TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

XÂY DỰNG VÀ ỨNG DỤNG KHO DỮ LIỆU(DATA WAREHOUSE) BÁN HÀNG TRÊN MÃ NGUỒN MỞ OFBIZ

Sinh viên thực hiện: **Nguyễn Việt Anh**

Lớp CNTT1 – K55

Giáo viên hướng dẫn: TS. **Phạm Văn Hải**

HÀ NỘI 05-2015

# **PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

1. Thông tin về sinh viên

Họ và tên sinh viên: Nguyễn Việt Anh

Điện thoại liên lạc: 01656589111 Email: vietanh.anlao@gmail.com

Lớp: CNTT1-K55 Hệ đào tạo: Chính quy

Đồ án tốt nghiệp được thực hiện tại: Đại Học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 01/01/2015 đến 29/05/2015.

2. Mục đích nội dung của ĐATN

Xây dựng và ứng dụng kho dữ liệu bán hàng dựa trên mã nguồn mở Ofbiz.

3. Các nhiệm vụ cụ thể của ĐATN

* Nghiên cứu, tìm hiểu về Data warehouse
* Thực hiện tìm hiểu Framework Ofbiz, Data model của Ofbiz
* Nghiên cứu nghiệp vụ bán hàng trong ofbiz
* Dựa vào dữ liệu hoạt động của Ofbiz để xây dựng Data Warehouse
* Từ Data Warehouse sẽ xử lí dữ liệu để đưa ra các báo cáo

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Việt Anh* – cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Phạm Văn Hải*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | Hà Nội, ngày 29 tháng 05 năm 2015  Tác giả ĐATN  *Nguyễn Việt Anh* |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng năm  Giáo viên hướng dẫn |

**TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP**

Mọi doanh nghiệp để có được sự thành công của mình, nhất thiết phải cần vào sự quyết định đúng đắn của các nhà quản trị của doanh nghiệp đó - việc quyết định của nhà quản trị phải dựa trên các số liệu liên quan, và một minh chứng khoa học để đi đến quyết định đó, và cách tốt nhất để có nhanh được các dữ liệu liên quan, và tìm kiếm các minh chứng khoa học này là nhà quản trị phải có một kho dữ liệu được sắp xếp một cách khoa học. Vì vậy trong đồ án này em xin đề xuất việc xây dựng kho dữ liệu về nghiệp vụ bán hàng cho các doanh nghiệp, nhà quản trị trên mã nguồn mở Ofbiz… trong việc kinh doanh sản phẩm. Sau đây là tóm tắt đồ án của em:

CHƯƠNG 1. GIỚI THIỆU: Trình bày nhiệm vụ đồ án, tóm tắt các nội dung được thực hiện trong đồ án.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT: Trình bày cơ sở lý thuyết và phương pháp áp dụng những lý thuyết đó vào việc giải quyết bài toán.

CHƯƠNG 3. MÔ TẢ BÀI TOÁN: Trình bày mô hình bài toán và định hướng giải quyết bài toán.

CHƯƠNG 4. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG: Trình bày chi tiết các kết quả phân tích thiết kế hệ thống, biểu đồ usecase, biểu đồ lớp, biểu đồ hoạt động, biểu đồ trình tự và thiết kế cơ sở dữ liệu.

CHƯƠNG 5. CÀI ĐẶT VÀ THỬ NGHIỆM CHƯƠNG TRÌNH: Trình bày các kết quả đạt được khi thực hiện chương trình cũng như đánh giá ưu nhược điểm của hệ thống.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN: Tổng kết các kết quả đạt được của đồ án và một số hạn chế cần khắc phục.

# **LỜI CẢM ƠN**

Trước tiên em xin gửi lời cảm ơn chân thành và sâu sắc nhất tới thầy Phạm Văn Hải, người đã tận tình giúp đỡ và hướng dẫn em trong thời gian hoàn thành đồ án cũng như trong thời gian học tập.

Đồng thời em xin bày tỏ lòng biết ơn sâu sắc tới các thầy cô giáo trong viện Công nghệ thông tin và truyền thông trường Đại học Bách Khoa Hà Nội, những người đã nhiệt tình giảng dạy và truyền đạt cho em những kiến thức nền tảng để em có thể vận dụng cho việc thực hiện đồ án cũng như trong quá trình công tác sau này.

Tiếp theo, em xin gửi lời cảm ơn chân thành nhất tới các anh chị là thành viên của công ty Olbius đã tận tình chỉ bảo em về các kỹ năng và kinh nghiệm quý báu trong thực tế, giúp em có nền tảng tốt để hoàn thành đồ án.

Cuối cùng, và quan trọng nhất, con xin bày tỏ lòng biết ơn sâu sắc tới ông bà, bố mẹ và em trai đã luôn là chỗ dựa tinh thần vững chắc nhất và tạo mọi điều kiện cho con ăn học nên người.

Do thời gian, kiến thức và kinh nghiệm của em còn hạn chế nên đồ án chắc chắn sẽ có những khiếm khuyết, em mong được các thầy cô, các anh chị và các bạn góp ý thêm để đồ án được hoàn thiện hơn.

Sinh viên

*Nguyễn Việt Anh*

MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](#_Toc420354035)

[LỜI CẢM ƠN 4](#_Toc420354036)

[DANH SÁCH CÁC HÌNH 8](#_Toc420354037)

[DANH SÁCH CÁC BẢNG 10](#_Toc420354038)

[BẢNG VIẾT TẮT 11](#_Toc420354039)

[CHƯƠNG 1: GIỚI THIỆU 12](#_Toc420354040)

[1.1 Tổng quan 12](#_Toc420354041)

[1.2 Phát biểu bài toán 12](#_Toc420354042)

[1.3 Vấn đề đặt ra cho bài toán 12](#_Toc420354043)

[1.4 Mục đích đồ án 12](#_Toc420354044)

[1.5 Nhiệm vụ đồ án 12](#_Toc420354045)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 13](#_Toc420354046)

[2.1 Giới thiệu tổng quan về kho dữ liệu (Data Warehouse) 13](#_Toc420354047)

[2.2 Data Warehouse là gì? 14](#_Toc420354048)

[2.2.1 Khái niệm 14](#_Toc420354049)

[2.2.2 Vì sao cần Data Warehouse 15](#_Toc420354050)

[2.3 Đặc điểm của Data Warehouse 16](#_Toc420354051)

[2.4 Các thành phần chính của Data Warehouse 17](#_Toc420354052)

[2.4.1 Tầng ETL 18](#_Toc420354053)

[2.4.2 Tầng data warehouse 19](#_Toc420354054)

[2.4.3 Tầng khai thác dữ liệu 19](#_Toc420354055)

[2.5 Mô hình dữ liệu thực thể của Data Warehouse 20](#_Toc420354056)

[2.5.1 Mô hình dạng sao 20](#_Toc420354057)

[2.5.2 Mô hình dữ liệu dạng bông tuyết 21](#_Toc420354058)

[2.6 Kĩ thuật phân tích OLAP 23](#_Toc420354059)

[2.6.1 Giới thiệu về OLAP 23](#_Toc420354060)

[2.6.2 Mô hình dữ liệu đa chiều 23](#_Toc420354061)

[2.6.3 Kiến trúc khối (Cube) của OLAP 25](#_Toc420354062)

[2.6.4 So sánh OLAP và OLTP 26](#_Toc420354063)

[2.6.5 Các thành phần của OLAP 26](#_Toc420354064)

[2.6.6 Các mô hình lưu trữ hỗ trợ OLAP 27](#_Toc420354065)

[2.6.7 So sánh các mô hình OLAP 30](#_Toc420354066)

[2.8 Giới thiệu về Ofbiz 32](#_Toc420354067)

[2.8.1 Giới thiệu Framework Ofbiz 32](#_Toc420354068)

[2.8.2 Mô hình cơ sở dữ liệu Ofbiz 35](#_Toc420354069)

[2.8.3 Kiến trúc 3 tầng và cách thức phát triển ứng dụng với Ofbiz 39](#_Toc420354070)

[2.8.4 Tìm hiểu thành phần sẵn có trên OFBIZ 40](#_Toc420354071)

[CHƯƠNG 3: MÔ TẢ BÀI TOÁN 43](#_Toc420354072)

[3.1 Mô hình hoạt động hệ thống 43](#_Toc420354073)

[3.2 Quy trình giải quyết bài toán 44](#_Toc420354074)

[3.3 Chi tiết các quy trình 44](#_Toc420354075)

[CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 50](#_Toc420354076)

[4.1 Phân tích hệ thống 50](#_Toc420354077)

[4.1.1 Biểu đồ Use Case 50](#_Toc420354078)

[4.1.2 Đặc tả Use case 52](#_Toc420354079)

[4.1.3 Biểu đồ trình tự 60](#_Toc420354080)

[4.2 Thiết kế hệ thống 64](#_Toc420354081)

[4.2.1 Thiết kế lớp 64](#_Toc420354082)

[4.2.2 Thiết kế CSDL 67](#_Toc420354083)

[CHƯƠNG 5: CÀI ĐẶT VÀ THỬ NGHIỆM HỆ THỐNG 73](#_Toc420354084)

[5.1 Môi trường cài đặt 73](#_Toc420354085)

[5.2 Cài đặt và thử nghiệm 73](#_Toc420354086)

[5.2.1 Cài đặt ofbiz 73](#_Toc420354087)

[5.2.2 Chức năng ETL trong Data warehouse qua pentaho 75](#_Toc420354088)

[5.2.3 Báo cáo tăng trưởng nhãn hàng 79](#_Toc420354089)

[5.2.4 Báo cáo tăng trưởng nhà phân phối 80](#_Toc420354090)

[5.2.5 Báo cáo chương trình khuyến mại 81](#_Toc420354091)

[5.2.6 Báo cáo doanh số bán hàng nhân viên 82](#_Toc420354092)

[CHƯƠNG 6: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 83](#_Toc420354093)

[6.1 Kết luận 83](#_Toc420354094)

[6.2 Hướng phát triển 84](#_Toc420354095)

[TÀI LIỆU THAM KHẢO 85](#_Toc420354096)

[PHỤ LỤC 86](#_Toc420354097)

**DANH SÁCH CÁC HÌNH**

[Hình 2. 1 Data warehouse 14](#_Toc420328936)

[Hình 2. 2 Vì sao cần Data warehouse? 15](#_Toc420328937)

[Hình 2. 3 Các thành phần trong data warehouse[7] 18](#_Toc420328938)

[Hình 2. 4 Mô hình dạng sao 20](file:///D:\Workspace\project\project\doc\Data%20warehouse.docx#_Toc420328939)

[Hình 2. 5 Mô hình dữ liệu dạng bông tuyết 22](file:///D:\Workspace\project\project\doc\Data%20warehouse.docx#_Toc420328940)

[Hình 2. 6 Mô hình dữ liệu đa chiều[6] 24](#_Toc420328941)

[Hình 2. 7 Mô hình MOLAP[8] 27](#_Toc420328942)

[Hình 2. 8 Mô hình ROLAP[8] 28](#_Toc420328943)

[Hình 2. 9 Mô hình HOLAP[8] 29](#_Toc420328944)

[Hình 2. 10 Các thành phần ERP 32](#_Toc420328945)

[Hình 2. 11 Kiến trúc Ofbiz 33](#_Toc420328946)

[Hình 2. 12 Quản lý người dùng, tổ chức trong ofbiz 36](#_Toc420328947)

[Hình 2. 13 Quản lý sản phẩm trong Ofbiz 37](#_Toc420328948)

[Hình 2. 14 Quản lý các đơn vị chuyển đổi trong ofbiz 38](#_Toc420328949)

[Hình 2. 15 Tổ chức thư mục Ofbiz 41](#_Toc420328950)

[Hình 2. 16 Cấu hình CDSL trong Ofbiz 73](#_Toc420328951)

[Hình 2. 17 Lưu trữ driver jdbc trong Ofbiz 74](#_Toc420328952)

[Hình 2. 18 Tạo component trong Ofbiz 75](#_Toc420328953)

[Hình 3. 1 Kiến trúc hệ thống 43](#_Toc420327403)

[Hình 3. 2 Tổ chức nội bộ Delys 45](#_Toc420327404)

[Hình 3. 3 Dữ liệu thực tế Delys trong Party Relationship 45](#_Toc420327405)

[Hình 3. 4 Quy trình xử lý đơn hàng tại Delys 46](#_Toc420327406)

[Hình 3. 5 Mô hình quan hệ trong Data warehouse [11] 47](#_Toc420327407)

[Hình 3. 6 Câu lện lấy dữ liệu chiều sản phẩm 47](#_Toc420327408)

[Hình 3. 7 Câu lệnh lấy dữu liệu chiều tiền tệ 47](#_Toc420327409)

[Hình 3. 8 Câu lện lấy dữ liệu chiều khuyến mại 48](#_Toc420327410)

[Hình 3. 9 Sinh giá trị ngày tự động trong Pentaho 48](#_Toc420327411)

[Hình 3. 10 Mốc thời gian tối đa khi sinh chiều thời gian 49](#_Toc420327412)

[Hình 3. 11 Câu lệnh lấy dữ liệu bảng sự kiện 49](#_Toc420327413)

[Hình 4. 1 Biểu đồ Use case tổng quan 50](#_Toc420327414)

[Hình 4. 2 Use case phân rã xem thống kê chương trình khuyến mại 51](#_Toc420327415)

[Hình 4. 3 Use case phân rã cấu hình pentaho 51](#_Toc420327416)

[Hình 4. 4 Use case phân rã Xem báo cáo tăng trưởng 52](#_Toc420327417)

[Hình 4. 5 Use cây phân rã Xem doanh số bán hàng 52](#_Toc420327418)

[Hình 4. 6 Biểu đồ hoạt động cho chức năng đăng nhập 53](#_Toc420327419)

[Hình 4. 7 biểu đồ hoạt động cho chức năng Xem thông tin cá nhân 54](#_Toc420327420)

[Hình 4. 8 Biểu đồ hoạt động cấu hình Pentaho sinh dữ liệu dimension 55](#_Toc420327421)

[Hình 4. 9 Biểu đồ hoạt động cấu hình Job Pentaho cập nhật dữ liệu Dimension DW 56](#_Toc420327422)

[Hình 4. 10 Biểu đồ hoạt động cấu hình Pentaho Dimension với Ofbiz Services 57](#_Toc420327423)

[Hình 4. 11 Biểu đồ hoạt động Xem báo cáo tăng trưởng NPP, Nhãn hàng 58](#_Toc420327424)

[Hình 4. 12 Biểu đồ hoạt động Xem thống kê chương trình khuyến mại 59](#_Toc420327425)

[Hình 4. 13 Biểu đồ hoạt động Xem doanh số bán hàng của Sales man 60](#_Toc420327426)

[Hình 4. 14 Biểu đồ trình tự cấu hình Pentaho Dimension với Ofbiz Services 61](#_Toc420327427)

[Hình 4. 15 Biểu đồ tình tự xem báo cáo tăng trưởng nhãn hàng 62](#_Toc420327428)

[Hình 4. 16 Biểu đồ trình tự xem thống kê chương trình khuyến mại 62](#_Toc420327429)

[Hình 4. 17 Biểu đồ trình tự xem thông tin người dùng 63](#_Toc420327430)

[Hình 4. 18 Biểu đồ tuần tự xem thống kê doanh số bán hàng nhân viên SalesMan 63](#_Toc420327431)

[Hình 4. 19 Biểu đồ tuần tự báo cáo tăng trưởng Nhà Phân Phối 64](#_Toc420327432)

[Hình 4. 20 Các lớp dùng chung 64](#_Toc420327433)

[Hình 4. 21 Các lớp cấu trúc tổ chức phòng ban 65](#_Toc420327434)

[Hình 4. 22 Các lớp xử lý Pentaho 66](#_Toc420327435)

[Hình 4. 23 Các bảng cấu hình pentaho 67](#_Toc420327436)

[Hình 4. 24 Cấu trúc các bảng trong DW 68](#_Toc420327437)

[Hình 5. 1 Kịch bản tạo dữ liệu chiểu product 76](#_Toc420327438)

[Hình 5. 2 Kịch bản tạo dữ liệu chiều thời gian 76](#_Toc420327439)

[Hình 5. 3 Kịch bản tạo dữ liệu chiều tiền tệ 77](#_Toc420327440)

[Hình 5. 4 Kịch bản tạo dữ liệu chiều chương trình Khuyến mại 78](#_Toc420327441)

[Hình 5. 5 Kịch bản tọa dữ liệu cho bảng fact Sales 79](#_Toc420327442)

[Hình 5. 6 Giao diện cấu hình Pentaho cho dimension 79](#_Toc420327443)

[Hình 5. 7 Giao diện cấu hình Pentaho cho bảng fact 79](#_Toc420327444)

[Hình 5. 8 Giao diện báo cáo tăng trưởng nhãn hàng 80](#_Toc420327445)

[Hình 5. 9 Báo cáo tăng trưởng nhà phân phối 81](#_Toc420327446)

