trả lời cho câu hỏi trên với k và l là những thứ nguyên của kho dữ liệu.

3.4.1.3. Chỉ số (Indexes)

Để tăng tốc độ xử lý các câu truy vấn, ta có thể sử dụng cấu trúc chỉ số B-cây (B-Tree: Balance-Tree). Ví dụ đối với Subcube(p,s), ta có thể xây dựng đánh chỉ số như sau:

- Ips: Tìm những chỉ số mà nó được ghép lại từ chiều p (part) với chiều s (sales).
- Isp: Tìm những chỉ số mà nó được ghép lại từ hai chiều s và p

Ở đây thứ tự các chiều là quan trọng. Cho trước một giá trị của p, ta có thể sử dụng Ips để tìm tất cả các hàng trong Subcube(p,s) mà nó có giá trị p. Tương tự, cho trước cặp (p,s) ta sử dụng Isp để tìm trong Subcube(p,s) những hàng, cột có cặp giá trị đó.

Sử dụng chỉ số B-cây sẽ giúp rút ngắn được thời gian trả lời cho các câu truy vấn. Đối với mỗi View ta có một số cách chỉ số hóa. Ví dụ với Subcube(p,s) ta có thể xây dựng 4 cách đánh chỉ số như sau: Ip(ps), Is(ps),

Ips(ps), Isp(ps).

Trong mỗi trường hợp ta liệt kê các thuộc tính khóa tìm kiếm như là chỉ số với Subcube(p,s) mà trong đó Index được xây dựng. Mỗi tập con các thuộc tính của một quan sát View, ta có thể xác định một chỉ số theo một thứ tự nào đó. Như vậy các chỉ số có thể của một khung nhìn View với m thuộc tính là:

$$\sum_{r=0}^m \binom{m}{r} r!$$

Như vậy, số các chỉ số là quá lớn. Nói chung, chỉ số có thể hỗ trợ trả lời các câu truy vấn. Để xử lý dữ liệu nhanh, chính xác thì phải xử lý trước về các lát cắt khi một tiền tố (Prefix) của các thuộc tính được chỉ số hóa tương ứng những thuộc tính lựa chọn (Selection Attribute) trong câu truy vấn hay chỉ số hóa các khung nhìn vào kho dữ liệu.

3.4.1.4. Quan hệ tính toán và phụ thuộc

Giữa các câu truy vấn (Queries) và các khung nhìn (Views), ta định nghĩa quan hệ tính toán << như sau:

Với mỗi câu hỏi Q và khung nhìn V:

Q << V nếu kết quả của Q (câu trả lời) tính toán được mà chỉ sử dụng các bộ (Tuples) có trong V. Q được trả lời trong V.

Xét ví dụ (2): Xét câu truy vấn Q1 = $\gamma c \delta s$ từ kho dữ liệu ở ví dụ (1). Câu truy vấn này có thể tính toán trong View V1 = sc và cũng có thể trong View V2 = psc. Vậy: Q1 << V1 và Q1 << V2.

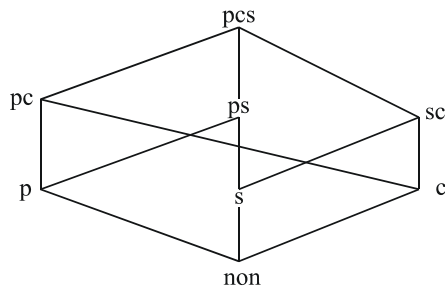
Nhưng V1 sẽ không có câu trả lời trong V3 = pc. Ta định nghĩa quan hệ thứ tự bộ phận $\leq$ đối với các View như sau:

V1 $\leq$ V2 khi và chỉ khi tập các thuộc tính của V1 là tập con của tập các thuộc tính của V2. V1 hẹp hơn V2.

Theo định nghĩa thì tất nhiên ta có part $\leq$ part và part $\leq$ (part,

customer), nhưng $\text{part} \leq \text{customer}$ và ngược lại cũng vậy.

Tập hợp các Subcube của Data Cube tạo ra các xác định trên $\leq$ sẽ được gọi là quan hệ phụ thuộc của các View. Ví dụ các Subcube của Data Cube đã nêu ở ví dụ (1) cùng với quan hệ $\leq$ tạo thành các Subcube của một CSDL như sau:



trong đó ‘non’ là tập rỗng

Chúng ta dễ dàng nhận thấy nếu $V1 \leq V2$ và $Q1 \ll V1$ thì $Q1 \ll V2$, nghĩa là tồn tại quan hệ phụ thuộc giữa $V1$ với $V2$ và quan hệ tính toán giữa $Q1$ với các View đó. Có thể sử dụng đồ thị để mô tả cho quan hệ tính toán được định nghĩa ở trên, trong đó tập đỉnh là tất cả các câu truy vấn và các View của một Data Cube. Nếu câu truy vấn Q tính toán được trong View V thì ta vẽ cạnh nối Q với V và bổ sung cạnh đó vào trong E . Mỗi cạnh (Q, V) lại có thể có trọng số f chính là phí tổn (thường là thời gian) để trả lời Q trong View V .

Vấn đề là ta tìm cách xây dựng cách đánh chỉ số trên khung nhìn V để sao cho có được câu trả lời nhanh hơn khi phân tích dữ liệu trực tuyến. Chỉ số đó phải không ảnh hưởng đến kết quả tính toán trong mọi cách, nhưng nó làm tăng tốc độ tính toán để xác định câu trả lời.

Để tiện lợi cho ký hiệu và xử lý, ta có thể gộp chỉ số i vào cùng với phí tổn phải mất để trả lời câu truy vấn Q khi sử dụng V thành một cặp nhân cho

mỗi cạnh (Q,V) . Từ những phân tích trên, ta có thể tổng quát hóa công thức đánh giá phí tổn những câu trả lời cho câu truy vấn Q khi sử dụng View V và chỉ số J .

Giả thiết Q là câu truy vấn $\gamma A \delta B$ trong đó A và B là các tập về chiều. $B = \emptyset$ khi và chỉ khi Q là câu truy vấn về Subcube và $B = \emptyset$ nghĩa là đề cập đến tất cả các chiều.

Giả thiết V là View C . Nếu $Q \ll V$ thì $A \cup B \subseteq C$, $C = \emptyset$ ký hiệu cho View rỗng (phần tử nhỏ nhất trong các View).

Giả thiết J là chỉ số $ID(V)$ và D là một thứ tự của các thuộc tính. $D = \langle \rangle$ (dãy trống) ký hiệu trường hợp không sử dụng chỉ số.

Ký hiệu E là tập con lớn nhất của B sao cho các thuộc tính của E tạo thành tiền tố (Prefix) của D . Phí tổn của câu trả lời cho Q khi sử dụng V có sử dụng chỉ số J sẽ được xác định như sau:

$$C(Q,V,J) = \frac{|(C)|}{|(E)|} \quad (*)$$

trong đó $|C|$ và $|E|$ ký hiệu số các dòng của các bảng dữ liệu tương ứng với View C và E

Công thức trên luôn tính được trong mọi trường hợp. Trường hợp $E = \emptyset$, nghĩa là không có chỉ số được đánh với V hoặc Q là câu truy vấn về Subcube, khi đó $|\emptyset| = 1$ do vậy $C(Q,V,J) = |V|$, nghĩa là chúng ta phải xử lý tất cả các dòng trong V mới có được câu trả lời cho Q .

Chúng ta hãy xét ví dụ về CSDL quản lý hàng bán ra với View $V = (p,s,c)$ bao gồm 6 triệu dòng, câu hỏi $Q = \gamma c \delta ps$ và chỉ số $J = Iscp$ trên Subcube (p,s,c) . Khi đó $C = (p,s,c)$ và $E = (s)$ bởi vì tập con lớn nhất tạo thành tiền tố của s là s . Giả thiết bảng s có 0.01 triệu dòng. Theo công thức thì:

$$C(Q, V, J) = \frac{6tr.}{0,01tr.}$$

3.4.2. Thuật toán chọn View và Index

Để thực hiện thuật toán ta cần biết cụ thể những thông tin sau:

- Kích thước của các View,
- Kích thước của từng chỉ số,
- Với mỗi bộ 3 (Query, View, Index), phí tổn của câu trả lời $C(Q, V, Y)$ là bao nhiêu.

Ở trên ta đã đưa ra công thức (*) để tính $C(Q, V, J)$. Vấn đề còn lại ở đây là cần xác định cụ thể kích thước của từng View và từng Index. Đây không phải là vấn đề đơn giản vì kích thước của chúng thường rất lớn.

3.4.2.1. Ước tính kích thước của mỗi View

Có nhiều cách xác định kích thước của View mà không cần thiết phải cụ thể cụ thể hóa tất cả các View. Ta có thể sử dụng phương pháp phân tích và lấy mẫu để xác định kích thước của View từ nhiều View khác mà chúng ta chỉ cần cụ thể hóa phần tử V_1 lớn nhất (View chứa tất cả các chiều) trong các View. Với một View, nếu các thuộc tính nhóm lại mà độc lập tính thì ta có thể xác định theo phương pháp giải tích theo kích thước của View. Ngược lại có thể dùng mẫu V_1 để tính kích thước của các View khác. Kích thước của một View là số các giá trị khác nhau của các thuộc tính mà chúng được nhóm lại.

3.4.2.2. Ước tính kích thước của chỉ số Index

Cho trước kích thước của mỗi View, ta hãy tính kích thước của chỉ số Index tương ứng. Thông thường kích thước của View trong mô hình là số dòng trong View đó. Kích thước của Index (B-cây) là số các lá của B-cây theo cách đánh chỉ số Index. Mặt khác số các nút lá của B-cây cho một Index xấp xỉ số dòng của một View tương ứng. Vậy ta có kết luận: kích thước của

một chỉ số Index trên View V cũng là kích thước của bản thân View V .

Ta hãy xét hai Index $J_1 = IA(V)$ và $J_2 = TB(V)$ đối với cùng một View V . Nếu B là tiền tố thực sự của A thì $C(Q, V, J_1) \leq C(Q, V, J_2)$ với mọi câu truy vấn Q .

Mặt khác, kích thước của J_1 và J_2 là cùng xấp xỉ với kích thước của một View theo cùng một độ chính xác nên ta có thể bỏ J_2 mà chỉ xét J_1 . Như vậy là với từng View, ta chỉ cần chọn chỉ số dài nhất để xử lý, đó là những chỉ số mà các thuộc tính khóa tìm kiếm của nó không phải là tiền tố thực sự của các thuộc tính tìm kiếm của các Index khác trong cùng một View.

Nếu V là View (C) thì tập các chỉ số Index sẽ là $\{ID(V) \mid D \text{ là một hoán vị của } C\}$.

3.4.2.3. Xác định bài toán

Như ở trên đã nêu, nhiệm vụ của ta là xây dựng các thuật toán để chọn các View và Index cụ thể để trả lời cho những câu truy vấn đối với một Data Cube cho trước. Ta có thể phát biểu một cách không hình thức: cho trước một tập các View, mỗi View lại xác định một tập các Index và một tập các câu truy vấn mà hệ thống cần phải trả lời. Mục đích của ta là chọn View và Index trong số đó để có được câu trả lời cho các câu truy vấn với phí tổn thấp nhất với một điều kiện ràng buộc là tập các View và tập các Index không chiếm nhiều không gian hơn một không gian cho trước S , đây là bài toán NP_đầy_đủ. Do vậy để giải quyết được bài toán trên, ta phải xây dựng những thuật toán có tính Heuristic, nhưng phải đảm bảo thuật toán thực hiện hiệu quả.

Trước tiên chúng ta tiến hành hình thức hóa bài toán nêu trên. Xét đồ thị lưỡng phân, $G = (V \cup Q, E)$ được gọi là đồ thị câu truy vấn - khung nhìn (Query - View Graph), V là tập các View còn Q chứa các câu truy vấn. Với mỗi $v_i \in V$ xác định tương ứng một bộ (S_i, I_i) trong đó S_i là không gian mà v_i

chiếm và I_i là tập các chỉ số trên v_i .

Ký hiệu I_{ik} là chỉ số thứ k của v_i .

- Với mỗi $q_i \in Q$ xác định tương ứng phí tổn T_i trả lời cho câu truy vấn q_i .
- Mỗi cạnh (q_i, v_j) có nhãn được gán tương ứng là (k, t_{ijk}) , trong đó t_{ijk} là phí tổn câu trả lời cho câu truy vấn q_i sử dụng View v_j và chỉ số thứ k của nó. Khi $k = 0$, t_{ijk} là phí tổn của câu trả lời cho q_i mà chỉ sử dụng v_j .

Bài toán: Cho tập các View V và tập các câu hỏi Q , cần xác định $M \subseteq V$, tập các View và các chỉ số cụ thể sao cho không gian mà các View và các chỉ số đó chiếm không vượt quá S (không gian giới hạn) đồng thời với cách chọn M cũng đảm bảo được cực tiểu hóa phí tổn toàn bộ để có được câu trả lời cho câu truy vấn Q từ một trong các View của M .

Nghĩa là ta cần cực tiểu hóa đại lượng sau sao cho tổng không gian mà các cấu trúc được lựa chọn từ M nhỏ hơn S :

$$\tau(G, M) = \sum_{i=1}^{|Q|} \min(T_i, \min t_{ijk}) \quad (**)$$

$$V_j, I_{jk} \in M$$

Bài toán trên dễ dàng tổng quát hóa thành bài toán xác định các View và Index trong Data Cube.

3.4.2.4. Giải quyết bài toán

Trước tiên chúng ta hãy định nghĩa một số ký hiệu. C - tập bất kỳ các View và Index trong đồ thị G . $S(C)$ là không gian các cấu trúc chiếm trong C . $B(C, M)$ là sinh lợi của C so với M và: $B(C, M) = \tau(G, M) - \tau(G, M \cup C)$; $B(C, \emptyset)$ là sinh lợi tuyệt đối của C .

Về không gian, lợi của C so với M sẽ là $B(C, M) / S(C)$.

a. Thuật toán r - cấu trúc

Cho trước: Đồ thị câu hỏi - khung nhìn G

Không gian hạn chế S

BEGIN

$M = \emptyset$; /* M = tập các cấu trúc đã được chọn */

While ($S(M) < S$)

BEGIN

Tìm tất cả các tập View và Index của một trong các dạng sau:

$\{v_i, I_{ij1}, I_{ij2}, \dots, I_{ijp}\}$ sao cho $v_i \notin M, I_{ijl} \notin M$ với

$1 \leq l, 0 \leq p < r$

hoặc

$\{I_{ij}\}$ sao cho v_i ứng với $I_{ij} \in M$ và $I_{ij} \notin M$.

Chọn C là một trong số các tập trên mà sinh lợi về không gian so với M là cực đại.

Đặt $M = M \cup C$;

END while

Return M ;

END;

Thuật toán r - cấu trúc thực hiện trong một số bước mà mỗi bước thì chọn tập con của C chứa nhiều nhất r cấu trúc. C là tập hợp gồm:

- Một View và một số chỉ số tương ứng của nó hoặc
- Một chỉ số mà View đã được chọn ở bước trước.

Vấn đề chính của thuật toán là chọn C ở mỗi bước sao cho sinh lợi của nó so với M là cực đại.

Đánh giá thuật toán: Giả thiết có n View trong Data Cube và mỗi View có nhiều nhất 1 chỉ số. Khi đó thuật toán r - cấu trúc phải thực hiện ở mỗi bước cần tính toán sinh lợi của $n \cdot 1 + n \cdot (1/r - 1)$ tập hợp. Như vậy độ phức tạp của thuật toán 1 sẽ là $\theta(kmr)$ trong đó m là số cấu trúc cho trước của đồ thị G và k là số cấu trúc được chọn trong thuật toán, trường hợp xấu nhất là bằng S .

b. Thuật toán tổng quát

Cũng như trên, mỗi bước của thuật toán cần chọn một tập con C bao gồm:

- Một View và một số chỉ số được chọn không bị hạn chế về số lượng hoặc
- Một chỉ số mà View tương ứng đã được chọn ở bước trước.

Cần lưu ý là kích thước của C sẽ không bị giới hạn bởi r như thuật toán trên. Mỗi bước của thuật toán phải thực hiện hai phần:

- Với mỗi View v_i chúng ta xây dựng tập IG_i mà lúc đầu chỉ chứa v_i . Sau đó bổ sung thêm dần các chỉ số vào IG_i cho đến khi sinh lợi về không gian của IG_i , tập các cấu trúc đã được chọn đạt tới cực đại.
- Tiếp theo là chọn chỉ số Index mà sinh lợi về không gian của View tương ứng so với M đạt được cực đại.

So sánh sinh lợi trên với sinh lợi của C với M , cái nào tốt hơn thì bổ sung vào M .

Thuật toán này được mô tả hình thức như sau:

Cho trước: Đồ thị query - view G

Không gian khổng lồ S

BEGIN

$M = \emptyset$; /* M = tập các cấu trúc đã được chọn */

While ($S(M) < S$)

BEGIN

$C = \emptyset$; /* tập tốt nhất chứa View và một số Index cho thời điểm xét */

For $v_i \in M \cap V_i$

BEGIN

$IG = \{v_i\}$; /* IG = tập chứa v_i và các cấu trúc chọn ra */

While ($S(IG) < S$)

BEGIN

Chọn I_{ic} là chỉ số của v_i mà sinh lợi so với $(M \cup IG)$ là cực đại;

```
IG = IG  $\cup$  Iic;  
END while;  
If (B(IG,M)/S(IG) > B(C,M)/|C| or C =  $\emptyset$ ) then C = IG;  
END for  
For Iij mà vi  $\in$  M  
If (B(Iij,M)/S(Iij) > B(C,M)/S(C)) then C = {Iij};  
M = M  $\cup$  C;  
END while  
Return M;  
END;
```

Đánh giá thuật toán: Độ phức tạp của thuật toán là $\theta(k^2mr)$, trong đó m là tổng số cấu trúc của đồ thị G và k là số cấu trúc cực đại để hợp với không gian S, trường hợp xấu nhất là bằng S.

3.3.5 Kết luận

Việc thực hiện các câu hỏi theo OLAP phụ thuộc rất nhiều vào việc tạo lập bảng tổng hợp theo các View trong các kho dữ liệu. Để tăng hiệu quả xử lý các câu hỏi, chúng ta có thể sử dụng chỉ số hóa (Index) trên các khung nhìn (View). Hai thuật toán trên mô tả cách chọn các View (hay Subcube) và xác định chỉ số Index cần phải tính toán trước để tăng hiệu quả xử lý các câu hỏi OLAP đối với kho dữ liệu.

Chương IV. Hệ trợ giúp quyết định dựa vào dữ liệu

4.1. Hệ trợ giúp quyết định

4.1.1. Giới thiệu

Ngay từ những năm 60 của thế kỷ trước, việc sử dụng các phương tiện tin học để tổ chức và khai thác các CSDL đã được tập trung nghiên cứu phát triển. Kể từ đó rất nhiều CSDL đã được tổ chức, phát triển và khai thác ở mọi qui mô và ở khắp các lĩnh vực hoạt động của con người và xã hội. Nhiều hệ quản trị CSDL mạnh với các công cụ phong phú và thuận tiện đã giúp cho con người khai thác có hiệu quả các nguồn tài nguyên dữ liệu. Mô hình CSDL quan hệ và ngôn ngữ vấn đáp chuẩn (SQL) đã có vai trò hết sức quan trọng trong việc tổ chức và khai thác các CSDL đó.

Giai đoạn này là thời kỳ của kỹ thuật thu thập dữ liệu, tiếp đó là thời kỳ của kỹ thuật truy nhập dữ liệu với những ứng dụng tập trung xử lý dữ liệu và thông tin theo các thủ tục có cấu trúc nhằm hỗ trợ điều khiển, dự báo và giám sát công việc. Đầu thập kỷ 70 của thế kỷ trước một loại hình ứng dụng mới ra đời, đó là Hệ trợ giúp quyết định (DSS) nhằm mục đích hỗ trợ các nhà quản lý cấp cao và ra quyết định điều hành.

