TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**KHAI THÁC TẬP DỮ LIỆU LỚN**

**BẢO VỆ HỆ THỐNG KHUYẾN NGHỊ BẰNG CÁCH XÁC ĐỊNH**

**CÁC XẾP HẠNG GIẢ**

*Người hướng dẫn*: **TS. Bùi Thanh Hùng**

*Người thực hiện*: **NGUYỄN DUY HÀN LÂM (MSSV: 196005004)**

Khoá  **: 2019**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**KHAI THÁC TẬP DỮ LIỆU LỚN**

**BẢO VỆ HỆ THỐNG KHUYẾN NGHỊ BẰNG CÁCH XÁC ĐỊNH**

**CÁC XẾP HẠNG GIẢ**

*Người hướng dẫn*: **TS. Bùi Thanh Hùng**

*Người thực hiện*: **NGUYỄN DUY HÀN LÂM (MSSV: 196005004)**

Khoá  **: 2019**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS Bùi Thanh Hùng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 28 tháng 9 năm 2016*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Duy Hàn Lâm*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

MỤC LỤC

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN ii](#_Toc23071040)

[MỤC LỤC 1](#_Toc23071041)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc23071042)

[CHƯƠNG 1: HỆ THỐNG GỢI Ý VÀ TÍNH BẢO MẬT CỦA HỆ THỐNG 6](#_Toc23071043)

[1.1 Hiện tượng long-tail (khởi nguồn ra đời của hệ thống gợi ý) 7](#_Toc23071044)

[1.2 Giới thiệu sơ lược về hệ thống gợi ý (Recommender System) 7](#_Toc23071045)

[1.3 Content-based Recommendation System 8](#_Toc23071046)

[1.3.1 Xây dựng utility matrix 8](#_Toc23071047)

[1.3.2 Item profile 8](#_Toc23071048)

[1.3.3 Xây dựng hàm mất mát 8](#_Toc23071049)

[1.3.4 Lập trình trên Python 8](#_Toc23071050)

[1.4 Neighborhood-Based Collaborative Filtering (User-User Collabrative Filtering) 8](#_Toc23071051)

[1.4.1 Xây dựng Similarity function 8](#_Toc23071052)

[1.4.2 Rating prediction 8](#_Toc23071053)

[1.4.3 Khuyết điểm 8](#_Toc23071054)

[1.5 Neighborhood-Based Collaborative Filtering (Item-Item Collabrative Filtering) 8](#_Toc23071055)

[1.5.1 Khắc phục khuyết điểm của User-User Collaborative Filtering 8](#_Toc23071056)

[1.5.2 Lập trình Neighborhood-Based Collaborative Filtering trên Python 8](#_Toc23071057)

[1.6 Matrix Factorization Collaborative Filtering 8](#_Toc23071058)

[1.6.1 Ý tưởng thuật toán 9](#_Toc23071059)

[1.6.2 Xây dựng và tối ưu hàm mất mát 9](#_Toc23071060)

[1.6.3 Bias 9](#_Toc23071061)

[1.6.4 Non-negative Matrix Factorization 9](#_Toc23071062)

[1.6.5 Incremental Matrix Factorization 9](#_Toc23071063)

[1.6.6 Các vấn đề khác 9](#_Toc23071064)

[1.6.7 Lập trình trên Python 9](#_Toc23071065)

[1.7 Các lược đồ cho CSDL đa chiều 9](#_Toc23071066)

[1.8 Các mô hình lưu trữ hỗ trợ OLAP 10](#_Toc23071067)

[1.8.1 Mô hình Multidimentional OLAP (MOLAP) 10](#_Toc23071068)

[1.8.2 Mô hình Relational OLAP (ROLAP) 12](#_Toc23071069)

[1.8.3 Mô hình Hybird OLAP (HOLAP) 13](#_Toc23071070)

[CHƯƠNG 2: OLAP VÀ KHAI PHÁ DỮ LIỆU 15](#_Toc23071071)

[2.1 Tóm tắt các khái niệm 16](#_Toc23071072)

[2.1.1 OLTP (Online Transaction Processing) 16](#_Toc23071073)

[2.1.2 OLAP (Online Analytical Processing) 16](#_Toc23071074)

[2.1.3 Sự khác nhau giữa OLAP và OLTP 17](#_Toc23071075)

[2.1.4 Chuyển đổi dữ liệu từ OLTP tới OLAP 18](#_Toc23071076)

[2.2 Quy trình thực hiện OLAP. 19](#_Toc23071077)

[2.3 Tóm tắt mô hình hóa OLAP. 20](#_Toc23071078)

[2.4 Tóm tắt các loại thao tác trong OLAP. 20](#_Toc23071079)

[2.4.1 Slice 20](#_Toc23071080)

[2.4.2 Dice 21](#_Toc23071081)

[2.4.3 Drill Down hay Drill Up 21](#_Toc23071082)

[2.4.4 Roll up 21](#_Toc23071083)

[2.4.5 Pivot 21](#_Toc23071084)

[CHƯƠNG 3: ỨNG DỤNG OLAP VÀO BÀI TOÁN CỤ THỂ 22](#_Toc23071085)

[3.1 Mô tả bài toán 22](#_Toc23071086)

[3.2 Mô hình thực thể ban đầu (CDM Model) 23](#_Toc23071087)

[3.3 Mô hình vật lý ban đầu của bài toán 24](#_Toc23071088)

[3.4 Triển khai OLAP cho bài toán 25](#_Toc23071089)

[3.5 Áp dụng OLAP trên hệ quản trị cơ sở dữ liệu 26](#_Toc23071090)

[3.6 Áp dụng OLAP trên chương trình 32](#_Toc23071091)

[TÀI LIỆU THAM KHẢO 39](#_Toc23071092)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1.1: Vị trí của OLAP 6](#_Toc482868018)

[Hình 1.2: Lược đồ hình sao 10](#_Toc482868019)

[Hình 1.3: Lược đồ bông tuyết 10](#_Toc482868020)

[Hình 1.4: Mô hình dữ liệu MOLAP 11](#_Toc482868021)

[Hình 1.5: Mô hình dữ liệu ROLAP 13](#_Toc482868022)

[Hình 2.1: Mô tả thao tác Slice[2] 21](#_Toc482868023)

[Hình 2.2: Mô tả thao tác Dice[2] 21](#_Toc482868024)

[Hình 2.3: Mô tả thao tác Drill Down/Up[2] 22](#_Toc482868025)

[Hình 2.4: Mô tả thao tác Pivot[2] 22](#_Toc482868026)

[Hình 3.1: Mô hình thực thể của bài toán 24](#_Toc482868027)

[Hình 3.2: Mô hình vật lý của bài toán 24](#_Toc482868028)

[Hình 3.3: Mô hình CDM sau khi thêm bảng sự thật 25](#_Toc482868029)

[Hình 3.4: Mô hình vật lý sau khi thêm bảng sự thật 26](#_Toc482868030)

[Hình 3.5: Dữ liệu minh họa cho bảng NHA\_CUNG\_CAP 26](#_Toc482868031)

[Hình 3.6: Dữ liệu minh họa cho bảng LOAI\_HANG 27](#_Toc482868032)

[Hình 3.7: Dữ liệu minh họa cho bảng NGUYEN\_VAT\_LIEU 27](#_Toc482868033)

[Hình 3.8: Dữ liệu minh họa cho bảng KHACH\_HANG 27](#_Toc482868034)

[Hình 3.9: Dữ liệu minh họa cho bảng sự thật SALE 28](#_Toc482868035)

[Hình 3.10: Câu lệnh Join các bảng của mô hình star schema 28](#_Toc482868036)

[Hình 3.11: Một phần kết quả của truy vấn join 28](#_Toc482868037)

[Hình 3.12: Áp dụng câu lệnh join cho truy vấn dùng OLAP 28](#_Toc482868038)

[Hình 3.13: Kết quả áp dụng kỹ thuật OLAP 29](#_Toc482868039)

[Hình 3.14: Câu lệnh group by đơn giản trên nền OLAP 29](#_Toc482868040)

[Hình 3.15: Kết quả sau khi dùng câu lệnh group by 29](#_Toc482868041)

[Hình 3.16: Ví dụ về Drill Down 29](#_Toc482868042)

[Hình 3.17: Kết quả của ví dụ về Drill Down 30](#_Toc482868043)

[Hình 3.18: Câu lệnh Slice 30](#_Toc482868044)

[Hình 3.19: Kết quả của câu lệnh Slice 30](#_Toc482868045)

[Hình 3.20: Câu lệnh thao tác Dice 31](#_Toc482868046)

[Hình 3.21: Kết quả câu lệnh Dice 31](#_Toc482868047)

[Hình 3.22: Câu lệnh Roll up 31](#_Toc482868048)

[Hình 3.23: Kết quả câu lệnh Roll Up 32](#_Toc482868049)

[Hình 3.24: Giao diện của chức năng 1 33](#_Toc482868050)

[Hình 3.25: Phương thức ở lớp Database phục vụ cho chức năng 1 33](#_Toc482868051)

[Hình 3.26: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 1 33](#_Toc482868052)

[Hình 3.27: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 1 34](#_Toc482868053)

[Hình 3.28: Phương thức phục vụ hiển thị kết quả của chức năng 1 34](#_Toc482868054)

[Hình 3.29: Kết quả của chức năng 1 34](#_Toc482868055)

[Hình 3.30: Phương thức ở lớp Database phục vụ cho chức năng 2 35](#_Toc482868056)

[Hình 3.31: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 2 35](#_Toc482868057)

[Hình 3.32: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 2 35](#_Toc482868058)

[Hình 3.33: Phương thức phục vụ hiển thị kết quả của chức năng 2 35](#_Toc482868059)

[Hình 3.34: Giao diện chung và kết quả chức năng 2 36](#_Toc482868060)

[Hình 3.35: Phương thức ở lớp Database phục vụ cho chức năng 3 36](#_Toc482868061)

[Hình 3.36: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 3 36](#_Toc482868062)

[Hình 3.37: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 3 37](#_Toc482868063)

[Hình 3.38: Giao diện chung và kết quả chức năng 3 37](#_Toc482868064)

[Hình 3.39: Phương thức ở lớp Database phục vụ cho chức năng 4 37](#_Toc482868065)

[Hình 3.40: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 4 37](#_Toc482868066)

[Hình 3.41: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 4 38](#_Toc482868067)

[Hình 3.42: Giao diện chung và kết quả chức năng 4 38](#_Toc482868068)

**DANH MỤC BẢNG BIỂU**

[Bảng 1: So sánh các mô hình OLAP 14](#_Toc482869075)

[Bảng 2: So sánh OLTP và OLAP 17](#_Toc482869076)

CHƯƠNG 1: HỆ THỐNG GỢI Ý VÀ TÍNH BẢO MẬT CỦA HỆ THỐNG

Ngày nay, có rất nhiều dữ liệu, thông tin có sẵn trên Internet nhưng rất khó để lọc ra những thông tin cần thiết từ nguồn dữ liệu, thông tin dồi dào này. Một trong những giải pháp để giải quyết vấn đề này, đó chính là sử dụng các thuật toán để đưa ra gợi ý giúp ta tìm được thông tin mà ta muốn, ta gọi các hệ thống đưa ra gợi ý này là “Recommender System”. Có 2 loại recommender system là: Content-based Recommendation System (Hệ thống đưa ra các gợi ý dựa trên các dữ liệu, thông tin, vật thể,…) và Collaborative filtering Recommendation System (Hệ thống đưa ra các gợi ý dựa trên hành vi, ý muốn của người dùng). Trong đó, Collaborative filtering Recommendation System được sử dụng phổ biến vì có tính khả thi cao do không cần phải phân loại nguồn đầu vào cho hệ thống (dữ liệu, thông tin, vật thể, đối tượng,…) vì đôi khi các nguồn rất khó hoặc không thể phân loại được. Collaborative filtering Recommendation System rất hiệu quả và hữu ích trong việc đưa ra các gợi ý cho người dùng, tuy niên hệ thống này có một khuyết điểm lớn, nằm ở chính nguồn đầu vào của hệ thống. Các nguồn đầu vào của hệ thống đôi khi sẽ bị sai lệch, không chính xác ảnh hưởng đến đầu ra là các gợi ý mà hệ thống đã đưa ra. Đôi khi các nguồn đầu vào này sai lệch không phải do khách quan, ngẫu nhiên mà có một số kẻ cố tình tạo dựng các nguồn dữ liệu giả nhằm tấn công vào hệ thống.