[Hình 5. 10 Giao diện danh sách chương trình khuyến mại 82](#_Toc420327447)

[Hình 5. 11 Giao diện chi tiết báo cao chương trình khuyến mại 82](#_Toc420327448)

[Hình 5. 12 Giao diện báo cáo bán hàng của nhân viên 83](#_Toc420327449)

[Hình 5. 13 Biểu đồ doanh số bán hàng nhân viên 83](#_Toc420327450)

# **DANH SÁCH CÁC BẢNG**

[Bảng 2. 1 So sánh các mô hình OLAP 30](#_Toc420327451)

[Bảng 4. 1 Đặc tả use case đăng nhập 53](#_Toc420327455)

[Bảng 4. 2 Đặc tả Use case xem thông tin cá nhân 54](#_Toc420327456)

[Bảng 4. 3 Đặc tả Use case cấu hình pentaho chạy tranfrom 55](#_Toc420327457)

[Bảng 4. 4 Đặc tả cấu hình Job Pentaho cập nhật dữ liệu Dimension DW 56](#_Toc420327458)

[Bảng 4. 5 Đặc tả Use case cấu hình Pentaho dimension với Ofbiz 57](#_Toc420327459)

[Bảng 4. 6 Đặc tả use case xem báo cáo tăng trưởng 58](#_Toc420327460)

[Bảng 4. 7 Đặc tả use case xem thống kê chương trình khuyến mại 59](#_Toc420327461)

[Bảng 4. 8 Đặc tả use case xem doanh số bán hàng Sales man 60](#_Toc420327462)

[Bảng 4. 9 Chi tiết các lớp 67](#_Toc420327463)

[Bảng 4. 10 Chi tiết các bảng trong CSDL 72](#_Toc420327464)

# **BẢNG VIẾT TẮT**

|  |  |
| --- | --- |
| OLAP | Online Analysis Processing |
| MOLAP | Multidimensional Online Analysis Processing |
| HOLAP | Hybird Online Analysis Processing |
| BI | Business Intelligence |
| OLAP | OnLine Transaction Processing |
| ROLAP | Relational Online Analysis Processing |
| DW | Data Warehouse |
| NPP | Nhà phân phối |
| OLTP | Online transaction processing |
| ERP | Enterprise Resource Planning |

# **CHƯƠNG 1: GIỚI THIỆU**

* 1. **Tổng quan**

Ngày nay, với sự phát triển của internet và ngành công nghệ thông tin, mỗi người chúng ta đều có cơ hội tiếp cận với một nguồn thông tin khổng lồ đến từ nhiều nơi, lĩnh vức khác nhau. Vì vậy cần phải có giải pháp để tổng hợp những thông tin cần thiết đối với bản thân.

Đặc biệt đối với mỗi nhà quản trị doanh nghiệp,việc quản lý và sử dụng nguồn thông tin của doanh nghiệp mình một cách hiệu quả sẽ góp phần không nhỏ tạo lên thành công của doanh nghiệp. Để làm được điều đó thì việc xây dựng một kho dữ liệu được sắp xếp khoa học sẽ góp phần không nhỏ trong việc hỗ trợ nhà quản trị đưa ra các quyết định đúng đắn dựa trên các số liệu và minh chứng khoa học liên quan.

Vì vậy trong đồ án này em xin đề xuất việc xây dựng và ứng dụng Data Warehouse-kho dữ liệu cho các doanh nghiệp, nhà quản trị… trong lĩnh vực kinh doanh sản phẩm trên mã nguồn mở Ofbiz

* 1. **Phát biểu bài toán**

Bài toán được phát biểu như sau: Doanh nghiệp hiện tại đang thực hiện triển khai ERP trên công nghệ mã nguồn mở OFBIZ, cần xây dựng và ứng dụng một kho dữ liệu phục vụ báo cáo, thống kê liên quan đến nghiệp vụ phân phối,bán hàng.

* 1. **Vấn đề đặt ra cho bài toán**

Những vấn đề đặt ra cho bài toán được phân tích như sau**:**

* Hiện tại doanh nghiệp đang triển khai quản lý trên FrameWork Ofbiz vì vậy cần thực hiện tìm hiểu luồng dữ liệu bán hàng để xây dựng cấu trúc của kho dữ liệu.
* Xây dựng cấu trúc cho kho dữ liệu từ OFbiz
* Quản lý, lập lịch thực hiện việc cập nhật dữ liệu cho kho dữ liệu từ dữ liệu hoạt động của doanh nghiệp, đảm bảo không ảnh hưởng đến quá trình hoạt động của doanh nghiệp.
* Tạo các báo cáo, biểu đồ trên Ofbiz lấy dữ liệu từ kho dữ liệu phù hợp với từng vị trí, cấp bậc người sử dụng
* Xây dựng giao diện thân thiện và dễ sử dụng với người dùng.
  1. **Mục đích đồ án**

Hỗ trợ người sử dụng đưa ra các quyết định trong kinh doanh dựa trên các số liệu thống kê, phân tích, biểu đồ..

* 1. **Nhiệm vụ đồ án**

Nhiệm vụ của đồ án: Xây dựng và ứng dụng kho dữ liệu dựa trên mã nguồn mở Ofbiz phục vụ nhu cầu báo cáo, thống kê cho doanh nghiệp.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

* 1. **Giới thiệu tổng quan về kho dữ liệu (Data Warehouse)**

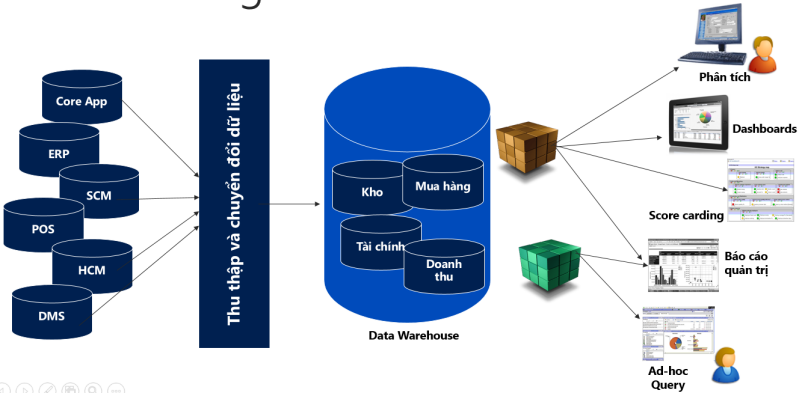
Trên thế giới có rất nhiều tổ chức hay công ty có những kho dữ liệu khổng lồ. Trong quá trình hoạt động, do yêu cầu của công việc nên những tổ chức hay công ty này phải lưu lại dữ liệu của khách hàng, sản phẩm, hóa đơn, ... Theo nghiên cứu thì cứ khoảng sau 5 năm khối lượng dữ liệu của doanh nghiệp lại tăng lên gấp đôi [6]. Tuy nhiên việc làm thế nào để có thể khai thác được những kho dữ liệu khổng lồ này lại là một vấn đề nan giải bởi dữ liệu thường không nhất quán và không được thiết kế với mục đích hỗ trợ quyết định. Vì vậy cái mà các tổ chức và công ty này cần là một công cụ cho phép họ tổng hợp dữ liệu một cách hiệu quả.

Sự ra đời của kho dữ liệu bắt nguồn từ sự kết hợp của hai nhân tố kinh tế và kĩ thuật. Các công ty, đặc biệt là các công ty lớn có nhiều chi nhánh khác nhau, rất muốn có được những thông tin tổng quan trên phạm vi toàn công ty, từ đó có thể giúp cho lãnh đạo của công ty ra quyết định. Tuy nhiên, các hệ thống dữ liệu tác nghiệp trước đây không thể hoàn thành được công việc này vì chúng được thiết kế với mục đích là phục vụ công việc nghiệp vụ hàng ngày, vì vậy cần phải có một công nghệ dữ liệu khác. Chính sự phát triển của công nghệ được sự thúc đẩy mạnh mẽ của nhu cầu của các doanh nghiệp trong môi trường kinh doanh đầy cạnh tranh đã tạo nên data warehouse.[6]

Data warehouse ra đời đã mang lại một số lợi ích nhất định cho các nhà kinh doanh :

* Cho phép các nhà kinh doanh theo dõi được sự biến động của môi trường kinh doanh : ở một khía cạnh nào đó, data warehouse có thể xem như là những "ảnh chụp" liên tiếp tình trạng kinh doanh của công ty, vì vậy nó cho phép người dùng cuối thấy được môi trường kinh doanh của công ty qua các thời kì.
* Câu trả lời cho các câu hỏi truy vấn dữ liệu có tính thương mại. Các nhà kinh doanh muốn rằng những câu hỏi họ đặt ra trong lĩnh vực kinh doanh sẽ được trả lời theo dạng mà họ có thể hiểu được.
* Cung cấp công cụ truy xuất dữ liệu nhằm mục đích phân tích và hỗ trợ quyết định. Trong những hệ thống cơ sở dữ liệu tác nghiệp trước đây thì điều này rất khó thực hiện ngay cả khi dữ liệu là đầy đủ và sẵn sàng, lÝ do đơn giản là vì hệ thống cơ sở dữ liệu tác nghiệp được thiết kế ra với mục đích phục vụ công việc tác nghiệp thường ngày chứ không phải được thiết kế ra với mục đích hỗ trợ quyết định.
* Thống nhất được dạng và cấu trúc dữ liệu phù hợp với yêu cầu của người dùng cuối. Đối với các công ty có nhiều chi nhánh khác nhau thì có thể dữ liệu được lưu trữ dưới các dạng khác nhau, ví dụ như cùng là đơn vị đo chiều dài nhưng có nơi dùng đơn vị là cm, có nơi dùng đơn vị là inch, ... do vậy gây khó khăn cho việc phân tích dữ liệu để hỗ trợ quyết định.
* Quản trị dễ dàng. Sau khi được xây dựng, người quản trị có thể lên kế hoạch để hệ thống vận hành tự động. Người quản trị chỉ có nhiệm vụ theo dõi, chỉnh sửa và nâng cấp khi có yêu cầu hoặc sự cố.
  1. **Data Warehouse là gì?**
     1. **Khái niệm**

Theo [10], Kho dữ liệu là kho lưu trữ dữ liệu, lưu trữ bằng thiết bị điện tử của một tổ chức. Các kho dữ liệu được thiết kế để hỗ trợ việc phân tích dữ liệu và lập báo cáo.



Hình 2. 1 Data warehouse

Data warehouse là một cơ sở dữ liệu quan hệ hay một cơ sở dữ liệu đa chiều được thiết kế cho mục đích truy vấn và phân tích hơn là xử lý các giao dịch. DWH tách riêng việc phân tích với việc xử lý giao dịch, và cho phép hợp nhất dữ liệu từ nhiều nguồn.

Data warehouse được sử dụng như là một nền tảng cho hệ thống hỗ trợ ra quyết định. Nó gắn liền với khái niệm Business Inteligence (Kinh doanh thông minh). Nó được dùng để giải quyết các vấn đề gặp phải khi một tổ chức cố gắng phân tích chiến lược từ số liệu trong một hệ thống database được dùng chung với hệ thống xử lý dữ liệu trực tuyến (Online Transaction Processing - OLTP).

* + 1. **Vì sao cần Data Warehouse**

Một hệ thống OLTP điển hình được đặc trưng bởi một hệ thống có nhiều người sử dụng đồng thời để thêm, sửa, xóa dữ liệu. Ví dụ như một hệ thống bán lẻ, có nhiều nhân viên bán hàng đồng thời cùng nhập và chỉnh sửa dữ liệu vào hệ thống. Trong một hệ thống nhỏ, việc sử dụng cùng một hệ thống database này cho mục đích lên báo cáo thống kê, phân tích có thể chấp nhận được do khối lượng dữ liệu ít, ít người dùng. Nhưng trong một hệ thống với khối lượng dữ liệu hàng chục triệu đến trăm triệu bản ghi trên một bảng thì việc khai thác dữ liệu chung trên cùng một hệ thống database OLTP là không thể chấp nhận được vì thời gian xử lý dữ liệu rất lâu, làm cho tài nguyên hệ thống trở nên quá tải và có thể làm cho toàn bộ hệ thống bị tê liệt.



Hình 2. 2 Vì sao cần Data warehouse?

Hệ thống OLTP được thiết kế cho mục đích thu thập dữ liệu, khối lượng dữ liệu càng càng càng lớn khiến cho nhu cầu phân tích càng trở nên cấp thiết hơn. Tuy nhiên database trong OLTP được thiết kế cho mục đích thu thập số liệu thường phải tối ưu hóa cả về không gian lưu trữ lẫn chuẩn hóa trong thiết kế tuân thủ nghiêm ngặt theo mô hình dữ liệu quan hệ. Còn khi nhu cầu phân tích, khai thác dữ liệu thông qua các báo cáo sẽ phát sinh hàng loạt các vấn đề:

- Người xây dựng báo cáo không hiểu mối quan hệ phức tạp giữa các bảng trong một loạt database của các ứng dụng khác nhau.

- Database của các ứng dụng có thể để rải rác trên nhiều server, rất khó cho việc tìm kiếm các bảng để xây dựng câu truy vấn.

- Việc phân quyền không cho phép người dùng có thể lấy dữ liệu chi tiết.

- DBA hạn chế việc chạy các câu query lớn nhằm ngăn chặn nguy cơ làm cho hệ thống bị tê liệt.

Bằng cách copy dữ liệu từ một hệ thống OLTP sang một máy chủ theo một lịch trình cho mục đích báo cáo, tổ chức có thể cải thiện thời gian phản hồi cho báo cáo và các câu truy vấn. Một database thiết kế trong hệ thống OLTP là không đủ linh hoạt cho một ứng dụng hỗ trợ ra quyết định, lý do chủ yếu do khối lượng dữ liệu quá lớn và sự phức tạp của các bảng trong các database có mô hình quan hệ.

* 1. **Đặc điểm của Data Warehouse**

***a) Hướng chủ đề :***

Data warehouse được thiết kế để giúp bạn phân tích dữ liệu nhằm trả lời các câu hỏi của người dùng cuối. Mặt khác, những câu hỏi của người dùng cuối lại có thể phân loại được theo một số chủ đề nào đó. Ví dụ người dùng cuối có thể đặt các câu hỏi như : "Khách hàng có độ tuổi nào mua nhiều bảo hiểm nhất trong năm nay ?", "Công ty con nào bán được nhiều bảo hiểm nhất trong tháng này ?" hay "Mặt hàng bảo hiểm nào được ưa chuộng nhất trong quý vừa rồi ?" để tìm hiểu rõ tình trạng bán bảo hiểm của công ty. Để trả lời những câu hỏi này, một data warehouse tập trung vào chủ đề hợp đồng bảo hiểm sẽ được xây dựng. Tương tự như vậy, data warehouse hướng vào các chủ đề khác cũng sẽ được hình thành.

***b) Tích hợp :***

Data warehouse được xây dựng từ nhiều nguồn dữ liệu khác nhau, các nguồn dữ liệu này có sự xung đột với nhau về tên trường, giá trị thể hiện, đơn vị đo lường, ... Ví dụ như cùng biểu diễn trường địa chỉ nhưng có cơ sở dữ liệu sử dụng tên trường là "Địa chỉ" trong khi cơ sở dữ liệu khác lại dùng là "Nơi ở". Ngoài ra các cơ sở dữ liệu cũng có thể có giá trị thể hiện khác nhau, ví dụ để biểu diễn giới tính của khách hàng, công ty con này dùng giá trị là 0 và 1, trong khi công ty con khác lại dùng giá trị là "M" và "W". Ngoài ra cũng còn phải kể đến sự sử dụng đơn vị đo lường khác nhau cũng dẫn tới việc khó thống nhất các cơ sở dữ liệu lại với nhau. Tóm lại, khi giải quyết được tất cả những sự xung đột dữ liệu này để dữ liệu có thể chuyển vào data warehouse theo một định dạng duy nhất, ta nói rằng chúng đã được tích hợp.

***c) Không thay đổi :***

Đặc tính này có ý nghĩa là khi dữ liệu đã được cập nhật vào data warehouse rồi thì dữ liệu không nên thay đổi nữa. Điều này cũng logic vì mục đích của data warehouse là cho phép bạn phân tích những gì đã xảy ra.

***d) Thời biến :***

Để phát hiện ra xu hướng kinh doanh, các nhà phân tích cần một khối lượng dữ liệu cực kì lớn, điều này trái ngược hẳn với hệ thống xử lý giao dịch trực tuyến, nơi mà tốc độ xử lý được đặt lên quan trọng hàng đầu còn thông tin lịch sử thì không được coi trọng. Thời biến ở đây có ý nghĩa là data warehouse phản ánh được sự thay đổi kinh doanh theo thời gian.