Khái niệm Hệ trợ giúp quyết định được Scott Morton đưa ra đầu những năm 70 với thuật ngữ Hệ thống hỗ trợ quản lý (MSS). Hệ thống được xác định như sau “Hệ thống dựa trên sự tương tác máy tính, giúp người ra quyết định dùng các dữ liệu và mô hình để giải các bài toán không có cấu trúc - những bài toán mờ, phức tạp với lời giải không hoàn chỉnh”. Theo Gorry và Scott Morton, các vấn đề xử lý có thể được phân chia thành có cấu trúc, nửa cấu trúc và không có cấu trúc. Trong đó các Hệ thông tin quản lý (MIS) được dùng để giải quyết loại bài toán thứ nhất còn lớp các bài toán thứ hai và thứ ba là phạm vi giải quyết của Hệ trợ giúp quyết định và Hệ chuyên gia.

Hệ trợ giúp quyết định là những hệ ứng dụng xây dựng trên máy tính nhằm giải quyết các bài toán, các vấn đề có cấu trúc kém. Vai trò chính của Hệ trợ giúp quyết định là nhằm mục đích giúp các nhà ra quyết định giải quyết những vấn đề trong những hoàn cảnh chưa được định nghĩa rõ ràng, các nhà ra quyết định có thể chưa biết rõ vấn đề cũng như giải pháp, tiêu chuẩn đánh giá sự thành công của lựa chọn.

Sự ra đời của Hệ trợ giúp quyết định đánh dấu bước phát triển quan trọng trong lĩnh vực ứng dụng tin học trong quản lý và điều hành công việc. Kể từ đó nó đã không ngừng được nghiên cứu và phát triển cả về lý thuyết và thực tế triển khai ứng dụng. Hệ trợ giúp quyết định tỏ ra có một thế mạnh nổi trội, rất cần thiết cho lãnh đạo và quản lý khiến nhiều tổ chức quan tâm nghiên cứu đầu tư xây dựng và phát triển.

4.1.2. Hệ trợ giúp quyết định

Hệ trợ giúp quyết định ban đầu rất thô sơ, được phát triển từ các phần mềm bảng tính. Các Hệ trợ giúp quyết định sau đó sử dụng các mô hình tối ưu của việc nghiên cứu các hoạt động nghiệp vụ và khoa học quản lý (OR/MS), sử dụng các kỹ thuật như qui hoạch tuyến tính. Phân tích “What if” đã trở nên đặc biệt phù hợp với các mô hình OR. Sử dụng cách tương tác “fron_ends”, những người làm quyết định có thể khám phá ra các khả năng và những gợi ý tốt hơn là những phán đoán cảm tính trong việc ra quyết định.

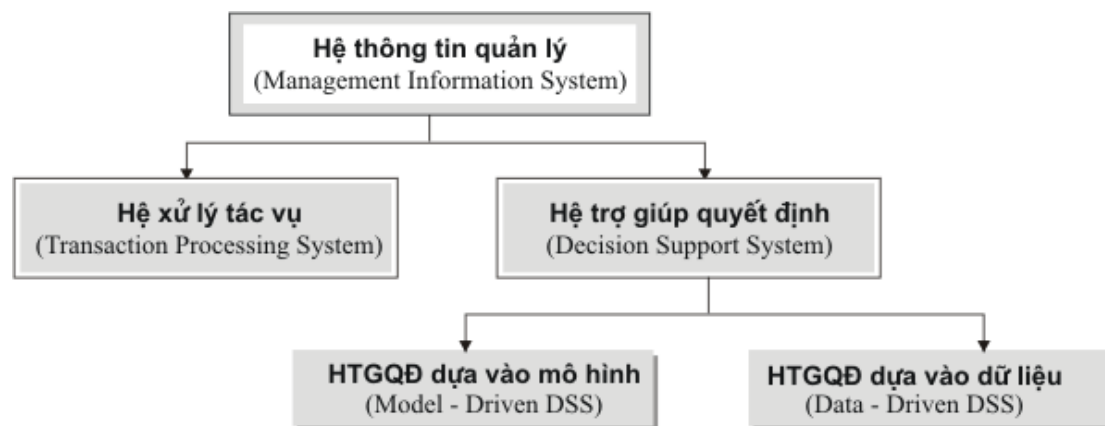
Hệ trợ giúp quyết định trở nên tinh thông hơn khi sử dụng các kỹ thuật trí tuệ nhân tạo. Những hệ thống này có thể được xem như những hệ thống dựa trên tri thức (Knowledge-Based Systems).

Với mục đích là làm như thế nào đó để có thể giúp người sử dụng thực hiện các phân tích nhằm đề xuất được các quyết định cần thiết, chúng ta có thể quan niệm rằng bất cứ loại hệ thống nào tuân thủ một mô hình tổ chức và xử lý riêng biệt của nó mà có thể trợ giúp việc ra quyết định thì đều được xem

là một Hệ trợ giúp quyết định.

Các loại Hệ trợ giúp quyết định truyền thống như sử dụng bảng tính, tối ưu toán học, phân tích số hay mô hình mô phỏng sẽ vẫn tiếp tục phát huy được tác dụng trong việc giải quyết nhiều vấn đề. Tuy nhiên những người làm những công việc sử dụng tri thức trong doanh nghiệp ngày càng yêu cầu hệ thống phải biết nhiều hơn và phải làm được nhiều hơn trong việc truy xuất, tổng hợp và phân tích thông tin. Họ sẽ càng ngày càng phụ thuộc nhiều hơn vào hệ thống để có thể ra các quyết định nhanh chóng với độ tin cậy cao hơn. Đây là xu hướng phát triển của hệ thống thông tin nói chung và của Hệ trợ giúp quyết định nói riêng.

4.1.3. Phân loại các hệ trợ giúp quyết định



Hình 4.1. Phân loại các Hệ thống tin quản lý

- Hệ xử lý tác vụ: mục đích chính của các Hệ xử lý tác vụ là giữ cho việc ghi nhận các giao tác được chính xác. Hệ thống này được xây dựng chỉ có thể làm ra những quyết định đơn giản trong việc xác định dữ liệu được ghi nhận là có hợp lệ hay không. Hệ xử lý tác vụ làm công việc hợp lệ hoá trước khi ghi nhận giao tác để CSDL được sạch hơn.

- Hệ trợ giúp quyết định: bao gồm những hệ thống được thiết kế để trợ giúp các nhà quản lý ra quyết định. Khác với Hệ xử lý tác vụ phục vụ cho các hoạt động hàng ngày, một Hệ trợ giúp quyết định phục vụ cho những mục tiêu dài hạn hơn và có thể cần đến một vài ý kiến, phán đoán đóng góp từ các chuyên gia. Nhu cầu cần thiết thêm các phán đoán của con người sẽ nhiều hơn nếu bài toán đặt ra không có cấu trúc chặt chẽ, khiến cho hệ thống khó có thể nắm bắt được tất cả những sắc thái của tình huống tạo ra quyết định.

Hiện nay hệ trợ giúp quyết định có thể được chia thành hai hướng cơ bản. Hướng đầu tiên dựa vào mô hình theo xu hướng của các Hệ trợ giúp quyết định cũ. Giá trị của hệ thống này là ở chất lượng của mô hình của nó. Khả năng phân tích của nó được dựa trên một lý thuyết hay trên một mô hình mạnh cùng với một giao diện tốt để làm cho mô hình dễ sử dụng.

Loại Hệ trợ giúp quyết định thứ hai là loại Hệ trợ giúp quyết định dựa vào dữ liệu. Giá trị của hệ thống này là ở khả năng tổ chức một lượng lớn dữ liệu và khả năng tổng hợp, phân tích dữ liệu của nó. Với sự phát triển cao của các kỹ thuật mạng và CSDL, Hệ trợ giúp quyết định hướng theo dữ liệu là một thành tựu lớn. Đây là một bước ngoặt thú vị từ tiếp cận truyền thống sang tiếp cận mới, trong đó cấu trúc và dữ liệu tách nhau ra và được tổ chức động trong kho dữ liệu, phản ánh bước tiến quan trọng về CSDL tương tác của các Hệ trợ giúp quyết định. Với tiếp cận mới này, dữ liệu đóng vai trò cung ứng và là động lực cho một Hệ trợ giúp quyết định đưa ra những thông tin cần thiết khác. Tuy nhiên dữ liệu được thu thập từ nhiều nguồn rất đa dạng và người sử dụng sẽ khó khăn với khối lượng dữ liệu phức tạp. Vì vậy yêu cầu lớn nhất đặt ra với Hệ trợ giúp quyết định loại này là khả năng xử lý, phân tích để phát hiện được những thông tin bổ ích từ các kho dữ liệu đó.

4.2. Hệ trợ giúp quyết định dựa vào dữ liệu

4.2.1. Tiếp cận kho dữ liệu và OLAP

Hoạt động xử lý thông tin có thể được phân thành hai loại: phân tích tác vụ (Operations Analysis) và phân tích hướng quyết định (Decision Oriented Analysis). Kho dữ liệu (Data Warehouse) và OLAP có thể được xem như là các thành phần của hoạt động xử lý thông tin hướng quyết định dựa trên phân tích (Analysis Based Decision Oriented Information Processing). Trong đó, kho dữ liệu đóng vai trò cung cấp dữ liệu và OLAP đóng vai trò phân tích, khai thác các dữ liệu này. Nói một cách khác, để có thể trợ giúp quyết định dựa vào dữ liệu cần xây dựng hai thành phần quan trọng là kho dữ liệu và OLAP.

Để có khả năng cung cấp những dữ liệu quyết định cho những người ra quyết định, cần sử dụng một cách lưu trữ dữ liệu cho phép họ quản lý, khai thác dữ liệu một cách dễ dàng. Cách lưu trữ dữ liệu kiểu này là kho dữ liệu. Một kho dữ liệu là một CSDL được thiết kế để trả lời các câu hỏi. Nó là nơi chứa nhiều loại dữ liệu từ các nguồn khác nhau (các hệ thống xử lý tác vụ). Dữ liệu từ những nguồn này được chuyển dịch vào trong kho dữ liệu, được đánh chỉ mục và được kết nối lại để có thể được truy xuất nhanh chóng và dễ dàng hơn, phục vụ cho các ứng dụng trợ giúp ra quyết định. Về thực tế, kho dữ liệu được hiểu như là một kho dữ liệu ổn định, phản ánh hoạt động của một đơn vị trong quá khứ.

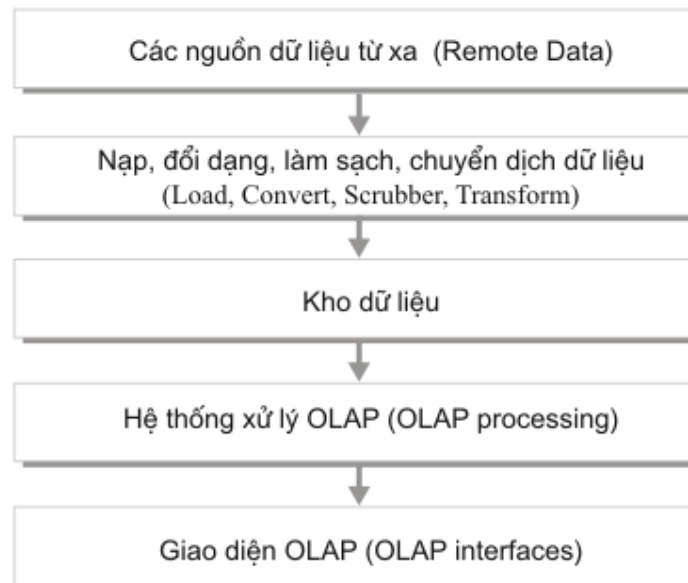
Một khi dữ liệu đã được thu thập, người sử dụng còn cần có một phương cách tốt để dễ dàng khai thác chúng nhằm truy xuất được các mẫu dữ liệu mà họ quan tâm. Hệ thống OLAP giúp cho họ làm điều này. Có vài cách tiếp cận khác nhau tới việc biểu diễn OLAP, nhưng chung nhất là tiếp cận lưu trữ dữ liệu đa chiều. Biểu diễn này cho ta một ma trận được định chiều của các ô. Sẽ có các ô chứa dữ liệu nhập từ ngoài (các phần tử dữ liệu cơ sở) và

các ô còn lại sẽ được tính toán từ các tiến trình gộp và chuyển dịch dữ liệu. Hệ thống OLAP là một hệ thống quản lý dữ liệu giàu năng lực, nó cho phép người sử dụng cắt lát dữ liệu theo nhiều khía cạnh khác nhau. Nếu người sử dụng cần thiết quan tâm chi tiết hơn về mẫu dữ liệu nào đó, họ có thể khoan sâu xuống (Drill_down) chi tiết của dữ liệu. Hệ thống OLAP cho phép người sử dụng “tiến sâu” vào dữ liệu và khám phá chúng ở nhiều mức. Người sử dụng có thể truy xuất được những dữ liệu cần thiết một cách nhanh chóng và dễ dàng mà không cần thực hiện lại công việc lập trình. Các yêu cầu chức năng chính của một hệ thống OLAP là: truy xuất và tính toán nhanh, có khả năng phân tích mạnh, linh hoạt (phân tích linh hoạt, giao diện linh hoạt, hiển thị dữ liệu linh hoạt) và hỗ trợ nhiều người sử dụng. Cũng như các hệ thống thông tin khác, các hệ thống OLAP vẫn yêu cầu phải có các chức năng như: sự chính xác và thích hợp với thời gian. Tuy nhiên chúng lại là các hệ thống duy nhất cố gắng cung cấp thêm các chức năng đặc biệt quan trọng đó là khả năng truy xuất nhanh, linh hoạt, thuận tiện tới số lượng lớn các dữ liệu được phát sinh từ các nguồn dữ liệu nhập có thể thay đổi thường xuyên và hỗ trợ nhiều người sử dụng.

OLAP nhằm tới việc đáp ứng xu hướng gia tăng số lượng và sự phức tạp của các dữ liệu cần thiết cho việc ra quyết định, tới việc gia tăng số người đang sử dụng một nguồn dữ liệu góp chung, tới việc gia tăng số lượng công việc cần thiết ra các quyết định không theo kế hoạch và tới sự gia tăng việc phân phối dữ liệu và xử lý liên quan đến một truy vấn.

Tóm lại, muốn có khả năng cùng lúc nhìn vào nhiều CSDL khác nhau qua việc kết hợp dữ liệu của chúng để làm cho chúng có thể được truy vấn dễ dàng hơn thì kho dữ liệu là một lựa chọn tốt. Nếu chúng ta muốn cung cấp cho người sử dụng khả năng phân tích dữ liệu nhanh chóng và phong phú thì giải pháp OLAP là thích hợp.

Sau đây là sơ đồ về hệ thống kho dữ liệu và OLAP: Đầu tiên dữ liệu từ các nguồn dữ liệu từ xa khác nhau (của các hệ thống xử lý tác vụ) được nạp vào. Trong quá trình nạp, dữ liệu cần được đổi sang dạng chung nhất, được làm sạch và được chuyển dịch thành những kết quả gộp tương đối có thể hữu dụng cho việc phân tích. Cuối cùng dữ liệu được đặt vào kho dữ liệu và được đánh chỉ mục để có thể truy xuất nhanh chóng. Một khi dữ liệu đã ở trong kho dữ liệu, xử lý OLAP trở nên quan trọng cho việc trả lời các truy vấn. Các hệ thống OLAP cho chúng ta khám phá dữ liệu trong những cách hướng tới việc ra quyết định. Các hệ thống OLAP cần có các giao diện đồ họa cho phép người sử dụng nhìn thấy dữ liệu trong dạng số (như bảng) và trong những dạng biểu diễn đồ họa (như biểu đồ). Người sử dụng có thể khoan sâu xuống bằng việc chọn vào các vùng trên màn hình để xem chi tiết hơn.



Hình 4.2. Kho dữ liệu và hệ thống OLAP

4.2.2. Trợ giúp quyết định dựa vào dữ liệu trên cơ sở kho dữ liệu và OLAP

Hệ thống OLAP cho chúng ta khám phá dữ liệu để hướng đến việc ra quyết định. Nó cho phép chúng ta truy xuất và xem dữ liệu từ nhiều khía cạnh

khác nhau. Nhưng quan trọng hơn là hệ thống sẽ cho chúng ta những lối vào bên trong dữ liệu để tìm hiểu, dựa trên chính những đặc tính của dữ liệu. Hệ thống cũng sẽ cho chúng ta khoan sâu vào trong dữ liệu để truy xuất được những thông tin chi tiết ở những mức độ khác nhau mà chúng ta có thể cần đến. Điểm quan trọng cuối cùng là những công cụ OLAP thường nhanh và dễ sử dụng. Chúng ta có thể lướt qua hàng Megabytes hay Gigabytes dữ liệu mà không phải đợi hàng giờ mới có được kết quả.

Hệ thống OLAP rất khác với hệ quản trị CSDL truyền thống: không chỉ dừng lại ở việc truy vấn tĩnh, người sử dụng còn có thể điều chỉnh việc tìm kiếm dữ liệu sao cho nó phù hợp với những nhu cầu chính xác của họ. Vì hệ thống OLAP cung cấp cho người sử dụng khả năng tiến sâu vào dữ liệu, cắt lát, khoan xuống các thông tin chi tiết nên người sử dụng sẽ có thể hiểu rõ hơn về dữ liệu để từ đó có thể ra các quyết định phù hợp một cách nhanh chóng. Hệ thống OLAP cho phép người sử dụng khiến cho “dữ liệu nói chuyện với chính chúng”.

Với mục đích nhằm hiểu được các loại dữ liệu để sử dụng và cách tổ chức chúng, ta cần làm việc với những nhà ra quyết định. Họ sẽ cho chúng ta biết họ sẽ sử dụng các dữ liệu như thế nào và những câu hỏi nào mà họ mong muốn được trả lời. Chính từ trong việc tổ chức mô hình OLAP, chúng ta sẽ khám phá ra được những loại dữ liệu nào đang sẵn có và những dữ liệu sơ cấp nào cần được thu thập để có thể phục vụ tốt cho việc tạo ra những quyết định hiệu quả.

Như vậy trợ giúp quyết định hướng theo dữ liệu nhằm vào việc tổ chức hiệu quả kho dữ liệu và sử dụng giải pháp OLAP để cung cấp tối đa các thông tin theo xu hướng quyết định cho người sử dụng, trợ giúp cho họ đưa ra được những quyết định phù hợp một cách dễ dàng và nhanh chóng nhất.

4.2.3. Tiến trình trợ giúp quyết định dựa vào dữ liệu cho bài toán cụ thể

Trước khi thiết kế mô hình OLAP cho một bài toán, chúng ta cần xác định rõ các vấn đề gặp phải trong các tình huống xem xét. Các vấn đề được nêu ra căn cứ vào việc tìm hiểu tình huống thực tế một cách khách quan và toàn diện. Trên cơ sở các nhận định này, chúng ta sẽ đặt ra những mục tiêu cần đạt tới ví dụ như: khắc phục hoặc giảm thiểu hạn chế, cải tiến hiệu quả ...



Hình 4.3. Tiến trình trợ giúp quyết định dựa vào dữ liệu cho bài toán cụ thể

Để ứng dụng OLAP, chúng ta cần xây dựng một mô hình phục vụ cho phân tích OLAP dựa trên tình huống của bài toán, các vấn đề và các mục tiêu đã xem xét. Đó là quá trình xác định các khối dữ liệu dự định tổ chức, định nghĩa cấu trúc các chiều và định nghĩa các công thức/luật cần thiết cho tính toán. Việc xác định các khối dữ liệu sẽ phát sinh ra vấn đề tại sao lại tổ chức n khối mà không là m khối, tổ chức những khối nào là có lợi nhất? Tương tự, việc tổ chức cấu trúc các chiều cũng như việc định nghĩa các công thức sẽ đặt ra những câu hỏi: phân cấp chiều như vậy đã phù hợp và đầy đủ chưa, các công thức định nghĩa như vậy đã đúng chưa, hợp lý không? Tất cả các vấn đề này phần lớn phụ thuộc vào việc cân nhắc hiệu quả xử lý đối với bài toán cụ thể, vào sự phân tích tình huống, hoàn cảnh thực tế một cách đầy đủ và cả trên điều kiện triển khai thực hiện mô hình: thiết bị phần cứng, ưu, khuyết của hệ thống OLAP sử dụng... Khó tạo lập được một chuẩn mực, phương pháp để xây dựng mô hình OLAP cho tất cả các bài toán, công việc này chủ yếu tùy thuộc vào bài toán cụ thể, vào môi trường triển khai và cả vào kinh nghiệm...