Những kẻ tấn công ẩn danh vào như những người dùng bình thường, họ tham gia vào quá trình xây dựng nguồn dữ liệu đầu vào cho các hệ thống gợi ý, tạo ra các hành vi sai, không đúng hay không hợp lý, từ đó khiến quá trình huấn luyện của hệ thống để đưa ra các gợi ý trở nên bị sai lệch. Từ đó làm giảm tính khách quan và độ chính xác của hệ thống. Điều này ảnh hưởng nghiêm trọng đến độ tin cậy của hệ thống. Tài liệu này nhằm mục đích thảo luận về các thể loại tấn công khác nhau có thể ảnh hưởng đến hệ thống recommender và các giải pháp được sử dụng để phát hiện các cuộc tấn công này.

* 1. Hiện tượng long-tail (khởi nguồn ra đời của hệ thống gợi ý)

Trong các cửa hàng hay siêu thị ta thường đa phần thấy có 2 tình trạng xảy ra. Một là có nhiều sản phẩm được rất rất nhiều người dùng mua và lựa chọn. Hai là có rất nhiều mặt hàng sản phẩm còn trên kệ, tuy nhiên chúng đôi khi vẫn có những khách hàng mua chúng. Ngoài ra ta còn thấy được đa phần số lượng các mặt hàng hay nhãn hàng được mua nhiều thì rất ít, còn các mặt hàng hay nhãn hàng được mua ít thì lại chiếm đa số trên kệ hàng.

Từ đó, ta có thể mô phỏng được hình dáng của đồ thị phân phối xác suất sẽ có 2 phần. Phần đầu là các đối tượng có xác suất xảy ra còn. Phần còn lại là một tập hợp dài các đối tượng có xác suất xảy ra thấp hơn hoặc thấp hơn rất nhiều. Ta đặt biệt danh cho tập hợp dài này là “long-tail”.

Tuy nhiên, trong kinh doanh, các chủ buôn thường không muốn điều này. Họ thường luôn muốn bán hết các mặt hàng có lượng mua ít càng nhanh càng tốt để có thể dành tài nguyên (kho, kệ hàng,…) cho các mặt hàng khác quan trọng hơn.

Để có thể đẩy nhanh được các mặt hàng có lượng kinh doanh thấp này, ta cần phải cho càng nhiều người dùng biết càng tốt vì có cơ hội họ sẽ mua chúng. Một trong những giải pháp đó chính là sử dụng hệ thống gợi ý. Hệ thống gợi ý sẽ dựa trên các đặc điểm của món hàng hay của người dùng mà từ đó có thể đưa ra các gợi ý phù hợp cho người dùng. Từ đó có thể nâng cao được cơ hội người mua sẽ chọn và mua các món hàng đó.

* 1. Giới thiệu sơ lược về hệ thống gợi ý (Recommender System)

Hệ thống gợi ý là hệ thống dựa trên tập dữ liệu đầu vào, qua quá trình huấn luyện, sẽ cho được ở đầu ra một kết quả. Kết quả này là một lời gợi ý dành cho các đối tượng trong tập đối tượng. Ví dụ ta có tập dữ liệu đầu vào là các bộ phim và tập đối tượng là những người xem phim, khi đưa vào hệ thống, sau khi huấn luyện, hệ thống sẽ đưa ra được các gợi ý là những bộ phim mà những người xem trong tập đối tượng nên xem.

Có 2 loại hệ thống gợi ý: Content-based và Collaborative filtering.

Content-based recommendation system: đánh giá các thuộc tính của item rồi từ đó phân loại các item. Dựa vào từng user mà sẽ đưa ra các gợi ý item phù hợp với ý muốn của người dùng. Ví dụ: có rất nhiều bộ phim với nhiều thể loại khác nhau như hình sự, kinh dị, trinh thám, hoạt hình, hài hước,… tùy theo sở thích của người xem ta sẽ gợi ý các bộ phim khác có thể loại tương tự với thể loại của các bộ phim mà người dùng đã xem trước đó.

Collaborative filtering recommendation system: đưa ra gợi ý cho người dùng dựa vào các khuôn mẫu hay hành vi mà người đó hành động. Ví dụ: người A thích xem phim kinh dị, bên cạnh đó người cũng thích xem phim hành động. Người B thích phim kinh dị, như vậy qua quá trình huấn luyện, hệ thống có thể gợi ý cho người B nên xem phim hành động.

* 1. Content-based Recommendation System

Ở mục này ta sẽ tìm hiểu tổng quát cách xây dựng 1 Content-based recommendation system.

* + 1. Xây dựng utility matrix

Các bài toán liên quan đến recommendation system đều có 2 phần là user (người dùng) và item (các đối tượng mà bài toán sẽ xử lý). Content-based recommendation system sẽ đưa ra gợi ý cho các user dựa trên mức độ quan tâm của từng user đối với từng item. Rồi từ đó sẽ đưa ra cái nhìn chung và cuối cùng là đưa ra gợi ý item khác cho các user.

Utility matrix là ma trận ở đó các hàng của nó sẽ là các item. Còn các cột sẽ là từng user. Giá trị của mỗi ô trong ma trận sẽ biểu thị mức độ quan tâm của user với item tương ứng. Có 2 loại giá trị biểu thị cho mức độ quan tâm là giá trị trống (nếu user không biểu thị sự quan tâm đến item) và 2 là mức độ quan tâm cụ thể.

Ta sẽ dựa trên các item được user đánh giá cụ thể, đưa vô mô hình để huấn luyện và đầu ra hệ thống sẽ gợi ý cho các user các item chưa được đánh giá mà user có khả năng “thích” chúng.

Càng nhiều item được đánh giá thì độ chính xác của mô hình huấn luyến càng cao.

Sau đây là một ví dụ điển hình cho utility matrix:

**Bảng 1: Ví dụ về ultility matrix**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| Batman | 5 | 5 | 0 | 0 | 1 | ? |
| Spyderman | 5 | ? | ? | 0 | ? | ? |
| Superman | ? | 4 | 1 | ? | ? | 1 |
| Tom & Jerry | 1 | 1 | 4 | 4 | 4 | ? |
| Doraemon | 1 | 0 | 5 | ? | ? | ? |

Theo như ví dụ trên, ta có 6 user lần lượt là A, B, C, D, E, F được đánh số từ 0 (không thích) cho đến 5 sao (rất thích) thể hiện mức độ quan tâm/đánh giá của user ứng với từng bộ phim. Ngoài ra các ô “?” biểu thị rằng user chưa đánh giá bộ phim. Nhiệm vụ của recommendation system chính là dự đoán giá trị tại các ô “?” rồi đưa ra gợi ý cho người dùng.

Batman, Spyder, Superman là các bộ phim hành động, Tom & Jerry, Doraemon là các bộ phim hoạt hình. Ta có thể thấy user A rất thích phim hành động vì Batman và Spyder đều được đánh giá 5 sao. User B cũng thích thể loại phim hành động vì Batman và Superman đều được đánh giá 5 sao và 4 sao. Tuy nhiên user A chưa đánh giá phim Superman, tuy nhiên Superman là phim hành động. Vì vậy hệ thống cần phân loại các bộ phim và sao cho có thể gợi ý phim Superman cho user A.

* + 1. Item profile

Như đã nói ở các phần trước, content-based recommendation system sẽ dựa trên nội dung, thuộc tính của các item để từ đó đưa ra gợi ý, vì vậy ta cần phải xây dựng 1 bộ hồ sơ (profile) dành cho mỗi item.

Ở ví dụ mục 1.3.1, một bộ phim thường có một số thuộc tính mà ta có thể liệt kê như: đạo diễn, diễn viên, thể loại, v.v. Ở đây, tài liệu sẽ xét theo thể loại của bộ phim. Ta chỉ xét 2 thể loại là hành động và hoạt động. Mỗi bộ phim sẽ được biểu diễn thành các vector 2 chiều, chiều thứ nhất biểu diễn mức độ hành động của bộ phim, và chiều thứ 2 biểu diễn mức độ hoạt hình của bộ phim.

Ta sẽ biểu diễn lại ví dụ ở mục 1.3.1 thành:

**Bảng 2: Ví dụ về item profile**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | Item feature vector |
| Batman | 5 | 5 | 0 | 0 | 1 | ? | x1 = [0.99, 0.02] |
| Spyderman | 5 | ? | ? | 0 | ? | ? | x2 = [0.91, 0.11] |
| Superman | ? | 4 | 1 | ? | ? | 1 | x3 = [0.95, 0.05] |
| Tom & Jerry | 1 | 1 | 4 | 4 | 4 | ? | x4 = [0.01, 0.99] |
| Doraemon | 1 | 0 | 5 | ? | ? | ? | x5 = [0.03, 0.98] |
| User model |  |  |  |  |  |  |  |

Bài toán của ta chính là đi tìm mô hình sao cho có thể fix được các có trong mô hình. Sau đó ta sẽ dự đoán các giá trị ở các ô “?”.

* + 1. Xây dựng hàm mất mát

Gọi số user là N, số item là M, utility matrix được mô tả là một ma trận Y. Thành phần ở hàng thứ m, cột thứ n của Y biểu diễn cho mức độ quan tâm của user n lên sản phẩm m mà hệ thống thu thập được. Các giá trị “?” ở bảng 2 ở trên khi biểu diễn ở ma trận Y sẽ là các ký thự trống. Ngoài ra ta sẽ biểu diễn một ma trận R, ma trận R sẽ cho ta biết một user đã đánh giá một item nào đó hay chưa, nếu sản phẩm thứ i được user j đánh giá thì ô đó sẽ có giá trị là 1, còn nếu chưa sẽ là 0.

Ta giả sử rằng, mức độ đánh giá item của các item sẽ là 1 một hàm y = f(x) ở đó x là các feature vector chúng ta sẽ cần tìm trọng số w và phần bù đại số (bias) b. Cụ thể ứng với mỗi user n, và item m ta sẽ có được độ đánh giá theo:

Sau khi ta đã huấn luyện xong, ta có được w và b phù hợp, ta sẽ thế vào mô hình để đi tìm các giá trị y ở các ô “?”. Để tính toán xem mức độ sai số ta sẽ sử dụng hàm mất mát cho từng user n như sau:

Tổng quát ta sẽ tính sai số dựa trên trung bình của các item m mà user thứ i đã đánh giá:

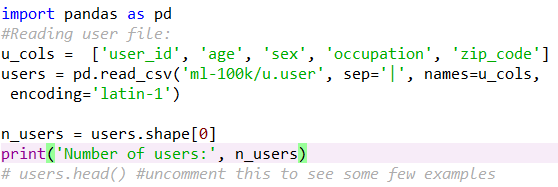
Trong đó:

Ta đặt là vector chứa các giá trị 1 tương ứng với các item mà user m đã đánh giá. Từ đó ta có thể viết lại biểu thức thành:

* + 1. Lập trình trên Python

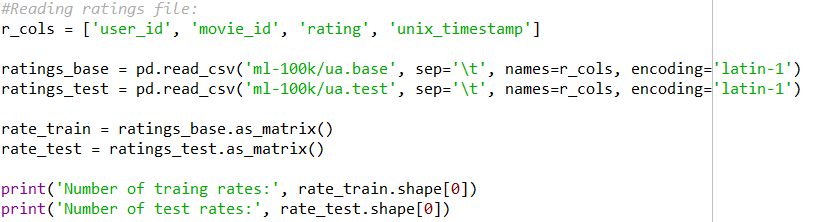
Bộ dữ liệu sử dụng: MovieLens 100k, năm 1998, bởi GroupLens.

Đọc file dữ liệu user:



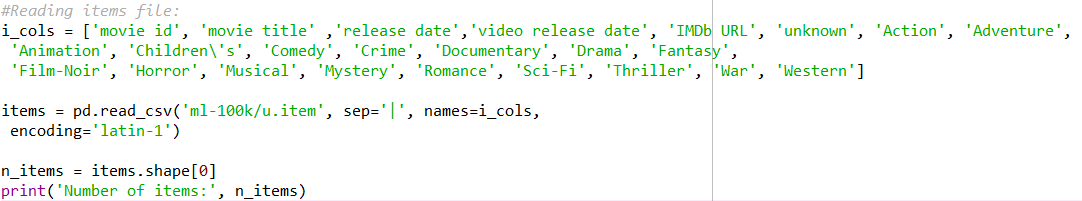
Hình 1.1 Code đọc file dữ liệu user bằng Python

Tiếp theo ta sẽ đọc file rating chứa dữ liệu item cũng như các đánh giá về item của các user, sau đó ta phân tập dữ liệu thành 2 phần là tập huấn luyện (train) để huấn luyện và xây dựng mô hình và tập kiểm thử (test) để kiểm tra tính đúng đắn của mô hình mà ta tạo được:



Hình 1.2 Đọc file rating và phân tập dữ liệu thành tập train, test

Tiếp theo ta sẽ xây dựng item profile, chứa item và các thuộc tính của item, ở đây ta sẽ chọn thuộc tính của các item chính là các thể loại phim của các item (cột 6 đến cột 24) và các thông tin cơ bản của item (cột 1 đến cột 5).