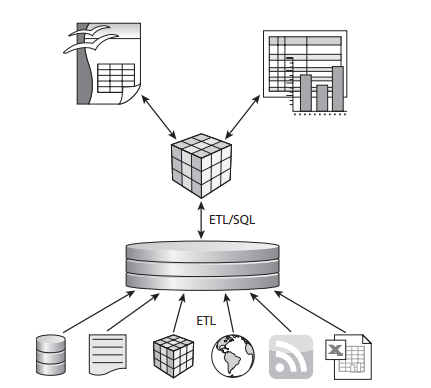
* 1. **Các thành phần chính của Data Warehouse**

Một hệ thống data warehouse bao gồm 3 thành phần [7] chính sau:

- Một bộ công cụ để thu thập dữ liệu từ hệ thống nghiệp vụ, chuẩn hoá chúng về định dạng dữ liệu đa chiều, nạp vào data warehouse (Extract-Transformation Loading – ETL).

- Một database dùng làm data warehouse để lưu trữ dữ liệu

- Một loạt các công cụ khai thác dữ liệu từ data warehouse như hệ thống OLAP, hệ thống báo cáo tĩnh, hệ thống data mining…



Hình 2. 3 Các thành phần trong data warehouse[7]

* + 1. **Tầng ETL**

Tầng ETL (Extract – Transform – Load) là tầng thấp nhất, ẩn đi với người dùng cuối, bao gồm 3 bước[8]:

- Bước thu thập (extract) gom góp dữ liệu từ nhiều khác nhau về. Các nguồn này có thể là database hệ thống nghiệp vụ (MS SQL, mySQL, Oracle, DB2…), cũng có thể là file ở các định dạng khác nhau (CSV, fix-length, excel, XML…), có thể là dữ liệu nội bộ doanh nghiệp hoặc từ bên ngoài. Một hệ thống ETL tốt phải đảm bảo tương thích với các nguồn dữ liệu thông dụng này.

- Bước chuẩn hoá (transform) biến đổi dữ liệu từ định dạng nguồn sang định dạng của data warehouse (định dạng dữ liệu đa chiều đã nói ở bước trên), bao gồm các bước nhỏ:

* Bước dọn dẹp (cleaning) xoá các bản ghi bị sai, lỗi và chuyển hoá dữ liệu về định dạng chuẩn chung.
* Bước tập hợp (integration) cắt gọt dữ liệu có chung ý nghĩa từ nhiều nguồn khác nhau về một khung duy nhất.
* Bước tổng hợp (aggregation) tổng hợp dữ liệu dựa vào độ chi tiết của data warehouse.

- Bước nạp dữ liệu (load) ghi dữ liệu đã được chuẩn hoá vào data warehouse. Bước này bao gồm cả quá trình cập nhật thay đổi từ hệ thống nghiệp vụ vào data warehouse, đảm bảo số liệu báo cáo luôn được cập nhật. Tuỳ thuộc vào chính sách

* + 1. **Tầng data warehouse**

Tầng data warehouse đứng ở trung tâm một hệ thống data warehouse làm nhiệm vụ lưu trữ dữ liệu bao quanh tất cả các hoạt động nghiệp vụ, các phòng ban của doanh nghiệp. Data warehouse thường bao gồm một hoặc nhiều data mart, với data mart chính là data warehouse thu nhỏ tập trung vào một nghiệp vụ nhất định nào đó của doanh nghiệp (ví dụ data mart về sale, data mart về kho bãi, data mart về nhân sự…)

Ngoài nhiệm vụ lưu trữ dữ liệu, tầng data warehouse còn có một thành phần khác rất quan trọng gọi là siêu dữ liệu (metadata). Siêu dữ liệu lại được chia làm 2 nhóm là nhóm siêu dữ liệu kỹ thu ật và siêu dữ liệu nghiệp vụ. Siêu dữ liệu nghiệp vụ (business metadata) mô tả ý nghĩa dữ liệu, các luật và ràng buộc tác động lên dữ liệu. Siêu dữ liệu kỹ thuật (technical metadata) mô tả cách thức tổ chức, lưu trữ và điều khiển dữ liệu trong hệ thố ng máy tính. Trong phạm vi data warehouse, siêu dữ liệu kỹ thuật được sử dụng để mô tả thông tin về data warehouse, về dữ liệu nguồn và các tiến trình ETL. Cụ thể:

- Siêu dữ liệu mô tả cấu trúc data warehouse và các data mart ở mức logic (mô tả bảng fact, bảng dimension, cây phân cấp, nguồn gốc dữ liệu) và mức vật lý (cấu trúc bảng, index, partition). Ngoài ra nó còn chứa thông tin bảo mật dữ liệu (xác thực, phân quyền người dùng) và các thông tin giám sát (thống kê hiệu năng sửdụng, báo cáo lỗi…)

- Siêu dữ liệu mô tả dữ liệu nguồn, cũng ở mức logic (cách thức và tham số kết nối lấy dữ liệu, tần suất cập nhật dữ liệu, ý nghĩa dữ liệu) và vật lý (cấu trúc dữ liệu)

- Siêu dữ liệu mô tả các tiến trình ETL, bao gồm cả gốc gác dữ liệu (truy được dữ liệu trên data warehouse về đến gốc gác của nó trong hệ thống nghiệp vụ), các luật thu thập, làm sạch, chuyển hoá dữ liệu.

* + 1. **Tầng khai thác dữ liệu**

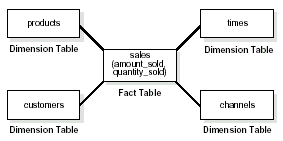
Tầng khai thác dữ liệu chứa các công cụ cho người dùng cuối khai thác, sử dụng các dữ liệu trong data warehouse. Một số công cụ chính:

* Báo cáo OLAP (OLAP tool) là báo cáo động cho phép người dùng sử dụng các tính năng c ủa OLAP (đã nói ở phần 1.3.3) để tạo báo cáo. Các truy vấn đột xuất này được gọi là truy vấn tuỳ biến (ad hoc query) vì hệ thống không hề được chuẩn bị trước cho thao tác của người dùng. Báo cáo OLAP được sử dụng khi người dùng muốn các thông tin cắt lớp, chuyên sâu hoặc toàn cảnh trước khi ra quyết đ ịnh
* Báo cáo tĩnh (reporting tool) là các báo cáo có cấu trúc, format, sử dụng truy vấn được định nghĩa trước đó, đôi khi bao gồm cả biểu đồ. Báo cáo tĩnh được sử dụng khi người dùng muốn xem các thông tin đánh giá, điều hành.
* Bộ công cụ khai phá dữ liệu (data mining) cho phép người dùng phân tích dữ

li ệu để tìm ra các thông tin quý giá còn bị ẩn dấu, ví dụ như các xu hướng, các mẫu chung

* 1. **Mô hình dữ liệu thực thể của Data Warehouse**

Cơ sở dữ liệu quan hệ đã tồn tại và phát triển trong một thời gian dài, nó đã chứng tỏ được thế mạnh của mình và thống lĩnh được thị trường, vì vậy hầu hết các hệ quản trị cơ sở dữ liệu đều sử dụng hoặc có nhân là công nghệ dữ liệu quan hệ. Tuy nhiên, để xây dựng được data warehouse thì dữ liệu cần phải là đa chiều, vì vậy ta có một số mô hình dữ liệu sau của data warehouse :

* + 1.  **Mô hình dạng sao**

Hình 2. 4 Mô hình dạng sao

Mô hình dạng sao, như cái tên của nó, có dạng hình sao. Trung tâm là một bảng sự kiện (fact table), xung quanh là các bảng chiều (dimension table) trỏ vào nó. Thông thường mỗi vùng chủ đề trong data warehouse thường xoay quanh một bảng sự kiện trung tâm. Các bảng vệ tinh xung quanh là các nhánh của bảng sự kiện có nhiệm vụ mô tả chi tiết hơn thành phần sự kiện trong bảng. Bảng sự kiện được phát hiện bằng cách phân tích công việc kinh doanh của công ty. Với mỗi công việc hoặc một tiến trình chính của công ty thường sẽ có một bảng sự kiện mô tả chi tiết tiến trình đó.

Trong ví dụ mô hình dạng sao đơn giản ở trên thì bảng sự kiện cho biết số lượng hàng bán ra và số tiền thu được ứng với các giá trị chiều là thời gian, loại sản phẩm, khách mua hàng và kênh phân phối hàng. Với mô hình như thế này thì ta có thể tính toán rất nhanh các thông số tổng hợp về số lượng bán hàng và tiền thu được, bên cạnh đó nó cũng cho ta biết được tình hình kinh doanh của công ty trong một thời gian dài. Bảng sự kiện tham chiếu đến các bảng khác thông qua các liên kết sử dụng khóa chính và khóa ngoại mà chúng ta đã rất quen thuộc.

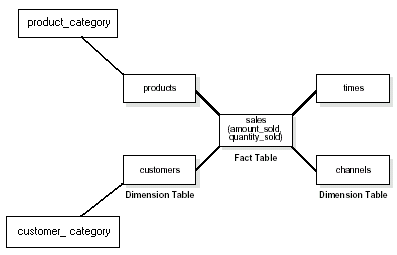
Như vậy, với mỗi thay đổi về giá trị tại các bảng chiều sẽ cho ta một giá trị mới trong bảng sự kiện. Chẳng hạn, nếu khách hàng mua hai sản phẩm trong cùng một lần thì sẽ có hai dòng trong bảng sự kiện mô tả giao dịch này. Mỗi dòng sẽ chứa lượng hàng bán và lượng tiền thu về, đồng thời tham chiếu đến các dòng chứa loại sản phẩm tương ứng trong bảng chiều sản phẩm. Nói chung là bảng sự kiện thường có dung lượng rất lớn và tăng kích cỡ liên tục, vì vậy khi thiết kế cần chú ý tới điều này.

Vậy thiết kế dữ liệu theo mô hình dạng sao rất khác với mô hình dữ liệu truyền thống vì rõ ràng mô hình dữ liệu truyền thống không cần bộ nhớ lưu trữ dữ liệu nhiều đến thế. Khác biệt này là do mục tiêu của data warehouse là cho phép người dùng cuối tìm kiếm, khai thác và xử lý thông tin một cách nhanh nhất. Người dùng cuối sẽ không quan tâm đến vấn đề giao dịch đó được diễn ra như thế nào, cái mà họ muốn biết là kết quả tổng hợp của chúng như : lượng hàng bán ra được theo vùng, doanh thu theo kì hay phân bố của nhóm khách hàng cùng mua một loại sản phẩm nào đó ... Để trả lời những câu hỏi kiểu như vậy cần một khối lượng dữ liệu rất lớn và cần thành lập những câu hỏi truy vấn phức tạp nếu ta vẫn cứ sử dụng mô hình dữ liệu kiểu truyền thống. Lúc đó vấn đề tốc độ sẽ trở nên quan trọng. Hệ quản trị cơ sở dữ liệu sẽ không thể đáp ứng được cùng một lúc nhiều câu hỏi truy vấn kiểu như vậy tại một thời điểm. Vì vậy cần tập hợp dữ liệu lại theo cấu trúc mà người dùng cần chứ không phải theo kiểu mô tả chính xác nhất các hoạt động của hệ thống.

* + 1. **Mô hình dữ liệu dạng bông tuyết**

Thực chất mô hình này chỉ là một biến thể của mô hình dạng sao. Thay vì các cánh sao chỉ gồm một bảng chiều thì các bảng chiều này lại được phân cấp thêm.

Ví dụ như các bảng chiều sản phẩm và khách hàng muốn phân cấp thêm thành loại sản phẩm và loại khách hàng.



Hình 2. 5 Mô hình dữ liệu dạng bông tuyết

Điểm khác biệt cơ bản giữa mô hình dạng bông tuyết và dạng sao là ở chỗ các chiều trong mô hình dạng bông tuyết được tách thành các cấp khác nhau. Nhờ vậy khối lượng cần lưu giữ sẽ được giảm đi. Tuy nhiên đó không phải là lợi ích chính của mô hình dạng bông tuyết. Lợi ích chính của mô hình dạng này là ở chỗ nó tạo ra một cái nhìn đầy đủ về phân cấp trong các chiều. Với mô hình này mỗi chiều được phân thành nhiều cấp và vì vậy việc khai thác dữ liệu ở mức chi tiết cũng như tổng hợp dữ liệu dễ dàng hơn. Người dùng không phải cố gắng hình dung các cấp trong mỗi chiều mà tự bản thân nó đã được mô hình hóa theo kiểu này.

Trở ngại lớn nhất khi triển khai mô hình dạng bông tuyết là vấn đề tốc độ. Đáng tiếc là nếu càng có nhiều quan hệ thì tốc độ truy vấn dữ liệu lại càng chậm, mà tốc độ lại là vấn đề sống còn khi thiết kế data warehouse. Do vậy, thông thường người ta tìm cách tránh sử dụng mô hình dạng bông tuyết. Tuy nhiên, đối với những người vốn đã quen với mô hình dữ liệu quan hệ và việc chuẩn hóa thì mô hình dạng bông tuyết gần gũi hơn và trên thực tế nhiều người thích sử dụng chúng hơn.

Cần phải nói lại rằng tiêu chí cho việc lùa chọn mô hình là tốc độ và dung lượng bộ nhớ lưu trữ. Ta có thể biến một mô hình dạng bông tuyết về dạng sao và ngược lại mà không làm ảnh hưởng gì tới ngữ nghĩa của dữ liệu. Nếu một khi đặt vấn đề tốc độ lên hàng đầu thì ta nên chọn mô hình dạng sao.

* 1. **Kĩ thuật phân tích OLAP**
     1. **Giới thiệu về OLAP**

OLAP là một kỹ thuật sử dụng các thể hiện dữ liệu đa chiều gọi là các khối (cube) nhằm cung cấp khả năng truy xuất nhanh đến dữ liệu của kho dữ liệu[6]. Tạo khối (cube) cho dữ liệu trong các bảng chiều (dimension table) và bảng sự kiện (fact table) trong kho dữ liệu và cung cấp khả năng thực hiện các truy vấn tinh vi và phân tích cho các ứng dụng client. Trong khi kho dữ liệu lưu trữ dữ liệu cho phân tích, thì OLAP là kỹ thuật cho phép các ứng dụng client truy xuất hiệu quả dữ liệu này. OLAP cung cấp nhiều lợi ích cho người phân tích, cho ví dụ như:

- Cung cấp mô hình dữ liệu đa chiều trực quan cho phép dễ dàng lựa chọn, định hướng và khám phá dữ liệu.

- Cung cấp một ngôn ngữ truy vấn phân tích, cung cấp sức mạnh để khám phá các mối quan hệtrong dữ liệu kinh doanh phức tạp.

- Dữ liệu được tính toán trước đối với các truy vấn thường xuyên nhằm làm cho thời gian trả lời rất nhanh đối với các truy vấn đặc biệt.

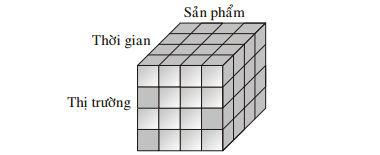
- Cung cấp các công cụmạnh giúp người dùng tạo các khung nhìn mới của dữ liệu dựa trên một tập các hàm tính toán đặc biệt.

OLAP được đặt ra để xử lý các truy vấn liên quan đến lượng dữ liệu rất lớn mà nếu cho thực thi các truy vấn này trong hệ thống OLTP sẽ không thể cho kết quả hoặc sẽ mất rất nhiều thời gian.

* + 1. **Mô hình dữ liệu đa chiều**

Các nhà quản lý kinh doanh có khuynh hướng suy nghĩ theo “nhiều chiều” (multidimensionally). Ví dụ như họ có khuynh hướng mô tả những gì mà công ty làm như sau: “Chúng tôi kinh doanh các sản phẩm trong nhiều thị trường khác nhau, và chúng tôi đánh giá hiệu quả thực hiện của chúng tôi qua thời gian”. Những người thiết kế kho dữ liệu thường lắng nghe cẩn thận những từ đó và họ thêm vào những nhấn mạnh đặc biệt của họ như: “Chúng tôi kinh doanh các sản phẩm trong nhiều thị trườngkhác nhau, và chúng tôi đánh giá hiệu quả thực hiện của chúng tôi qua thời gian”.