Trên cơ sở mô hình OLAP, CSDL OLAP đã được thiết lập, người khai thác hệ thống sẽ điều chỉnh việc tìm kiếm thông tin của mình bằng cách liên tục đặt ra những yêu cầu truy vấn, thực hiện, rồi nhận xét kết quả, nhằm tìm hiểu rõ dần những nội dung tiềm ẩn của dữ liệu nguồn (thu thập được theo những vấn đề đang quan tâm) để tiến tới chỗ có đủ cơ sở nhận định, từ đó ra được các quyết định cần thiết.

Việc trợ giúp ra quyết định không nhằm đưa ra cho người sử dụng một số phương án khả dĩ hiệu quả để giúp họ lựa chọn hoặc đưa hẳn ra một phương án khả dĩ tối ưu để giúp họ quyết định như các hệ trợ giúp quyết định dựa vào mô hình thường làm, nó tạo phương tiện để cung cấp nhiều nhất các thông tin phong phú, đa dạng, trên các khía cạnh, ở các mức khác nhau một cách nhanh chóng, giúp cho người khai thác có thể điều chỉnh việc tìm kiếm

dữ liệu để nắm bắt được tối đa những gì họ cần hiểu rõ, để chính họ sẽ ra những quyết định phù hợp. Như vậy người khai thác cần là chuyên gia về lĩnh vực của bài toán ứng dụng cụ thể. Tuy nhiên không như khuynh hướng của hệ trợ giúp quyết định dựa vào mô hình là phần lớn phạm vi ứng dụng hẹp cho từng vấn đề cụ thể và chỉ giúp quyết định được một vài vấn đề, hệ thống trợ giúp quyết định dựa vào dữ liệu có phạm vi ứng dụng là rất rộng và có thể giúp ra nhiều quyết định khác nhau. Trong mỗi ứng dụng cụ thể, chúng ta chỉ cần thiết lập mô hình OLAP tương ứng cho nó là có thể hỗ trợ ra quyết định.

4.3. Xây dựng cấu trúc thông tin hỗ trợ việc ra quyết định

4.3.1. Vai trò của cấu trúc thông tin

Các hệ thống thông tin và cấu trúc thông tin của nó cần hỗ trợ mỗi giai đoạn của quá trình ra quyết định. Môi trường ra quyết định đòi hỏi một số mô hình trợ giúp quyết định phụ thuộc vào các điều kiện khác nhau. Nếu quá trình ra quyết định có thể không tự động hoá hoàn toàn thì nó cũng có thể được tự động hoá một phần nào đó.

Để ra quyết định qua việc phân tích và mô hình hoá, các ứng dụng dữ liệu cần được liên kết từ nhiều cơ sở dữ liệu truyền thống và được cơ cấu lại để hỗ trợ ứng dụng. Sau đó nó cần được định dạng lại và được đưa ra trong dạng thức có thể trợ giúp để làm việc hiểu nó dễ dàng hơn.

Hiện nay bất kỳ tổ chức nào cũng có các hệ thống dựa trên thực tế như các hệ thống xử lý giao dịch (Transaction Processing Systems), các hệ thống tự động hoá văn phòng (Office Automation Systems), các hệ trợ giúp quyết định (Decision Support Systems) và các hệ thống xử lý phân tích trực tuyến (OLAP). Các hệ thống này được hỗ trợ bởi các công nghệ khai thác dữ liệu, kho dữ liệu, ảo hoá dữ liệu và các hệ thống thông tin địa lý, mỗi hệ thống thông tin này có các cấu trúc thông tin riêng của nó. Để cung cấp môi trường

trợ giúp quyết định với các cấu trúc thông tin ở trên cần thực hiện các bước sau:

- Thống nhất dữ liệu từ các cấu trúc thông tin khác nhau trong các nguồn dữ liệu khác nhau. Quá trình này được gọi là trích xuất, chuyển đổi và nạp dữ liệu (Extract Transformation Load - ETL).
- Xử lý bằng các công cụ phân tích và mô hình hoá.
- Định dạng lại dữ liệu gốc, cung cấp các năng lực truy vấn, đồ hoạ và báo cáo.

4.3.2. Các yếu tố ảnh hưởng

Có hai tham số cho thấy tính hiệu quả của hệ thống ra quyết định:

- Các yêu cầu thông tin: sự hỗ trợ của cấu trúc thông tin đối với các yêu cầu thông tin của người ra quyết định.
- Mức độ tích hợp: đây là sự mở rộng các cấu trúc thông tin trong tổ chức được tích hợp.

4.3.2.1. Các yêu cầu thông tin

Yêu cầu quan trọng trong quá trình ra quyết định là sự có mặt của thông tin được yêu cầu trong dạng thức đúng. Tính hiệu quả của các cấu trúc thông tin được sử dụng để ra quyết định được đo bằng các yếu tố sau:

- Độ chính xác của thông tin.
- Độ tin cậy của thông tin.
- Độ tương quan của thông tin.
- Dạng thức của thông tin.
- Tính chất hợp thời của thông tin.

Sự kết hợp mức độ ra quyết định bao hàm các cấu trúc thông tin từ các tính năng khác. Nếu không được cấu trúc, các yêu cầu chính xác của người ra quyết định không được giải quyết và cũng có thể biến đổi đa dạng hơn. Do đó

người ra quyết định có thể không hiểu đầy đủ các nhu cầu thông tin thực tế. Khi các hệ thống OLAP được xây dựng sử dụng các hệ thống OTLP, chức năng của hệ thống đích phụ thuộc vào dữ liệu nguồn. Sự phụ thuộc này có thể được hạn chế. Chất lượng cấu trúc thông tin của hệ trợ giúp quyết định phụ thuộc vào chất lượng cấu trúc thông tin của các hệ thống nguồn.

Các hậu quả của việc không thực hiện các yêu cầu thông tin của người ra quyết định là:

- Khi thông tin không chính xác ở cấp có thể chấp nhận được, người ra quyết định cần đưa ra các nhận xét. Đây là việc tiêu tốn thời gian và dễ mắc lỗi. Trong một số các trường hợp, họ cần loại bỏ thông tin từ hệ thống và tìm kiếm thông tin từ các nguồn khác.
- Khi thông tin không liên quan đến mục đích, người ra quyết định cần xuyên suốt quá trình để có được thông tin liên quan. Điều này có thể bao gồm việc xử lý lại dữ liệu hoặc chuyển qua các quá trình khác cho đến khi thông tin liên quan được thu thập.
- Khi thông tin không đúng dạng thức, người ra quyết định phải xuyên suốt quá trình để đưa nó về dạng thức chính xác. Điều này lần nữa lại làm tiêu tốn thời gian.
- Đã có các trường hợp các thông tin được yêu cầu không có mặt đúng lúc để cho ra các quyết định quan trọng. Các quyết định quan trọng này được cho ra khi sử dụng các tổng hợp bằng tay.

Mặc dù không liên quan trực tiếp đến các yêu cầu thông tin, hai yếu tố sau lại hạn chế việc thu được các lợi ích tối đa từ các hệ trợ giúp quyết định:

- Việc thiếu các công cụ phân tích hợp lý làm giảm khả năng phân tích và đánh giá các lựa chọn thay thế. Người ra quyết định không thể thực hiện phân tích lợi ích/chi phí hay phân tích ảnh hưởng.
- Khi hệ thống hạn chế số lượng các lựa chọn thay thế, người ra quyết

định không thể lựa chọn giải pháp tốt nhất.

4.3.2.2. Mức độ tích hợp

Một yếu tố rõ ràng trong môi trường ra quyết định là số lượng nguồn thông tin ngày càng tăng. Việc ra quyết định đòi hỏi khả năng từ nhiều tính năng. Điều này sinh ra yêu cầu tích hợp một vài cấu trúc thông tin trong các hệ thống khác nhau. Trong đa số các trường hợp, không thể truy cập hay đưa ra tất cả thông tin được yêu cầu. Thậm chí nếu tất cả các yêu cầu đã được xác định thì cũng không thể thực hiện khi sử dụng dữ liệu từ các hệ thống thông tin hiện có. Trích xuất dữ liệu từ một vài OLTP hay các nguồn dữ liệu hợp lệ thành một nhóm chung và chuyển nó thành dạng thức tương thích với hệ thống OLAP đòi hỏi các nỗ lực đáng kể.

Với các công cụ thông minh, kho dữ liệu có sẵn, việc tích hợp tất cả hoặc tích hợp một vài tính năng không phải là nhiệm vụ dễ dàng. Nó không thể điều tiết dữ liệu từ một vài hệ thống nội tại. Một số nguyên nhân của các vấn đề tích hợp là:

- Việc tích hợp tự động không thể được cung cấp theo tính chất của hạ tầng công nghệ. Các tổ chức đầu tư vào các công nghệ khác nhau vì thế cần yêu cầu tích hợp.
- Tính chất của hệ thống không cho phép cung cấp việc tích hợp hệ thống. Các ứng dụng từ các nhà cung cấp khác nhau hoạt động theo các thông số khác nhau. Vì vậy về kỹ thuật, có thể không hợp lý khi tiến hành tích hợp các hệ thống này.
- Các kỹ năng cần thiết cần cho tích hợp các hệ thống này không có trong tổ chức.
- Yêu cầu tích hợp tự động bị bỏ qua.

Các hậu quả của điều này là:

- Kết quả nhận được từ các hệ thống đa dạng có các định dạng khác

nhau. Cần xem xét để đưa nó vào một dạng thức chung.

- Việc thực hiện bằng tay nhằm kết hợp các kết quả này là vô cùng nhiều. Người ra quyết định có thể không mất thời gian cho các nhiệm vụ này.
- Khi thực hiện các bước cần thiết để kết hợp các kết quả từ các nguồn khác nhau, cơ hội để nhận thấy các lỗi ở mỗi bước là hiển nhiên. Điều này đòi hỏi nỗ lực bổ sung để thông qua và sửa các lỗi này.
- Người ra quyết định có thể cần sự hỗ trợ kỹ thuật để kết hợp các kết quả từ các hệ thống khác nhau. Sự phụ thuộc về kỹ thuật ở mức độ này là có thể chấp nhận được.
- Sự không nhất quán trong các kết quả từ các nguồn khác nhau có thể không được phát hiện. Trong trường hợp này kết quả cuối có thể chứa lỗi.

4.3.3. Mô hình tổ chức thông tin

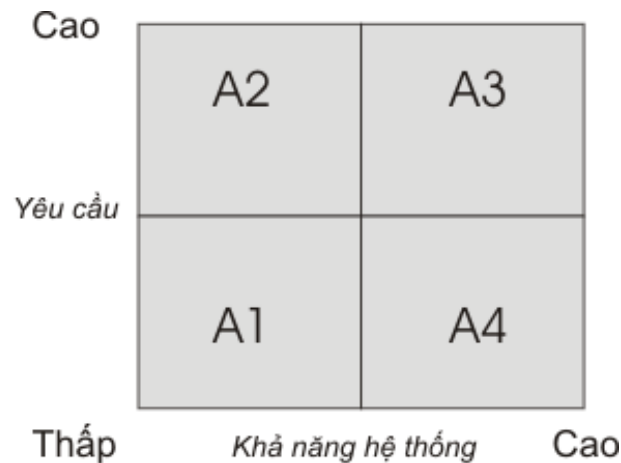
Không có một biện pháp tốt nhất hay một mô hình tốt nhất để xây dựng các cấu trúc thông tin để có thể đáp ứng các yêu cầu của người ra quyết định. Một tiêu chí có thể được cung cấp để việc lựa chọn cấu trúc thông tin là phù hợp nhất. Điều này dựa trên hai yếu tố đã được nêu ở trên là các yêu cầu thông tin và mức độ tích hợp. Các yếu tố này được sử dụng để hình thành chiến lược xây dựng cấu trúc thông tin cho việc ra quyết định.

4.3.3.1. Các yêu cầu thông tin và năng lực của hệ thống thông tin

Bước đầu tiên là phải thiết lập một sự tương đồng giữa các yêu cầu thông tin và hệ thống hỗ trợ thông tin. Các yêu cầu thông tin chính xác có thể không được xác định dựa theo tính chất phi cấu trúc của quá trình ra quyết định, các hạn chế dựa trên hiểu biết kinh nghiệm của người ra quyết định. Các hệ trợ giúp quyết định hiện đại cung cấp một số năng lực nhằm hỗ trợ người

ra quyết định trong quá trình chuẩn bị ra quyết định. Tuy nhiên, không phải tất cả các tiện ích này đều do người ra quyết định yêu cầu. Vì thế cần gợi ý để nhận dạng các yêu cầu ở một cấp nào đó, sau đó các tiện ích từ hệ trợ giúp quyết định có thể được thiết kế phù hợp với các yêu cầu này.

Bằng việc sử dụng hai thông số là mức độ của việc nhận dạng yêu cầu thông tin và các năng lực phần mềm trợ giúp quyết định, ma trận Yêu cầu/Năng lực được xác định.



Hình 4.4. Ma trận Yêu cầu/Năng lực

Ma trận thể hiện 4 khả năng được ký hiệu A1, A2, A3 và A4. Mỗi khả năng cung cấp các mức độ khác nhau của việc nhận dạng các yêu cầu và các năng lực của hệ thống.

- A1: Chỉ các yêu cầu thông tin tối thiểu được nhận dạng và phần mềm chỉ cung cấp các tiện ích tối thiểu. Chỉ nhận dạng các yêu cầu tối thiểu của người ra quyết định đồng thời hệ thống chỉ cung cấp một số chức năng có ích. Nó là hệ thống tự động đơn giản như một bảng tính. Trong trường hợp này người ra quyết định cần thực hiện công việc bằng tay nhiều hơn trong quá trình ra quyết định.

- A2: Phần lớn các yêu cầu đã được nhận dạng. Nhưng hệ thống không thể cung cấp tất cả các tiện ích này. Nguyên nhân có thể là do việc đánh giá hệ thống không hợp lý, có thể là do không có thông tin liên quan, không có người chuyên nghiệp có kỹ năng để đánh giá hoặc do xu hướng của các sản phẩm thương mại.
- A3: Tất cả các yêu cầu của người ra quyết định đã được nhận dạng và hệ thống hỗ trợ quyết định có khả năng cung cấp các tiện ích. Người ra quyết định có thể sử dụng toàn bộ các tiện ích này. Đây là trường hợp lý tưởng cho việc ra quyết định do có một sự tương đồng giữa các yêu cầu và các tiện ích hệ thống.
- A4: Chỉ một phần các yêu cầu được nhận dạng. Hệ thống có các tiện ích để cung cấp tất cả các chức năng được yêu cầu trong quá trình ra quyết định. Tuy nhiên các tiện ích không được sử dụng bởi những người ra quyết định. Những người ra quyết định không biết làm thế nào để sử dụng hệ thống nhằm thu được các lợi ích tối đa. Kiểu này làm phức tạp các hệ trợ giúp quyết định được cài đặt mà không phân biệt tất cả các yêu cầu.

A3 là tình huống lý tưởng và trên thực tế điều này hiếm khi tồn tại. Tuy nhiên đa số các tổ chức có các hệ thống nằm trong khả năng A2 hoặc A4 và trên thực tế sẽ đạt được mức vừa phải giữa A2 và A4.

4.3.3.2. Mức độ tích hợp hệ thống

Có 5 mức độ tích hợp hệ thống được xác định. Các mức độ này biến đổi từ kết hợp bằng tay hoàn thành đến tích hợp toàn bộ.

- Mức 1: Kết quả nhận được từ các hệ thống khác nhau được tích hợp cao và không thể kết hợp chúng một cách tự động. Đây là trường hợp không có sự tích hợp và người ra quyết định cần xử lý các kết quả bằng tay.

- Mức 2: Đây là trường hợp tồi nhất. Các kết quả từ các hệ thống được xem xét bởi người ra quyết định và chỉ thông tin liên quan được trích xuất từ báo cáo. Ở đây có sự xử lý bằng tay để có được kết quả mong muốn. Ví dụ: có thể có một số hệ thống riêng biệt đưa ra các kết quả trong dạng thức các báo cáo in.
- Mức 3: Như mức 2, các kết quả nhận được từ các hệ thống riêng khác nhau như các báo cáo in. Đây là việc truy xuất lại hệ thống tự động khác để thu được kết quả mong muốn. Ví dụ: các kết quả nhận được từ các hệ thống khác nhau được nhập vào bảng tính Excel để xử lý.
- Mức 4: Trong trường hợp này các kết quả có mặt trong dạng thức ‘điện tử’. Tuy nhiên sự can thiệp bằng tay là cần thiết để đưa các kết quả này vào quá trình kết hợp được tự động hoá. Ví dụ: các kết quả từ nhiều hệ thống có mặt như các file riêng, người ra quyết định cần tải các file này vào thư mục liên quan bằng cách sử dụng các thủ tục.
- Mức 5: Đại diện cho việc tích hợp toàn bộ hệ thống. Hệ thống tự động thu thập các kết quả từ các hệ thống khác và kết hợp xử lý chúng để đạt được các kết quả mong muốn. Trên thực tế không thể cung cấp khả năng này do các lý do kỹ thuật như sự không tương thích giữa các cấu trúc cơ sở dữ liệu. Trong một số trường hợp, nỗ lực phát triển phần mềm yêu cầu cung cấp tiện ích này.

4.3.4. Kết luận

Có một số yếu tố ảnh hưởng đến việc xây dựng các cấu trúc thông tin để ra quyết định. Các yêu cầu thông tin của người ra quyết định và việc tích hợp một vài hệ thống đóng vai trò sống còn. Không có một biện pháp hoàn hảo nào để cung cấp hỗ trợ tự động dành cho việc ra quyết định. Ma trận Yêu cầu/Năng lực và mức độ tích hợp sẽ cung cấp tiêu chí để đạt được thành công trong việc hỗ trợ ra quyết định.

4.4. Dịch vụ trợ giúp quyết định của Microsoft

OLAP là một công nghệ được tăng cường có thể cải thiện các khả năng phân tích nhưng nó lại được coi như một công cụ đắt tiền, khó hoàn chỉnh và không mềm dẻo khi triển khai. Microsoft đã giải quyết các vấn đề này của OLAP và đưa ra giải pháp phân tích đa chiều.

Microsoft DSS (Microsoft Decision Support Services) là giải pháp OLAP mới với đầy đủ các tính năng được đưa ra như một phần của công cụ Microsoft SQL Server. DSS gồm một Server cho phép người sử dụng đưa ra các phân tích phức tạp với một khối dữ liệu lớn trong một khoảng thời gian chấp nhận được. Thành phần phía Client của Microsoft DSS là Client Cache và công cụ tính toán được gọi là Microsoft PivotTable Service giúp tăng cường khả năng thực thi và giảm lưu thông dữ liệu trên mạng. PivotTable Service cũng có khả năng cho phép người sử dụng kiểm soát các phân tích khi mạng bị ngắt.

OLAP là thành phần chủ yếu trong quá trình xây dựng triển khai các kho dữ liệu, Microsoft DSS cung cấp các tính năng cốt lõi hỗ trợ nhiều ứng dụng từ lập báo cáo đến hỗ trợ ra quyết định. Các tính năng OLAP trong SQL Server sẽ tạo ra các phân tích đa chiều đầy đủ hơn và đem lại các thuận lợi cho người sử dụng. Nó không chỉ được sử dụng cho trong tổ chức mà còn có thể áp dụng cho các nhóm và cá nhân.

Do sự đa dạng của các công cụ và ứng dụng hỗ trợ OLAP qua Microsoft OLE DB cho giao diện OLAP mà Microsoft DSS có thể giúp tăng số lượng các tổ chức truy nhập vào các công cụ phân tích phức tạp này và làm giảm chi phí cho kho dữ liệu.