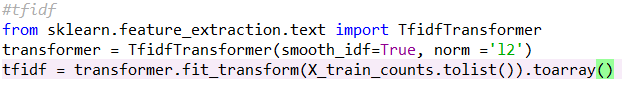
Hình 1.3 Xây dựng item profile

Ta lọc ra, bỏ ra các thông tin cơ bản (cột 1 đến cột 5) và chỉ lấy 19 cột thể loại phim (cột 6 đến cột 24) để đem đi xử lý.



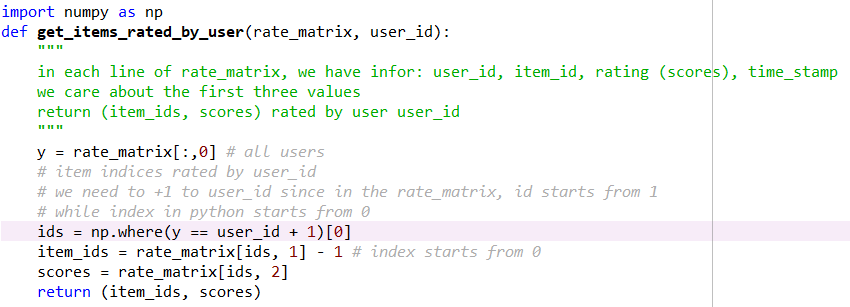
Hình 1.4 Lấy thông tin thể loại phim của từng item

Ta sử TF-IDF để xây dựng bộ feature vector cho từng item.



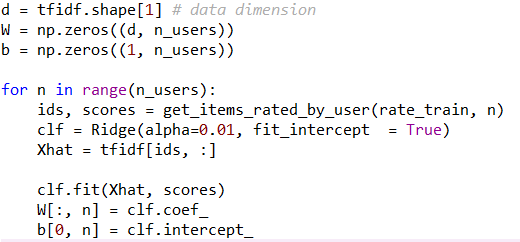
Hình 1.5 Xây dụng bộ feature vector cho item

Ta sẽ xây dựng hàm lấy danh sách các item mà user đánh giá. Ta sẽ lấy theo tất cả user (ma trận R).



Hình 1.6 Hàm lấy danh sách các item mà user đã đánh giá

Tiếp theo, ta sẽ tính Ridge Regression cho mỗi user:



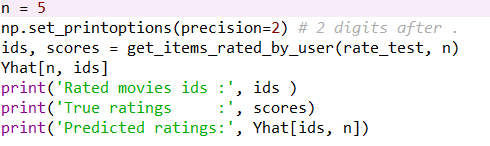
Hình 1.7 Tính Ridge Regression cho từng user

Cuối cùng ta sẽ hoàn thiện bộ feature vector bằng cách dự đoán các giá trị bằng công thức:



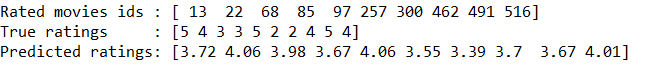
Hình 1.8 Dự đoán các giá trị

Thực hiện chạy chương trình trên user có id = 5



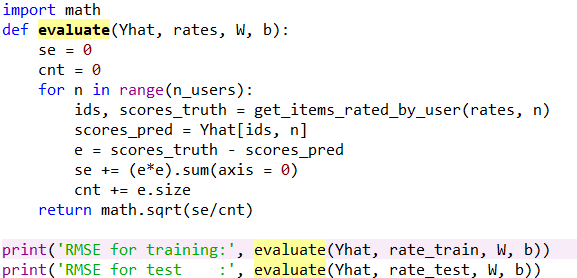
Hình 1.9 Chạy chương trình hệ thống trên user có id = 5

Sau khi chạy chương trình, ta sẽ có kết quả tương ứng:



Hình 1.10 Kết quả sau khi chạy thử chương trình

Cuối cùng, ta sẽ tiến hành đánh giá sai số bằng RMSE:



Hình 1.11 Thực hiện đánh giá sai số RMSE

Kết quả của sai số RMSE:



Hình 1.12 Kết quả sai số RMSE

Dựa trên các giá trị dự đoán đánh giá cao mà ta sẽ đưa ra được gợi ý các bộ phim mà user có id = 5 nên xem.

* 1. Neighborhood-Based Collaborative Filtering (User-User Collabrative Filtering)

Ở mục 1.3 ta đã tìm hiểu xong về Content-based Recommendation System. Cốt lõi của content-based chính là xây dựng một bộ các feature vector để từ đó ta tiến hành đi xây dựng mô hình, dự đoán và đưa ra đánh giá sai số. Tuy nhiên cách làm này có 2 khuyết điểm cơ bản. Thứ nhất, khi xây dựng mô hình, ta không thể tham khảo giá trị đánh giá của user khác. Thứ hai, không phải lúc nào ta cũng có được một bản mô tả mối liên hệ giữa item và user.

Để tránh được tình trạng không thể phân loại được item, hay không thể xây dựng được một bộ item profile, feature vector tốt. Ta có thể sử dụng một cách làm khác để có thể đưa ra 1 gợi ý cho user. Ta sẽ tham khảo các user khác, dựa vào chuỗi hành vi của user đang xét, ta sẽ tiến hành đi tìm xem có chuỗi hành vi của user nào khác có điểm chung (tương đồng) với user đang xét. Sau khi tìm được, ta sẽ gợi ý cho user đang xét những item dựa trên user khác đã tìm được.

Ví dụ, A thích xem phim Batman và Spyderman (đều đánh giá 5 sao), B thích xem phim Batman và Superman (đánh giá lần lượt là 5 sao, 4 sao). Từ đó ta suy ra được A và B đều thích xem phim hành động và đều cùng xem phim Batman (điểm chung). Nếu A là user đang được xét để đưa ra gợi ý, dựa vào hệ thống ta tìm được user B có nhiều điểm chung so với user A. Và hệ thống thấy rằng user B ngoài điểm chung thì user B có xem thêm phim Superman. Vì vậy có khả năng rằng user A có thể thích phim Superman.

Để xây dựng được 1 hệ thống như vậy, ta cần phải giải quyết 2 vấn đề:

* Cách xác định sự giống nhau giữa 2 user
* Khi đã xác định được 2 user nào giống nhau, từ đó làm cách nào để có thể dự đoán được mức độ đánh giá của user đang xét với các item.

Ở mục này, ta sẽ đi tìm xem cách để có thể tìm ra được sự giống nhau giữa 2 user. Ta gọi quá trình này là “User-user Collaborative Filtering”.

* + 1. Xây dựng Similarity function

Như đã đề cập, ta thấy rằng cốt lõi của Collaborative Filtering chính là đi tìm sự giống nhau giữa 2 user. Vì vậy việc đầu tiên mà ta cần làm đó chính là đi xây dựng hàm đánh giá mức độ giống nhau giữa 2 user.

Xét ví dụ bảng 1 ở trên ta có 6 user (A, B, C, D, E, F) và có 5 bộ phim (Batman, Spyderman, Superman, Tom & Jerry, Doraemon). Trong mỗi ô là giá trị đánh giá của các user dành cho các bộ phim mà họ đã xem. Như đã nói ở mục 1.3 điều ta cần làm chính là làm sao có thể điền các giá trị còn thiếu vào các ô có giá trị “?”. Tuy nhiên ta sẽ không đi xây dựng item profile có chứa item feature vector nữa mà ta sẽ đi tìm sự tương đồng giữa 2 user.

**Bảng 3: Nhắc lại về Utility Matrix**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| Batman | 5 | 5 | 0 | 0 | 1 | ? |
| Spyderman | 5 | ? | ? | 0 | ? | ? |
| Superman | ? | 4 | 1 | ? | ? | 1 |
| Tom & Jerry | 1 | 1 | 4 | 4 | 4 | ? |
| Doraemon | 1 | 0 | 5 | ? | ? | ? |

Điều đầu tiên ta làm đó là đặt sự tương đồng giữa 2 user ui, uj là sim (ui, uj). Nhìn bảng 3 ta thấy rằng user A và user B đều thích các bộ phim hành động Batman, Spyderman, Superman (đều được đánh giá 4 sao và 5 sao) và không thích các bộ phim hoạt hình như Tom & Jerry hay Doraemon cho lắm (vì chúng chỉ được đánh giá từ 0 cho đến 1 sao). Và điều này lại xảy ra ngược lại đối với các user D, E, F. Vì vậy ta cần phải xây dựng 1 hàm tìm điểm chung (similarity function) tốt để có thể đảm bảo rằng:

Hay nói một cách tổng quát:

Để xác định mức độ đánh giá của user A với bộ phim Superman, ta cần phải dựa vào user B. Ta may mắn vì user B không những tương đồng với user A mà còn đã đánh giá phim Superman. Vì vậy ta không cần tìm user khác.

Để đo được độ tương quan (similarity) giữa 2 user, ta sẽ đi xây dựng feature vector cho từng user (khác với content-based là ta đi xây dựng feature vector cho từng item). Feature vector sẽ được xây dựng dựa trên user, do đó sẽ rất khó khăn vì không phải item đều có giá trị đánh giá của user. Có 1 cách khắc phục tình trạng này đó là ta sẽ điền các giá trị tạm thời vào chỗ còn thiếu. Tuy nhiên, giá trị tạm thời này như thế nào là hợp lý để điền vào chỗ trống, là một vấn đề mà ta phải cân nhắc.

Nếu ta điền vào chỗ trống là 0, giá trị này là không tốt vì trong một số trường hợp 0 là giá trị nhỏ nhất, khiến kết quả không được khách quan, cũng như bị lệch, sai số nhiều trong một vài trường hợp. Một giải pháp nữa đó là ta sẽ điền giá trị trung bình được tính dựa trên giá trị đánh giá cao nhất và giá trị đánh giá thấp nhất. Tuy nhiên, ta làm như vậy một vài trường user sẽ không đúng vì đôi khi có một vài user sẽ rất dễ tính hay khó tính sẽ đánh giá nghiên về một bên của giá trị trung bình, nghĩa là các user này sẽ đánh giá quá thấp hoặc đánh giá quá cao so với mức ngưỡng trung bình. Vì vậy, trường hợp tối ưu đó là ta sẽ tính trung bình cộng các giá trị đánh giá của từng user sau đó ta sẽ lấy giá trị này điền tạm thời vào các ô còn thiếu của từng user.

Để hiểu rõ hơn quá trình thực hiện, ta sẽ xét ví dụ utility matrix sau:

**Bảng 4: Một ví dụ khác về utility matrix**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 5 | 5 | 2 | 0 | 1 | ? | ? |
| I1 | 4 | ? | ? | 0 | ? | 2 | ? |
| I2 | ? | 4 | 1 | ? | ? | 1 | 1 |
| I3 | 2 | 2 | 3 | 4 | 4 | ? | 4 |
| I4 | 2 | 0 | 4 | ? | ? | ? | 5 |
| Mean rating | 3.25 | 2.75 | 2.5 | 1.33 | 2.5 | 1.5 | 3.33 |

Sau đó, ta sẽ đi chuẩn hóa dữ liệu bằng cách trừ các ô có giá trị cho giá mean rating. Còn các ô “?” ta sẽ điền 0.

**Bảng 5: Một ultility matrix đã được chuẩn hóa**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 1.75 | 2.25 | -0.5 | -1.33 | -1.5 | 0 | 0 |
| I1 | 0.75 | 0 | 0 | -1.33 | 0 | 0.5 | 0 |
| I2 | 0 | 1.25 | -1.5 | 0 | 0 | -0.5 | -2.33 |
| I3 | -1.25 | -0.75 | 0.5 | 2.67 | 1.5 | 0 | 0.67 |
| I4 | -1.25 | -2.75 | 1.5 | 0 | 0 | 0 | 1.67 |

Từ đây ta có thể thấy rõ hơn rằng giá trị càng âm thì càng cho thấy user không đánh cao item. Ngược lại nếu giá trị càng dương thì càng cho thấy user đánh giá cao item. Còn các giá trị 0 biểu thị user chưa đánh giá item đó. Như vậy là đáp ứng được tối ưu các điều kiện cần của bài toán đặt ra.

Tiếp theo ta sẽ tính toán độ tương đồng giữa 2 user bằng hàm cosine hay cos (u1, u2).

Từ bảng 5 ta đi tính độ tương đồng cho các user, ta được kết quả sau:

**Bảng 6: Similarity giữa các user**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| U0 | 1 | 0.83 | -0.58 | -0.79 | -0.82 | 0.2 | -0.38 |
| U1 | 0.83 | 1 | -0.87 | -0.40 | -0.55 | -0.23 | -0.71 |
| U2 | -0.58 | -0.87 | 1 | 0.27 | 0.32 | 0.47 | 0.96 |
| U3 | -0.79 | -0.40 | 0.27 | 1 | 0.87 | -0.29 | 0.18 |
| U4 | -0.82 | -0.55 | 0.32 | 0.87 | 1 | 0 | 0.16 |
| U5 | 0.2 | -0.23 | 0.47 | -0.29 | 0 | 1 | 0.56 |
| U6 | -0.38 | -0.71 | 0.96 | 0.18 | 0.16 | 0.56 | 1 |

Sau khi tính xong độ tương đồng giữa các user, ta có thể thấy user u0 có nhiều nét tương đồng so với user u1 vì cos của u0 với u1 cao hơn so với cos của u0 với các user còn lại.