Suy nghĩ một cách trực giác, việc kinh doanh như một khối(cube) dữ liệu, với các nhãn trên mỗi cạnh của khối (xem hình bên dưới). Các điểm bên trong khối là các giao điểm của các cạnh. Với mô tả kinh doanh ở trên, các cạnh của khối là Sản phẩm, Thị trường, và Thời gian. Hầu hết mọi người đều có thể nhanh chóng hiểu và tưởng tượng rằng các điểm bên trong khối là các độ đo hiệu quả kinh doanh mà được kết hợp giữa các giá trị Sản phẩm, Thị trường và Thời gian:



Hình 2. 6 Mô hình dữ liệu đa chiều[6]

Mô phỏng các chiều trong kinh doanh Một khối dữ liệu (datacube) thì không nhất thiết phải có cấu trúc 3 chiều (3-D), nhưng về cơ bản là có thể có N chiều (N-D). Những cạnh của khối được gọi là các chiều (dimensions), mà đó là các mặt hoặc các thực thể ứng với những khía cạnh mà tổ chức muốn ghi nhận. Mỗi chiều có thể kết hợp với một bảng chiều (dimension table) nhằm mô tả cho chiều đó. Ví dụ, một bảng chiều của Sản phẩm có thể chứa những thuộc tính như Ma\_sanpham, Mo\_ta, Ten\_sanpham, Loai\_SP,… mà có thể được chỉ ra bởi nhà quản trị hoặc các nhà phân tích dữ liệu. Với những chiều không được phân loại, như là Thời gian, hệ thống kho dữ liệu sẽ có thể tự động phát sinh tương ứng với bảng chiều (dimension table) dựa trên loại dữ liệu. Cần nói thêm rằng, chiều Thời gian trên thực tế có ý nghĩa đặc biệt đối với việc hỗ trợ quyết định cho các khuynh hướng phân tích. Thường thì nó được mong muốn có một vài tri thức gắn liền với lịch và những mặt khác của chiều thời gian. Hơn nữa, một khối dữ liệu trong kho dữ liệu phần lớn được xây dựng để đo hiệu quả của công ty. Do đó một mô hình dữ liệu đa chiều đặc thù được tổ chức xung quanh một chủ đề mà được thể hiện bởi một bảng sự kiện (fact table) của nhiều độ đo số học (là các đối tượng của phân tích). Ví dụ, một bảng sự kiện có thể chứa số mặt hàng bán, thu nhập, tồn kho, ngân sách,… Mỗi độ đo số học phụ thuộc vào một tập các chiều cung cấp ngữ cảnh cho độ đo đó.

Vì thế, các chiều kết hợp với nhau được xem như xác định duy nhất độ đo, là một giá trị trong không gian đa chiều. Ví dụ như một kết hợp của Sản phẩm, Thời gian, Thị trường vào 1 thời điểm là một độ đo duy nhất so với các kết hợp khác. Các chiều được phân cấp theo loại. Ví dụ như chiều Thời gian có thể được mô tả bởi các thuộc tính như Năm, Quý, Tháng và Ngày. Mặt khác, các thuộc tính của một chiều có thể được tổ chức vào một lưới mà chỉ ra một phần trật tự của chiều. Vì thế, cũng với chiều Thời gian có thể được tổ chức thành Năm, Quý, Tháng, Tuần và Ngày. Với sự sắp xếp này, chiều Thời gian không còn phân cấp vì có những tuần trong năm có thể thuộc về nhiều tháng khác nhau.

Vì vậy, nếu mỗi chiều chứa nhiều mức trừu tượng, dữ liệu có thể được xem từ nhiều khung nhìn linh động khác nhau. Một số thao tác điển hình của khối dữ liệu như roll-up (tăng mức độ trừu tượng), drill-down (giảm mức độ trừu tượng hoặc tăng mức chi tiết), slice and dice (chọn và chiếu), và pivot (định hướng lại khung nhìn đa chiều của dữ liệu), cho phép tương tác truy vấn và phân tích dữ liệu rất tiện lợi. Những thao tác đó được biết như Xử lý phân tích trực tuyến (OnLine Analytical Processing – OLAP). Những nhà ra quyết định thường có những câu hỏi có dạng như “tính toán và xếp hạng tổng số lượng hàng hoá bán được theo mỗi quốc gia (hoặc theo mỗi năm)”. Họ cũng muốn so sánh hai độ đo số học như số lượng hàng bán và ngân sách được tổng hợp bởi cùng các chiều. Như vậy, một đặc tính để phân biệt của mô hình dữ liệu đa chiều là nó nhấn mạnh sự tổng hợp của các độ đo bởi một hoặc nhiều chiều, mà đó là một trong những thao tác chính yếu để tăng tốc độ xử lý truy vấn.

* + 1. **Kiến trúc khối (Cube) của OLAP**

Đối tượng chính của OLAP là khối (cube), một thể hiện đa chiều của dữ liệu chi tiết và tổng hợp. Một khối bao gồm một nguồn dữ liệu (Data source), các chiều (Dimensions), các độ đo (Measures) và các phần dành riêng (Partitions). Các khối được thiết kế dựa trên yêu cầu phân tích của người dùng. Một kho dữ liệu có thể hỗ trợ nhiều khối khác nhau như khối Bán hàng, khối Bảng kiểm kê…

Dữ liệu nguồn của một khối chỉ ra nơi chứa kho dữ liệu cung cấp dữ liệu cho khối. Các chiều (dimension) được ánh xạ từ các thông tin của các bảng chiều (dimension table) trong kho dữ liệu vào các mức phân cấp, ví dụ như chiều Địa lý thì gồm các mức như Lục địa, Quốc gia, Tỉnh-Thành phố. Các chiều có thể được tạo một cách độc lập và có thể chia sẻ giữa các khối nhằm xây dựng các khối dễ dàng và để chắc chắn rằng thông tin tổng hợp cho phân tích luôn ổn định. Ví dụ, nếu một chiều chia sẻ một phân cấp sản phẩm và được sử dụng trong tất cả các khối thì cấu tạo của thông tin tổng hợp về sản phẩm sẽ ổn định giữa các khối sử dụng chiều đó.

Một chiều ảo (virtual dimension) là một dạng đặc biệt của chiều mà ánh xạ các thuộc tính từ các thành viên (member) của một chiều khác để sau đó có thể được sử dụng trong các khối. Ví dụ, một chiều ảo của thuộc tính kích thước sản phẩm cho phép một khối (cube) tổng hợp dữ liệu như số lượng sản phẩm bán được theo kích thước, hoặc như số lượng áo bán được theo kiểu và theo kích thước. Các chiều ảo (virtual dimension) và các thuộc tính thành viên được đánh giá là cần thiết cho các truy vấn và chúng không đòi hỏi phải có các khối lưu trữ vật lý.

Các độ đo (measure) xác định các giá trị số từ bảng sự kiện (fact table) mà được tổng hợp cho phân tích như giá bán, chi phí hoặc số lượng bán.

Các phần dành riêng (partition) là các vật chứa lưu trữ đa chiều, giữ dữ liệu của khối. Mỗi khối chứa ít nhất một partition, và dữ liệu của khối có thể kết hợp từ nhiều partition. Mỗi partition có thể lấy dữ liệu một nguồn dữ liệu khác nhau và có thể lưu trong một vị trí riêng biệt (separate).Dữ liệu của một partition có thể được cập nhật độc lập với các partition khác trong một khối. Ví dụ, dữ liệu của một khối có thể được chia theo thời gian, với một partition chứa dữ liệu của năm hiện hành, một partition khác chứa dữ liệu của năm trước, và một partition thứ ba chứa tất cả dữ liệu của các năm trước nữa.

Các partition của một khối có thể được lưu trữ độc lập trong các cách thức khác nhau với các mức độ tổng kết khác nhau. Các partition không thể hiện đối với người dùng, đối với họ một khối (cube) là một đối tượng đơn, và chúng cung cấp các tuỳ chọn đa dạng để quản lý dữ liệu OLAP. Một khối ảo (virtual cube) là một khung nhìn luận lý (logic) của các phần chia của một hoặc nhiều khối.

Một khối ảo có thể được sử dụng để nối (join) các khối khác nhau để chia sẻ một chiều chung nào đó, ví dụ như có thể kết giữa khối Bán hàng và khối Kho nhằm các mục đích phân tích đặc biệt nào đó trong khi duy trì các khối tách biệt cho đơn giản. Các chiều (dimension) và các độ đo (measure) có thể được chọn từ các khối được kết để thể hiện trong khối ảo.

* + 1. **So sánh OLAP và OLTP**

Đặc trưng của các ứng dụng OLTP (On-Line Transaction Processing) là các tác vụ xử lý tự động ghi chép dữ liệu xử lý tác vụ của một tổ chức như ghi nhận đơn đặt hàng và các giao dịch ngân hàng (chúng là những công việc hàng ngày của tổ chức thương mại) mà cần phải đọc hoặc cập nhật một vài mẩu tin dựa trên khoá chính của chúng. Những tác vụ đó có cấu trúc, được lặp lại, bao gồm các giao dịch ngắn, tối giản và tách biệt, yêu cầu dữ liệu chi tiết và mới cập nhật. Các cơ sở dữ liệu tác nghiệp có xu hướng từ vài trăm megabyte đến hàng gigabyte kích thước và chỉ lưu trữ các dữ liệu hiện hành. Tính nhất quán và khả năng phục hồi của cơ sở dữ liệu là then chốt, và tối đa thông lượng giao dịch là thước đo chính yếu. Vì thế cơ sở dữ liệu được thiết kế để tối thiểu các xung đột trùng lặp.

Còn kho dữ liệu, mục tiêu là hỗ trợ quyết định cho các nhà quản lý. Tính chi tiết và riêng lẻ của các mẩu tin thì ít quan trọng hơn tính lịch sử, tổng kết và hợp nhất của dữ liệu. Do đó, kho dữ liệu thường chứa dữ liệu hợp nhất từ một hoặc nhiều cơ sở dữ liệu tác nghiệp và được thu thập qua một thời gian dài. Kết quả là kích thước kho dữ liệu có khuynh hướng từ vài trăm gigabyte đến hàng terabyte so với các cơ sở dữ liệu tác nghiệp. Kho dữ liệu hỗ trợ các truy vấn phức tạp với thời gian hồi đáp nhanh, các truy vấn phức tạp có thể truy xuất hàng triệu mẩu tin và thực hiện nhiều lần các thao tác quét, kết và tổng hợp. Đối với kho dữ liệu, số lượng truy vấn đưa vào và thời gian hồi đáp quan trọng hơn số lượng giao dịch đưa vào. Mà OLAP là một trong những công cụ cho phép thực hiện hiệu quả các truy vấn này.

Căn cứ vào đó, các cơ sở dữ liệu tác nghiệp được xây dựng để hỗ trợ tốt các tác vụ OLTP, vì thế nếu cố gắng thực thi các truy vấn OLAP phức tạp đối với các cơ sở dữ liệu tác nghiệp sẽ cho kết quả là hiệu quả thực hiện không thể chấp nhận được.

* + 1. **Các thành phần của OLAP**

Những thành phần mà OLAP sử dụng để thực hiện các dịch vụ bao gồm:

- Nguồn dữ liệu: Các cơ sở dữ liệu OLTP và các nguồn dữ liệu hợp lệ khác chứa các dữ liệu có thể chuyển đổi thành dữ liệu OLAP trong kho lưu trữ.

- Kho trung gian: là nơi lưu trữ và xử lý dữ liệu được tập hợp, sau đó được sắp xếp, sàng lọc, chuyển đổi thành dữ liệu OLAP hữu ích.

- Máy chủ lưu trữ: Các máy tính chạy cơ sở dữ liệu liên kết chứa các kho dữ liệu cho kho lưu trữ, và các máy chủ quản lý dữ liệu OLAP (warehouse server).

- Ứng dụng thông minh: Các bộ công cụ và ứng dụng thực hiện truy vấn dữ liệu OLAP và cung cấp các báo cáo và thông tin cho người ra quyết định của doanh nghiệp (Business Intelligence).

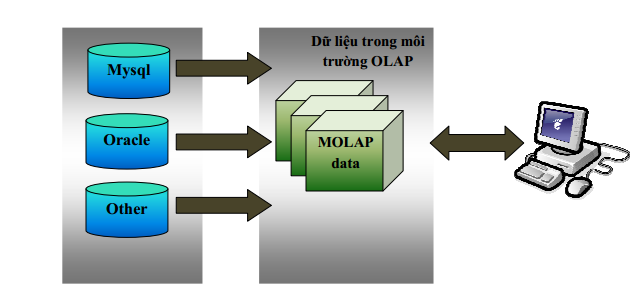
- Siêu dữ liệu: Các đối tượng như các bảng biểu trong cơ sở dữ liệu OLTP, các khối trong kho lưu trữ dữ liệu, và các bản ghi mà ứng dụng tham chiếu tới các đoạn dữ liệu khác nhau.

* + 1. **Các mô hình lưu trữ hỗ trợ OLAP**

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

1. Mô hình Multidimentional OLAP (MOLAP)

Mô hình OLAP đa chiều (MOLAP) lưu trữ dữ liệu cơ sở (là dữ liệu từ các bảng của kho dữ liệu hoặc data mart) và thông tin tổng hợp (là các độ đo được tính toán từ các bảng) trong các cấu trúc đa chiều gọi là các khối (cube). Các cấu trúc này được lưu bên ngoài cơ sở dữ liệu data mart hoặc kho dữ liệu.



Hình 2. 7 Mô hình MOLAP[8]

Lưu trữ các khối (cube) trong cấu trúc MOLAP là tốt nhất cho các truy vấn tổng hợp dữ liệu thường xuyên mà cần thời gian hồi đáp nhanh. Ví dụ, tổng sản phẩm bán được của tất cả các vùng theo quý.

* Ưu điểm của mô hình MOLAP:

- Thực thi nhanh: 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 được tạo ra trước khi khối tạo ra.

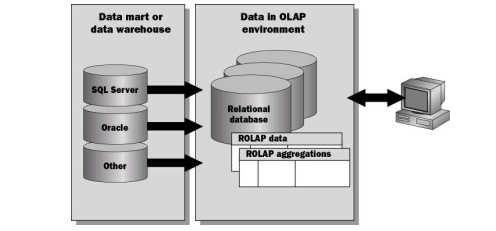
* Nhược điểm của mô hình MOLAP:

- Giới hạn lượng dữ liệu có thể xử lý: Bởi vì tất cả các tính toán được sinh ra khi xây dựng khối, do đó nó không thể bao gồm lượng dữ liệu lớn trong khối của chính nó. Điều này không có nghĩa là dữ liệu từ khối không thể được xây dựng từ một lượng dữ liệu lớn. Điều này có thể, nhưng nó chỉ tóm tắt thông tin chứa trong chính nó.

- Yêu cầu đầu tư thêm: Công nghệ tạo khối thường được độc quyền và không tồn tại trong tổ chức nào. Vì vậy, để sử dụng công nghệ MOLAP cần phải đầu tư bổ sung thêm vốn và nhân lực

1. Mô hình Relational OLAP (ROLAP)

Mô hình OLAP quan hệ (ROLAP) lưu trữ dữ liệu cơ sở và thông tin tổng hợp trong các bảng quan hệ. Các bảng này được lưu trữ trong cùng cơ sở dữ liệu như là các bảng của kho dữ liệu.



Hình 2. 8 Mô hình ROLAP[8]

Lưu trữ các khối trong cấu trúc ROLAP là tốt nhất cho các truy vấn dữ liệu không thường xuyên. Ví dụ như nếu 80% người dùng truy vấn chỉ dữ liệu trong vòng một năm trở lại đây, các dữ liệu cũ hơn một năm sẽ được đưa vào một cấu trúc ROLAP để giảm không gian đĩa bị chiếm dụng, hơn nữa còn để loại trừ dữ liệu trùng lặp.

* Ưu điểm của mô hình ROLAP:

- Có thể xử lý lượng dữ liệu lớn: Kích thước giới hạn của ROLAP phụ thuộc vào kích thước của cơ sở dữ liệu ngồn. Nói cách khác, bản thân công nghệ ROLAP không có giới hạn về kích thước dữ liệu.

- Có thể vận dụng chức năng vốn có của cơ sở dữ liệu quan hệ: Cơ sở dữ liệu quan hệ thường đi kèm với rất nhiều chức năng. Công nghệ ROLAP có thể tận dụng các chức năng này, tiết kiệm chi phí.

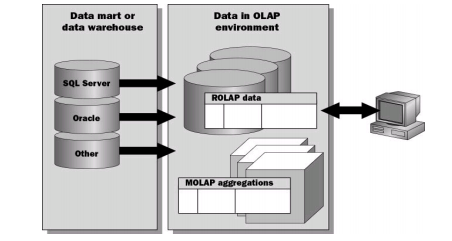
* Nhược điểm của ROLAP:

- Hiệu suất xử lý thấp: Mỗi báo cáo ROLAP thường được tập hợp dữ liệu từ nhiều bảng khác nhau, điều này sẽ làm cho hiệu quả của ROLAP thấp khi dữ liệu lớn, phân tán.

- Giới hạn bởi chức năng của SQL: Bởi vì công nghệ ROLAP chủ yếu dựa vào việc tạo ra các câu lệnh SQL để truy vấn cơ sở dữ liệu. Mà báo cáo dựa trên truy vấn SQL trong một số trường hợp không đạt được hiệu quả mong muốn. Các nhà phát triển đã khắc phục điều này bằng các tạo ra các công cụ hỗ trợ ngoài giúp người dùng tạo ra các chức năng của riêng họ

1. Mô hình Hybird OLAP (HOLAP)

Mô hình OLAP lai (HOLAP) là sự kết hợp giữa MOLAP và ROLAP.