4.4.1. Kho dữ liệu Microsoft

Trong lịch sử, việc tập trung đầu tư vào tính toán để có được các hệ thống xử lý dữ liệu như kế toán, đặt hàng, sản xuất kinh doanh hay hệ thống

thông tin khách hàng là yêu cầu thiết yếu. Các tổ chức đầu tư ngày càng nhiều vào các ứng dụng và công nghệ để tập trung xử lý các dữ liệu đã được thu thập. Kho dữ liệu chính là kết quả của quá trình thu thập từ các nguồn khác nhau của các hệ thống để từ đó đưa ra các thông tin phân tích và báo cáo cho người sử dụng. Kho dữ liệu được sử dụng để lưu trữ, mô tả và tổng hợp các thông tin cho người sử dụng.

Vài năm trước, Microsoft đã tiến hành một vài cải tiến với mục đích mở rộng khả năng kho dữ liệu và tính năng trợ giúp quyết định. Hai sáng kiến đã được Microsoft đưa ra là Microsoft Data Warehousing Framework gồm các chỉ dẫn cho việc phát triển các sản phẩm Microsoft và Microsoft Alliance For Data Warehousing là liên minh các nhà công nghiệp hưởng ứng các nền tảng Microsoft và Data Warehousing Framework nhằm mục đích phát triển và kinh doanh. Các sáng kiến này dựa trên xu hướng chủ đạo của Microsoft nhằm vào tiến trình kho dữ liệu:

- Giảm giá việc thực hiện, duy trì các kho dữ liệu.
- Xác định lại việc phân cấp các hệ thống kho dữ liệu, không chỉ những hệ thống rất lớn mà cả những cấp thấp hơn với những người sử dụng riêng biệt.
- Tăng cường tích hợp các công cụ kho dữ liệu cung cấp bởi những nhà cung cấp khác.

4.4.1.1. Microsoft Data Warehousing Framework

Data Warehousing Framework là một kiến trúc mở dùng để mô tả các cơ chế chia sẻ dữ liệu và siêu dữ liệu trong việc xây dựng và quản lý các kho dữ liệu và kho dữ liệu chủ đề. Các công nghệ cơ bản nằm trong Framework là các giao diện dữ liệu OLE DB và Microsoft Repository chạy trên SQL Server. Microsoft Repository là một cơ sở dữ liệu chứa thông tin mô tả về các thành phần phần mềm và các mối quan hệ giữa chúng (siêu dữ liệu). Các mô hình

siêu dữ liệu được định nghĩa trong Microsoft Repository cho các giản đồ cơ sở dữ liệu, biến đổi dữ liệu và các giản đồ cơ sở dữ liệu OLAP.

Các thành phần trong Framework miêu tả các bước đầy đủ trong tiến trình kho dữ liệu, một vài bước trong số này được thực hiện bởi Microsoft nhưng có thể dễ dàng mở rộng bởi các đối tác của Microsoft sử dụng công nghệ được chọn. SQL Server cung cấp nhiều thành phần cơ bản được yêu cầu cho việc xây dựng và duy trì kho dữ liệu như: thiết kế cơ sở dữ liệu với công cụ đồ hoạ thiết kế giản đồ, khả năng lưu trữ dữ liệu cao, các tính năng chuyển đổi dữ liệu qua Data Transformation Services (DTS), các tính năng OLAP với DSS.

4.4.1.2. Sự phức tạp của dữ liệu

Tuỳ thuộc vào tiến trình kho dữ liệu, dữ liệu được chuẩn bị cho người sử dụng và hầu hết thông tin trong một kho dữ liệu quan hệ không dễ khai thác. Thông thường cấu trúc dữ liệu rất khó để nhận thức đối với người sử dụng doanh nghiệp hoặc các câu hỏi kinh doanh (như “Ai là người bán hàng giỏi nhất trong mỗi khu vực trong từng tháng của năm ngoái?”) là phức tạp khi cần đáp ứng nhanh trong ngôn ngữ truy vấn quan hệ SQL. Một vài yêu cầu có thể được định sẵn với các công cụ truy vấn mở rộng ẩn trong sự phức tạp cơ sở dữ liệu từ người sử dụng. Trong lớp rộng các ứng dụng cho phép người sử dụng hiển thị dữ liệu đa chiều thì giải pháp tối ưu nhất là công nghệ OLAP.

Các tổ chức kinh doanh thường có dữ liệu đa chiều và sự phức tạp là tất yếu. Thậm chí các công ty nhỏ nhất cũng muốn kiểm soát bán hàng qua sản phẩm, người bán, khu vực địa lý, khách hàng và thời gian. Mỗi loại đặc trưng cho một chiều trong mô hình OLAP. Các tổ chức có các công cụ tìm kiếm để truy nhập, tìm kiếm và phân tích dữ liệu đa chiều bằng những cách tự nhiên và dễ dàng.

OLAP không phải là một khái niệm mới, các ứng dụng OLAP cần đưa ra những phân tích nhanh từ các thông tin đa chiều được chia sẻ:

- Nhanh: Thông tin chuyển đến người sử dụng là liên tục và hầu hết các truy vấn được đáp ứng dưới 5 giây.
- Phân tích: Có thể thực hiện các phân tích thống kê và tính toán cơ bản của dữ liệu.
- Chia sẻ: Thực hiện các yêu cầu bảo mật cần thiết cho việc chia sẻ dữ liệu bảo mật có tính tiềm năng trong nhiều người sử dụng.
- Đa chiều: là đặc tính cốt lõi của OLAP.
- Thông tin: cần để có thể truy nhập tất cả dữ liệu và thông tin cần thiết liên quan đến ứng dụng.

4.4.1.3. Lợi ích đối với việc kinh doanh

OLAP cung cấp cho các tổ chức khả năng truy nhập, hiển thị và phân tích dữ liệu kinh doanh một cách mềm dẻo. Trước tiên, OLAP đưa dữ liệu tới người sử dụng qua một mô hình dữ liệu trực giác tự nhiên. Người sử dụng có thể thấy và hiểu thông tin trong kho dữ liệu một cách hiệu quả hơn và do đó cho phép các tổ chức nhận thấy rõ hơn giá trị các dữ liệu của họ. OLAP tăng tốc việc chuyển tải thông tin tới người sử dụng, hiển thị các cấu trúc đa chiều bằng việc sẵn sàng tính toán một vài giá trị dữ liệu. Sự kết hợp giữa tiếp cận dễ dàng và thực thi nhanh chóng cho phép người sử dụng xem và phân tích dữ liệu của họ nhanh hơn và hiệu quả hơn khi chỉ dùng công nghệ cơ sở dữ liệu quan hệ. Kết quả là cần nhiều thời gian để phân tích dữ liệu và tồn ít thời gian phân tích cơ sở dữ liệu.

4.4.1.4. Mô hình dữ liệu

Trong một mô hình dữ liệu OLAP, thông tin được khái niệm như các khối gồm các kiểu mô tả (các chiều) và các giá trị định lượng (đơn vị đo). Mô

hình dữ liệu đa chiều làm cho nó đơn giản đối với những người sử dụng bằng cách công thức hoá các truy vấn phức tạp, sắp xếp dữ liệu trong các báo cáo, thay đổi từ tổng hợp dữ liệu chi tiết và lọc dữ liệu đưa vào các tập con. Ví dụ, các chiều điển hình trong khối chứa thông tin bán hàng gồm Time, Geography, Product, Channel, Organization và Scenario. Các đơn vị đo điển hình sẽ gồm Dollar_Sales, Unit_Sales, Inventory, Headcount, Income và Expense.

Trong mỗi chiều của một mô hình dữ liệu OLAP, dữ liệu có thể được tổ chức trong cấu trúc phân cấp đại diện bởi các mức (Level) chi tiết trong dữ liệu. Ví dụ, trong chiều Time, ta có thể có mức Years, Months và Days, tương tự chiều Geography có thể có các mức Country, Region, State/Province và City. Một trường hợp cụ thể của mô hình OLAP sẽ cho các giá trị riêng biệt của mỗi mức trong cấu trúc phân cấp. Người sử dụng xem dữ liệu OLAP sẽ chuyển lên hoặc xuống giữa các mức dữ liệu để xem chi tiết hơn hoặc tổng hợp thông tin

4.4.1.5. Các hình thức lưu trữ

Các khối, chiều, phân cấp và đơn vị đo là các vấn đề chính khi tiếp cận đa chiều OLAP. Bằng cách mô tả dữ liệu kiểu này, người sử dụng có thể dễ dàng tiếp cận trực quan qua một tập phức tạp dữ liệu. Kiểu mô tả đơn giản mô hình dữ liệu đặc trưng làm thông tin được chuyển tải tới người sử dụng nhanh hơn. Nguyên lý của OLAP là người sử dụng nên xem xét thời gian đáp ứng thích hợp cho mỗi khung nhìn của dữ liệu mà họ yêu cầu. Do dữ liệu thường được thu thập chỉ ở mức chi tiết, thông tin tổng hợp sẽ được tính toán thuận lợi. Các giá trị được tính trước là các lợi ích của OLAP.

Trong thời kỳ đầu của công nghệ OLAP, hầu hết các nhà cung cấp giả định rằng giải pháp khả thi cho các ứng dụng OLAP chỉ được sử dụng trong mô hình lưu trữ phi quan hệ. Sau này các nhà cung cấp khác khám phá ra qua

việc sử dụng các cấu trúc cơ sở dữ liệu (giản đồ hình sao và hình tuyết rơi), đánh chỉ mục, lưu trữ tập hợp và hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) có thể được sử dụng cho OLAP. Những nhà cung cấp này gọi công nghệ này là OLAP quan hệ (ROLAP). Những nhà cung cấp OLAP trước kia thông qua mô hình MOLAP cho OLAP đa chiều.

Trong một vài năm trước, tranh luận giữa MOLAP và ROLAP rất gay gắt. Việc xử lý MOLAP thường tốt hơn công nghệ quan hệ nhưng có các vấn đề về khả năng cân bằng. Mặt khác việc xử lý ROLAP cân bằng hơn và thường hấp dẫn người dùng do họ đầu tư vào công nghệ cơ sở dữ liệu quan hệ.

Ngày nay, người ta lại tập trung vào các giải pháp OLAP lai thường được gọi là HOLAP, là sự kết hợp giữa kiến trúc ROLAP và MOLAP nhằm mang lại một giải pháp có các đặc điểm tốt nhất của cả hai mô hình là: tính thực hiện cao với khả năng cân bằng lớn. Cách tiếp cận HOLAP duy trì các tổng hợp chi tiết trong cơ sở dữ liệu quan hệ khi duy trì các tập hợp riêng rẽ.

4.4.2. Kiến trúc dịch vụ trợ giúp ra quyết định của Microsoft

Microsoft DSS được thiết kế từ với mục đích tối thiểu hoá hầu hết phí tổn cho việc xây dựng và duy trì các ứng dụng OLAP. Microsoft DSS phù hợp với cả các thành phần phần mềm phía Server và Client.

Trên Server, DSS Analysis Server hoạt động như hệ điều hành Windows và cung cấp các tính năng lõi. Chương trình truy nhập đến các tính năng quản trị trong Analysis Server qua một mô hình đối tượng gọi là Decision Support Object (DSO). Giao diện người dùng được tích hợp cho Microsoft DSS (OLAP Manager) cũng được phát triển với DSO và cung cấp nhiều tiện lợi mà không cần lập trình. OLAP Manager có thể thực hiện trên một máy tính phân tán từ Analysis Services cho phép quản trị cơ sở dữ liệu thiết kế cho các mô hình dữ liệu OLAP, truy nhập thông tin trong các kho cơ

sở dữ liệu quan hệ, thiết kế các tập hợp và xác định các kho dữ liệu OLAP và các tính năng khác. Các định nghĩa siêu dữ liệu OLAP được lưu trữ trong các kho riêng nhưng có thể được xuất ra Microsoft Repository với mô hình thông tin OLAP bằng một tiện ích đơn giản.

Microsoft DSS có thể truy nhập nguồn dữ liệu trong bất kỳ nguồn cung cấp dữ liệu OLE DB nào không chỉ SQL Server mà còn bao gồm các cơ sở dữ liệu Microsoft Access, Microsoft FoxPro, Oracle, Sybase, Informix. Bất kỳ nguồn dữ liệu nào cung cấp một giao diện ODBC đều có thể truy nhập được qua tiện ích trong OLE DB mà ODBC điều khiển nếu chúng thuộc các giao diện OLE DB. Những nguồn dữ liệu này có thể nằm trên những hệ nền (Platform) khác hệ điều hành Windows như UNIX hoặc các hệ thống Mainframe và các cơ sở dữ liệu như DB2 hoặc Teradata. Với các năng lực hoạt động trên nhiều hệ nền khác nhau của OLE DB, dữ liệu có thể được truy nhập từ nhiều hệ thống ngay cả khi chúng là cục bộ với DSS Server.

Những nguồn dữ liệu này cũng bao gồm một thành phần Client được gọi là PivotTable Service. PivotTable Service là tiện ích kết nối các ứng dụng Client OLAP với DSS Server. Tất cả các truy nhập dữ liệu được quản lý bởi DSS, các chương trình và ứng dụng Client được thực hiện qua OLE DB cho giao diện OLAP cung cấp bởi PivotTable Service.

Cả các thành phần Client và Server của Microsoft DSS đều có khả năng mở rộng tính năng. Các nhà cung cấp phần mềm độc lập và các nhà tư vấn có thể mở rộng khả năng tính toán, quản lý dữ liệu hoặc tính năng ứng dụng sử dụng các đặc tính DSO.

4.4.3. Các vấn đề trong việc triển khai Microsoft DSS

4.4.3.1. Xây dựng mô hình dữ liệu OLAP cho Microsoft DSS

Một vấn đề cơ bản trong triển khai OLAP là ánh xạ cơ sở giản đồ dữ

liệu ban đầu với mô hình đa chiều. Trong sự phát triển của các sản phẩm OLAP, quá trình thiết kế cơ sở dữ liệu OLAP trở thành quan trọng đặc biệt, liên quan một cách phức tạp đến công nghệ OLAP cụ thể sẽ được triển khai. Do đó, công cụ phát triển cơ sở dữ liệu OLAP phải chuyên dụng, có khả năng phát triển ứng dụng và tăng cường hỗ trợ cho việc thiết kế dữ liệu

Trong hầu hết các vấn đề khi triển khai OLAP có vẻ như dữ liệu đã được chuẩn bị cho việc phân tích qua quá trình thực hiện kho dữ liệu nơi mà thông tin được trích chọn từ các hệ thống tác nghiệp, được làm sạch, hợp lý hoá và tổng hợp để đưa vào các ứng dụng OLAP. Đây là bước cần thiết trong quá trình để đảm bảo rằng dữ liệu được hiển thị bởi người sử dụng OLAP là chính xác, nhất quán và phù hợp.

Ngoài ra, thông tin trong kho dữ liệu được tổ chức theo giản đồ hình sao hoặc hình tuyết rơi dễ dàng giúp người sử dụng hiểu được dữ liệu, tối đa hoá khả năng truy xuất dữ liệu cho các ứng dụng hỗ trợ ra quyết định và giảm thiểu khối lượng lưu trữ đối với các dữ liệu lớn. Các giản đồ này là sự tương đối về mặt quan hệ của mô hình dữ liệu OLAP và là bước đầu cho việc định nghĩa các khối OLAP. Một vài sản phẩm OLAP tạo sự thuận lợi cho xu hướng này. Nó thường không cung cấp các công cụ để ánh xạ một giản đồ hình sao tới một mô hình OLAP làm cho chi phí của việc xây dựng các mô hình OLAP rất cao và thời gian phát triển kéo dài không cần thiết.

Một sự khác biệt quan trọng trong Microsoft DSS là giao diện người sử dụng OLAP Manager được tạo với phần quản trị dữ liệu OLAP (OLAP Database Administrator - DBA). Microsoft DSS OLAP Manager được thực hiện từ Microsoft Management Console (MMC) và chia sẻ cùng một giao diện quản trị với SQL Server. Lợi ích rõ ràng là OLAP DBA có thể làm tốt hơn việc dịch từ SQL Server đến các sản phẩm Microsoft khác. Microsoft DSS bao gồm nhiều tính năng giúp những người ít kinh nghiệm hoặc những

người ít sử dụng thực hiện được các tác vụ chính. Microsoft DSS còn có đầy đủ các tính năng trợ giúp OLAP và các hướng dẫn theo từng bước để xây dựng một khối. Các Wizard có sẵn có thể tự động thực hiện các tác vụ chính như tạo các chiều.

Hơn nữa, Microsoft DSS được đánh giá là có thể phát triển trong môi trường kho dữ liệu được thiết kế theo cấu trúc giản đồ hình sao hay tuyết rơi. Wizard tạo khối đặc biệt thích hợp cho việc tạo các mô hình đa chiều một cách nhanh chóng. Ngoài ra Microsoft DSS cũng có thể dễ dàng cung cấp các kiểu giản đồ khác.

4.4.3.2. Lưu trữ mềm dẻo

Thu thập dữ liệu là chính sách của hầu hết các sản phẩm OLAP. Bước thu thập trước sẽ cho các kết quả có ý nghĩa: có thể dễ dàng được loại bỏ từ đầu một số tập hợp và việc độ lớn của dữ liệu lưu trữ đột ngột tăng lên.

Các ví dụ thực về ảnh hưởng của bùng nổ dữ liệu là rất nhiều. Một báo cáo chính thức từ một thử nghiệm chuẩn của một sản phẩm OLAP cho kết quả là dữ liệu bị tăng tới 240 lần và cần 2,4 GB dung lượng đĩa chỉ để lưu trữ 10 MB dữ liệu nhập vào ban đầu.

Việc cung cấp dung lượng lưu trữ thích hợp để đối phó với bùng nổ dữ liệu là rất quan trọng để triển khai OLAP và tạo ra các giới hạn khác biệt đối với một tổ chức để phân tích tất cả các mức dữ liệu được thiết kế.

Do đặc trưng của bùng nổ dữ liệu, các ứng dụng OLAP có thể hứng chịu nhiều hơn khi nguồn dữ liệu hay chi tiết dữ liệu được phân tán qua khối đa chiều. Mất dữ liệu hoặc dữ liệu không hợp lệ có thể xuất hiện rải rác trong mô hình dữ liệu OLAP. Trong trường hợp xấu nhất một sản phẩm OLAP sẽ không lưu lại một giá trị rỗng. Ví dụ, một công ty có thể không bán tất cả sản phẩm của họ trong tất cả các khu vực bởi vậy giá trị rỗng sẽ xuất hiện ở điểm giao của các sản phẩm không được bán trong khu vực cụ thể.

Dữ liệu rải rác gây nên sự khó khăn cho những người cung cấp OLAP, một số trường hợp sẽ cho kết quả là cơ sở dữ liệu sẽ lưu trữ các giá trị rỗng, mật độ lưu trữ thấp và tốn chỗ cũng như tài nguyên. Microsoft DSS không lưu trữ giá trị rỗng và kết quả là các khối không bị tăng kích cỡ. Khi điều này thường xuyên xuất hiện như yếu tố quyết định trong kiến trúc OLAP, sự khác biệt giữa các bộ xung từ nhà cung cấp là nhỏ so với sự bùng nổ dữ liệu ngày càng lớn do việc tính toán trước quá nhiều tập hợp.

Microsoft DSS đưa ra giải pháp mềm dẻo để cho phép OLAP DBA quyết định kiểu lưu trữ nào là thích hợp. Microsoft DSS hỗ trợ đầy đủ việc xử lý MOLAP, ROLAP hoặc giải pháp kết hợp trong đó các tập hợp được lưu trữ cả đa chiều và quan hệ. Ví dụ, quản trị cơ sở dữ liệu có thể lựa chọn các dữ liệu được truy cập thường xuyên như giá trị năm (Year) trong MOLAP và dữ liệu lịch sử có các vấn đề về phân cấp trong ROLAP.