* + 1. Rating prediction

Ta đã hoàn thành thao tác tìm cặp user có mối quan hệ tương đồng với nhau. Ở mục này ta sẽ đi hoàn thiện nốt bài toán bằng việc đi dự đoán các giá trị còn thiếu.

Để dự đoán giá trị cho một ô còn thiếu ta làm các bước sau:

* Chọn k tương ứng với số dữ liệu cần tham khảo để dự đoán giá trị đang xét
* Lấy ra tập các user đã đánh giá item đang xét.
* Tính điểm tương đồng giữa user có item đang xét với các user này, chọn ra k user có điểm tương đồng cao nhất với user đang xét, ta gọi tập này .
* Tính tổng các tích được tính theo giá trị đánh giá của user tương đồng nhân với lại độ tương đồng giữa user đang xét với user đã đánh giá.
* Sau đó ta lấy tổng trên chia cho tổng k độ tương đồng (lấy dương độ tương đồng).

Nói ngắn gọn hơn ta dự đoán các giá trị bằng công thức:

Như vậy là ta đã có dự đoán và điền được hoàn thiện các chỗ còn thiếu:

**Bảng 7: Thay các giá trị chưa đánh giá (0) bằng các giá trị dự đoán**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 1.75 | 2.25 | -0.5 | -1.33 | -1.5 | **0.18** | **-0.63** |
| I1 | 0.75 | **0.48** | **-0.17** | -1.33 | **-1.33** | 0.5 | **0.05** |
| I2 | **0.91** | 1.25 | -1.5 | **-1.84** | **-1.78** | -0.5 | -2.33 |
| I3 | -1.25 | -0.75 | 0.5 | 2.67 | 1.5 | **0.59** | 0.67 |
| I4 | -1.25 | -2.75 | 1.5 | **1.57** | **1.56** | **1.59** | 1.67 |

Như vậy từ đây ta sẽ cộng lại các giá trị đánh giá cho giá trị mean hồi nãy mà ta đã trừ đi, ta sẽ khôi phục lại được các giá trị đánh giá gốc. Từ đây ta có thể đưa ra các gợi ý cho user dựa vào các điểm đánh giá.

* + 1. Khuyết điểm

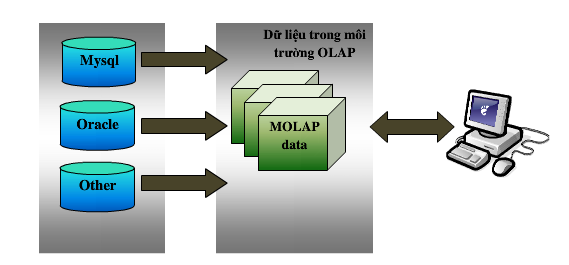
Trong thực tế thường có rất nhiều user việc đi tìm similarity giữa các user sẽ rất nhiều, việc chứa các kết quả similarity này đôi khi gặp khó khăn vì tài nguyên bộ nhớ có hạn khiến không thể lưu trữ hết các tính toán similarity.

Ngoài ra, utility matrix thường rất thưa (rất ít đánh giá). Dẫn tới bảng chuẩn hóa bị thưa và cần rất nhiều tính toán để có thể điền hết utility matrix.

* 1. Neighborhood-Based Collaborative Filtering (Item-Item Collabrative Filtering)
     1. Khắc phục khuyết điểm của User-User Collaborative Filtering
     2. Lập trình Neighborhood-Based Collaborative Filtering trên Python
  2. Matrix Factorization Collaborative Filtering
     1. Ý tưởng thuật toán
     2. Xây dựng và tối ưu hàm mất mát
     3. Bias
     4. Non-negative Matrix Factorization
     5. Incremental Matrix Factorization
     6. Các vấn đề khác
     7. Lập trình trên Python
  3. Các vấn đề về bảo mật
  4. Tổng kết chương 1

OLAP hỗ trợ nhiều mô hình lưu trữ dữ liệu khác nhau, mỗi mô hình có ưu và khuyết điểm riêng, được sử dụng tùy mục đích khai thác.

* + 1. Mô hình Multidimentional OLAP (MOLAP)
* Mô hình OLAP đa chiều lưu trữ dữ liệu dưới dạng file có cấu trúc đặc thù (như cấu trúc mảng) và thực hiện các tính năng OLAP trên cấu trúc này. Dù bị hạn chế về lượng dữ liệu lưu trữ và xử lý so với ROLAP. Tuy nhiên với các phép truy vấn hoặc tổng hợp số liệu lại cho hiệu năng tốt hơn do dữ liệu được thiết kế tối ưu cho truy vấn OLAP trong khi ROLAP phải thông qua database.
* MOLAP thích hợp cho các truy vấn tổng hợp dữ liệu thường xuyên, theo thời gian dài, cần thời gian hồi đáp nhanh. Ví dụ: tổng hợp doanh thu từng chi nhánh trong vòng 2 năm trở lại.

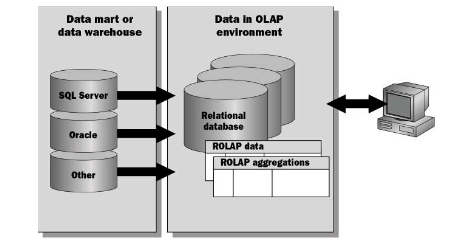


Hình 1.4: Mô hình dữ liệu MOLAP

* Ưu điểm:
* Thực thi nhanh: các khối trong MOLAP thu hồi dữ liệu nhanh và tối ưu hóa hoạt động.
* Có thể thực hiện các phép toán phức tạp: mọi tính toán tạo ra trước khi khối tạo ra.
* Nhược điểm:
* Giới hạn lượng dữ liệu xử lí: vì tất cả tính toán được sinh ra khi xây dựng khối nên không thể chứa lượng dữ liệu lớn trong khối chính nó.
* Yêu cầu đầu tư thêm: sử dụng MOLAP cần đầu tư bổ sung thêm vốn và nhân lực do công nghệ tạo khối thường độc quyền và không tồn tại trong tổ chức nào.
* Mô hình MOLAP cho phép thực hiện các truy vấn phân tích dữ liệu tốt nhất vì:
* Thông tin tổng hợp và dữ liệu cơ sở được lưu trữ trong cấu trúc đa chiều.
* Không cần dùng thao tác join – một trong những thao tác tốn chi phí nhất của mô hình quan hệ.
* Sử dụng các thuật toán nén dữ liệu cho phép lưu trữ với ít không gian hơn.
* Sử dụng chỉ mục bitmap cho hiệu quả thực thi tốt hơn.
* Lấy dữ liệu trong khối rất nhanh bằng cách sử dụng các xử lí truy vấn tốc độ cao và cache dữ liệu. Thông tin nhận được từ khối và các bảng OLAP cơ sở chỉ được truy xuất thông tin chi tiết.
* Có thể nạp trước vào bộ nhớ cache.
* Dữ liệu có thể dễ dàng sao chép đến client phục vụ cho việc phân tích không trực tuyến.

1. * 1. Mô hình Relational OLAP (ROLAP)

* Mô hình OLAP quan hệ lưu trữ dữ liệu trong CSDL quan hệ, dùng câu lệnh SQL để thực hiện các tính năng của OLAP.
* ROLAP thích hợp cho các truy vấn dữ liệu theo thời gian thực, xảy ra thường xuyên và đòi hỏi độ chính xác thức thời. Ví dụ: xem thông tin chứng khoán.



Hình 1.5: Mô hình dữ liệu ROLAP

* Ưu điểm:
* Có thể xử lí dữ liệu lớn do không bị giới hạn về kích thước dữ liệu.
* Có thể vận dụng chức năng vốn có của CSDL quan hệ để tiết kiệm chi phí.
* Nhược điểm:
* Hiệu suất xử lí thấp: do mỗi báo cáo ROLAP thường tập hợp dữ liệu từ nhiều bảng khác nhau tạo ra hiệu quả thấp khi dữ liệu lớn, phân tán.
* Giới hạn bởi chức năng của SQL: vì ROLAP chủ yếu dựa vào việc tạo ra các câu lệnh SQL truy vấn CSDL. Nhưng do báo cáo dựa trên truy vấn trong một số trường hợp không đạt hiệu quả mong muốn nên được khắc phục bằng cách tạo ra các công cụ hỗ trợ ngoài giúp người dùng tạo ra chức năng của riêng họ.
* Lưu trữ dữ liệu trong ROLAP cung cấp các lợi ích:
* Cho phép cube builder tự động tạo chỉ mục.
* Ánh xạ tổng hợp có sẵn từ kho dữ liệu. OLAP Manager sử dụng tổng hợp có sẵn mà không cần tính toán lại cho mỗi truy vấn.
* Tạo đòn bẩy cho hệ quản trị CSDL quan hệ nhằm giúp các nhà quản trị hệ thống duy trì hiệu quả hơn.
* Hỗ trợ SQL Server, Oracle, Access và Open Database Connectivity(ODBC).
  + 2. Mô hình Hybird OLAP (HOLAP)
* Mô hình OLAP lai là sự kết hợp giữa MOLAP và ROLAP, kết hợp những ưu điểm và loại bỏ những khuyết điểm của 2 mô hình nhằm tạo ra mô hình tối ưu. HOLAP tận dụng khả năng lưu trữ của OLAP và khả năng xử lý của MOLAP.
* Lưu trữ các khối trong HOLAP là tốt nhất cho các truy vấn tổng hợp dữ liệu thường xuyên dựa trên một lượng lớn dữ liệu cơ sở.
* Lợi ích của việc lưu trữ trong HOLAP:
* Lưu dữ liệu tổng hợp trong khối (theo MOLAP) để tăng tốc độ xử lý các truy vấn phức tạp.
* Lưu trữ dữ liệu cơ sở trong các bảng quan hệ (theo ROLAP) nhằm tiết kiệm không gian lưu trữ).
* Tiêu thụ ít không gian lưu trữ hơn MOLAP.
* Tránh trùng lặp dữ liệu.
* Ví dụ: trong một công ty chứng khoán
* Dữ liệu về thông tin nhân viên, doanh thu 2 năm trở về trước => lưu trong cube.
* Dữ liệu về thông tin chứng khoán, giá cổ phiếu hằng ngày => lưu trong bảng quan hệ.
* Tối ưu hóa vấn đề lưu trữ và xử lý hệ thống.
  + 1. ***So sánh các mô hình***

|  |  |  |  |
| --- | --- | --- | --- |
|  | MOLAP | ROLAP | HOLAP |
| Số chiếu | >2 | 2 | >2 |
| Lưu trữ dữ liện cơ sở | Khối | Bảng quan hệ | Bảng quan hệ |
| Lưu trữ thông tin tổng hợp | Khối | Bảng quan hệ | Khối |
| Hiệu suất thực hiện truy vấn | Nhanh nhất | Chậm nhất | Nhanh |
| Tiêu thụ không gian lưu trữ | Nhiều | Thấp | Trung bình |
| Chi phí bảo trì | Cao | Thấp | Trung bình |

Bảng 1: So sánh các mô hình OLAP

* 1. **Dịch vụ OLAP của SQL Server**
* Dịch vụ OLAP là server tầng giữa phục vụ cho OLAP.
* Hệ thống dịch vụ OLAP là công cụ mạnh trong việc xây dựng các khối đa chiều của dữ liệu, phục vụ cho phân tích và cung cấp khả năng truy xuất nhanh đến thông tin khối cho các client.
* Kiến trúc dịch vụ OLAP chia thành 2 phần:
* Phần server: đại diện bởi OLAP server.
* Phần client: là dịch vụ PivotTable.
* Cả dịch vụ OLAP và PivotTable đều cho phép thiết kế, tạo mới và quản lí các khối từ kho dữ liệu (data warehouse) và cho phép client truy xuất đến dữ liệu OLAP.
* OLAP server quản lí dữ liệu còn PivotTable làm việc với server để client truy xuất dữ liệu.
* Các đặc điểm của dịch vụ OLAP:
* Dễ sử dụng: bằng cách cung cấp các giao diện người dùng và các trợ giúp thực hiện.
* Linh động: mô hình dữ liệu mạnh cho định nghĩa khối và lưu trữ.
* Các khối có thể ghi: cho các kịch bản phân tích dạng “What if”.
* Kiến trúc có thể co dãn: cung cấp sự đa dạng lưu trữ và giải pháp tự động đối với “hội chứng bùng nổ dữ liệu” gây khó chịu cho các kĩ thuật OLAP.
* Tích hợp các công cụ quản tri, bảo mật, nguồn dữ liệu và client/server caching.
* Hỗ trợ rộng rãi các hàm API và kiến trúc mở để hỗ trợ các ứng dụng tùy ý.