Hình 2. 9 Mô hình HOLAP[8]

Lưu trữ các khối (cube) trong cấu trúc HOLAP là tốt nhất cho các truy vấn tổng hợp dữ liệu thường xuyên dựa trên một lượng lớn dữ liệu cơ sở. Ví dụ, chúng ta sẽ lưu trữ dữ liệu bán hàng theo hàng quý, hàng năm trong cấu trong MOLAP và dữ liệu hàng tháng, hàng tuần và hàng ngày trong cấu trúc ROLAP.

* Lợi ích của việc lưu trữ trong cấu trúc HOLAP là:

- Lấy dữ liệu trong khối (cube) nhanh hơn bằng cách sử dụng xử lý truy vấn tốc độ cao của MOLAP.

- Tiêu thụ ít không gian lưu trữ hơn MOLAP.

- Tránh trùng lặp dữ liệu.

* + 1. **So sánh các mô hình OLAP**

Bảng 2. 1 So sánh các mô hình OLAP

|  |  |  |  |
| --- | --- | --- | --- |
|  | MOLAP | ROLAP | HOLAP |
| Lưu trữ dữ liệu 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 |

* + 1. **Giới thiệu bộ công cụ Pentaho**a) Giới thiệu

Bộ công cụ Pentaho open BI cung cấp một cái nhìn toàn cảnh về khả năng kinh doanh thông minh (BI) của doanh nghiệp bao gồm: lập biểu báo, phân tích, biểu đồ, tích hợp dữ liệu, và là một hệ BI mã nguồn mởphổbiến nhất thế giới. Sản phẩm của

Pentaho được các doanh nghiệp hàng đầu sử dụng như: MySql, Motorola, Terra

Industries, DivX…[6]

Bộ công cụ pentaho có các công cụ:

- Report designer

- Design studio

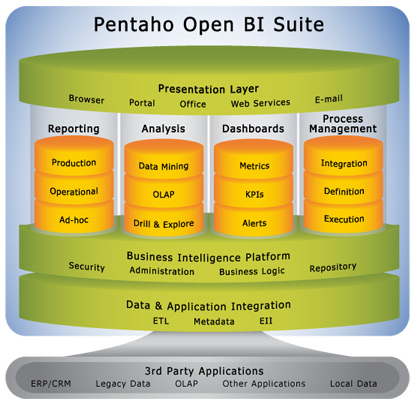
- Aggregation designer

- Metadata editer

- Pentaho data integartion

- Schema wordbench

Cấu trúc của Pentaho:

****

Hình 3. 1 Pentaho suite [5]

**b) Công cụ tích hợp dữ liệu của Pentaho**

Bộ Tích hợp dữ liệu Pentaho (PDI) là thành phần của Pentaho chịu trách nhiệm về quá trình Extract, Transform và Load (ETL). Mặc dù các công cụ ETL được thường xuyên nhất được sử dụng trong các môi trường kho dữ liệu, nhưng PDI cũng có thể được sử dụng cho các mục đích khác:

* Chuyển dữ liệu giữa các ứng dụng hoặc cơ sở dữ liệu
* Xuất dữ liệu từ cơ sở dữ liệu đến các tập tin không có cấu trúc
* Tải một lượng dữ liệu lớn vào cơ sở dữ liệu
* Làm sạch dữ liệu
* Tích hợp ứng dụng

PDI cung cấp giao diện người dùng cho phép thực hiện các quá trình mà không cần phải biết code của chương trình, có thể nói PDI là công cụ siêu dữ liệu theo định hướng[5].

PDI có thể được sử dụng như một ứng dụng độc lập, hoặc nó có thể được sử dụng như một phần của sự lớn Pentaho Suite. Là một công cụ ETL, nó là công cụ mã nguồn mở phổ biến nhất hiện nay. PDI hỗ trợ một mảng rộng lớn của đầu vào và đầu ra định dạng, bao gồm các file văn bản, bảng dữ liệu, cơ sở dữ liệu và các công cụ thương mại và miễn phí. Hơn nữa, khả năng chuyển đổi của PDI cho phép bạn thao tác dữ liệu với rất ít những hạn chế.

* 1. **Giới thiệu về Ofbiz**
     1. **Giới thiệu Framework Ofbiz**

OFbiz là một mã nguồn mở cung cấp framework cho việc phát triển một ứng

dụng web trên nền Java. OFbiz cung cấp sẵn các thành phần cần thiết cho ứng dụng.

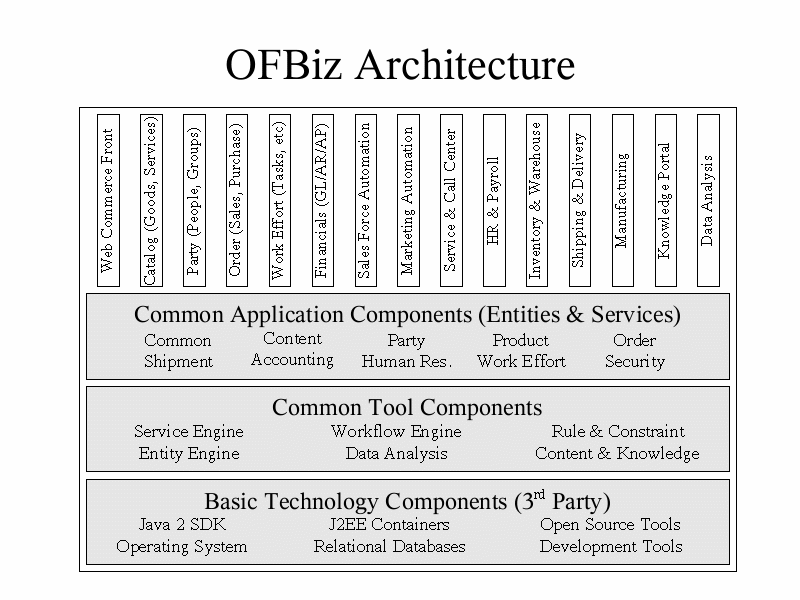
Phát triển ứng dụng dựa trên OFbiz là việc viết mã sử dụng các thành phần có sẵn để phối hợp chúng hoạt động thành một ứng dụng hoàn chỉnh. Công việc xây dựng một ứng dụng web trên OFbiz cũng tương tự xây một ngôi nhà, ngôi nhà đã có sẵn nền móng và các thành phần khác như tường, cửa, v.v… Công việc của người thợ là kết hợp các bức tường theo một cách nào đó và đặt các ô cửa ở các vị trí thích hợp.

OFBiz là framework mã nguồn mở được thiết kế để tạo thuận lợi cho việc xây dựng ERP software (ERP là tên chung cho bất kì hệ thống nào mà tích hợp tất cả quy trình business và dữ liệu thành 1 hệ thống duy nhất).



Hình 2. 10 Các thành phần ERP

Kiến trúc OFbiz gồm 04 lớp: Công nghệ cơ sở (Basic technology), Công cụ chung (Common tool), ứng dụng chung (Common application) và ứng dụng (Application)



Hình 2. 11 Kiến trúc Ofbiz

* Lớp công nghệ cơ sở cung cấp các công nghệ mới nhất để phát triển phần mềm trên J2EE, OFbiz hỗ trợ JavaBean, CORBA, Web Service, các cơ sở dữ liệu quan hệ, các công cụ để phát triển và quản trị hệ thống. Lớp thứ nhất của OFbiz là chức năng tương tự như một Application Server phục vụ các chức năng cơ bản của máy chủ dữ liệu.
* Lớp công cụ chung cung cấp các dịch vụ hỗ trợ cho việc phát triển phần mềm. Entity Engine cung cấp một cách thức dễ dàng để truy xuất vào cơ sở dữ liệu, không cần phải viết các câu lệnh SQL mà chỉ cần dùng các phương thức của Entity Engine để truy xuất và thao tác dữ liệu. Entity Engine thay thế cho việc dùng JDBC và các lệnh SQL chuẩn. Service Engine cung cấp cho lập trình viên thao tác lập trình trên mô hình MVC được dễ dàng và thuận lợi. Workflow engine dựa trên đặc tả WFMC và OMG. Nó là một phần của Services framework và được tích hợp chặt chẽ với Entity Engine, OFbiz Workflow Engine sử dụng XPDL làm ngôn ngữ định nghĩa tiến trình (process definition language). OFbiz Rule Engine dựa trên kỹ thuật lập trình logic đã tồn tại hàng thập kỉ. Ngôn ngữ lập trình logic được nhiều người biết đến là Prolog, mặc dù có nhiều ngôn ngữ khác nữa, ý tưởng chính ở đây là thay vì đưa ra một tập các chỉ thị thực thi theo một trình tự cố định như các ngôn ngữ truyền thống, chúng ta chỉ cần định nghĩa các sự kiện (fact) và các luật (rule).
* Lớp ứng dụng chung là các ứng dụng cơ bản nhất đối một doanh nghiệp, OFbiz đã xây dựng các ứng dụng này thành các thành phần cơ bản, khi phát triển ứng dụng các nhà phát triển phần mềm chỉ khảo sát thực tế, tích hợp các thành phần ứng dụng này lại theo đúng yêu cầu khảo sát là có thể sử dụng được. Việc này giúp cho việc phát triển phần mềm trở nên nhanh chóng và dễ nâng cấp, bảo trì.
* Lớp ứng dụng trên cùng là các ứng dụng chuẩn của doanh nghiệp, các doanh nghiệp chỉ cần chỉnh lại cấu hình để đáp ứng đúng thực tế của doanh nghiệp. Các ứng dụng này hầu như đã đáp ứng được hết các yêu cầu của doanh nghiệp.

Như đã phân tích ở trên, kiến trúc OFbiz là một flatform, một framework để cho việc phát triển một ứng dụng trên web được nhanh chóng và dễ dàng. Tùy theo yêu cầu cụ thể của doanh nghiệp, nếu sát với lớp trên cùng thì sử dụng ngay lớp đó, nếu không thì phải hiệu chỉnh lại lớp ứng dụng, nếu vẫn không sát thực tế thì phải phát triển ứng dụng trên lớp công cụ chung, nếu vẫn còn chưa được thì nhà phát triển phần mềm phải sử dụng lớp công nghệ cơ sở để phát triển ứng dụng. Càng về lớp trên thời gian phát triển càng nhanh nhưng độ linh hoạt càng thấp, lớp dưới cùng thì thời gian phát triển ứng dụng lâu hơn nhưng giải quyết được nhiều việc hơn.

Các thành phần trong Ofbiz bao gồm :

* Accounting (Agreements, Invoicing, Vendor Management, General Ledger ): Phần mềm kế toán cho công ty
* Asset Maintenance: Bảo vệ tài sản cho nhân viên trong công ty
* Catalogue and Product Management: Quản lý sản phẩm và calatogue
* Facility and Warehouse Management: Quản lý kho và cơ sở vật chất
* Manufacturing: Quản lý sản xuất của công ty
* Order Processing: Xử lý thông tin đặt hàng của khách hàng
* Inventory Management, automated stock replenishment: Kiểm toán, tự động bổ sung cổ phiếu
* Content management system (CMS): Quản lý nội dung hệ thống
* Human resources (HR): Quản trị nhân sự trong công ty
* People and Group Management: Quản lý cá nhân và nhóm người lao động
* Project Management: Quản lý dự án
* Sales Force Automation: Bán hàng tự động
* Work Effort Management: Quản lý kết quả làm việc
* Electronic point of sale (ePOS): Quản lý chi nhánh báng hàng điện tử
* Electronic commerce (eCommerce): Thương mại điện tử
  + 1. **Mô hình cơ sở dữ liệu Ofbiz**

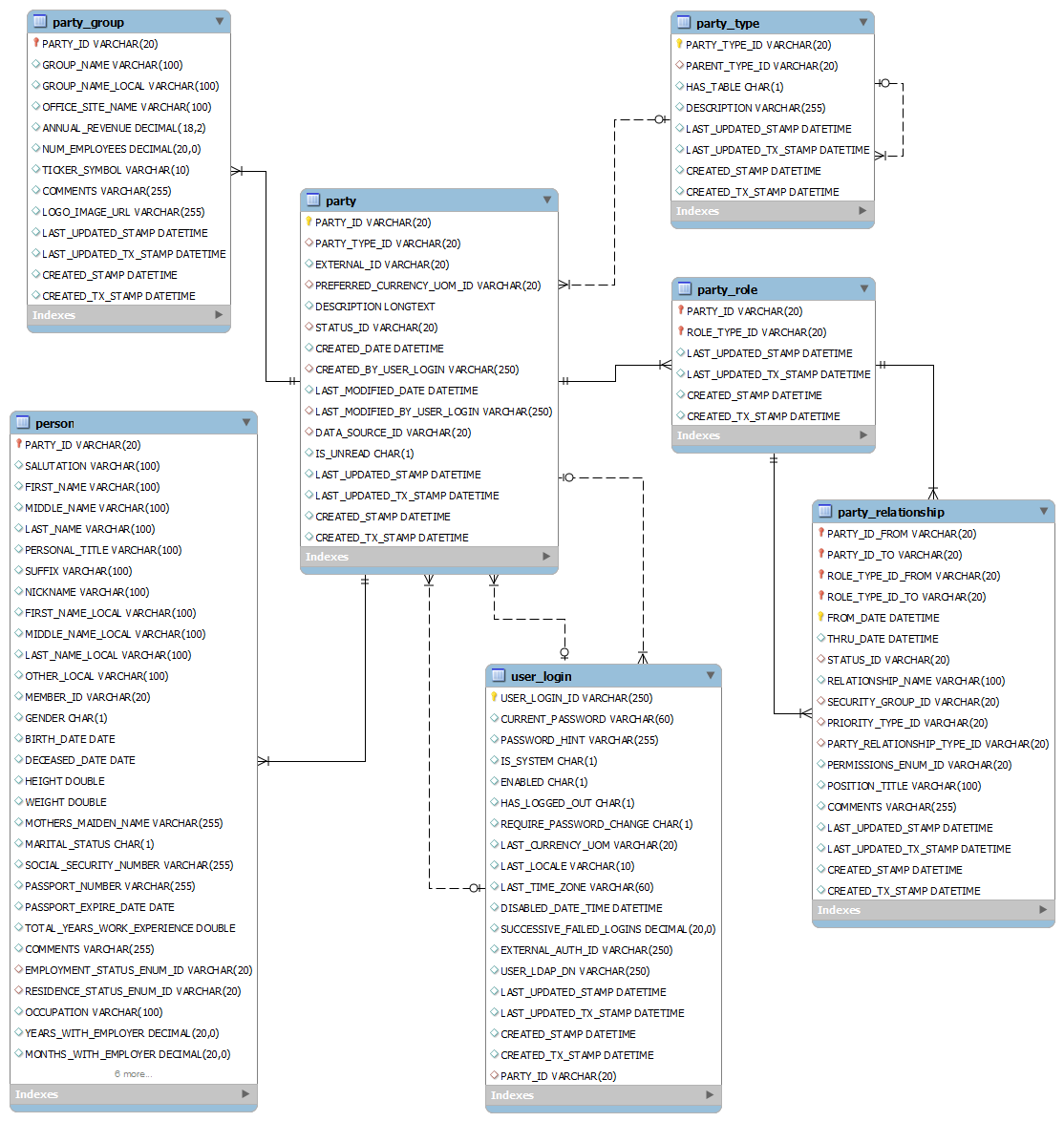
DataModel của OFBiz được xây dựng dựa trên mô hình phổ quát dữ liệu Universal Data Model (mô hình thực thể - quan hệ) và được Len Silverston viết thành cuốn The Data Model Resource Book, là một mô hình rất khoa học, dễ phát triển.

Mô hình phổ quát dữ liệu UDM làm giảm đáng kể thời gian và chi phí cho việc xây dựng mô hình và dữ liệu kho dữ liệu thông qua các mẫu tái sử dụng các khuôn mẫu(template) và mô hình(model) đã được sử dụng và chứng minh trên thực tế. Đáp ứng được nhu cầu thay đổi của doanh nghiệp, tổ chức với các đặc thù nghề nghiệp hoạt động khác nhau từ doanh nghiệp kinh doanh phân phối đến doanh nghiệp viễn thông, tổ chức y tế bệnh viện bảo hiểm, thậm chí cả kinh doanh online ecommerce. Và đặc biệt là đáp ứng các đặc thù của các tổ chức trên phạm vi toàn thế giới.

Dựa trên kinh nghiệm đúc kết được qua thiết kế triển khai cho các doanh nghiệp và tổ chức, UDM đã được ra đời nhằm mục đích tái sử dụng các mô hình (model), khuôn mẫu (template) và công cụ (tool) một cách nhanh chóng và tiết kiệm, ngoài ra UDM cũng sẽ là 1 mô hình phổ biến để các Quản trị viên dữ liệu (Data Administrator), Chuyên viên mô hình dữ liệu (Data modeler), Chuyên viên phân tích dữ liệu (Data Analyst), Chuyên viên thiết kế Cơ sở dữ liệu (Data Designer), Quản trị kho dữ liệu (Data warehouse administrator), Chuyên viên thiết kế kho dữ liệu (Data Warehouse Designer) hoặc bất cứ ai (gọi chung là Người thiết kế cơ sở dữ liệu Database Designer) cần phân tích, quản lý hoặc tích hợp cấu trúc dữ liệu. Bằng việc sử dụng UDM, người thiết kế dữ liệu có thể loại bỏ và phòng tránh các dư thừa dữ liệu, đây là việc hết sức cần thiết vì sẽ ảnh hưởng đến tính chính xác của thông tin và sự hiệu quả của hệ thống.