Dù sao mô hình dữ liệu cơ bản là hoàn toàn vô hình đối với ứng dụng Client và người sử dụng chỉ tiếp xúc với các khối. Dù chọn giải pháp nào để thực hiện một mô hình dữ liệu (MOLAP, ROLAP hay HOLAP) thì sự tích hợp của Microsoft DSS với các cơ sở dữ liệu quan hệ vẫn là chủ đạo. Với công cụ thiết kế GUI và Wizard liên kết thẳng tới OLE DB, Microsoft DSS duy trì liên kết chặt chẽ giữa nguồn dữ liệu, siêu dữ liệu đa chiều OLAP và tập hợp của chúng.

Khi thực hiện các mô hình dữ liệu ROLAP, Microsoft DSS sẽ định nghĩa, tạo lập và duy trì tất cả các cấu trúc cơ sở dữ liệu quan hệ. Tính chất này giúp công cụ phát triển tự do trong việc thực hiện các tác vụ hoặc quản lý các truy vấn phức tạp.

Microsoft DSS cũng tối thiểu hoá vấn đề cơ bản của công nghệ OLAP là bùng nổ dữ liệu do sự thu thập trước dữ liệu quá mức. Sự bùng nổ dữ liệu OLAP là kết quả của việc thu thập trước đa chiều. Trong các hệ thống OLAP

truyền thống, dữ liệu không được thu thập trước sẽ không được dùng cho mục đích phân tích và báo cáo trừ khi được tính toán khi chạy. Việc tính toán trước và lưu trữ tất cả tập hợp phức tạp (Ví dụ: tổng của tất cả các sản phẩm và các mức sản phẩm qua tất cả các giai đoạn, qua tất cả các tổ chức, qua tất cả các kênh phân phối), các sản phẩm OLAP truyền thống sẽ gây ra một sự bùng nổ về dữ liệu một cách ô ạt.

Trái ngược với việc buộc phải tiếp cận tính toán trên tất cả các tập hợp có thể, Microsoft DSS xác định rõ các tập hợp nào cung cấp các cải tiến tốt nhất cho việc thực hiện nhưng cũng có thể tạo ra sự cân bằng giữa tốc độ hệ thống và dung lượng lưu trữ yêu cầu để quản lý các tập hợp. Nếu công cụ phát triển tính toán trước tất cả các tập hợp, dung lượng lưu trữ yêu cầu sẽ được tối đa hoá (đây là biểu hiện bùng nổ dữ liệu). Mặt khác nếu công cụ phát triển không thực hiện tính toán trước, dung lượng lưu trữ yêu cầu có thể sẽ là không nhưng việc xử lý sẽ không được cải thiện.

Trong hầu hết các trường hợp, Microsoft DSS có thể cải thiện được thêm 80% các yêu cầu truy vấn mà không cần tính toán trước trên các tập hợp: bùng nổ dữ liệu thường xuất hiện trong 20% số tập hợp còn lại. DSS phân tích mô hình siêu dữ liệu OLAP và sử dụng phương pháp đánh giá (Heuristics) để xác định tập các tập hợp là gốc của tất cả các tập hợp khác. Kết quả là Microsoft DSS nhận được dữ liệu không có tính tập hợp từ một vài tập đã tồn tại hơn là phải quét toàn bộ kho dữ liệu.

Phương pháp đánh giá của Microsoft DSS là khá tốt, nó dựa trên các mô hình toán học có thể phù hợp hoặc không phù hợp với các mô hình được sử dụng trên thực tế. Để đánh giá việc xử lý theo các mô hình thực tế, Microsoft DSS có thể ghi lại (Log) các truy vấn được gửi tới Server. Các Log này sau đó có thể được sử dụng để tìm ra tập các tập hợp mà Microsoft DSS duy trì. Ví dụ: một Wizard đơn giản có thể làm DBA yêu cầu Microsoft DSS

tạo mới một tập các tập hợp cho tất cả các truy vấn cần nhiều hơn n giây để trả lời (n có thể là 10 giây hoặc hơn).

Trong nhiều tổ chức, thời gian thực hiện được quan tâm hơn dung lượng đĩa cần cho lưu trữ. Người ta có thể chịu bỏ tiền mua thêm dung lượng lưu trữ hơn là phải mất hàng ngày để có được dữ liệu. Giải pháp Microsoft DSS về vấn đề bùng nổ dữ liệu đầu sao cũng giảm bớt được thời gian yêu cầu cho quá trình nạp dữ liệu ban đầu và các cập nhật có lợi như giảm thiểu khối lượng đĩa lưu trữ cần thiết. Nếu một ứng dụng bắt đầu với 10 GB kho dữ liệu và sinh ra 10 GB các tập hợp, thời gian của quá trình yêu cầu là một phần nhỏ của tiến trình bùng nổ toàn phần tập các tập hợp.

Microsoft DSS cũng tạo ra một cách tiếp cận mới với vấn đề dữ liệu bị dàn trải. Khi các chi tiết thực hiện nội tại được sở hữu, các kết quả thực hiện trên cả MOLAP và ROLAP quản lý lưu trữ cực kỳ tốt và trên thực tế các cơ sở dữ liệu với yêu cầu lưu trữ OLAP là nhỏ hơn dữ liệu chi tiết ban đầu.

Các khối ảo có thể được sử dụng trong bất kỳ tình huống nào, nơi mà người sử dụng muốn có một khung nhìn về thông tin kết hợp từ hai khối khác nhau chia sẻ một vài chiều chung. Tương tự trong khái niệm về khung nhìn, các khối ảo là hai hay nhiều khối liên kết theo một hoặc nhiều chiều chung. Lợi thế của các khối ảo áp dụng trong các tình huống dữ liệu dàn trải là vấn đề đáng kể. Ví dụ, một khối chứa các đơn vị đo cho giá bán một sản phẩm và giá bán thực tế có thể có một đơn vị đo bằng bảng giá để thực hiện giảm giá nhưng giá trị bảng giá có thể được lặp lại nhiều lần. Bằng cách xây dựng khối bảng giá được kết hợp trong khối ảo với thông tin giá bán thực tế, DBA có thể loại trừ khá nhiều dữ liệu dư thừa. Khả năng tạo các khối ảo có nghĩa là nhiều giá trị không cần thiết có thể được loại trừ khỏi lưu trữ dữ liệu OLAP.

Việc thực hiện cụ thể một ứng dụng OLAP cần tính năng của một vài yếu tố bao gồm kích cỡ cơ sở dữ liệu, sức mạnh phần cứng hệ thống và dung

lượng đĩa cấp cho các dữ liệu tập hợp trước. Các ứng dụng Microsoft DSS đáp ứng được hầu hết các yêu cầu truy vấn trong thời gian dưới 5 giây và gần như tất cả các truy vấn trong 10 giây.

Các bổ xung mới của Microsoft DSS là các khối phân chia, làm cho công nghệ này được nâng lên một bậc. Khối phân chia có thể là một khối Logic của dữ liệu được dàn trải qua các khối vật lý thậm chí rải rác trên các Server riêng biệt. Để trả lời các truy vấn từ người sử dụng, Microsoft DSS phân tán các truy vấn đến các Server, kích hoạt dữ liệu và có thể nhận lại song song.

Ví dụ, trường hợp một ứng dụng dò tìm các cuộc điện thoại trong 10 khu vực địa lý nơi có hàng triệu cuộc điện thoại mỗi ngày. Do mục đích của phân tích này, dữ liệu có thể nằm trong 10 Server mà mỗi cái chứa dữ liệu một khu vực riêng và thường chỉ có một khối dữ liệu Logic đơn giản. Để trả lời yêu cầu từ người sử dụng về thông tin này, Microsoft DSS chuyển các yêu cầu truy vấn thích hợp cho mỗi Server và nhận lại các kết quả riêng lẻ. Mỗi cơ sở dữ liệu riêng biệt cũng sẵn sàng cho các truy xuất riêng biệt để phân tích chỉ thông tin tìm kiếm cho từng khu vực. Khả năng của Microsoft DSS quản lý hiệu quả dữ liệu phân chia trên các Server làm cho công nghệ này tiến xa hơn.

4.4.3.3. Chuyển thông tin tới người sử dụng

Công nghệ OLAP Server được kết hợp chặt chẽ với công nghệ Client, có nghĩa là khách hàng có rất ít lựa chọn cho sản phẩm. Điều này dẫn đến giá của giải pháp cao và thường có các lựa chọn không tương xứng cho các ứng dụng Client/Server và chuyển tải thông tin OLAP dựa trên nền Web. Vài năm trước trong thị trường cơ sở dữ liệu quan hệ, một giao diện chung là cần thiết để phát triển trong lựa chọn các ứng dụng và cơ sở dữ liệu là ODBC.

Các công cụ OLAP lần đầu tiên được áp dụng vào năm 1996 khi nhóm

các nhà cung cấp gọi là Hội đồng OLAP công bố chuẩn MDAPI để mở ra một thị trường cho sự tham gia của các nhà cung cấp lớn. Cộng đồng các nhà cung cấp gồm các thành viên Hội đồng OLAP.

Nhận ra sự cần thiết phải có một chuẩn thống nhất, Microsoft đã đưa ra một định nghĩa mở rộng của API dùng để truy nhập dữ liệu OLE DB đang tồn tại bao gồm các tính năng đa chiều. Microsoft đã đưa ra hai phác thảo của API, tìm kiếm các ý kiến của các nhà cung cấp và đưa ra phiên bản cuối cùng đã được chứng thực bởi 18 nhà cung cấp trong bản beta. Ngày nay OLE DB cho OLAP API đã được hỗ trợ từ 30 nhà cung cấp trong số này có tất cả các thành viên hiện tại của Hội đồng OLAP. Rất nhiều trong số này đã sẵn sàng đưa ra các sản phẩm dạng beta chi tiết cho người sử dụng.

Nhiều nhà phân tích kinh doanh cần dữ liệu phân tích đa chiều ngay cả khi đã ngắt kết nối mạng trong lúc họ đang đi công tác chẳng hạn. Người sử dụng lưu động thường muốn xem và phân tích chỉ những lát nhỏ của khối, chẳng hạn như trường hợp người quản lý bán hàng muốn biết tổng doanh thu cho từng khu vực khi đến thăm các văn phòng công ty ở từng khu vực. DOLAP (Desktop OLAP) là công nghệ không đòi hỏi một máy chủ chia sẻ để truy cập dữ liệu đa chiều.

Hầu hết các công nghệ máy chủ OLAP ngày nay không cung cấp việc tạo các khối DOLAP trong suốt. Công việc này được dành cho các nỗ lực phát triển đòi hỏi sự tập trung hoặc để các công cụ Client sẽ thêm vào tính năng OLAP hỗ trợ sử dụng trên máy để bàn.

Hầu hết công cụ hiện thị hiện nay đều là dùng các dạng thông tin trên Web và OLAP không phải là ngoại lệ. Để giảm chi phí cho mỗi người sử dụng các ứng dụng OLAP, trình duyệt Web hứa hẹn mở ra một cách truy cập đa chiều. Hiện tại đã có một vài sản phẩm và công cụ chuyển giao dữ liệu OLAP qua mạng nội bộ nhưng chưa có một cơ chế dễ dàng cho công cụ phát

triển ứng dụng để tạo ra các công cụ hiển thị OLAP theo ý muốn.

DSS Server lưu cả các truy vấn của người sử dụng, siêu dữ liệu cũng như dữ liệu. Việc xác định các truy vấn đã lưu và siêu dữ liệu làm Microsoft DSS có khả năng trả lời các truy vấn mới bằng cách tính toán dữ liệu đã được lưu thay vì truy xuất từ đĩa. Ví dụ, một người sử dụng yêu cầu dữ liệu bán hàng trong các tháng 1, 2 và 3, người khác yêu cầu dữ liệu bán hàng trong quý I, khi đó Microsoft DSS có thể lấy tổng số từ tháng 1 đến 3 từ RAM nhanh hơn tìm dữ liệu quý I từ đĩa. Không có gì khác biệt từ hầu hết các máy chủ OLAP khác.

Microsoft DSS là giải pháp duy nhất có khả năng cung cấp nhiều tính năng cho phía Client. Mọi Client kết nối tới DSS Server sẽ dùng một thành phần gọi là Microsoft PivotTable Service. PivotTable Service có tính năng điều khiển và quản lý kết nối giữa Client và Server. PivotTable Service chia sẻ nhiều đoạn mã với DSS Server, chuyển thành phần tính toán đa chiều của Server, Cache và quản lý truy vấn đến Client. Kết quả sẽ cho ta một mô hình quản lý dữ liệu Client/Server mới có khả năng thực hiện yêu cầu cao và giảm thiểu lưu thông mạng. Điều này đem lại lợi ích là dung lượng đĩa yêu cầu của PivotTable Service chỉ xấp xỉ 2 MB và bộ nhớ yêu cầu là 500 KB.

Kiến trúc DSS Client/Server thông minh này có khả năng xác định cách thức để trả lời một yêu cầu từ người sử dụng nhanh nhất và loại trừ sự dư thừa trong lưu thông mạng. Điểm cốt lõi của kiến trúc này là chia sẻ siêu dữ liệu giữa Client và Server. Khi người sử dụng yêu cầu thông tin từ Server, cả dữ liệu và siêu dữ liệu (định nghĩa bởi cấu trúc khối) được tải xuống Client. Siêu dữ liệu khối trên Client cho phép PivotTable Service quyết định yêu cầu nào cần được trả lại cho Server.

Xét một ví dụ khi ai đó cần xem xét dữ liệu bán hàng trong 3 tháng. Cho rằng cả DSS Server và ứng dụng phía Client đã được kích hoạt. Khi

người sử dụng yêu cầu dữ liệu bán hàng của tháng 1, 2 và 3, dữ liệu sẽ được lưu trên Cache ở cả 2 phía Server và Client. Nếu sau đó người sử dụng yêu cầu dữ liệu trong quý I, PivotTable Service sẽ lấy kết quả trên máy Client mà không cần gửi yêu cầu tới Server. Nếu người sử dụng sau đó lại yêu cầu dữ liệu quý I của năm đó để so sánh với năm ngoái, PivotTable Service sẽ đủ thông minh và chỉ truy cập vào dữ liệu năm ngoái trên Server.

PivotTable Service cũng cung cấp cơ chế cho việc sử dụng lưu động. Các phần của một khối được định nghĩa và truy nhập từ Server có thể được lưu trên Client để dùng cho truy nhập sau này khi ngắt kết nối mạng. Theo cách này, những người sử dụng là các nhà kinh doanh có thể dùng những phần trong cơ sở dữ liệu của họ khi đang lưu động và có các phân tích hoàn chỉnh khi không cần phải ngồi tại văn phòng. Hơn nữa, PivotTable Service cũng cho phép người sử dụng tạo các mô hình OLAP đơn giản trên máy Client, truy nhập các thông tin trong nguồn dữ liệu OLE DB từ các file cho đến các cơ sở dữ liệu trên máy để bàn.

PivotTable Service cũng cung cấp kết nối cho các ứng dụng dựa trên Web. Khi OLE DB cho OLAP là một giao diện chương trình mức thấp, một ActiveX Data Object (ADO) được phát triển để cung cấp truy nhập dữ liệu đa chiều. Nó sẽ gọi ADO/MD, có thể dễ dàng sử dụng để tạo ActiveX điều khiển trong Microsoft Visual Basic để xem, lập báo cáo về dữ liệu DSS từ trang Web. ADO/MD là công cụ lập trình ứng dụng hợp tác cho phép truy xuất tới tất cả các tính năng DSS.

4.4.3.4. Khả năng của các công cụ OLAP

Microsoft nhận thấy OLAP chính là sự mở rộng tự nhiên của công nghệ cơ sở dữ liệu và đã đưa DSS vào SQL Server. SQL Server bao gồm rất nhiều đặc tính bổ xung khác hỗ trợ tiến trình kho dữ liệu bao gồm:

- Visual Database Tools: dùng để tạo các giản đồ cơ sở dữ liệu.

- Data Transformation Services: dùng để trích xuất và chuyển dữ liệu tác nghiệp vào kho dữ liệu.
- Microsoft Repository: cung cấp tính năng lưu trữ siêu dữ liệu thống nhất trong SQL Server.

Trong phiên bản sắp tới của Microsoft Office, Microsoft sẽ đưa ra nhiều năng lực OLAP tương thích với DSS. Xây dựng OLE DB cho giao diện OLAP, năng lực mới này sẽ hỗ trợ truy nhập trực tiếp vào DSS Server, khả năng sử dụng khi đã ngắt mạng và tính năng truy nhập qua giao diện Web. Đầu tiên, trong MS Excel, tính năng PivotTable mới có thể kết nối bảng tính MS Excel với OLE DB cho việc cung cấp dữ liệu OLAP. Trong trường hợp này, nó sẽ có thêm các năng lực như khả năng tạo khối cục bộ từ một phần của khối trên Server.

Công cụ PivotTable hiện tại trên MS Excel được thay thế với công cụ PivotTable Service từ Microsoft DSS cho phép người dùng máy để bàn nhiều khả năng mềm dẻo hơn khi tạo các cấu trúc đa chiều mà không phải giới hạn bộ nhớ.

Một số tính năng mới được gọi là Web Components trong phiên bản sắp tới của Office sẽ cung cấp việc truy xuất OLAP cơ bản và tính năng lập bảng qua ActiveX để dễ dàng thêm vào các trang Web. Xây dựng OLE DB cho OLAP có thể được sử dụng với bất kỳ cung cấp OLAP tương thích trong Microsoft DSS. Với việc cải tiến và đưa ra các tính năng phân tích và hiển thị OLAP cơ bản, Microsoft đang cố gắng tạo ra một bước chuyển của xu thế kho dữ liệu để nhằm giảm giá công cụ ra quyết định.

Sự kết hợp OLE DB và OLAP cũng tạo ra sự đa dạng của các sản phẩm từ các nhà cung cấp phần mềm cho việc sử dụng DSS. Do vậy nhiều công cụ Client mới được xây dựng để truy nhập thông tin trong DSS, người sử dụng có nhiều lựa chọn cho ứng dụng.

4.5. Hướng nghiên cứu phát triển: Hệ trợ giúp quyết định phân tán

Khi xây dựng các hệ trợ giúp quyết định, nói chung người ta cũng đã quan tâm đến các công nghệ phân tán nhưng việc tính toán phân tán thường bị hạn chế ở các tác vụ phụ trợ. Các tính toán phân tán có khả năng tích hợp hệ trợ giúp quyết định với các hệ thống E-mail, nhiều nghiên cứu về hệ trợ giúp quyết định đã tận dụng các lợi thế của công nghệ mạng và phân tán để mở rộng với nhiều công cụ như Video tương tác hai chiều, Chat, Bulletin Board... Nói chung, công nghệ phân tán được ứng dụng trong các hệ trợ giúp quyết định thường được xem như một cách thuận tiện để truy cập vào các cơ sở dữ liệu bên ngoài hay truy cập vào hệ thống Client/Server. Các hệ trợ giúp quyết định dựa trên Web được cung cấp như các thành phần Client để kết nối với Server có Hosting ứng dụng trợ giúp quyết định.

Tuy nhiên, tiềm năng to lớn của các công nghệ phân tán không chỉ được sử dụng bởi các nhà thiết kế trong việc mở rộng các hệ trợ giúp quyết định truyền thống với các công cụ hỗ trợ, nó còn có thể thay thế các chiến lược hỗ trợ quyết định kiểu hợp tác hay cộng tác phân tán mới có ảnh hưởng tới cấu trúc lõi của các hệ trợ giúp quyết định. “Về cơ bản, nhiều hệ thống được liệt vào loại hệ trợ giúp quyết định phân tán đa phần vẫn là các ứng dụng tập trung cho ra các kết quả trong mô hình Client/Server. Tuy nhiên, hệ trợ giúp quyết định phân tán gần gũi hơn với các quá trình có nhiều người tham gia nếu nó có thể tránh được cấu trúc tập trung và thực hiện các chức năng hỗ trợ và ra quyết định cho người sử dụng”[4].

Hệ thống cần phải hỗ trợ càng nhiều càng tốt cho các quá trình quyết định mà không áp đặt bất kỳ sự ép buộc nào đối với những người ra quyết định. Về cơ bản, hai yêu cầu cần thiết để coi hệ trợ giúp quyết định là sự mở rộng tự nhiên của các năng lực ra quyết định là:

- Hệ trợ giúp quyết định cần bắt chước hành động ra quyết định của

người ra quyết định,

- Hệ trợ giúp quyết định phải luôn gần gũi với người ra quyết định, sẵn sàng được sử dụng bất kỳ đâu, bất kỳ lúc nào.