CHƯƠNG 2: OLAP VÀ KHAI PHÁ DỮ LIỆU

OLAP là một kiểu của khai phá dữ liệu (data mining). Kỹ thuật này dùng để khai thác những dữ liệu thu thập được để giúp cho những nhà quản lý thực hiện những quyết định phù hợp tương ứng với nghiệp vụ mà mình đang quản lý.

Trong chương này sẽ gồm 2 phần cơ bản đó là tóm tắt lại các định nghĩa, thuật ngữ liên quan đến OLAP và thực hiện ví dụ thực tiễn cơ bản nhằm giúp người đọc có thể hình dùng được kỹ thuật OLAP.

* 1. Tóm tắt các khái niệm

Nhìn chung các hoạt động liên quan đến cơ sở dữ liệu có thể được chia thành 2 lớp rộng là: OLTP (Online Transaction Processing) và OLAP (Online Analytical Processing).

* + 1. OLTP (Online Transaction Processing)

Là kiểu khai thác cơ sở dữ liệu (CSDL) truyền thống mà chúng ta thường hay sử dụng. Các hoạt động khai thác thường là các câu truy vấn hay các câu lệnh cập nhật nhỏ, đơn giản. Thường các câu lệnh này chỉ ảnh hưởng đến một phần nhỏ trong dữ liệu và môi trường có thể luôn được thực hiện thường xuyên.

* + 1. OLAP (Online Analytical Processing)

Đối lập với OLTP, trong OLAP chúng ta sẽ thực hiện các thao tác tác lớn hơn và phức tạp hơn trong vấn đề phân tích dữ liệu. Trong công việc khai phá dữ liệu (data mining), OLAP thực hiện các công việc phức tạp và các công việc này sẽ thực hiện trên một mảng dữ liệu lớn khác biệt so với OLTP. Đồng thời chúng ta có thể không cần cập nhật dữ liệu một cách thường xuyên như OLTP. Tiếp theo ta sẽ tóm tắt một vài thuật ngữ liên quan tới OLAP.

* Data warehousing: là một cơ sở dữ liệu được tổng hợp từ các nguồn dữ liệu trong OLTP. Trong OLAP, data warehousing được dùng để thực hiện các công việc phân tích dữ liệu nhằm phục vụ các nghiệp vụ thực tiễn.
* Decision support system (DSS): là một hệ thống cơ sở hạ tầng được dùng để phân tích dữ liệu.
* Star Schema: là mô hình thiết kế mà OLAP được xây dựng. Bao gồm 2 thành phần cơ bản là **Fact table** (bảng sự thật) và **Dimension tables** (các bảng con)
* Fact table (trong OLAP thường chỉ có 1 bảng Fact table): là bảng chứa nhiều dữ liệu được cập nhật thường xuyên
* Dimension tables: là các bảng con và các dữ liệu trong các bảng này ít khi được cập nhật thường xuyên. Và các dữ liệu này cũng không lớn. Mỗi bảng con thể hiện một chiều trong bảng sự thật.

Ta có ví dụ[1] sau:

* Sales(storeID, itemID, custID, qty, price)
* Store(storeID, city, state)
* Item(itemID, category, brand, color, size)
* Customer(custID, name, address)

Trong đó Sales sẽ là bảng sự thật chứa đựng những dữ liệu đang diễn ra đó là nghiệp vụ bán hàng diễn ra, có thể được miêu tả như sau: sản phẩm có mã là itemID được bán cho khác hàng là custID ở cửa hàng storeID với số lượng là qty, bán với giá là Price. Miêu tả cho các dữ liệu trên là các bảng Store (bảng cửa hàng), Item (danh sách các món hàng mà doanh nghiệp bán), Customer (bảng khác hàng lưu những khách hàng mua và đã mua sản phẩm của doanh nghiệp). Mỗi bảng con thể hiện một chiều cho bảng sự thật, ví dụ: bảng Store sẽ miêu tả chi tiết thông tin của storeID được lưu trữ trong bảng sự thật Sales. Tương tự đối với itemID và custID. Do đó bảng sự thật sẽ có 3 chiều ứng với nó là 3 bảng khác nhau.

Trong ví dụ trên bảng sự thật có 3 chiều tương ứng với 3 thuộc tính là storeID, itemID, custID. Các thuộc tính này được gọi là thuộc tính chiều (Dimension attribute). 2 thuộc tính còn lại trong bảng sự thật là: qty và price được gọi là thuộc tính phụ thuộc (Dependent attribute).

* + 1. Sự khác nhau giữa OLAP và OLTP

OLAP có thể xử lí các truy vấn liên quan đến lượng dữ liệu rất lớn mà OLTP không thể cho kết quả hoặc mất nhiều thời gian để thực thi.

|  |  |  |
| --- | --- | --- |
|  | **OLTP** | **OLAP** |
| Ứng dụng | Vận hành: ERP, CRM, ứng dụng truyền thống, tin tức, weblog, ... | Hệ thống quản lý thông tin, Hệ thống hỗ trợ ra quyết định,... |
| Người dùng chủ yếu | Người dùng, nhân viên | Quản lý, điều hành |
| Phạm vi xử lý | Hàng tuần/tháng | Hàng năm |
| Mức độ làm mới | Ngay tức thì | Định kỳ |
| Mô hình dữ liệu | Entity-relationship, hoặc thích thì có thể BigData, key-value,... | Multi-dimensional |
| Schema | Normalized (Chuẩn hoá dữ liệu) | Star hoặc snowflake |
| Nhấn mạnh | Cập nhật | Truy hồi thông tin |
| Số lượng người dùng | Hàng ngàn tới hàng triệu | Hàng trăm |
| Dung lượng ổ đĩa | MB tới GB | GB tới TB |
| Đo lường | Số lượng giao dịch/người dùng đồng thời | Số lượng câu truy vấn và thời gian phản hồi |

Bảng 2: So sánh OLTP và OLAP

* + 1. Chuyển đổi dữ liệu từ OLTP tới OLAP

Thực hiện thông qua các qui trình sau:

* Hợp nhất dữ liệu: tất cả dữ liệu liên quan tới các mục đặc trưng phải có khả năng hợp nhất từ nhiều hệ thống OLTP tới một hệ thống OLAP đơn. Qui trình hợp nhất phải giải quyết được sự khác nhau về mã hóa giữa các hệ thống OLAP, phù hợp với dữ liệu chung bằng cách so sánh các trường tương tự, có thể biến đổi dữ liệu lưu trữ từ nhiều loại khác nhau trong OLTP thành một loại duy nhất sử dụng trong OLAP. Các hệ thống cung cấp dữ liệu đầu vào không nhất thiết phải là OLTP truyền thống mà có thể lưu trữ ở nhiều dạng hợp lệ.
* Quét dữ liệu: quá trình hợp nhất dữ liệu OLTP vào kho dữ liệu có thể gây ra lỗi chính tả. Việc quét dữ liệu được tạo ra nhằm điều chỉnh sự không thống nhất trước khi dữ liệu được nhập vào kho lưu trữ phục vụ cho OLAP.
* Tập hợp dữ liệu: OLAP chỉ truy vấn những dữ liệu tổng kết cần thiết, hoặc các dữ liệu được tập hợp bằng một số quy tắc nhất định.
* Sắp xếp dữ liệu: dữ liệu OLTP chuyển vào kho lưu trữ sẽ được biến đổi theo cách sắp xếp hợp lí hơn đối với nhu cầu phân tích nhằm đưa ra quyết định hạn chế tiêu phí thời gian. Quá trình thiết lập kho lưu trữ bao gồm việc sắp xếp lại dữ liệu OLTP, lưu trong các bảng biểu liên kết thành dữ liệu OLAP lưu trong các khối đa chiều. Sau đó dữ liệu được tải vào kho lưu trữ.
* Truy cập và phân tích dữ liệu: khi dữ liệu tải vào kho, OLAP cung cấp khả năng truy cập, xem và phân tích dữ liệu với độ linh hoạt và hiệu quả cao. OLAP trình bày dữ liệu thông qua mô hình dữ liệu tự nhiên và trực quan, giúp cho người sử dụng xem và hiểu một cách tốt nhất các thông tin trong kho lưu trữ. Từ đó giúp người sử dụng nhận biết giá trị dữ liệu.
  1. Quy trình thực hiện OLAP.

Các câu truy vấn trên mô hình OLAP sẽ thực hiện theo trình tự: join, lọc, nhóm, và tổng hợp. Join là thao tác truy vấn nhiều bảng thông qua các mối quan hệ giữa bảng sự thật với các bảng con (khóa chính, khóa ngoại). Thao tác này có thể giúp mở rộng các sự thật trong bảng sự thật, ví dụ như ta có thể lấy được các món hàng mà cửa hàng storeID bán,... Tiếp tục là giai đoạn lọc, ở giai đoạn ta có thêm vào câu truy vấn thao tác lọc những món hàng itemID được bán ở cửa hàng storeID mà custID mua. Sau đó ta sẽ gom nhóm lại và tổng hợp tạo thành bảng kết quả. Qua quy trình này ta có thể lấy được những món hàng itemID mà custID hay mua ở storeID để từ đó ta có thể đưa ra các chính sách phục vụ hợp lý cho custID.

Việc thực hiện truy vấn trên những dữ liệu lớn như thế này có thể sẽ ảnh hưởng đến hiệu suất của chương trình vì các truy vấn sẽ thực hiện càng chậm nếu dữ liệu càng lớn.

* 1. Tóm tắt mô hình hóa OLAP.

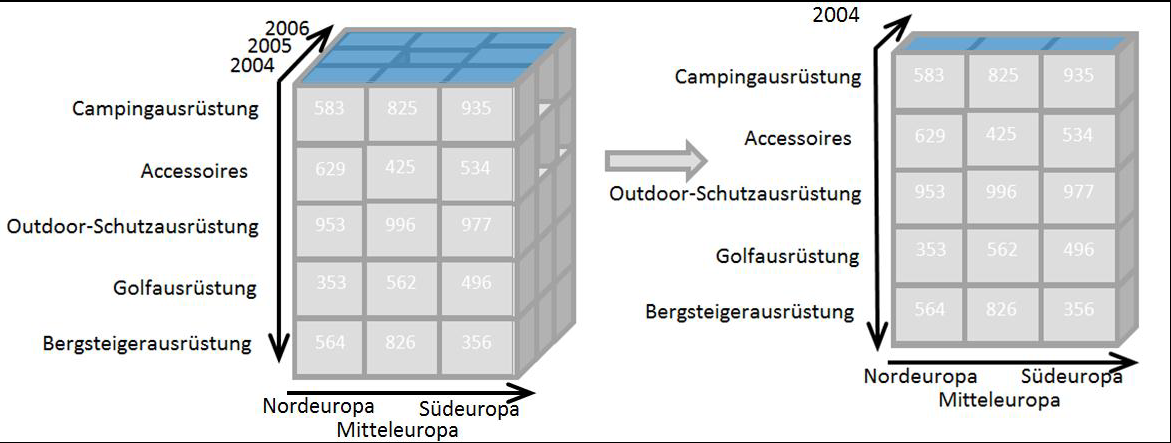
OLAP được mô hình hóa thành các khối dữ liệu lập phương. Trong đó các bảng con sẽ tạo thành các chiều của khối lập phương, các dữ liệu liên quan sẽ nằm trên các dòng của khối lập phương. Cuối cùng các dữ liệu được tổng hợp thành từ các câu truy vấn sẽ nằm ở các rìa, cạnh và đỉnh của khối lập phương. Dữ liệu trong OLAP không phải là khối lập phương 3 chiều mà có thể là một dạng hình có kích thước n chiều (phụ thuộc vào số chiều mà bảng sự thật có).

Các thuộc tính chiều trong bảng sự thật phải là độc nhất (phải là khóa (khóa chính ở bảng sự thật và ở các bảng con)). Nếu các thuộc tính chiều không phải là khóa thì nó cần phải được lập.

Một điều cần chú ý nữa đó là nếu đưa thuộc tính Date vào bảng sự thật thì cần được cân nhắc xem nó nên là thuộc tính chiều hay là thuộc tính phụ thuộc.