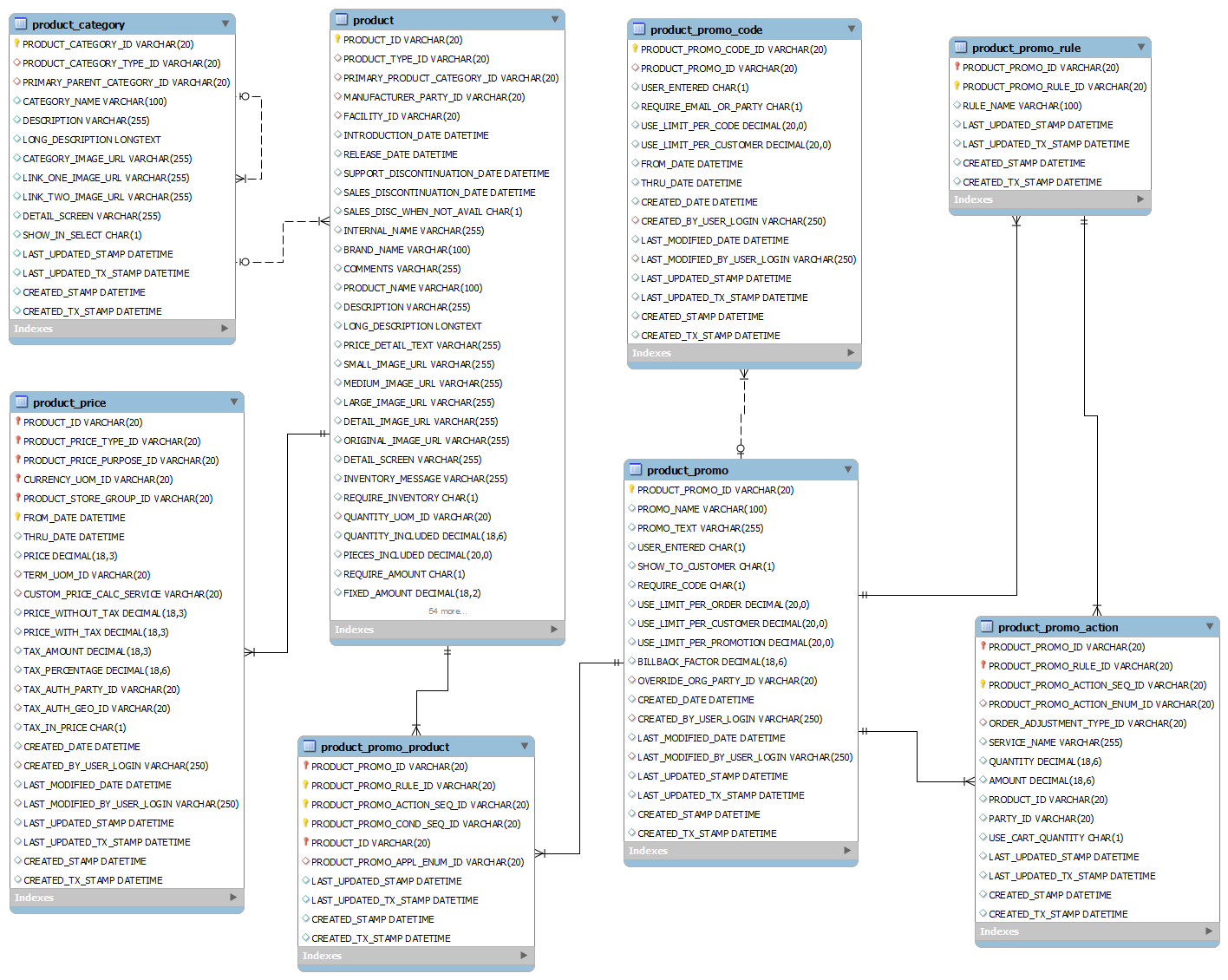
Một trong các thách thức lớn nhất khi xây dựng một hệ thống hoạt động hiệu quả chính là việc tích hợp. Các hệ thống thường được xây dựng một cách tách rời để phù hợp với các nhu cầu riêng biệt tại các thời điểm khác nhau. Các doanh nghiệp và tổ chức cần xây dựng nhiều hệ thống khác nhau như: Quản lý chăm sóc khách hàng, Quản lý hệ thống bán hàng, Quản lý dự án, Hệ thống kế toán, Quản lý ngân sách, Quản lý mua hàng, Quản lý nhân sự... Một ví dụ thường thấy là khi các hệ thống được xây dựng riêng biệt, các thông tin giống nhau sẽ bị lặp lại ở các hệ thống đó, cụ thể như thống tin về tổ chức, nhân viên, con người, vị trí địa lý hoặc thông tin sản phẩm. Điều đó đồng nghĩa với việc từng hệ thống riêng rẽ đó sẽ cần xây dựng và sử dụng nguồn thông tin riêng. Một vấn đề lớn với cách tiếp này là việc đảm bảo sự đồng bộ và chính xác về dữ liệu của các hệ thống rời rạc đó. Thông tin sẽ không được xuyên suốt qua các hệ thống, để có được điều đó thì cũng phải trả một giá khá đắt khi xây dựng các giải pháp tích hợp các hệ thống đó lại.

UDM ngoài việc định ra các template và data model tương ứng còn đưa ra các ví dụ rất sinh động, phản ánh đầy đủ nhu cầu của từng tổ chức về mặt hoạt động và thay đổi (nếu có). Do đó từ việc thiết kế đến triển khai cho các tổ chức khác nhau sẽ không tốn kém thời gian và đảm bảo tính phù hợp cho hoạt động và phát triển của doanh nghiệp. Từ đó tiết kiệm chi phí thiết kế, bảo trì và nâng cấp (nếu có).



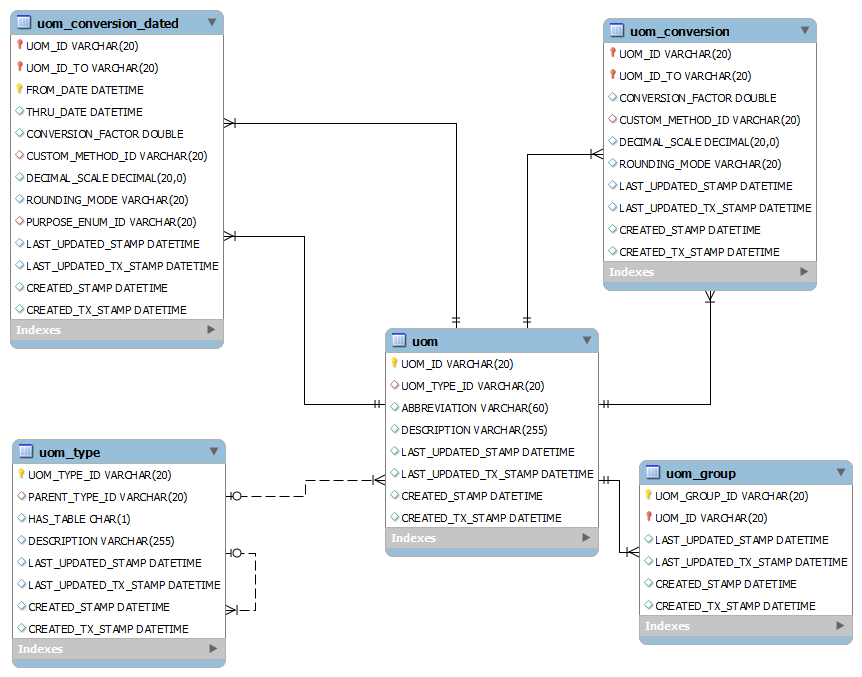
Hình 2. 12 Quản lý người dùng, tổ chức trong ofbiz

* Danh mục các bảng chính quản lý người dùng, tổ chức trong Ofbiz[7]:
  + Party: Lưu trữ thông tin cơ bản của người dùng, tổ chức
  + PartyType: Party được phân ra nhiều kiểu: Tổ chức, người..
  + Person: nếu PartyType là Person sẽ được lưu thông tin chia tiếp tại đây
  + PartyGroup: nếu PartyType là PartyGroup sẽ được lưu thông tin chi tiết tại dây
  + PartyRole: các quyền của party trong hệ thống
  + UserLogin: thông tin tài khoản truy cập hệ thống
  + PartyRelationship: Lưu thông tin quan hệ giữa các party



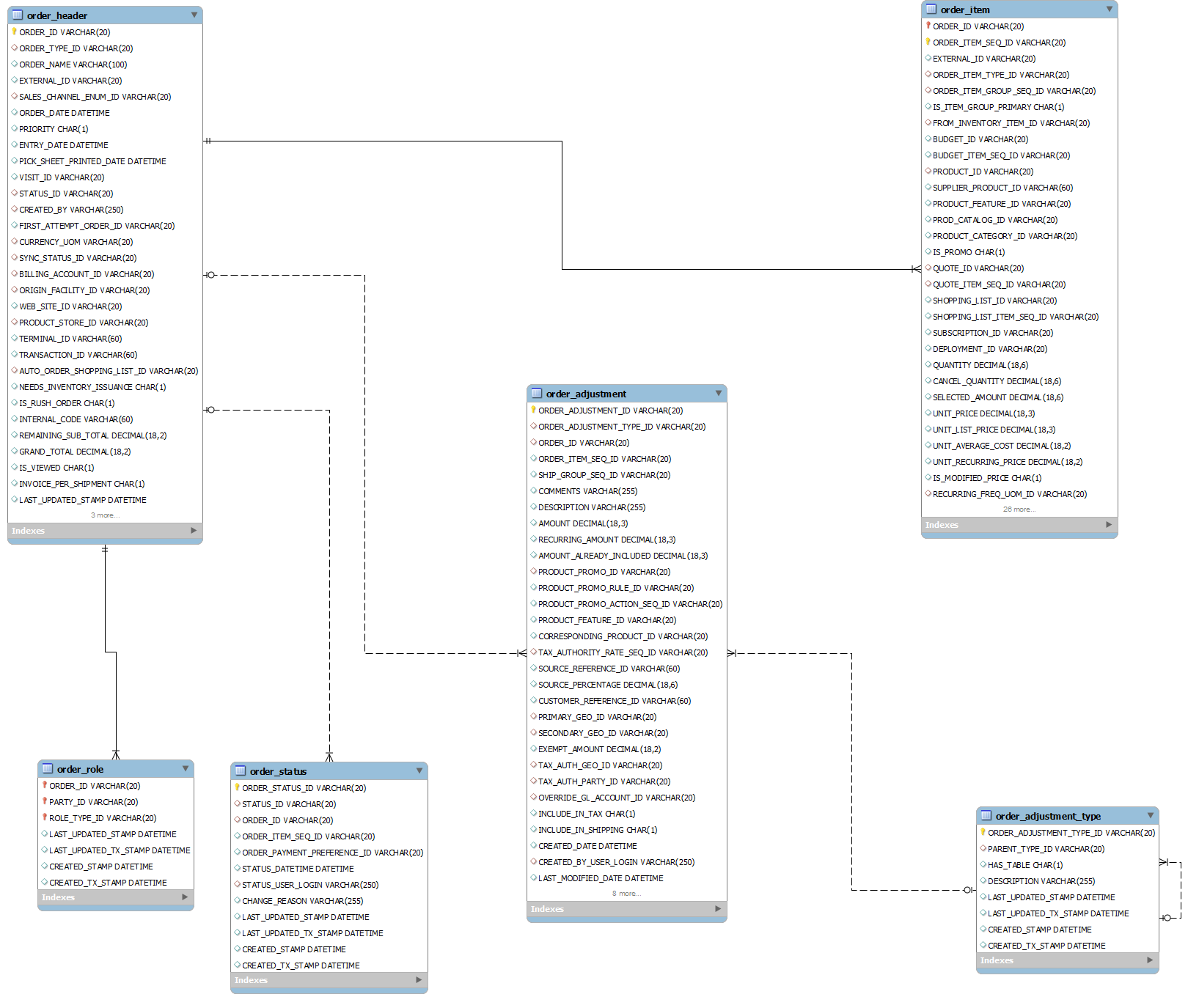
Hình 2. 13 Quản lý sản phẩm trong Ofbiz

* Danh muc các bản quản lý sản phẩm chính trong Ofbiz [7]:
  + Product: lưu thông tin sản phẩm
  + ProductPrice: giá của sản phẩm
  + ProductCategory: Category của sản phẩm
  + ProductPromo: Chương trình khuyến mại liên quan đến sản phẩm
  + ProductPromoRule: Luật để khách hàng được áp dụng chương trình khuyến mại
  + ProductPromoAction: Hình thức khuyến mại khi khách hàng đáp ứng các yêu cầu cửa chương trình khuyến mại
  + ProductPromoProduct: Sản phẩm áp dụng khuyến mại
  + ProductPromocode: Mã nhập chương trình khuyến mại



Hình 2. 14 Quản lý các đơn vị chuyển đổi trong ofbiz

* Danh mục các bảng chình trong quản lý đơn vị Ofbiz:
  + Uom: Tất cả các đơn vị sẽ được lưu tại đây.
  + Uom Type: Kiểu của đơn vị. Ví du: Đơn vị tiền tệ, Khối lượng..
  + Uom group: nhóm các đơn vị
  + Uom Conversion: Quy đổi từ đơn vị này sang đơn vị kia và giá trị quy đổi. Sử dụng cho các đơn vị cố định, không thay đổi theo thời gian. Vd: Km->mm
  + Uom Conversion Date: Quy đổi từ đơn vị này sang đơn vị kia và giá trị quy đổi. Sử dụng cho các đơn vị không cố định, thay đổi theo thời gian. Vd: tiền tệ…



* Dữ liệu được lưu trữ bán hàng Ofbiz:
* Order Header: Lưu các thông tin về các Order đã được tạo trên hệ thống (Kênh bán hàng, giá trị đơn hàng, trạng thái đơn hàng, người tạo đơn hàng…)
* Order Item: Lưu trữ chi tiết các sản phẩm trong đơn hàng
* Order status: Lưu trữ trạng thái, thời gian của Order, Item trong Order theo thời gian. Từ khi tạo đơn hàng đến khi kết thúc. (Thành công hoặc bị hủy)
* Order Adjustment:Lưu trữ sự thay đổi giá trị của đơn hàng so với giá trị ban đầu.Ví dụ:Một đơn hàng có tổng giá trị sản phẩm là 500.000 VNĐ, Khi tính tiền đơn hàng sẽ phải chịu thêm một khoản thuế VAT là 10% thì giá trị này sẽ được lưu tại bảng Order Adjustment
* Order Role: Lưu thông tin các cá nhân, tổ chức liên quan đến đơn hàng. Ví dụ: người đặt hàng, người thanh toán, người tạo hóa đơn …
  + 1. **Kiến trúc 3 tầng và cách thức phát triển ứng dụng với Ofbiz**

Framework Ofbiz sử dụng mô hình 3 tầng chung cho tất cả các ứng dụng của nó. Đó là tầng Data Layer, Business Logic Layer và Presentation Layer. Các tầng Data Layer và Service Layer có những cách thức riêng chịu trách nhiệm tương tác với từng tầng[3]

* **Data Model Layer**: Tầng này đại diện cho Cơ sở dữ liệu. Nó bao gồm 1 thực thể chức năng (Entity Engine) có trách nhiệm kết nối cơ sở dữ liệu, phục hồi cơ sở dữ liệu và lưu trữ… Nó sử dụng lớp Java Generic Delegator để kết nối với cơ sở dữ liệu và nó sử dụng Java Generic Value đại diện cho những thực thể được chèn vào cơ sở dữ liệu.
* **Business Logic Layer**: Tầng này đại diện cho logic, hoặc các dịch vụ cung cấp cho người dùng và thực hiện trên Cơ sở dữ liệu của tầng Data layer. Có thể có nhiều loại dịch vụ như java, SOAP, simple, workflow… và mỗi lại dịch vụ có cách thức riêng của mình. Có một Service Engine có trách nhiệm phân chia và gọi các dịch vụ..v.v..
* **Presentation Layer**: Ofbiz đã chuyển sang sử dung “Screens” để đại diện cho từng trang của Ofbiz. Do đó mỗi trang nên được biểu diễn bình thường như một screen. Một trang Ofbiz bao gồm nhiều thành phần như hearders, Footers, appheader… do đó khi dựng hình các trang, đây là sự kết hợp có trình tự đã được sắp đặt, hoặc bao gồm, trong một screen
  + 1. **Tìm hiểu thành phần sẵn có trên OFBIZ**

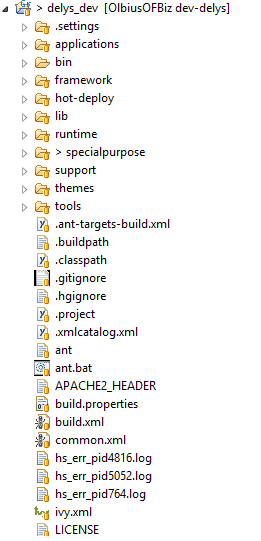
1. ***Các thành phần chính của OFBIZ***

Download Ofbiz tại đây: http://ofbiz.apache.org/download.html

Sau khi download và giải nén chúng ta sẽ có một folder OFBiz. Trước tiên chúng ta xem các thành phần trong thư mục này.