Bắt chước các hành vi ra quyết định về cơ bản là nhiệm vụ chính của bất kỳ hệ trợ giúp quyết định nào. Tuy nhiên, để hoàn thành nhiệm vụ này tốt nhất điều quan trọng là cần hiểu rằng những người ra quyết định hiện nay hiếm khi ra quyết định một mình. Vì vậy hệ trợ giúp quyết định cần bắt chước các hành động ra quyết định với nhiều người tham gia. Việc chuyển từ hệ trợ giúp quyết định với một người sử dụng sang hệ trợ giúp quyết định nhiều người sử dụng cần được chia thành hai hướng riêng:

- Thiết kế kiến trúc phân tán mới cho hệ trợ giúp quyết định,
- Hỗ trợ kết hợp các tính năng khác nhau của hệ trợ giúp quyết định phân tán.

Để bắt chước hành động ra quyết định, điều vô cùng quan trọng là phải giải phóng người ra quyết định khỏi các thúc ép về kỹ thuật, nói cách khác người ra quyết định cần tránh các nhiệm vụ kỹ thuật kiểu cấu hình hoặc sửa lỗi. Các tình huống về y tế là các ví dụ tốt cho yêu cầu này: các hệ thống có thể kết nối với bất kỳ loại thiết bị y tế nào mà không đòi hỏi các thầy thuốc phải bận tâm về cấu hình dữ liệu hay phần mềm. Để đạt được mục tiêu này việc sử dụng hệ trợ giúp quyết định phân tán là giải pháp linh hoạt nhất.

Để đảm bảo rằng hệ trợ giúp quyết định luôn gần gũi với người ra quyết định và những người hỗ trợ, sẵn sàng được sử dụng bất kỳ lúc nào, bất kỳ đâu, trên bất kỳ thiết bị nào, người ra quyết định và những người hỗ trợ có thể làm việc với hệ thống từ bất kỳ đâu thì việc truy cập thẳng vào hệ trợ giúp quyết định là luôn cần thiết để tạo sự linh hoạt. Mô hình mạng tập trung không phù hợp cho việc hỗ trợ quá trình ra quyết định với đông người tham gia. Hơn nữa, khi hệ trợ giúp quyết định phân tán chạy ở mọi nơi thì nó không

được phụ thuộc vào vị trí. Thêm vào đó trong một số môi trường làm việc, những người ra quyết định và những người hỗ trợ sẽ làm việc độc nên hệ trợ giúp quyết định phân tán cần có tính năng chạy cả ở chế độ Offline. Nói cách khác, hệ trợ giúp quyết định phân tán không phụ thuộc vào bất kỳ máy chủ trung tâm nào.

Mô hình liên kết

Các kiến trúc phân tán truyền thống được sử dụng trong quá trình thiết kế hệ trợ giúp quyết định phân tán thường là kiểu mô hình mạng tập trung. Mặc dù lợi thế của các hệ thống tập trung là sự đơn giản nhưng hạn chế chính của chúng là tập trung một chỗ. Do vậy hệ thống không có khả năng chịu lỗi, không có sự độc lập về vị trí, khó mở rộng và cân bằng. Sự phụ thuộc vào các nguồn tập trung là không thể chấp nhận được đối với cách tiếp cận của chúng ta. Nói cách khác, kiến trúc không tập trung là mô hình thích hợp để đáp ứng các yêu cầu. Ta có thể đưa ra một mô hình liên kết tận dụng lợi thế của mô hình mạng không tập trung. Trong mô hình này, mỗi người sử dụng hệ trợ giúp quyết định nhận được môi trường làm việc phù hợp với vai trò và có thể liên tục thay đổi. Môi trường làm việc này chứa cả thông tin của người sử dụng hệ trợ giúp quyết định và các dịch vụ cung cấp cho người sử dụng trong việc liên kết với các thành viên khác trong quá trình hỗ trợ quyết định. Môi trường làm việc là Node hay Peer có thể giao tiếp với các Node khác khá linh hoạt. Mỗi môi trường làm việc chứa tối thiểu các dịch vụ hạ tầng cần để chạy hệ trợ giúp quyết định. Các dịch vụ này đều cung cấp các tính năng cơ bản như tra cứu (Lookup), quản lý giao dịch, liên kết các quá trình...

Khi người sử dụng được quyền tham gia vào hệ trợ giúp quyết định, môi trường làm việc của họ sẽ thông báo sự hiện diện của họ trên mạng. Môi trường làm việc của những người sử dụng đã tồn tại sẽ thông báo và hỏi môi trường làm việc mới để chia sẻ các chính sách. Thông tin trao đổi được hiển

thị trong thời gian thật trên màn hình của thiết bị. Mỗi người sử dụng hệ trợ giúp quyết định biết về sự hiện diện của người sử dụng khác và có thể truy cập vào thông tin chung. Theo nghĩa này, cộng đồng sử dụng vừa linh hoạt vừa tự hình thành. Môi trường làm việc không chỉ chứa dữ liệu mà còn chứa các dịch vụ mà tất cả các thành viên đều có thể sử dụng. Trong khi mỗi môi trường làm việc chứa một bộ tối thiểu các dịch vụ hạ tầng cần để chạy hệ trợ giúp quyết định thì các dịch vụ mở rộng chỉ được cung cấp cho các Roles đặc trưng.

Hệ thống còn có thể tự quản trị. Tính mềm dẻo cho phép các môi trường làm việc phát hiện một cách tự động sự biến mất của môi trường làm việc khác và cập nhật thông tin tới người sử dụng. Các môi trường làm việc còn lại sẽ loại bỏ các thông tin liên quan đến các môi trường vừa biến mất và người sử dụng sẽ được biết về sự vắng mặt của người sử dụng khác. Thông tin này sẽ được hiển thị trở lại ngay khi người sử dụng quay lại mạng. Môi trường làm việc bị biến mất cho thấy một ví dụ về dịch vụ hạ tầng cần để chạy hệ trợ giúp quyết định, điều này cho thấy cộng đồng sử dụng có tính tự sửa lỗi.

Trong mô hình liên kết, người sử dụng hệ trợ giúp quyết định có thể làm việc cục bộ, đơn lẻ với môi trường làm việc của người đó. Tuy nhiên, cơ sở thông tin của toàn bộ hệ thống tăng tỉ lệ trực tiếp với số người sử dụng chia sẻ các môi trường làm việc. Mỗi người sử dụng được trang bị các năng lực cần để sử dụng trong chế độ cục bộ và phân tán (các năng lực mạng và dịch vụ hạ tầng). Thông tin của mỗi người sử dụng được quản lý theo cách mà những người tham gia kết nối có thể chia sẻ, nếu họ được cho phép. Sự khác biệt quan trọng nhất giữa kiến trúc này và kiến trúc tập trung là sự vắng mặt của các máy chủ trung tâm và/hoặc các hệ thống quản trị cơ sở dữ liệu.

Chương V. Xây dựng hệ thống trợ giúp quyết định dựa vào dữ liệu bằng công cụ Analysis Services

5.1. Mục tiêu của hệ thống

Hệ thống sẽ cung cấp một công cụ để xử lý mô hình OLAP của một ứng dụng cụ thể: cho phép tổ chức CSDL đa chiều và trên CSDL đã được thiết lập này cung cấp khả năng phân tích dữ liệu một cách dễ dàng, linh hoạt và nhanh chóng nhằm trợ giúp cho việc ra quyết định.

Để thực hiện được những chức năng này, luận văn sử dụng công cụ Analysis Services của Microsoft SQL Server 2000 với mô hình đối tượng DSO (Decision Support Objects) cho phép tạo ra những ứng dụng quản lý: Databases, DataSources, Dimensions, Cubes và Roles. Ngoài ra ta cũng có thể quản trị an ninh, xử lý Cubes và khai thác mẫu...

5.2. Yêu cầu về hệ thống

Hệ thống phải dễ dàng cho người sử dụng, không cần thiết họ phải là người hiểu biết nhiều về máy tính. Có thể ứng dụng để trợ giúp quyết định trong một phạm vi rộng các bài toán, trong các lĩnh vực khác nhau mà dữ liệu có khuynh hướng đa chiều.

Các quyết định được thực hiện của người sử dụng là hoàn toàn khách quan và đủ căn cứ, dựa trên tiềm tàng của khối lượng lớn các dữ liệu cơ sở sẵn có (là kết quả của các hoạt động tác nghiệp).

Hệ thống cung cấp một khả năng phân tích khá mạnh, cho phép người sử dụng có thể:

- Cắt lát các khối đa chiều theo các mức, các chiều tùy chọn.
- Khoan sâu xuống các mức dữ liệu chi tiết.
- Cuộn lên các mức dữ liệu tổng hợp hơn (theo phân cấp chiều).

Hệ thống có thể giúp người sử dụng dễ dàng mô tả, thay đổi yêu cầu

phân tích dữ liệu của mình ở các giao diện và màn hình thể hiện kết quả đa chiều. Với hệ trợ giúp quyết định này, người sử dụng sẽ dễ dàng, thuận tiện và nhanh chóng trong việc ra các quyết định cho bài toán ứng dụng thực tế của mình bằng cách thực hiện các công việc sau:

(1). Xác định yêu cầu của bài toán ứng dụng cụ thể.

(2). Xây dựng mô hình OLAP cho bài toán, hướng đến việc giải quyết các yêu cầu này. Đây là công việc phụ thuộc chủ yếu vào nội dung, yêu cầu của bài toán và rất dễ thực hiện khi sử dụng hệ thống.

(3). Thực hiện một vài thao tác đơn giản để tạo lập cơ sở dữ liệu đa chiều.

(4). Thực hiện việc phân tích dữ liệu thuận tiện, linh hoạt và trực quan. Người sử dụng dễ dàng, linh động trong việc mô tả, thay đổi các yêu cầu truy vấn thông tin (trên bất kỳ mức nào, trên bất kỳ khía cạnh nào của các dữ liệu chứa trong khối) bằng các thao tác chọn lựa đơn giản trên một màn hình thân thiện.

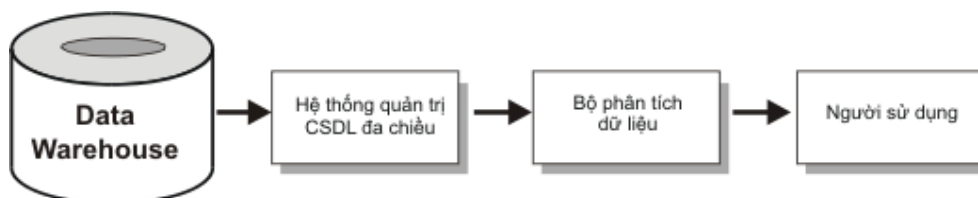
Người dùng sẽ thấy những kết quả phân tích mong muốn dưới dạng các bảng báo cáo tổng hợp rất tiện lợi và linh hoạt trong việc uốn nắn hoạt động phân tích dữ liệu của mình để làm sáng tỏ dần tất cả các thông tin cần thiết, đủ để dễ dàng ra các quyết định tốt nhất.

5.3. Chức năng chính của hệ thống

Một hệ trợ giúp quyết định dựa vào dữ liệu sử dụng hai thành tố chính là Kho dữ liệu và OLAP sẽ có đầu vào là các dữ liệu thu được từ các hoạt động tác nghiệp (của các hệ thống tác nghiệp) và đầu ra là các báo cáo, thông tin phân tích dồi dào và phong phú được hiển thị một cách trực quan và linh hoạt. Hoạt động phân tích có thể được tiến hành một cách dễ dàng, thuận tiện và nhanh chóng. Dựa trên những kết quả của các phân tích này, các nhà quản lý sẽ có đủ các thông tin cần thiết để có thể ra được những quyết định hợp lý

nhất.

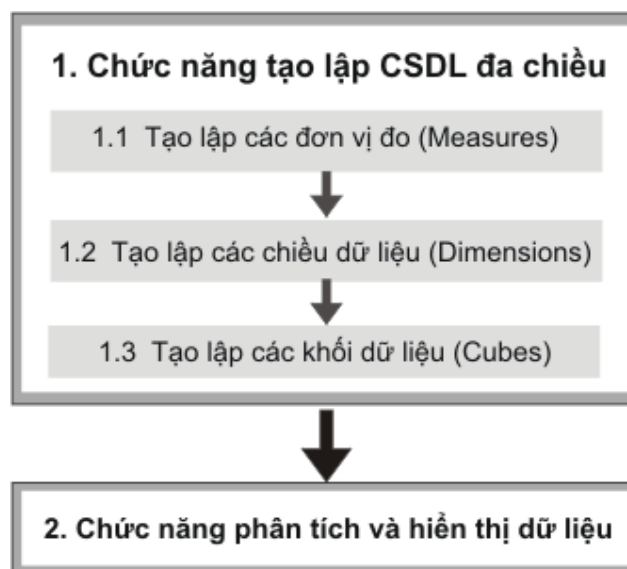
Dưới đây là sơ đồ thể hiện kiến trúc của hệ thống trợ giúp quyết định dựa vào dữ liệu:



Hình 5.1. Kiến trúc hệ trợ giúp quyết định dựa vào dữ liệu

Hệ thống sẽ được giới thiệu tập chung vào những khối chức năng quan trọng sau:

- Chức năng tạo lập CSDL đa chiều (Cubes).
- Chức năng phân tích, hiển thị dữ liệu.



Hình 5.2. Chức năng hệ trợ giúp quyết định dựa vào dữ liệu

5.3.1. Chức năng tạo lập CSDL đa chiều

Là công cụ giúp tạo lập nên các khối và cấu trúc các chiều từ mô hình OLAP của ứng dụng cụ thể. Dữ liệu nguồn được chứa trong bảng Fact. Để tạo khối cần thực hiện các bước như sau:

- Chọn bảng Fact cho khối cần tạo.
- Tạo các đơn vị đo (Measures): chọn các cột (có kiểu dữ liệu là kiểu số) trong bảng Fact để làm các đơn vị đo.
- Tạo lập các chiều dữ liệu: mỗi chiều tương ứng với một bảng (bảng Dimension) trong sơ đồ hình sao mà bảng Fact ở trên là bảng trung tâm. Trong mỗi bảng chiều (Dimension) chọn các cột làm các mức (Level) của chiều đó. Chúng ta cần chọn các mức sao cho có thể đáp ứng được việc tham chiếu “Roll_up”, “Drill_down” theo mức.
- Xử lý khối.

Ngoài ra chức năng tạo khối ảo giúp ta tạo ra một khối mới từ các khối đã có bằng cách sử dụng lại các đơn vị đo và các chiều của các khối cũ mà không cần tốn thêm không gian nhớ:

- Chọn các khối mà ta cần tạo khối ảo từ nó.
- Chọn các đơn vị đo cho khối ảo.
- Chọn các chiều cho khối ảo.
- Xử lý khối.

5.3.2. Chức năng phân tích và hiển thị dữ liệu

Sau khi khối đã được tạo, dữ liệu trong khối sẽ được phân tích qua thao tác xử lý khối. Sau đó người sử dụng có thể tham khảo được kết quả phân tích một cách dễ dàng bằng cách thực hiện các thao tác “Roll_up”, “Drill_down” theo mức trên mỗi chiều của khối đã chọn. Với chức năng này ta có thể:

- Cắt lát các khối đa chiều theo các mức, các chiều tùy chọn.

- Khoan sâu xuống các mức dữ liệu chi tiết khi cần.
- Cuộn lên các mức dữ liệu tổng hợp hơn (theo phân cấp chiều) nếu muốn.
- Tạo ra sự linh hoạt và tiện ích cho người sử dụng: dễ dàng mô tả, thay đổi yêu cầu phân tích dữ liệu.

5.4. Giới thiệu hệ thống

5.4.1. Khởi động Analysis Manager

Khi cài đặt MS SQL 2000 Analysis Services, Analysis Manager cũng được cài đặt như một công cụ quản trị cho Analysis Server. Giống như Microsoft SQL Server Enterprise Manager, giao diện khi ta sử dụng Analysis Manager cũng được tích hợp các tiện ích cho việc quản trị. Một cấu trúc cây sẽ xuất hiện bên trái của giao diện quản trị như cấu trúc phân cấp thường thấy trong Windows Explorer.

Khởi động Analysis Manager, ta sẽ thấy giao diện Analysis Manager với tất cả các Analysis Server được thiết lập trong môi trường phía bên trái giao diện. Tên của Analysis Server được tự động gán khi cài đặt MS SQL Server 2000 và giống tên của máy.

5.4.2. Cài đặt cơ sở dữ liệu và nguồn dữ liệu (Database & Data Source)

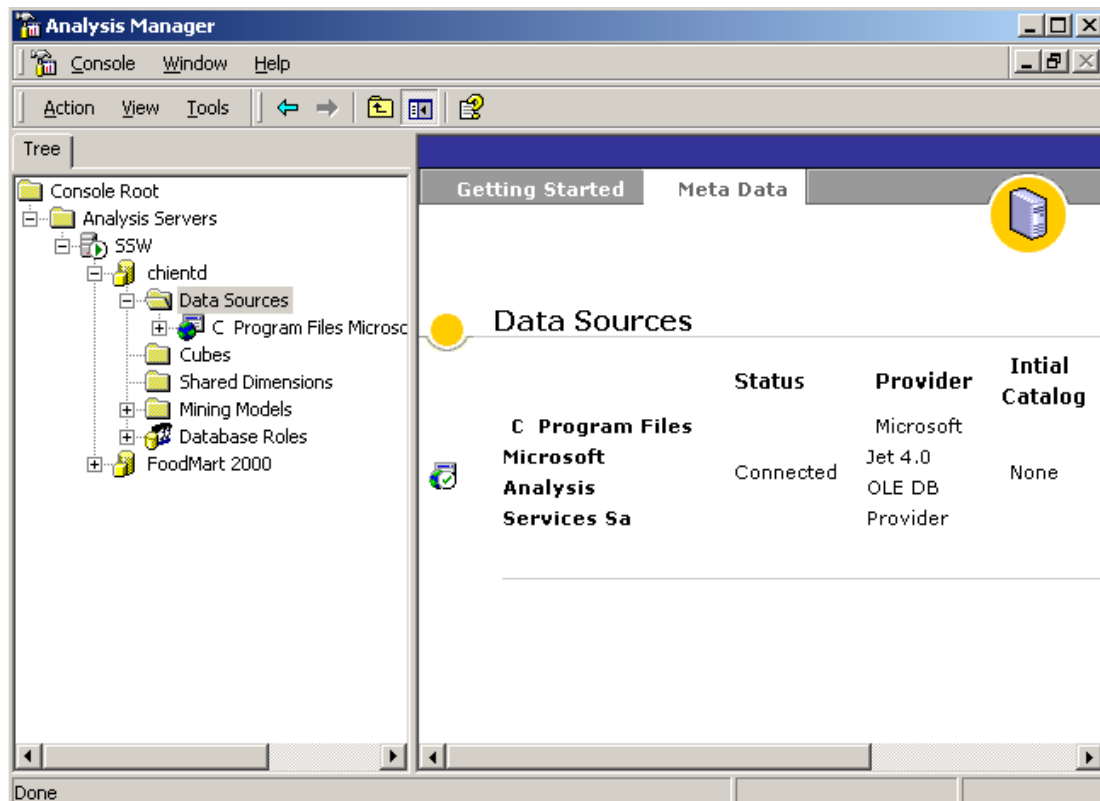
Trước khi thiết kế một khối mới, ta cần cài đặt cơ sở dữ liệu riêng biệt trong Analysis Manager gọi là cơ sở dữ liệu OLAP. Với cơ sở dữ liệu OLAP được tạo ta có thể tổ chức các khối, các Roles, nguồn dữ liệu, các chiều chia sẻ và các đối tượng khác. Trong các đối tượng khác, cơ sở dữ liệu có thể chứa đa khối mà mỗi khối có nguồn dữ liệu riêng. Để cài đặt cơ sở dữ liệu, ta sẽ liên kết nguồn dữ liệu với cơ sở dữ liệu trước khi tiến hành cấu trúc khối.