* 1. Tóm tắt các loại thao tác trong OLAP.
     1. Slice

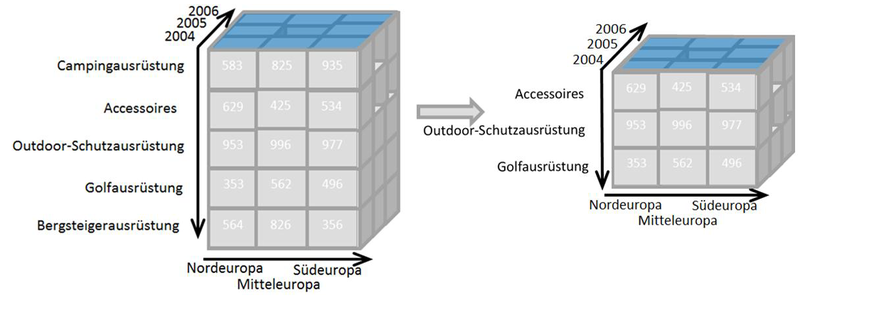
Slice là hành động chọn một tập hợp con (1 chiều) của khối lập phương bằng cách chọn một giá trị đơn cho một trong các chiều của khối lập phương, tạo ra một hình khối mới với một kích thước nhỏ hơn. Như hình vẽ sau:



Hình 2.1: Mô tả thao tác Slice[2]

* + 1. Dice

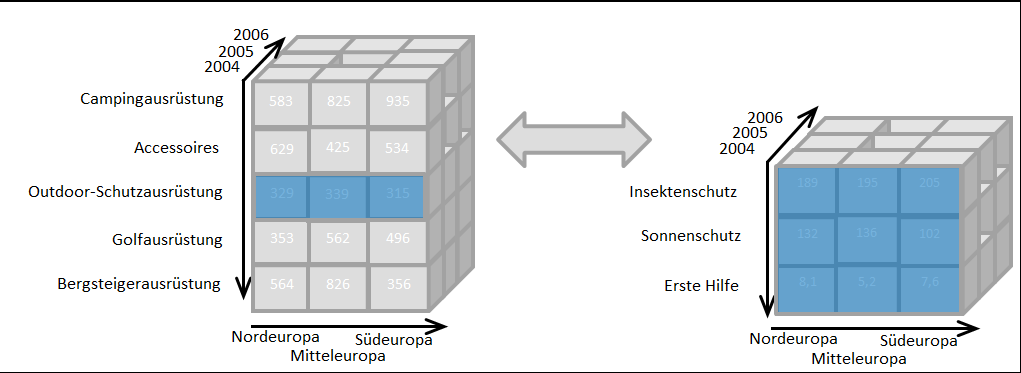
Dice: tạo ra một khối con bằng cách cho phép các nhà phân tích lựa chọn các giá trị cụ thể thuộc nhiều chiều khác nhau.



Hình 2.2: Mô tả thao tác Dice[2]

* + 1. Drill Down hay Drill Up

Cho phép người dùng điều hướng giữa các mức dữ liệu khác nhau, từ phần tổng quát hóa nhất (chiều đi lên) đến chi tiết hóa nhất (chiều đi xuống giảm).



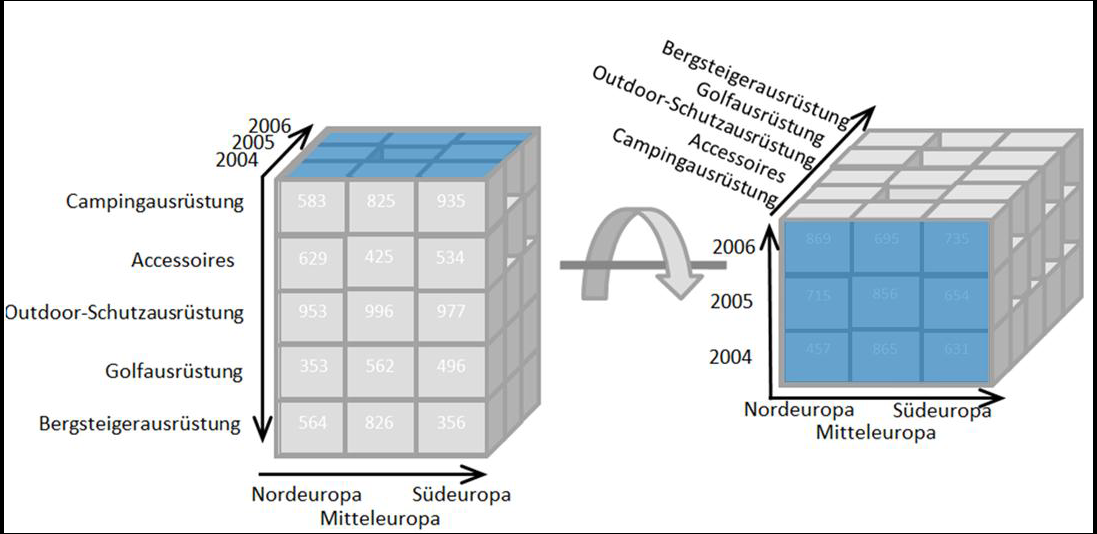
Hình 2.3: Mô tả thao tác Drill Down/Up[2]

* + 1. Roll up

Roll up: Một Roll up bao gồm những dữ liệu được tổng hợp theo một chiều. Quy tắc tổng hợp có thể là tổng số tính toán theo một hệ thống phân cấp hoặc áp dụng một bộ công thức như "lợi nhuận = doanh thu - chi tiêu".

* + 1. Pivot

Pivot cho phép nhà phân tích xoay khối lập phương trong không gian để xem các mặt khác nhau của nó. Ví dụ: các thành phố có thể được sắp xếp theo chiều dọc và sản phẩm theo chiều ngang khi xem dữ liệu cho một phần tư cụ thể. Pivot có thể thay thế sản phẩm bằng các khoảng thời gian để xem dữ liệu theo thời gian cho một sản phẩm.[2]



Hình 2.4: Mô tả thao tác Pivot[2]

CHƯƠNG 3: ỨNG DỤNG OLAP VÀO BÀI TOÁN CỤ THỂ

* 1. Mô tả bài toán

Một cửa hàng vật liệu xây dựng muốn xây dựng CSDL để quản lý hoạt động kinh doanh của cửa hàng.

Cửa hàng mua nguyên vật liệu từ các nhà cung cấp. Mỗi lần cần mua nguyên vật liệu, cửa hàng cần lập đơn đặt hàng ghi rõ các mặt hàng cần mua, số lượng. Thông tin về nguyên vật liệu là mã vật liệu, tên vật liệu, đơn vị tính. Trong đơn đặt hàng cần ghi rõ mã hóa đơn, ngày lập và nhân viên lập. Khi nhà cung cấp giao hàng, cửa hàng sẽ lập phiếu nhập hàng gồm các thông tin mã phiếu nhập, ngày giờ nhập, nhập theo đơn hàng nào, các mặt hàng nhập, số lượng và đơn giá nhập. Một đơn đặt hàng có thể phải nhập hàng nhiều lần nhưng mỗi lần nhập hàng chỉ nhập theo một đơn đặt hàng. Các mặt hàng được phân theo loại hàng để tiện quản lý. Thông tin của loại hàng gồm mã loại, tên loại.

Các khách hàng sẽ đến cửa hàng mua vật liệu. Mỗi lần khách hàng mua vật liệu nhân viên của cửa hàng sẽ lập hoá đơn, trong đó ghi rõ ngày lập, xuất cho khách hàng nào, nhân viên lập, danh sách chi tiết các mặt hàng cùng số lượng, đơn giá xuất, tổng thành tiền.

Khách hàng có thể thanh toán nhiều lần cho mỗi lần mua hàng và tối đa là 5 lần cho một hoá đơn. Mỗi lần thanh toán phải nhiều hơn 20% tổng số tiền trên hoá đơn, thời gian thanh toán cho một hoá đơn là không quá 60 ngày. Khách hàng chỉ được mua hàng lần tiếp theo khi đã thanh toán xong hoá đơn của lần mua trước đó.

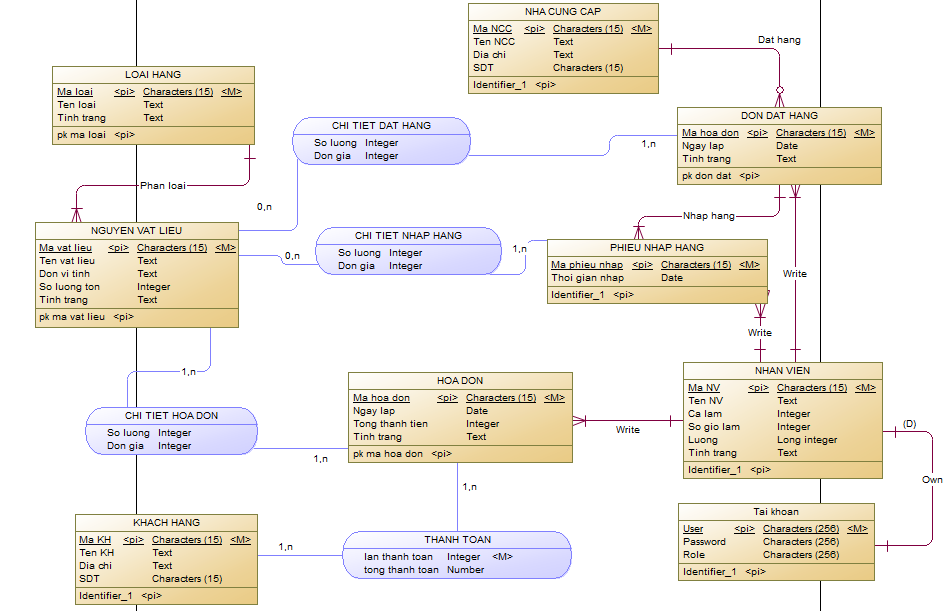
Cuối tháng, bộ phận thu ngân sẽ lập báo cáo thống kê về thu, chi của cửa hàng. Thống kê mặt hàng ưa chuộng, khách hàng thường xuyên, ...

* 1. Mô hình thực thể ban đầu (CDM Model)

Các thực thể bao gồm:

* Nguyên vật liệu: Mã vật liệu, tên vật liệu, đơn vị tính.
* Đơn đặt hàng: Mã hóa đơn, ngày lập.
* Nhân viên: Mã nhân viên, Tên nhân viên, Ca làm, số giờ làm và lương
* Phiếu nhập hàng: Mã phiếu nhập, ngày giờ nhập.
* Nhà cung cấp: Mã nhà cung cấp, Tên nhà cung cấp, Địa chỉ, Số điện thoại
* Loại hàng: Mã loại, tên loại
* Hóa đơn: Mã hóa đơn, ngày lập, Tổng thành tiền
* Phiếu thanh toán: Mã thanh toán, Ngày thanh toán, Tổng thanh toán

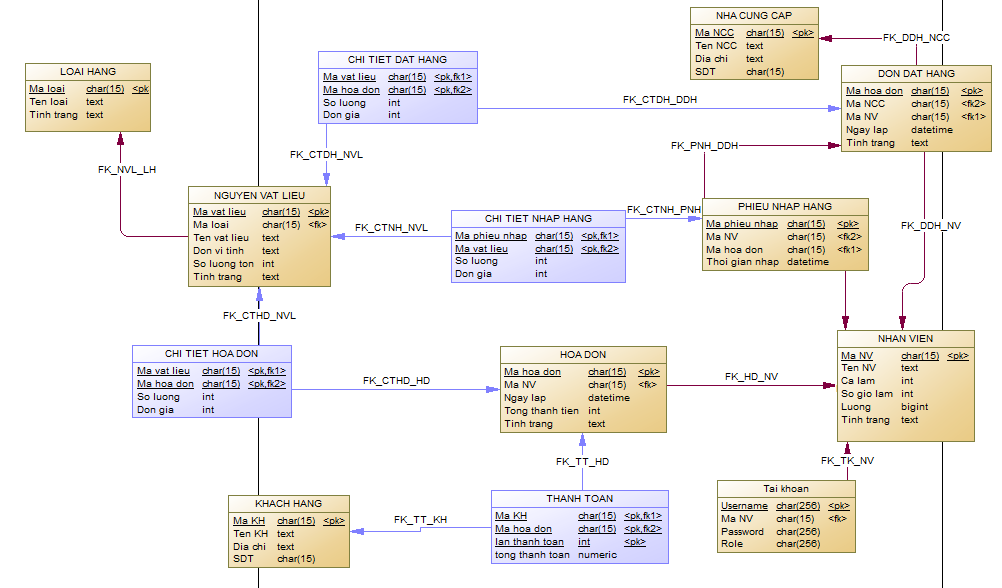
Khách hàng: Mã khách hàng, tên khách hàng, Địa chỉ, Số điện thoại



Hình 3.1: Mô hình thực thể của bài toán

* 1. Mô hình vật lý ban đầu của bài toán

Sau khi xây dựng xong mô hình thực thể, ta có thể sinh ra được mô hình vật lý như sau:



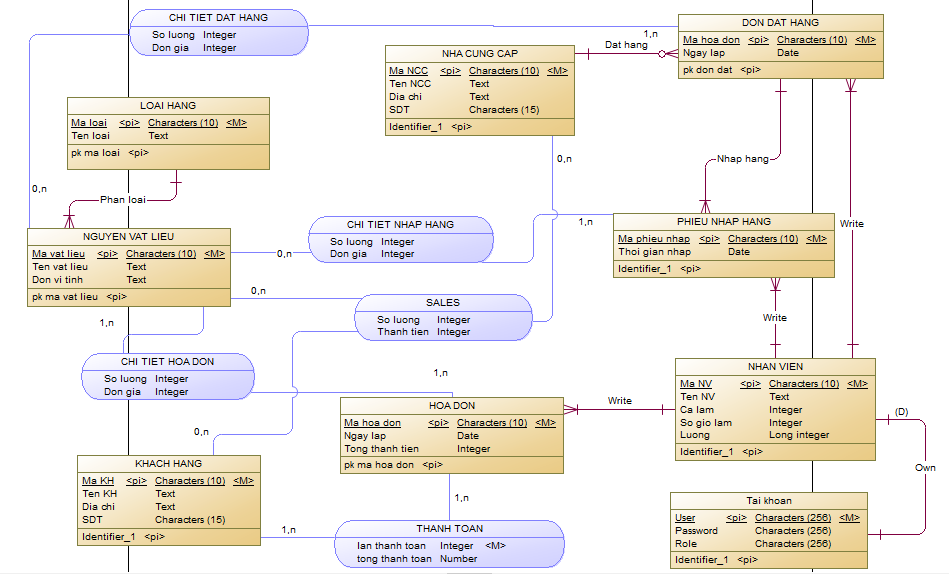
Hình 3.2: Mô hình vật lý của bài toán

* 1. Triển khai OLAP cho bài toán

Như ta đã thấy cơ sở dữ liệu ở đây chỉ dừng ở dạng truyền thống là OLTP, chỉ có những bảng dữ liệu đơn giản. Để chuyển từ OLTP sang OLAP, ta sẽ phải lập mô hình star schema cho bài toán. Để có lập star schema, ta cần phải có bảng sự thật và bảng con của nó. Ở đây, ta chỉ mới có các bảng con.

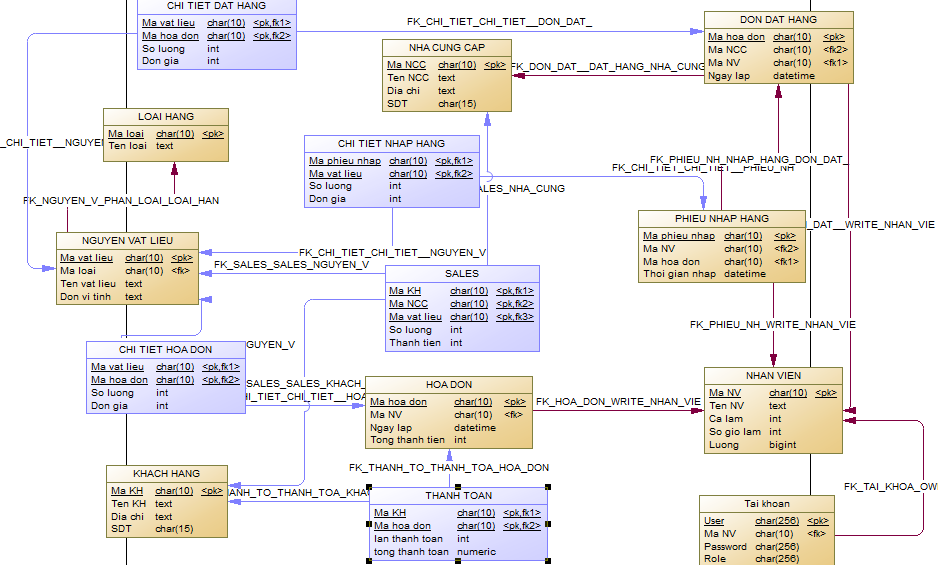
Tùy vào mục đích quản lý của nhà quản trị mà ta có thể xây dựng nhiều loại bảng sự thật khác nhau. Ví dụ ở đây ta sẽ triển khai bảng sự thật (đặt tên bảng là SALES) như sau: Bảng sự thật sẽ gồm các lưu thông tin về các nhà cung cấp mã là MA\_NCC, các khách hàng có mã MA\_KH, và các món hàng có mã là MA\_VAT\_LIEU dùng để theo dõi tình hình mua bán của cửa hàng. Từ các dữ liệu ta có thể thu thập một số thông tin như vật liệu nào được khác hàng có mã là MA\_KH mua nhiều hay nhà cung cấp có mã là MA\_NCC có vật liệu được nhiều khác hàng mua,... Như vậy bảng sự thật đã được hình thành với 3 thuộc tính chiều là: MA\_NCC, MA\_KH, MA\_VAT\_LIEU tạo thành 3 chiều cho bảng sự thật tương ứng cho 3 bảng con là NHA\_CUNG\_CAP, KHACH\_HANG và NGUYEN\_VAT\_LIEU để mô tả cho 3 chiều của bảng sự thật. Ngoài ra để hỗ trợ ta cần thêm 2 thuộc tính phụ thuộc đó là: số lượng và thành tiền cho bảng sự thật.

Mô hình CDM ban đầu sau khi thêm bảng sự thật sẽ trở thành:



Hình 3.3: Mô hình CDM sau khi thêm bảng sự thật

Và mô hình vật lý tương ứng sẽ là:



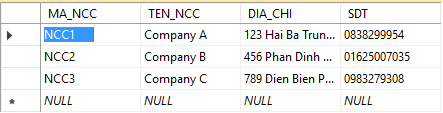
Hình 3.4: Mô hình vật lý sau khi thêm bảng sự thật

Về cơ bản là ta đã xây dựng star schema để triển khai kỹ thuật OLAP.

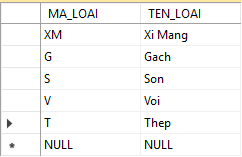
* 1. Áp dụng OLAP trên hệ quản trị cơ sở dữ liệu

Để hình dung về OLAP rõ hơn, ta sẽ tạo cơ sở dữ liệu này trên SQL Sever.

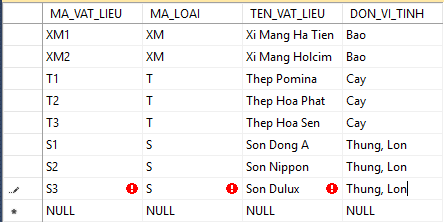
Sau khi tạo xong cơ sở dữ liệu, ta tiến hành tạo một số bộ giá trị cho các bảng nhằm xây dựng hoàn tất ví dụ minh họa. Một số giá trị minh họa của một số bảng như sau:



Hình 3.5: Dữ liệu minh họa cho bảng NHA\_CUNG\_CAP



Hình 3.6: Dữ liệu minh họa cho bảng LOAI\_HANG

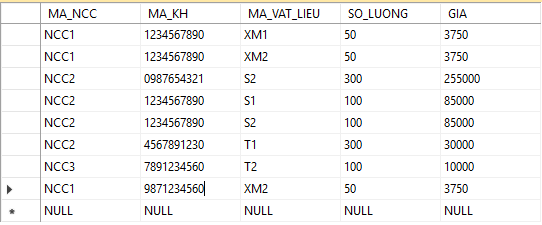


Hình 3.7: Dữ liệu minh họa cho bảng NGUYEN\_VAT\_LIEU



Hình 3.8: Dữ liệu minh họa cho bảng KHACH\_HANG

Như vậy cơ bản là ta đã xây dựng xong bộ giá trị (dữ liệu) minh họa cho các bảng con. Tiếp theo sẽ là bộ giá trị cho bảng sự thật.



Hình 3.9: Dữ liệu minh họa cho bảng sự thật SALE

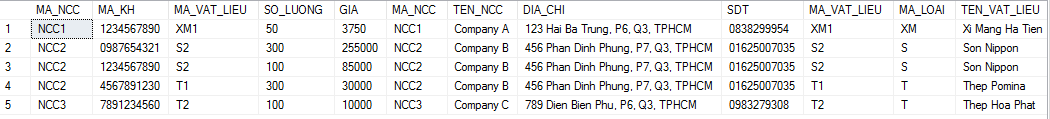
Sau khi ta thực hiện xong việc tạo bộ dữ liệu, ta sẽ tiếp tục đến phần áp dụng kỹ thuật OLAP.

Bước đầu tiên của việc sử dụng OLAP đó là ta sẽ tiến hành join các bảng lại với nhau qua câu lệnh truy vấn đơn giản sau:



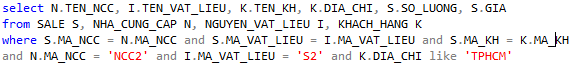
Hình 3.10: Câu lệnh Join các bảng của mô hình star schema

Sau khi thực hiện câu lệnh trên ta được kết quả là một bảng gồm tất cả các trường của cả 4 bảng bao gồm bảng sự thật và 3 bảng con.



Hình 3.11: Một phần kết quả của truy vấn join

Từ đây, ta có thể thực hiện các thao tác khác dựa trên câu lệnh join. Như ví dụ sau:



Hình 3.12: Áp dụng câu lệnh join cho truy vấn dùng OLAP

Sau khi thực thi truy vấn hệ quản trị cơ sở dữ liệu sẽ trả về kết quả như sau:



Hình 3.13: Kết quả áp dụng kỹ thuật OLAP

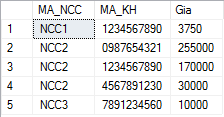
Ta sẽ thu được danh sách có chứa các dữ liệu thỏa mãn nhà cung cấp có MA\_NCC là ‘NCC2’, tên vật liệu có mã là ‘S2’, khách hàng có nơi ở là ‘TPHCM’, ta thu được kết quả đó là khách hàng ‘Nguyen Van B’ sống ở ‘TPHCM’ mua ‘Sơn Nippon’ của Công ty B với số lượng là 300 và giá là 255 triệu đồng.

Bên cạnh join ta có dùng group by để nhóm các đối tượng lại với nhau để có giúp nhà tổng kết tình hình của một nghiệp vụ nào đó, ví dụ ta có câu lệnh sau:



Hình 3.14: Câu lệnh group by đơn giản trên nền OLAP

Câu lệnh này sẽ trả về kết quả là báo cáo tổng kết số tiền thu theo các khách hàng, khác nhau từ việc bán các sản phẩm được phân phối theo các nhà sản xuất tương ứng. Và đây là kết quả trả về từ câu lệnh trên:



Hình 3.15: Kết quả sau khi dùng câu lệnh group by

Tiếp tục, ta sẽ thực hiện các thao tác khác của OLAP:

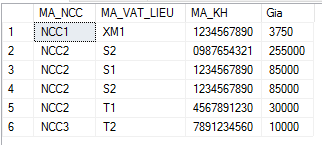
* **Drill Down:**

Như ta đã biết Drill Down là quá trình đi từ tổng quát hóa đến chi tiết hóa trong OLAP nên ta sẽ có ví dụ như sau:



Hình 3.16: Ví dụ về Drill Down

Và ta sẽ thu được kết quả là:

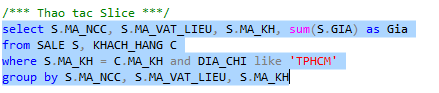


Hình 3.17: Kết quả của ví dụ về Drill Down

Trả về cụ thể danh sách các dữ liệu được gom nhóm theo MA\_NCC, MA\_VAT\_LIEU, MA\_KHACH\_HANG. Qua đó ta có thể thấy chi tiết hóa rõ ràng đó là khách hàng ‘1234567890’ đã mua sơn S1 và sơn S2 của nhà cung cấp ‘NCC2’ với giá lần lượt là 85 triệu đồng và 85 triệu đồng. Thay vì là 170 triệu như bảng ở hình 1.19.

* **Slice:**

Là thao tác cắt khối OLAP ra một phần và cắt theo một chiều nhất định. Trong ví dụ sau ta sẽ cắt khối OLAP theo chiều khách hàng với ràng buộc là chỉ lấy ra danh sách các khách hàng ở TPHCM đã mua hàng của cửa hàng.



Hình 3.18: Câu lệnh Slice

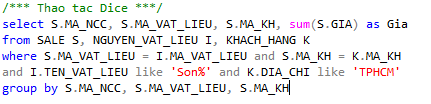
Kết quả là:



Hình 3.19: Kết quả của câu lệnh Slice

* **Dice:**

Là thao tác ta sẽ cắt khối OLAP theo nhiều mà ta muốn. Kết quả ta thu được sẽ là một khối con nhỏ thuộc khối OLAP. Như ví dụ sau, ta sẽ cắt khối OLAP theo chiều khách hàng với ràng buộc là chỉ lấy ra danh sách các khách hàng ở TPHCM đã mua hàng của cửa hàng và cắt theo chiều thứ 2 là nguyên vật liệu với ràng buộc là chỉ lấy các nguyên vật liệu là sơn.