Trong đó nội dung từng thành phần như sau:

* application folder: Bao gồm các ứng dụng có trong OFBiz, khi chúng ta tự tạo một ứng dụng riêng (Chứa core application của OFBIZ). Đại diện cho những application liên quan tới business của OFBiz. Có thể tìm thấy những manufacturing, content management, order management …. trong thư mục này.
* framework folder: Bao gồm các thành phần của framework OFBiz, giống như Entity Engine, Service Engine, thư mục chứa các file chung cho tất cả mọi ứng dụng.
* hot-deploy folder: Trong thư mục này có thể có một số folder, nơi các thành phần của ứng dụng được tải tự động mà không cần tải chúng một cách rõ ràng như chúng ta vẫn thấy ở một số ứng dụng khác đặt trong Application folder.
* specialpurpose folder: Bao gồm một số ứng dụng bổ sung như “ecommerce” và “workflow”, tuy nhiên chúng không phải một phần của OFBiz core. Các component do người dùng phát triển nên đặt ở đây.
* themes folder: Bao gồm các file có chức năng thay đổi giao diện người.



Hình 2. 15 Tổ chức thư mục Ofbiz

Những thư mục như application, framework, specialpurpose, hot-deploy, themes phải được khai báo trong /framework/base/config/componentload.xml.

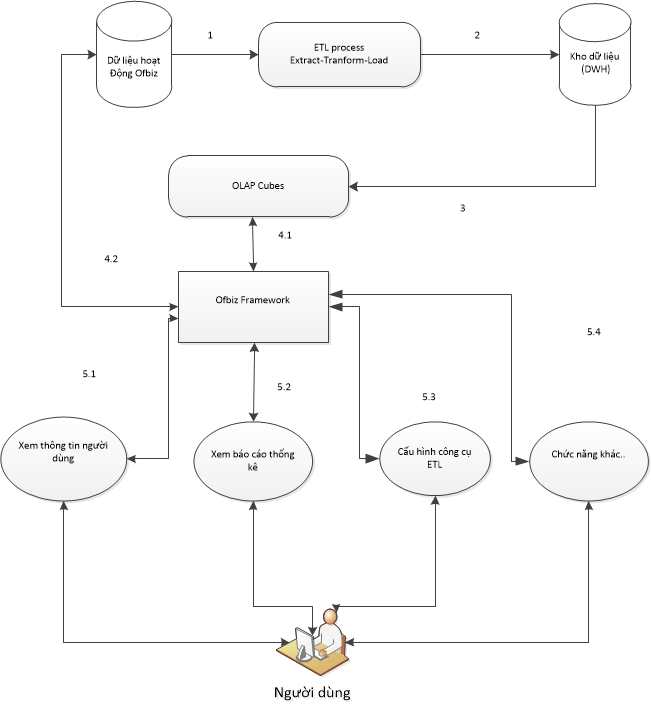
1. ***Cấu trúc của một component OFBIZ***

Cấu trúc của 1 component OFBiz điển hình gồm có:

* configuration files (config): Thường được sử dụng cho data configuaration, ví dụ nó được dùng để hỗ trợ các giao diện ngôn ngữ, bên trong nó chúng ta sẽ tìm được một số file cho các ngôn ngữ khác nhau, và dựa trên ngôn ngữ giao diện người dùng, một trong số các file này sẽ được sử dụng.
* seed data definition (data): Chứa đựng dữ liệu load về khi OFBiz được khởi động và dữ liệu demo.
* entity definition (entitydef): Chứa việc thực thi lớp dữ liệu, ví dụ bản quan hệ CSDL và các mối quan hệ. Bên trong thư mục này có 2 file chính, 1 cho việc định nghĩa và 1 cho việc thực thi : entitygroup.xml và entitymodel.xml.
* flow logic (script): Chứa đựng việc thực thi cho các service sử dụng OFBiz mini-language và nó chứa đựng một số script. Bên trong thư mục này còn chứa việc thực thi các service cho các ứng dụng khác nhau.
* service definition (servicedef): Định nghĩa các service được dùng trong lớp business logic, nó chứa các file service.xml – định nghĩa services, service\_view.xml – dùng cho view (Khi add 1 file service, cần phải được định nghĩa thêm trong ofbiz component.xml).
* java code (src): Chứa các file source cho các service – các file được thực thi với Java.
* webapp: Chứa các form và ứng dụng web. Với OFBiz các trang được chia thành các thành phần nhỏ hơn, những cái được kết hợp để tạo lên sản phẩm mới. Do đó nhiều trang có thể chia sẻ các thành phần chung như header, sidebar, và navigation bars.
* widget: Là lớp trình diễn OFBiz gần nhất, các trang được định nghĩa như “screen”. Thư mục widgets cho giao diện màn hình người dùng. OFBiz cho phép thiết kế giao diện người dùng để tạo như “generic screens” có thể được sử dụng cho những nền khác. Nội dung thư mục widget phản ánh những thứ trên webapp. Do đó mỗi ứng dụng sẽ có màn hình riêng của nó. Screen được chia làm 2 phần: action và widget. Action chịu trách nhiệm cho việc lấy dữ liệu về trong khi widget chịu trách nhiệm hiển thị dữ liệu.

**CHƯƠNG 3: MÔ HÌNH ĐỀ XUẤT**

* 1. **Mô hình hoạt động hệ thống**



Hình 3. 2 Kiến trúc hệ thống

* **Mô tả chi tiết sơ đồ hệ thống:**
  + Dữ liệu hoạt động Ofbiz: là nới lưu trữ dữ liệu của FrameWork Ofbiz
  + ETL process: Là bước thực hiện chuyển dữ liệu từ dữ liệu hoạt dộng sang kho dữ liệu (DWH)
  + Kho dữ liệu (DWH): Là kho cơ sở dữ liệu của hệ thống
  + Người dùng: thực hiện kết nối và gửi các yêu cầu đến server, đồng thời chứa giao diện tương ứng với từng chức năng.
  + Chức năng xem báo cáo thống kê: là chức năng chính mà hệ thống sẽ đáp ứng cho người dùng.
  + Chức năng cấu hình công cụ ETL: cung cấp các chưc năng cơ bản giúp người dùng thực hiện cài đặt bước ETL.
  1. **Quy trình giải quyết bài toán**

Với các cơ sở lý thuyết ở trên đồ án xin đề xuất phương án áp dụng những lý thuyết này vào giải quyết bài toán như sau:

Bước 1: Tìm hiểu Data model, dữ liệu thực tế, nghiệp vụ của doanh nghiệp trong Ofbiz

Bước 2: Thiết kế kiến trúc Data Warehouse

Bước 3: Sử dụng mô hình ETL của công cụ mã nguồn mở Pentaho để thêm mới và cập nhật dữ liệu cho Kho dữ liệu

Bước 4: Sử dụng kĩ thuật OLAP trên Ofbiz đối với Kho dữ liệu

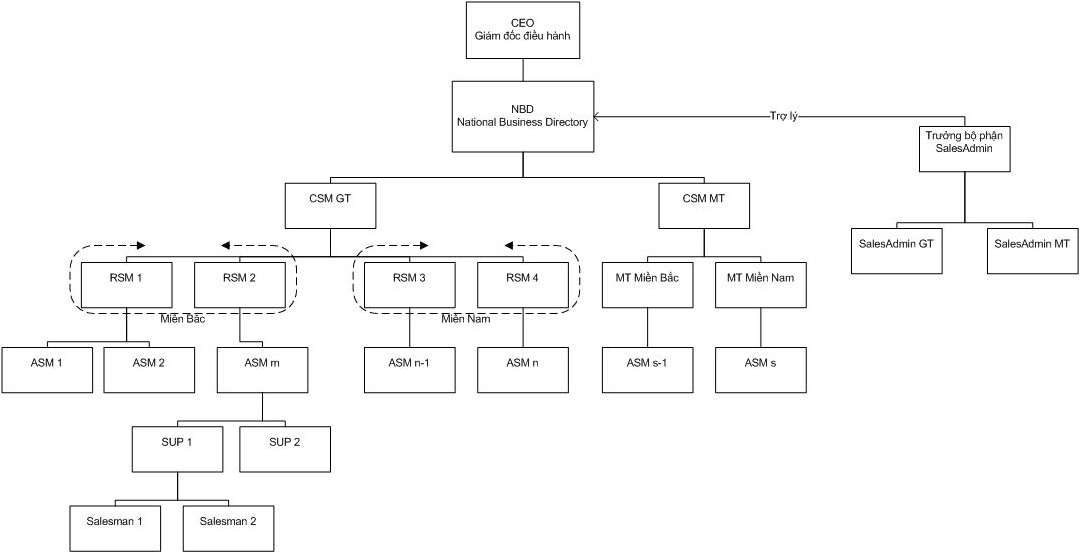
Bước 5: Đưa các biểu đồ, báo cáo lên giao diện web của ofbiz

* 1. **Chi tiết các quy trình**

1. Tìm hiểu Data model, dữ liệu thực tế, nghiệp vụ của doanh nghiệp trong Ofbiz

Công ty Delys chuyên nhập khẩu và phân phối thực phẩm cao cấp của châu Âu, đặc biệt là các sản phẩm về sữa như váng sữa và sữa chua nhập khẩu. Sản phẩm váng sữa nổi tiếng nhất Việt Nam với tên gọi Monte được công ty nhập khẩu và phân phối độc quyền từ đối tác Zott GMBH & Co. KG (CHLB Đức) – doanh nghiệp sản xuất sữa hàng đầu châu Âu với bề dầy lịch sử từ năm 1926.

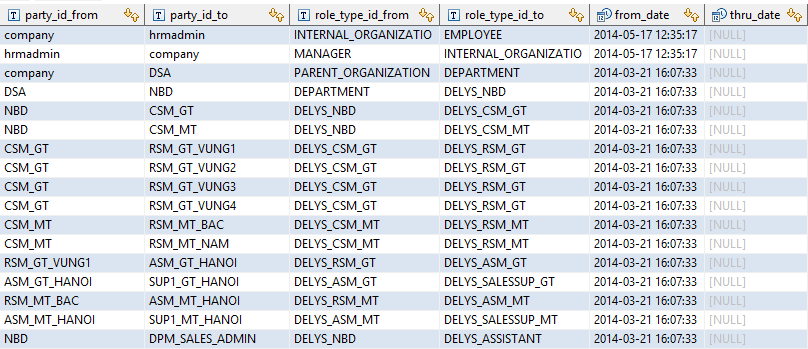
* Tổ chức nội bộ



Hình 3. 3 Tổ chức nội bộ Delys

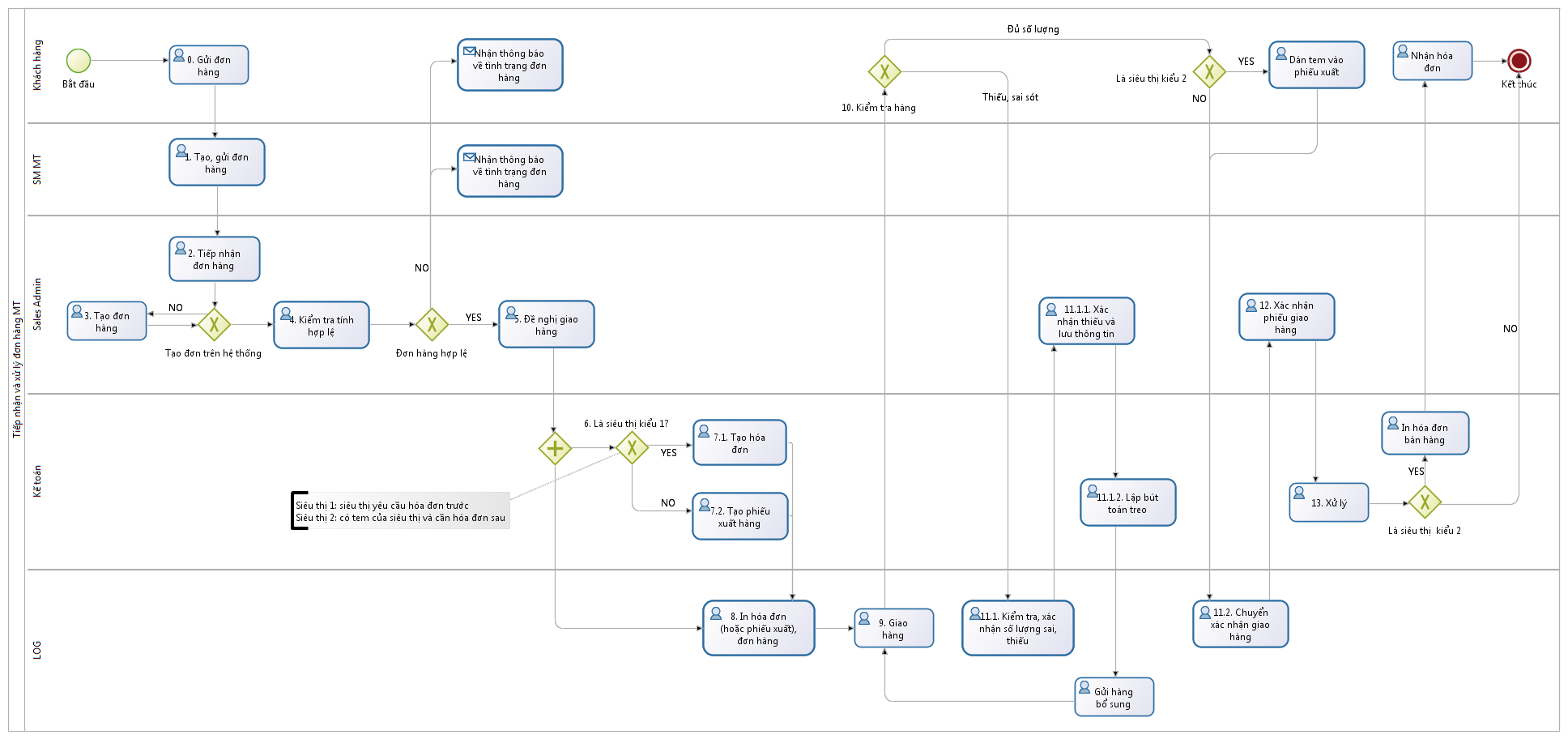
Các mối quan hệ giữa các phòng ban được lưu trong bảng PartyRelationship của Ofbiz.

* Dữ liệu thực tế:



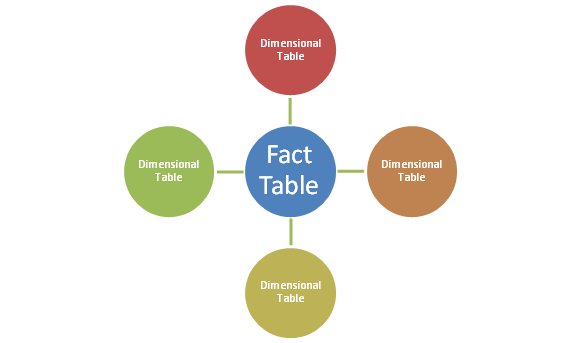
Hình 3. 4 Dữ liệu thực tế Delys trong Party Relationship

* **Quy trình xử lý đơn hàng**



Hình 3. 5 Quy trình xử lý đơn hàng tại Delys

1. Thiết kế kiến trúc Data Warehouse  
     
    Áp dụng những kiến thức cơ sở lý thuyết về Data warehouse, thực hiện việc xây dựng kiến trúc của kho dữ liệu bán hàng theo mô hình cơ sở dữ liệu quan hệ gồm:

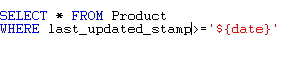
* Bảng chiều dữ liệu: Chiều thời gian, chiều tiền tệ, chiều sản phẩm, chiều chương trình khuyến mại
* Bảng sự kiện: 1 bảng sự kiến về dữ liệu liệu bán hàng  
  ****

Hình 3. 6 Mô hình quan hệ trong Data warehouse [11]

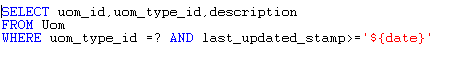
1. Sử dụng bộ công cụ Pentaho để thực hiện thêm mới và cập nhật kho dữ liệu

Ở Delys, dữ liệu từ các công ty con đã được tập hợp vào kho dữ liệu tập trung và ở dưới dạng dữ liệu quan hệ, vì vậy đây là một thuận lợi rất lớn cho những người xây dựng data warehouse.