Nguồn dữ liệu cần phải tồn tại qua các OLE DB được sử dụng, với Microsoft OLE DB Provider for ODBC Drivers thì có thể chọn các hệ thống

cũ hơn. Chọn Microsoft Jet 4.0 OLE DB Provider. OLE DB Drivers for ODBC thường cung cấp cách thứ cấp để kết nối, OLE DB Driver tối ưu hơn cho việc lựa chọn.

5.4.3. Tạo khối

- Chọn lệnh ‘New Database’ để tạo một Database để lưu, chứa các khối và đặt tên cho Database này (ví dụ ‘chientd’). Chọn ‘DataSource’ để tạo nguồn dữ liệu cho các khối trong Database (ví dụ chọn bộ dữ liệu có sẵn của Microsoft là FoodMart 2000).

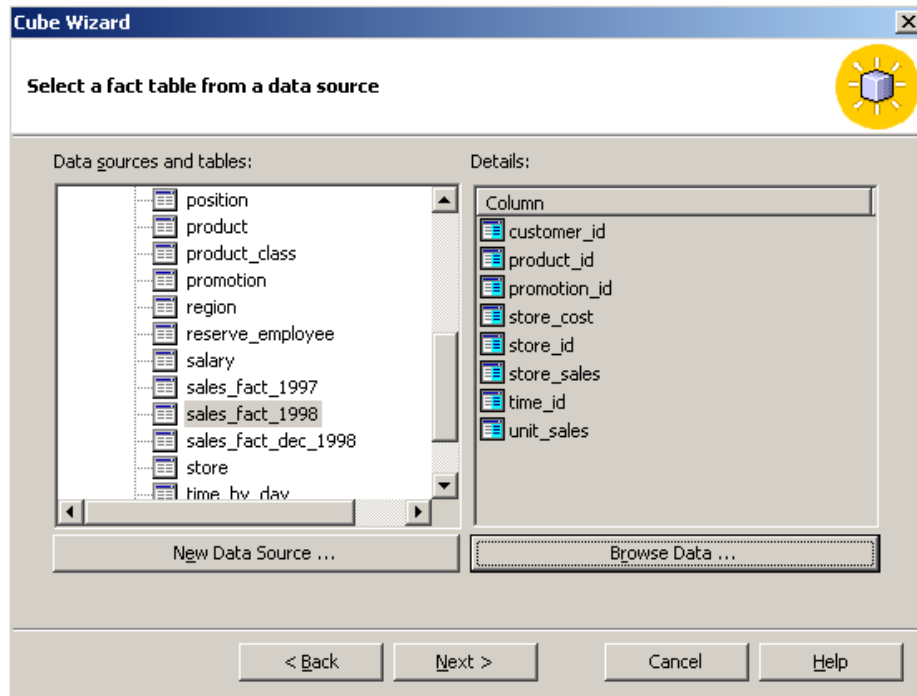


Hình 5.3. Tạo DataSource cho các khối trong Database

- Chọn ‘New Cube’ để tạo các khối.
- Chọn bảng Fact cho khối: Chọn bảng Fact trong Datasource ở cửa sổ bên trái, việc chọn bảng nào là tùy thuộc vào yêu cầu truy vấn (ví dụ

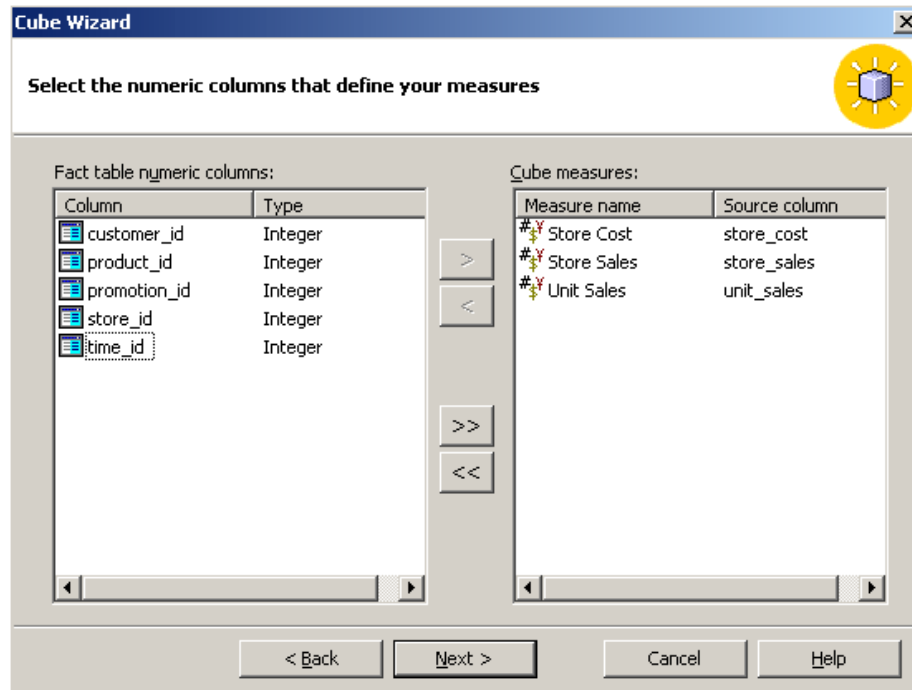
chọn bảng 'sales_fact_1998').

- Cửa sổ bên phải hiển thị các trường dữ liệu của bảng Fact đã chọn. Để kiểm tra dữ liệu của bảng Fact, kích vào nút 'Browse Data'. Để tạo mới hoặc tạo thêm nguồn dữ liệu cho khối chọn nút 'New Data Source'.

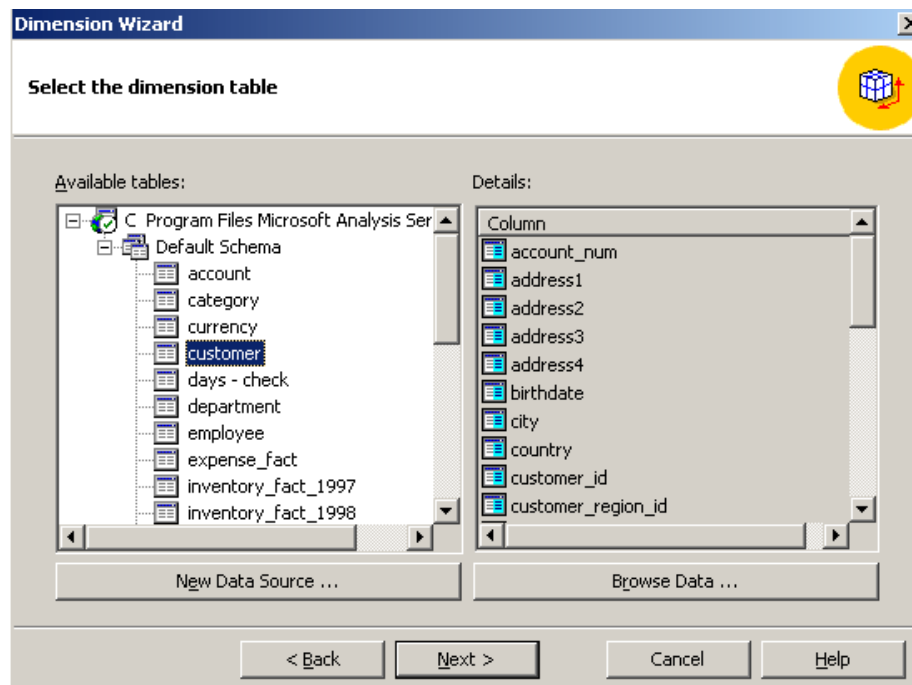


Hình 5.4. Chọn bảng Fact

- Tạo các đơn vị đo: Chọn các cột số của bảng Fact (sales_fact_1998) ở cửa sổ trái để làm các đơn vị đo, cửa sổ bên phải hiển thị các cột số được chọn.
- Tạo các chiều: Để định nghĩa một chiều ta cũng chọn bảng chiều trong danh sách các bảng của nguồn dữ liệu ở cửa sổ trái (ví dụ chọn bảng customer).

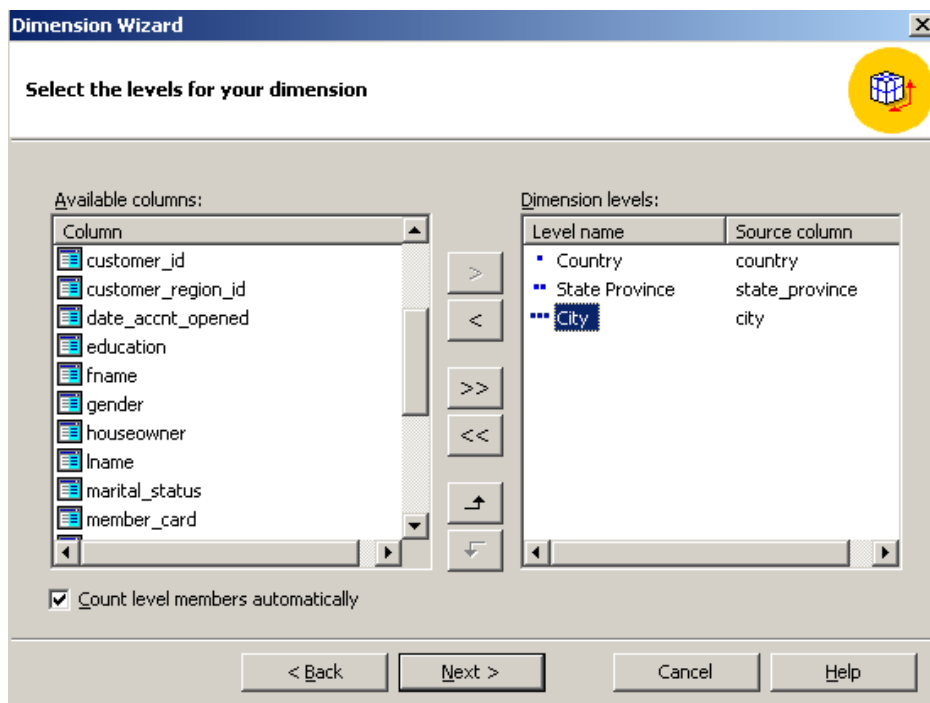


Hình 5.5. Chọn đơn vị đo



Hình 5.6. Tạo chiều

- Tạo các mức cho mỗi chiều: Chọn các cột của bảng chiều ở cửa sổ bên trái làm các mức của chiều. Lưu ý ta cần chọn các mức sao cho có thể đáp ứng được việc tham chiếu “Roll_up”, “Drill_down” theo mức (ví dụ với bảng chiều ‘customer’ trên ta chọn ba cột ‘country’, ‘state_province’, ‘city’ làm ba mức của chiều).



Hình 5.7. Chọn các mức của chiều

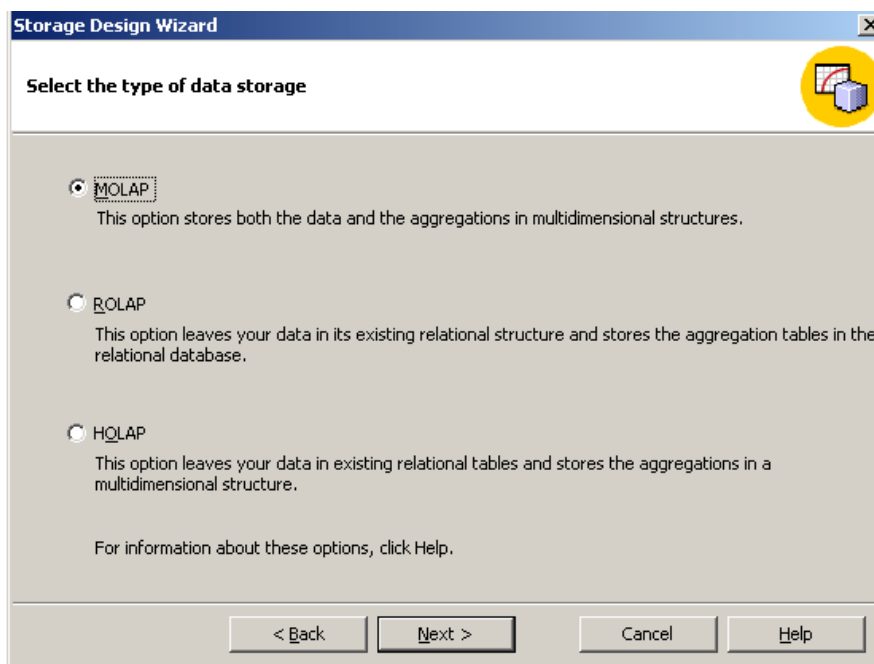
- Chọn ‘Edit’ để thêm, sửa hay xoá các khối, các đơn vị đo và các chiều trong khối.

5.4.4. Lưu trữ và xử lý khối

Analysis Services thường đưa ra nhiều lựa chọn lưu trữ dữ liệu và các tập dữ liệu trong các khối. Các kiểu lưu trữ gồm: MOLAP, ROLAP và HOLAP.

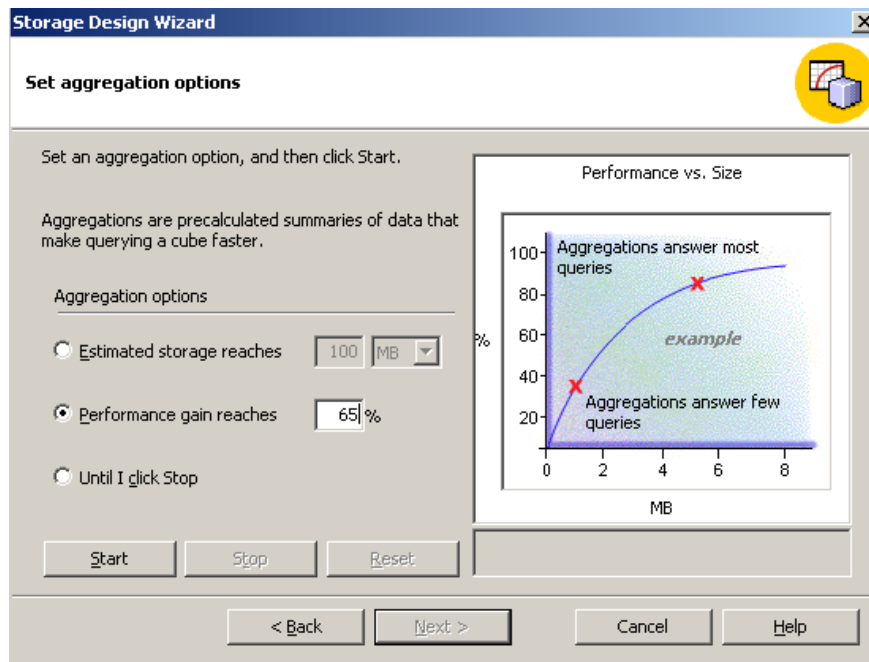
Khi kết thúc việc thiết kế cấu trúc của khối, ta cần thiết kế mô hình lưu trữ theo một trong 3 kiểu trên và cho Analysis Services biết cách sinh ra các

tập hợp hoặc tính toán trước các tổng hợp để cải thiện việc thực hiện các truy vấn cũng như cải tiến năng lực của các khối. Tiếp theo ta sẽ xử lý các khối, nạp dữ liệu từ nguồn dữ liệu đã thiết kế để sinh ra các tính toán tổng hợp mà ta định nghĩa trong chỉ dẫn của tập hợp.

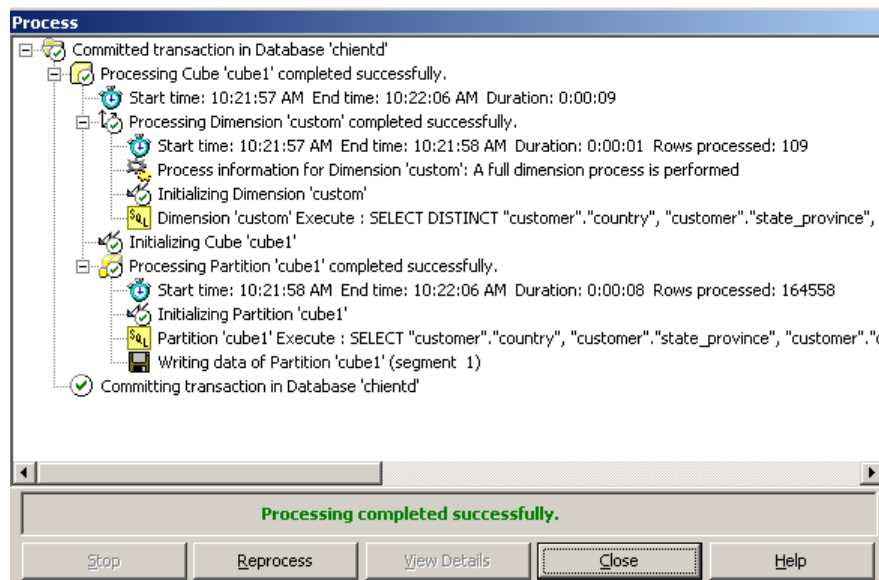


Hình 5.8. Chọn kiểu lưu trữ

- Để thiết kế cách lưu trữ, sử dụng 'Storage Design Wizard', chọn 'Type of Data Storage', kiểu mặc định để lưu trữ dữ liệu là MOLAP. Khi xuất hiện 'Set Aggregation Options', có thể yêu cầu Analysis Services tăng tốc độ xử lý mà không cần quan tâm dung lượng lưu trữ yêu cầu là bao nhiêu bằng cách chọn 'Performance Gain Reaches' và gõ tỉ lệ chọn (ví dụ 65%). Sự cân bằng giữa dung lượng lưu trữ sử dụng và khả năng đáp ứng của khối có thể được quản lý bằng tính năng này.



Hình 5.9. Tăng tốc độ thực hiện



Hình 5.10. Xử lý khối

- Cửa sổ tiến trình xuất hiện cho phép ta giám sát tiến trình của khối qua từng các giai đoạn và toàn tiến trình. Khi tiến trình kết thúc xuất hiện

thông báo “Processing completed successfully”

5.4.5. Khối ảo tăng cường khả năng xử lý và bảo mật

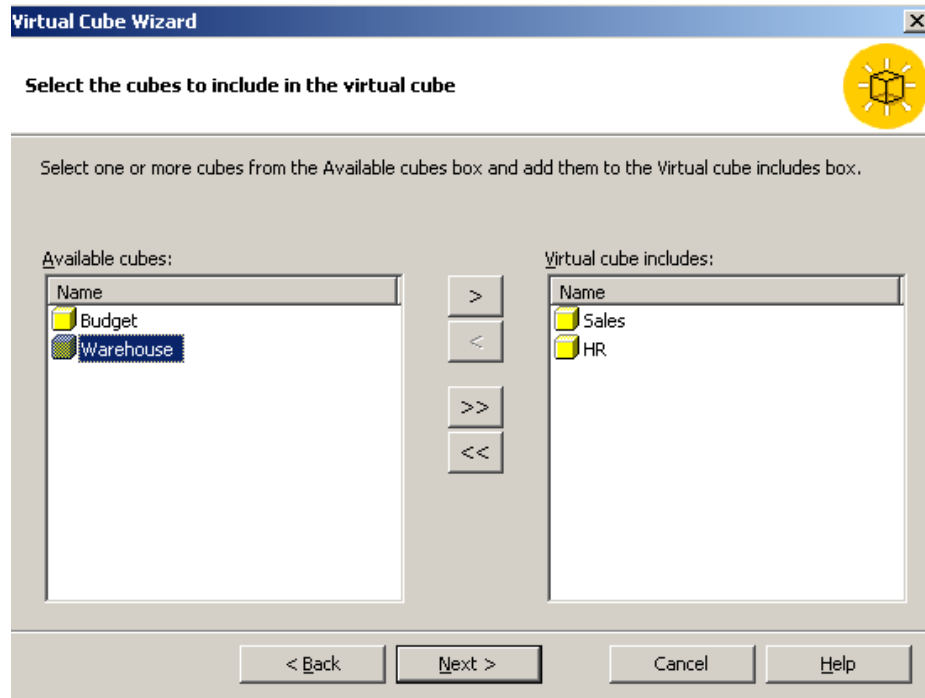
Có thể tạo ra các khung nhìn từ việc kết hợp các bảng một cách Logic trong một cơ sở dữ liệu quan hệ, một khối ảo là một khối Logic được tạo ra qua việc kết hợp các khối. Ta có thể chọn các chiều và các đơn vị đo của khối ảo từ các chiều và đơn vị đo đã tồn tại. Không cần thiết phải sử dụng tất cả các chiều và đơn vị đo nhưng có thể chọn những chiều cung cấp cho ta những khung nhìn tốt nhất của dữ liệu mà ta đang cố đưa ra từ những nguồn dữ liệu khác nhau. Khối ảo xuất hiện như một khối đơn lẻ từ các yêu cầu thông tin người dùng.