Hình 3.20: Câu lệnh thao tác Dice

Kết quả là:



Hình 3.21: Kết quả câu lệnh Dice

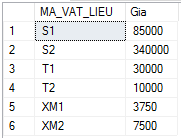
* Roll up:

Roll up là thao tác đối lập với Drill Down, nghĩa là ta sẽ biểu diễn thông tin truy vấn từ cụ thể hóa thành tổng quát hóa. Như ví dụ sau:



Hình 3.22: Câu lệnh Roll up

Lấy ra danh sách các dữ liệu về vật liệu, tiền mà doanh nghiệp thu được từ vật liệu đó.



Hình 3.23: Kết quả câu lệnh Roll Up

* 1. Áp dụng OLAP trên chương trình

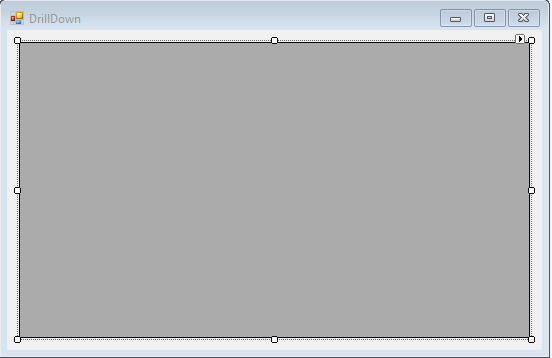
Từ mục 3.6 ta sẽ áp dụng vô chương trình 4 chức năng cơ bản. Một là, hiển thị toàn bộ thông tin kinh doanh (dùng thao tác Drill Down). Hai là, lọc ra các sản phẩm ở tỉnh được chọn có khách hàng mua (dùng thao tác Slice). Ba là, lọc ra danh sách các sản phẩm có tên loại hàng được chọn mà khách hàng ở tỉnh được chọn mua (dùng thao tác Dice). Bốn là, tổng quát kết quả mua của từng sản phẩm (dùng thao tác Roll Up). Ngoài ra OLAP còn có thể áp dụng vào chương trình nhiều chức năng khác, nhưng do phạm vi ứng dụng quá rộng nên báo cáo này xin rút ngọn để minh họa cho 4 thao tác cơ bản đã được nêu trên.

Do OLAP hỗ trợ cho các quyết định của nhà quản lý nên OLAP chỉ được đưa vào phân quyền “admin” (quản trị) của chương trình.

Chương trình được thiết kế theo mô hình 3 lớp. Ở mỗi lớp, bài báo cáo sẽ trình bày ngắn gọn các lớp được gọi và phương thức chính phục vụ cho hoạt động của chức năng. Cuối cùng, sẽ là phần trình bày kết quả thu được của chương trình

* **Chức năng 1**: Hiển thị toàn bộ thông tin kinh doanh.

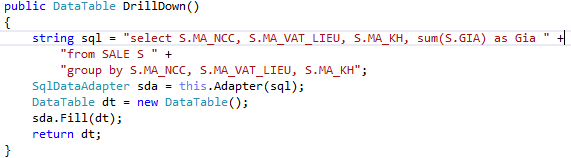
Giao diện chính của chứng năng:



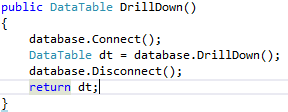
Hình 3.24: Giao diện của chức năng 1

Lớp DAO của mô hình sẽ có 2 lớp chính phục vụ cho chương trình đó là:

* Database: chứa các hàm có câu lệnh liên quan đến vấn đề xử lý, truy vấn trên SQL Server.
* Lớp OLAP\_DAO: Gọi các phương thức xử lý cơ sơ dữ liệu trong lớp Database.

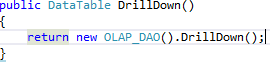


Hình 3.25: Phương thức ở lớp Database phục vụ cho chức năng 1



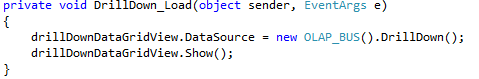
Hình 3.26: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 1

Tiếp theo là lớp BUS của mô hình:



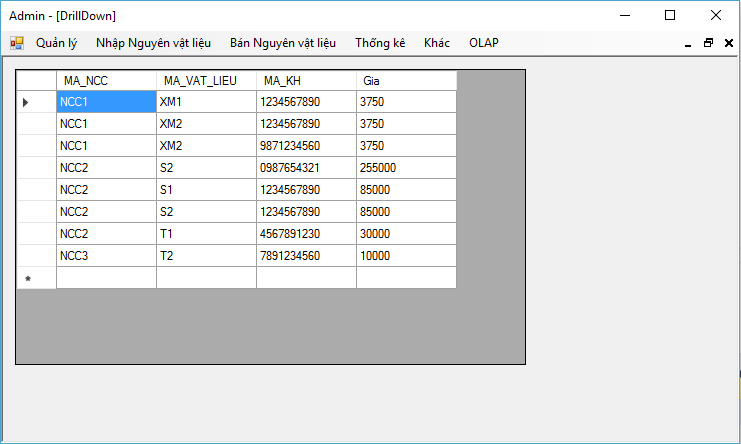
Hình 3.27: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 1

Cuối cùng là phần trên của chương trình (giao diện):



Hình 3.28: Phương thức phục vụ hiển thị kết quả của chức năng 1

Kết quả của chương trình:

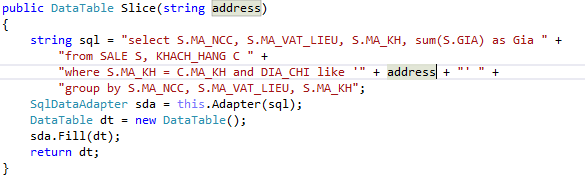


Hình 3.29: Kết quả của chức năng 1

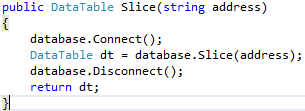
* **Chức năng 2**: Lọc danh sách các sản phẩm (thao tác Slice)

Chức năng này có tác dụng đó là giúp nhà quản trị có thể biết được các sản phẩm nào được mua ở các tỉnh được chọn. Từ đó nhà quản trị có thể đưa ra các chính sách như khuyến mãi, bảo hành, chăm sóc khách hàng,... sao cho phù hợp.

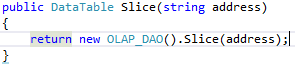
Tương tự như chức năng 1, chương trình sẽ có các phần qua trọng sau:



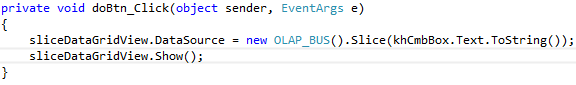
Hình 3.30: Phương thức ở lớp Database phục vụ cho chức năng 2



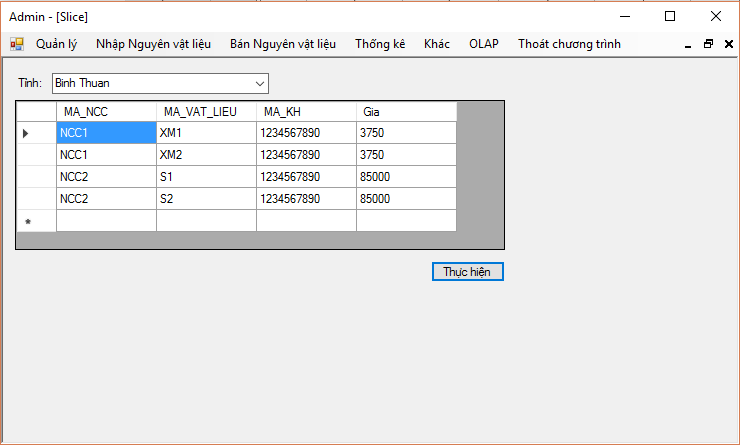
Hình 3.31: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 2



Hình 3.32: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 2



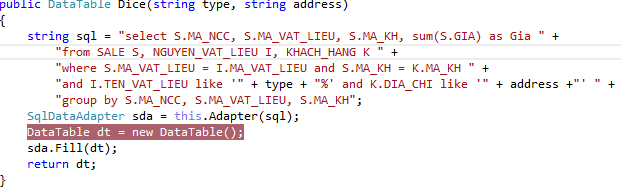
Hình 3.33: Phương thức phục vụ hiển thị kết quả của chức năng 2



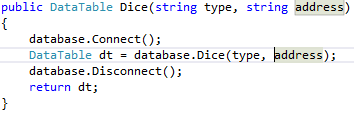
Hình 3.34: Giao diện chung và kết quả chức năng 2

* **Chức năng 3**: Lọc danh sách các sản phẩm (thao tác Dice)

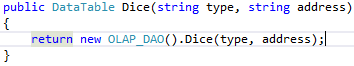
Chức năng này giúp lấy được danh sách chi tiết sản phẩm có tên loại hàng được chọn mà khách hàng ở tỉnh được mua. Chức năng này có thể hỗ trợ nhà quản lý giới hạn tên các mặt hàng thuộc loại hàng được chọn để đưa ra các quyết định như áp dụng các chương trình khuyến mãi, đầu tư,...



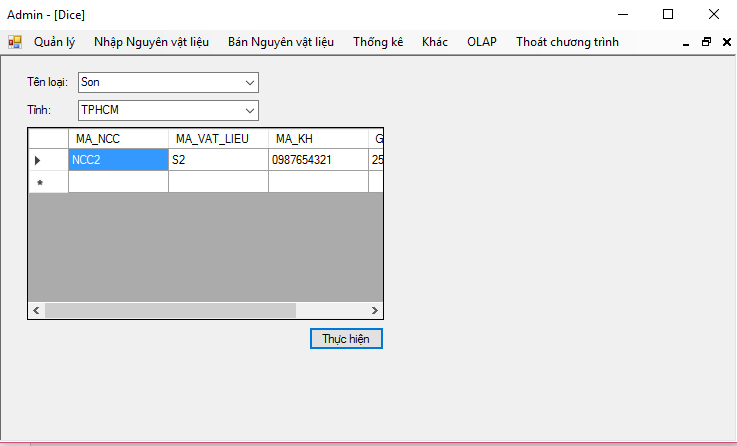
Hình 3.35: Phương thức ở lớp Database phục vụ cho chức năng 3



Hình 3.36: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 3



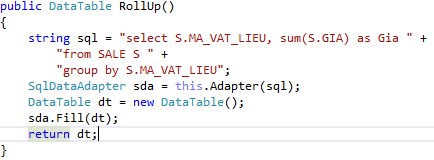
Hình 3.37: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 3



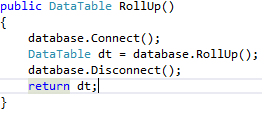
Hình 3.38: Giao diện chung và kết quả chức năng 3

* **Chức năng 4**: Tổng quát kết quả kinh doanh của từng sản phẩm

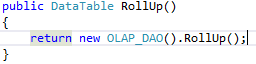
Tổng kết giá trị thu được theo các sản phẩm mà cửa hàng bán.



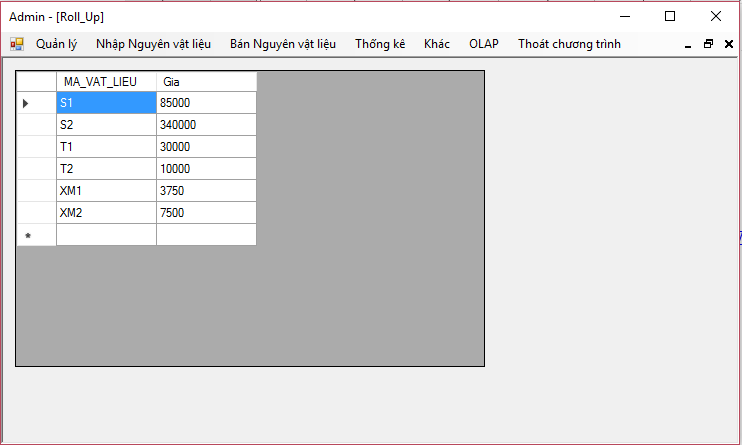
Hình 3.39: Phương thức ở lớp Database phục vụ cho chức năng 4



Hình 3.40: Phương thức ở lớp OLAP\_DAO phục vụ cho chức năng 4



Hình 3.41: Phương thức ở lớp OLAP\_BUS phục vụ cho chức năng 4



Hình 3.42: Giao diện chung và kết quả chức năng 4

TÀI LIỆU THAM KHẢO

**Tiếng Anh:**

1. Saakshi Kapoor, Dr.Vishal Gupta and Rohit Kumar (2017), A review of attacks and its detection attributes on collaborative recommender systems.
2. DDD