Các khối ảo thường gồm các chiều và đơn vị đo được lựa chọn từ các khối. Ta thường gặp các tình huống yêu cầu kết hợp dữ liệu từ nhiều nguồn dữ liệu OLAP và muốn hạn chế dữ liệu đưa tới người dùng chỉ trong một tập hợp con các chiều và đơn vị đo trong một khối đơn lẻ. Trong tình huống này khối ảo có thể cung cấp một lựa chọn tốt.

Sức mạnh của khối ảo bao gồm tính mềm dẻo, tính khả chuyển và dễ dàng trong việc tạo hay sửa khối. Yêu cầu lưu trữ tối thiểu khá mềm dẻo, có thể cho phép kết hợp và đa dạng hoá việc tạo các khối. Sức mạnh của khối ảo nằm trong khả năng bảo mật của nó. Thuộc tính này tạo ra sự hữu dụng của khối ảo khi một phần hoặc các phần của thông tin khối là nhạy cảm và không thích hợp để cung cấp cho tất cả người sử dụng. Một khối ảo cần bỏ qua một số thông tin nhạy cảm khi cung cấp. Có thể tạo hai Role bảo mật: cái thứ nhất chứa những người dùng được phép xem các thông tin nhạy cảm và cái thứ hai chứa những người dùng còn lại. Ta dễ dàng cấp quyền cho Role thứ nhất truy cập toàn bộ khối và Role thứ hai chỉ truy cập khối ảo.

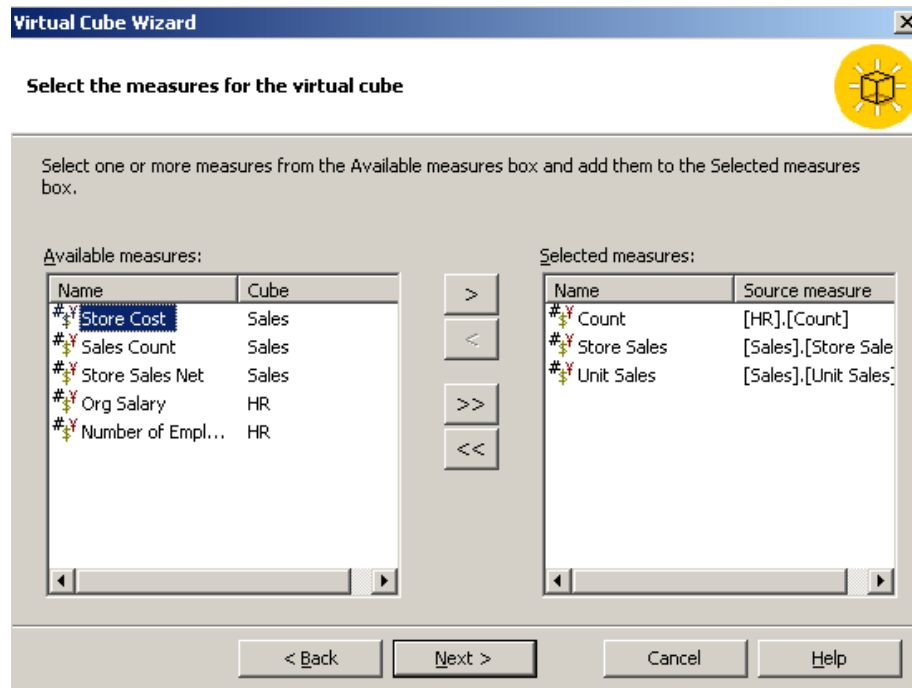
5.4.6. Tạo khối ảo

- Chọn ‘New Virtual Cube’ để tạo khối ảo từ các khối đã có.
- Chọn các khối cho khối ảo: cửa sổ trái liệt kê tất cả các khối có trong FoodMart 2000. Cửa sổ phải hiển thị các khối được chọn (ví dụ ta chọn hai khối ‘Sales’ và ‘HR’).

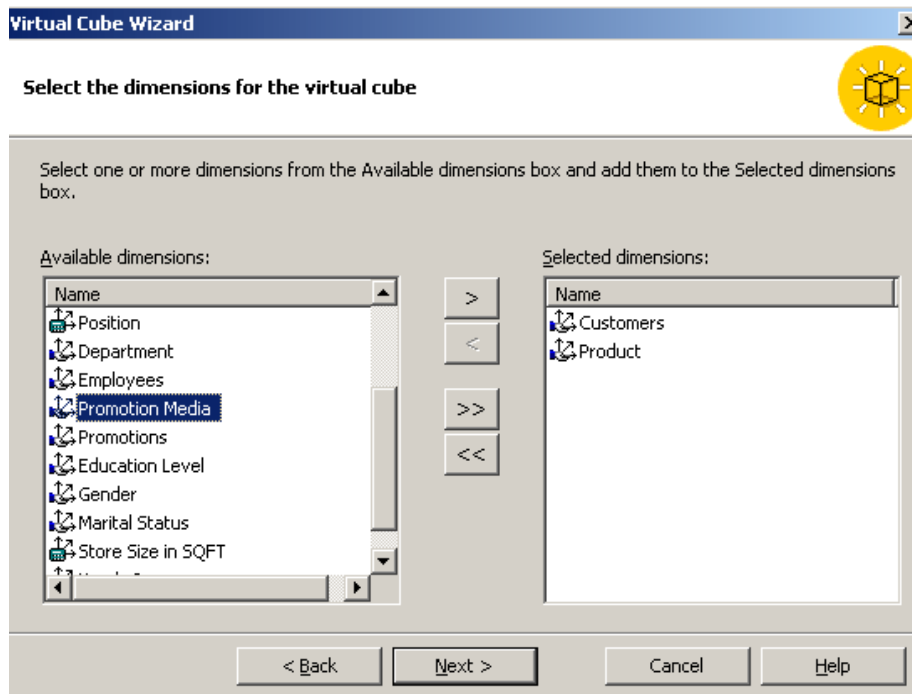


Hình 5.11. Chọn các khối cho khối ảo

- Chọn các đơn vị đo cho khối ảo: màn hình tiếp theo sẽ liệt kê tất cả các đơn vị đo của các khối đã chọn (khối ‘Sales’ và ‘HR’) trên cửa sổ trái. Thực hiện việc chọn các đơn vị đo có sẵn để làm đơn vị đo cho khối ảo. Cửa sổ phải sẽ hiển thị các đơn vị đo được chọn.
- Chọn chiều cho khối ảo: thực hiện việc chọn chiều cho khối ảo tương tự như việc chọn các đơn vị đo.



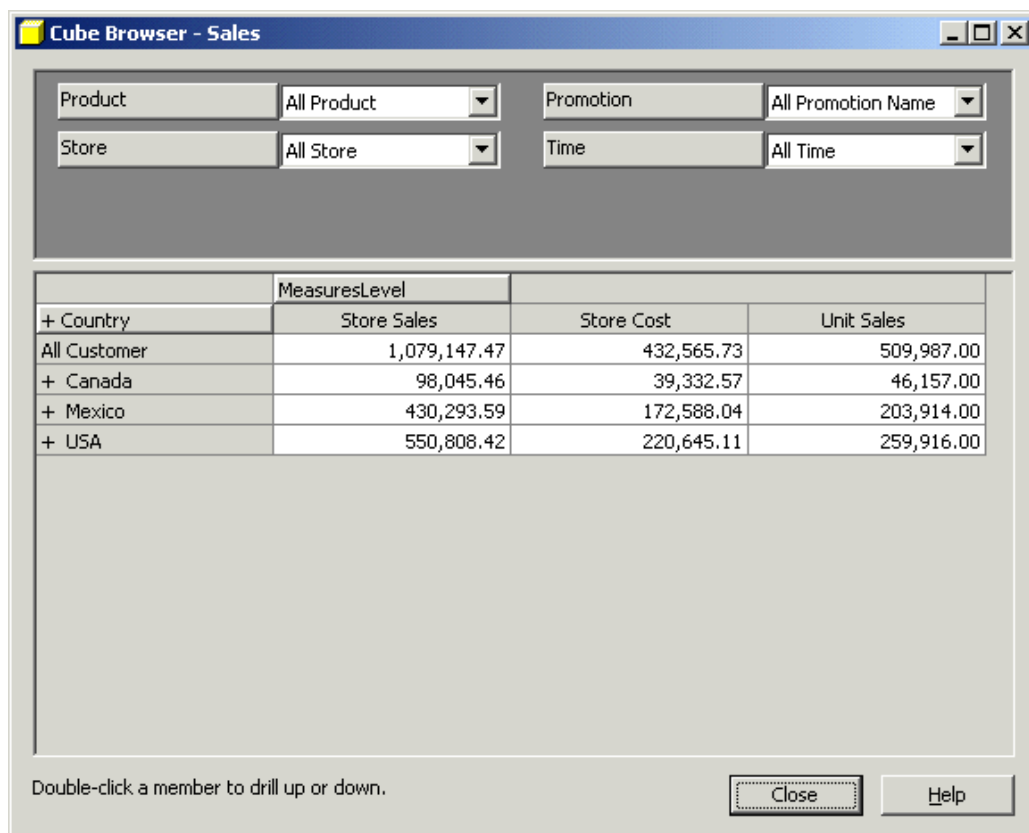
Hình 5.12. Chọn đơn vị đo cho khối ảo



Hình 5.13. Chọn chiều cho khối ảo

5.4.7. *Hiển thị dữ liệu khối*

- Chọn ‘Browse Data’ để hiển thị dữ liệu của khối cần phân tích (ví dụ khối ‘Sales’), màn hình phân tích dữ liệu của khối sẽ xuất hiện.
- Cửa sổ bên dưới hiển thị dữ liệu của một chiều nào đó, ở đây là chiều Store (lúc này ta đang nhìn thấy dữ liệu của khối Sales theo chiều Store). Tại đây, ta có thể khoan sâu xuống (Drill_down) các mức dữ liệu chi tiết hoặc cuộn lên (Roll_up) các mức dữ liệu tổng hợp hơn theo phân cấp chiều sẽ có được dữ liệu phân tích theo yêu cầu. Cửa sổ bên trên hiển thị các chiều còn lại của khối.
- Để xoay khối ‘Sales’ sang chiều khác (nhìn dữ liệu của khối theo chiều khác) ta chỉ việc chọn tên chiều ở cửa sổ bên trên.



Hình 5.14. Hiển thị dữ liệu khối

- Muốn cắt lát một khối, ta sẽ cố định một số chiều tại giá trị cụ thể bằng cách kích chuột vào nút lệnh cạnh mỗi tên chiều.

5.4.8. Ví dụ minh họa

Dựa vào bộ dữ liệu mẫu FoodMart 2000 của Microsoft ta có thể đưa ra một ví dụ như sau: Giám đốc doanh nghiệp muốn biết “Mặt hàng nào bán chạy nhất vào quý 4 năm 1998 từ các kho thuộc thành phố Los Angeles của Mỹ, tầng lớp khách hàng (nhóm tuổi) nào tiêu thụ mạnh nhất?”. Đây là câu hỏi có 4 chiều: **mặt hàng, thời gian, vùng, nhóm tuổi**. Không dễ gì có được câu trả lời cho những câu hỏi nhiều chiều như vậy bằng cách sử dụng các kỹ thuật truy vấn truyền thống trong các mô hình dữ liệu quan hệ như SQL mà phải dựa trên những kết quả phân tích nhiều chiều. Hơn nữa, do yêu cầu của người sử dụng liên tục thay đổi, đòi hỏi các câu trả lời phải được xử lý theo thứ tự khác nhau lúc theo vùng, khi thì theo thời gian, khi khác lại theo nhóm tuổi... Để trả lời được câu hỏi này, ta sẽ xây dựng khối ‘Test’ như sau:

- Bảng Fact: chọn bảng ‘sales_fact_1998’
- Đơn vị đo: ‘unit_sales’
- Chiều:
 - o Sản_phẩm: chọn bảng ‘product’
Các mức: ‘product_name’
 - o Thời_gian: chọn bảng ‘time_by_day’
Các mức: ‘the_year’, ‘quarter’, ‘the_month’
 - o Vùng: chọn bảng ‘store’
Các mức: ‘store_country’, ‘store_state’, ‘store_city’
 - o Nhóm_tuổi: chọn bảng ‘customer’
Các mức: ‘country’, ‘state_province’, ‘city’, ‘birthdate’

Xoay khối ‘Test’ theo chiều Sản_phẩm, cố định giá trị Q4 ở chiều Thời_gian, giá trị Los Angeles ở chiều Vùng chúng ta sẽ có câu trả lời.

Phần kết luận

Sau một thời gian nghiên cứu và hoàn thành luận văn, tác giả đã nắm bắt được các khái niệm tổng quát và các lý thuyết căn bản về OLAP, tiếp cận và phân tích đa chiều. Đề tài cũng đã nêu rõ các chi tiết để áp dụng những cơ sở lý luận này vào phát triển mô hình cụ thể của một hệ trợ giúp quyết định dựa vào dữ liệu, của tiến trình trợ giúp ra quyết định, triển khai tổ chức hệ thống và xây dựng những thành phần chính yếu nhất của một hệ trợ giúp quyết định dựa vào dữ liệu.

Luận văn đã thực hiện được tất cả các nội dung và đạt được các mục tiêu đề ra như trong bản đề cương đã được duyệt. Các kết quả đạt được bao gồm:

- Nắm bắt được các lý thuyết về khai thác dữ liệu và xử lý phân tích trực tuyến.
- Nắm bắt được lý thuyết về tổ chức kho dữ liệu.
- Nắm bắt được cách tiếp cận và phân tích dữ liệu đa chiều.
- Tìm hiểu về Hệ thống trợ giúp quyết định dựa vào dữ liệu trên cơ sở phương pháp luận xử lý phân tích trực tuyến.
- Xây dựng hệ trợ giúp quyết định dựa vào dữ liệu bằng công cụ Analysis Services.

Tuy nhiên vẫn còn một số vấn đề mà luận văn chưa đề cập đến. Một số hướng phát triển khác nữa có thể mở rộng như: việc tổ chức và quản lý kho dữ liệu trên mạng và thực hiện những liên kết giữa các khối đa chiều với kho dữ liệu hay trực tiếp đến các hệ xử lý tác vụ để tự động hoá việc cập nhật dữ liệu và cấu trúc chiều cho các khối đa chiều; việc phối hợp giữa các khối đa chiều để khai thác tối đa khả năng của các khối đa chiều; nghiên cứu tăng cường khả năng hiển thị kết quả, giúp cho việc mô tả, thay đổi các yêu cầu truy vấn thông tin thuận lợi hơn, linh hoạt hơn.

Mặc dù đã cố gắng trong nghiên cứu và thực hiện đề tài, nhưng vì thời gian và trình độ có hạn, chắc chắn luận văn không tránh khỏi nhiều thiếu sót. Em xin bày tỏ lòng biết ơn sâu sắc tới GS.TS.Nguyễn Thúc Hải, người đã tận tình giảng dạy và hướng dẫn em hoàn thành bản luận văn này. Em cũng xin bày tỏ lòng biết ơn tới các thầy, cô và các anh, chị ở khoa Công nghệ Thông tin và Trung tâm Đào tạo sau Đại học đã nhiệt tình giảng dạy và giúp đỡ em trong suốt thời gian học tập vừa qua. Xin chân thành cảm ơn các bạn học và đồng nghiệp đã giúp đỡ tôi trong quá trình học tập, nghiên cứu và thử nghiệm đề tài.

Tài liệu tham khảo

- [1] Viện Công nghệ Thông tin (1997), *Kho dữ liệu - Data Warehouse*, Hà Nội.
- [2] Surajit Chaudhuri (1997), *An Overview of Data Warehouse and OLAP Technology*, <http://research.microsoft.com>.
- [3] Ching T.H., Agrawal R., Megiddo N., Srikant R. (1997), *Range Queries in OLAP Data Cubes*, Proceeding ACM SIGMOD.
- [4] Alexandre Gachet (2003), *Distributed Decision Support System: A Federalist Model of Cooperation*, University of Fribourg.
- [5] William H.Inmon (2005), *Building the Data Warehouse - Fourth Edition*, Wiley Publishing Inc.
- [6] Intelligent Science, *Intelligent Decision Support System - IDSS*, <http://www.intsci.ac.cn/en/research/idss.html>.
- [7] Ralph Kimball (1996), *Dealing with Dirty Data*, DBMS Online Magazine Sep 1996.
- [8] Patrick O'Neil, Dallen Quass (1997), *Improved Query Performance with Variant Indexes*, Proceeding ACM SIGMOD.
- [9] Chrisantha Silva (2004), *Building Info-Structures for Decision Making*, IT Business Systems.
- [10] Erik Thomsen (2002), *OLAP Solutions: Building Multidimensional Information Systems - Second Edition*, Wiley Computer Publishing.
- [11] E.Turban, J.E.Aronson (2001), *Decision Support Systems and Intelligent Systems - 6th Edition*, Prentice Hall.

Một số Website tham khảo:

<http://www.databasejournal.com/features/mssql/article.php/1429671>

<http://www.certtutor.net/Articles/Index.cfm?ArticleID=7150>

Tóm tắt luận văn

Luận văn đề cập đến việc nghiên cứu xây dựng một hệ trợ giúp quyết định dựa vào dữ liệu, sử dụng phương pháp luận xử lý phân tích trực tuyến (OLAP). Hệ trợ giúp quyết định theo tiếp cận này có thể giúp các nhà quản lý thiết lập một mô hình OLAP cho ứng dụng cụ thể của mình trong việc tổ chức cơ sở dữ liệu đa chiều và dễ dàng điều chỉnh hoạt động phân tích, tìm kiếm thông tin theo những khía cạnh khác nhau của dữ liệu nhằm thu thập được tối đa dữ liệu cần thiết để từ đó đưa được những quyết định tốt nhất một cách nhanh chóng.

Không giống với các hệ trợ giúp quyết định truyền thống thường được xây dựng với mục đích đưa ra giải pháp tối ưu cho một bài toán cụ thể, trong một phạm vi ứng dụng hẹp, hệ trợ giúp quyết định dựa vào dữ liệu hướng đến việc giúp người sử dụng có thể khai thác được tối đa khả năng tiềm ẩn của một khối lượng dữ liệu lớn, nhằm thu được những thông tin tổng hợp ở đủ các khía cạnh khác nhau của dữ liệu, để từ đó có thể ra các quyết định đúng một cách nhanh chóng. Do đặc điểm này, phạm vi ứng dụng của hệ trợ giúp quyết định dựa vào dữ liệu là rộng. Nó có thể được sử dụng để trợ giúp quyết định cho các bài toán khác nhau, trong những lĩnh vực khác nhau.

Luận văn trình bày các nội dung chính về phương pháp tiếp cận và phân tích đa chiều trong xử lý phân tích trực tuyến, xây dựng hệ trợ giúp quyết định dựa vào dữ liệu với hai thành phần chính là kho dữ liệu và xử lý phân tích trực tuyến, tiến trình trợ giúp quyết định dựa vào dữ liệu, xây dựng cấu trúc thông tin để hỗ trợ việc ra quyết định và xây dựng hệ thống với chức năng tạo lập cơ sở dữ liệu đa chiều và phân tích hiển thị dữ liệu.

Các từ khoá (Key Word): xử lý phân tích trực tuyến (OLAP), kho dữ liệu (Data Warehouse), hệ trợ giúp quyết định (DSS), khối (Cube) và đa chiều (Multidimensional).