* Tổng hợp dữ liệu hiện có trong dữ liệu hoạt động vào data warehouse:
  + Lấy dữ liệu chiều sản phẩm:

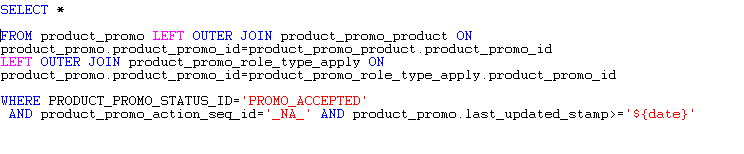


Hình 3. 7 Câu lện lấy dữ liệu chiều sản phẩm

* + Lấy dữ liệu chiều tiền tệ:  
    

Hình 3. 8 Câu lệnh lấy dữu liệu chiều tiền tệ

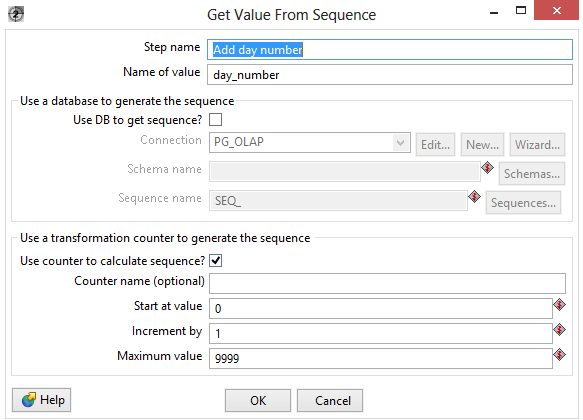
* + Lấy dữ liệu chiều chương trình khuyến mại:



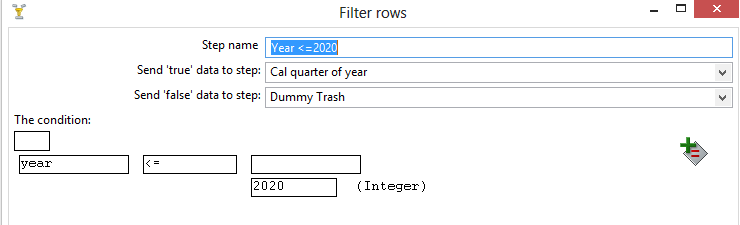
Hình 3. 9 Câu lện lấy dữ liệu chiều khuyến mại

* + Sinh dữ liệu chiều thời gian: Truyền vào giá trị một ngày bất kì, sử dụng các hàm tính toán có sẵn của Pentaho để tự sinh ra giá trị các ngày đến một mốc cho trước



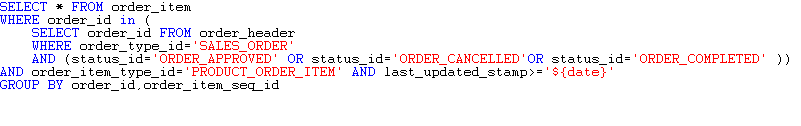


Hình 3. 10 Sinh giá trị ngày tự động trong Pentaho



Hình 3. 11 Mốc thời gian tối đa khi sinh chiều thời gian

* + Tổng hợp bảng sự kiện Order:



Hình 3. 12 Câu lệnh lấy dữ liệu bảng sự kiện

1. Sử dụng kĩ thuật OLAP trên Ofbiz đối với Kho dữ liệu

Để sử dụng được kĩ thuật OLAP, ta xây dựng một bảng view dữ liệu trên ofbiz kết nối bảng sử kiện với tất cả các chiều trong kho dữ liệu. Như vậy sẽ tạo ra một cube có nhiều chiều khác nhau, giúp cho người sử dụng thuận tiện khi truy suất.

1. Đưa các biểu đồ, báo cáo lên giao diện web của ofbiz

Để đưa ra được các báo cáo thống kê ra giao diện web:

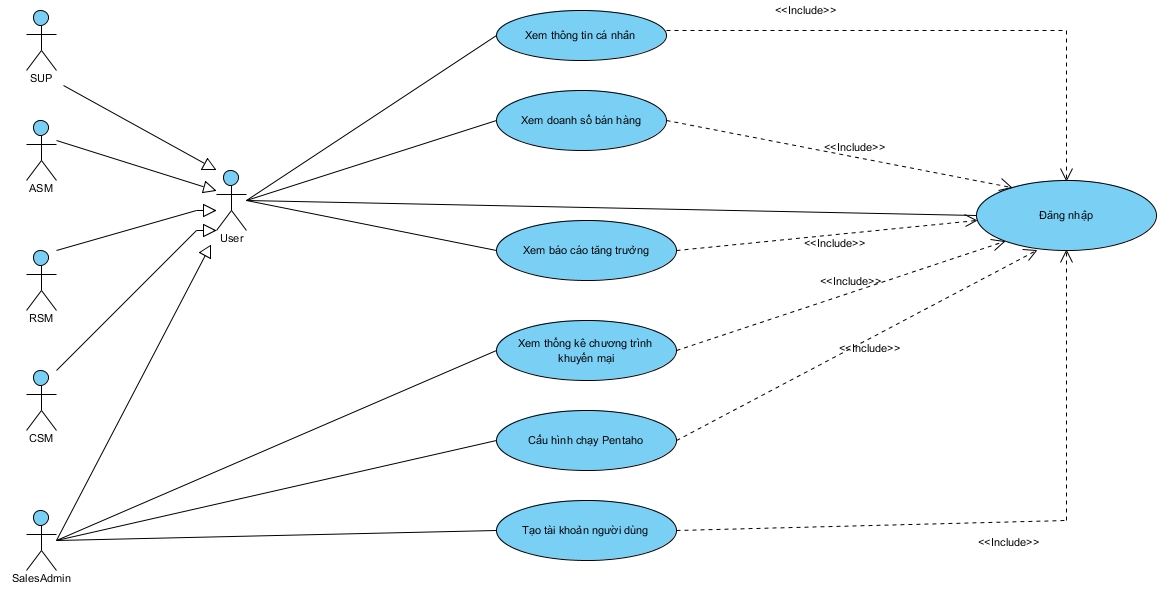
* + Truy vấn kho dữ liệu trên Ofbiz bằng các class java trên ofbiz
  + Sử dụng kĩ thuật Json để trả dữ liệu từ server về trình duyệt
  + Sử dụng framework jquuery,high chart để vẽ biểu đồ từ các dữ liệu Json trả về.

# **CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

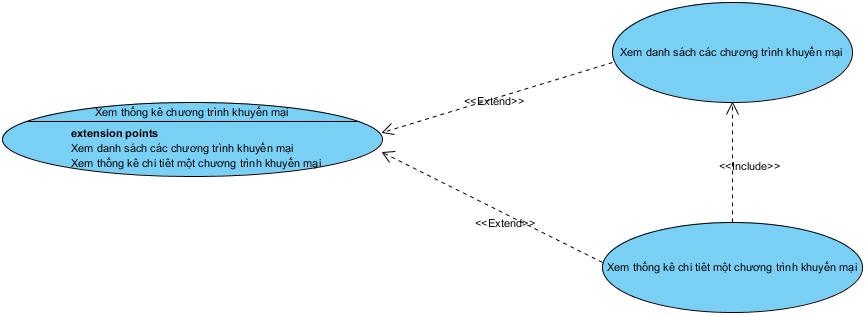
* 1. **Phân tích hệ thống**

Đồ án sử dụng phương pháp phân tích hướng đối tượng, sử dụng các cú pháp của UML 2.0 và công cụ Visual Paradigm để mô tả hệ thống.

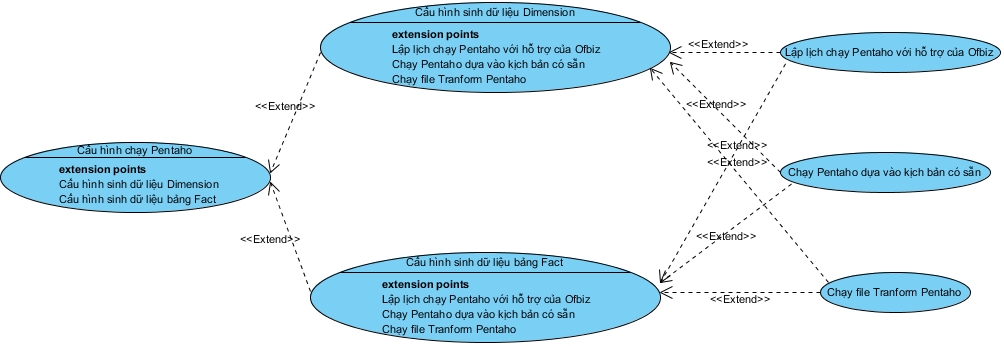
* + 1. **Biểu đồ Use Case**



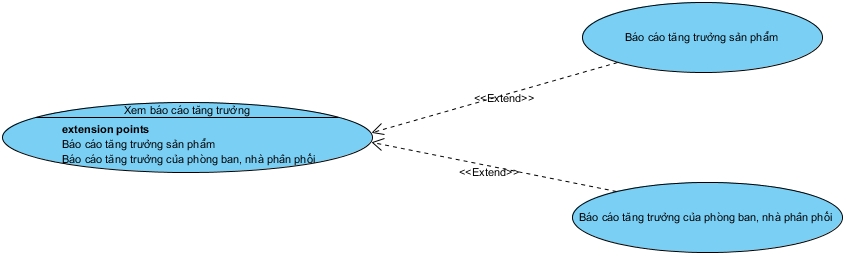
Hình 4. 1 Biểu đồ Use case tổng quan



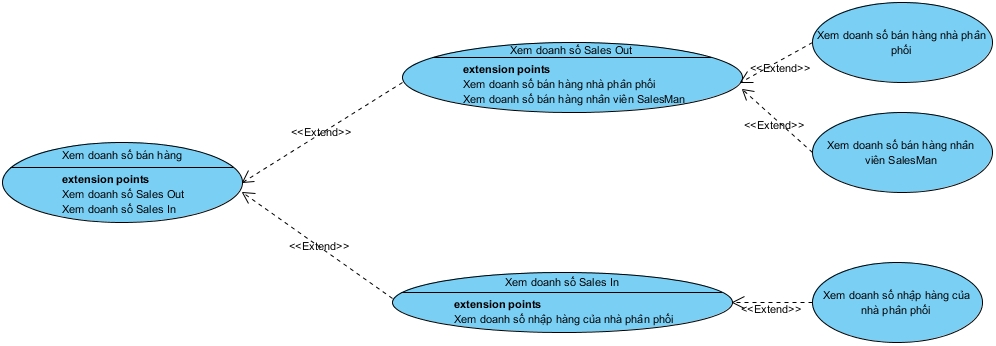
Hình 4. 2 Use case phân rã xem thống kê chương trình khuyến mại



Hình 4. 3 Use case phân rã cấu hình pentaho



Hình 4. 4 Use case phân rã Xem báo cáo tăng trưởng



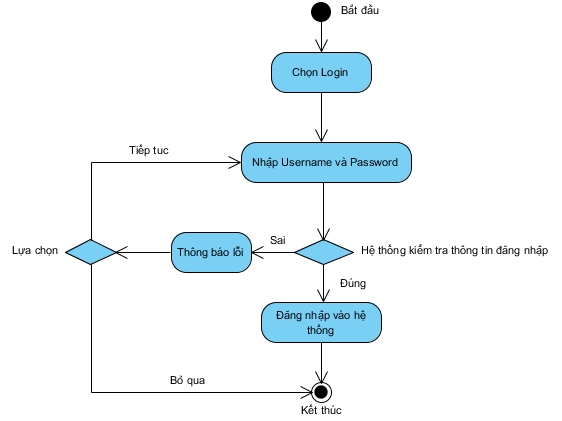
Hình 4. 5 Use cây phân rã Xem doanh số bán hàng

* + 1. **Đặc tả Use case**

1. Đặc tả Use case đăng nhập

Bảng 4. 1 Đặc tả use case đăng nhập

|  |  |
| --- | --- |
| Tên Use Case | Đăng nhập |
| Tên thực thể | Người dùng |
| Mô tả | Use Case đăng nhập sẽ cung cấp cho người dùng 1 form để đăng nhập vào hệ thống |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống đăng nhập thành công nếu người dùng nhập đúng * Hệ thống cũng sẽ đưa ra thông báo lỗi nếu gặp sự cố và yêu cầu người dùng nhập lại |
| Kịch bản | * Điền thông tin * Nếu thông tin đúng thì login thành công nếu không đúng thì yêu cầu người dùng nhập lại |

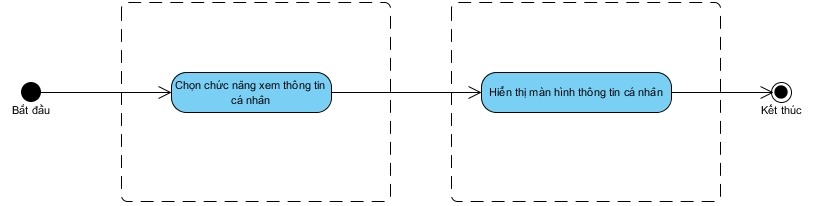


Hình 4. 6 Biểu đồ hoạt động cho chức năng đăng nhập

1. Đặc tả Use case Xem thông tin cá nhân

Bảng 4. 2 Đặc tả Use case xem thông tin cá nhân

|  |  |
| --- | --- |
| Tên Use Case | Xem thông tin cá nhân |
| Tên thực thể | Người dùng |
| Mô tả | Use Case xem thông tin cá nhân sẽ cung cấp cho người dùng một màn hình hiển thị cho phép người dùng xem thông tin cá nhân. |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống hiển thị màn hình thông tin cá nhân cho người dùng |
| Kịch bản | * Chọn chức năng xem thông tin cá nhân * Hiển thị màn hình thông tin cá nhân |

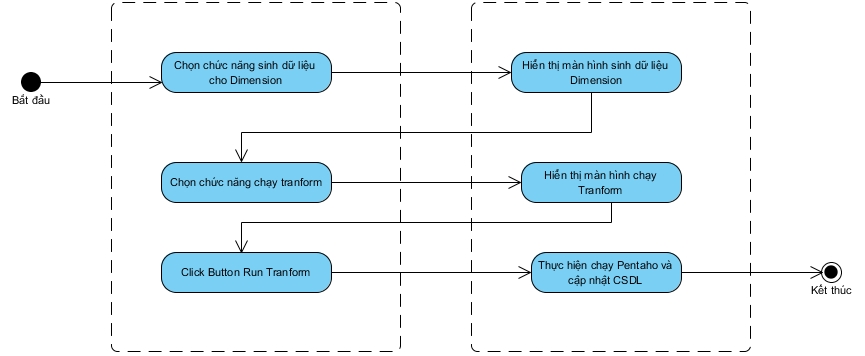


Hình 4. 7 biểu đồ hoạt động cho chức năng Xem thông tin cá nhân

1. Đặc tả Use case Cấu hình Pentaho sinh dữ liệu dimension chạy trực tiếp file tranform

Bảng 4. 3 Đặc tả Use case cấu hình pentaho chạy tranfrom

|  |  |
| --- | --- |
| Tên Use Case | Cấu hình Pentaho sinh dữ liệu dimension chạy trực tiếp file tranform |
| Tên thực thể | Sales Admin |
| Mô tả | Use Case Cấu hình Pentaho sinh dữ liệu dimension chạy trực tiếp file tranform sẽ cung cấp cho Sales admin màn hình để thực hiện việc lấy dữ liệu dimension bằng file tranform |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người dùng có quyền Sales Admin * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống sẽ sinh và cập nhật dữ liệu Dimension của DW * Thông báo thành công |
| Kịch bản | * Sales Admin chọn chức năng chạy tranform * Click button Start Tranform của dimension cần thực hiện |

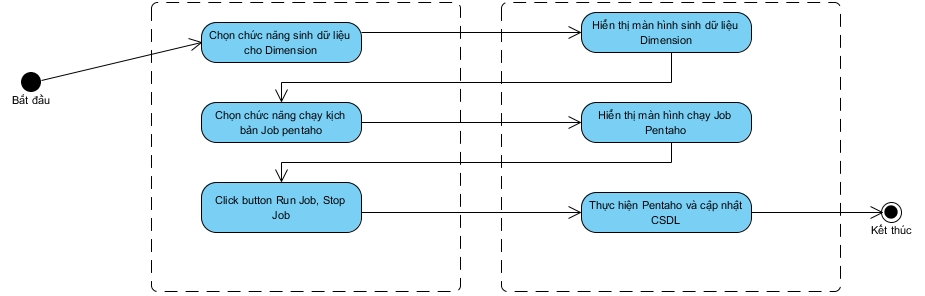


Hình 4. 8 Biểu đồ hoạt động cấu hình Pentaho sinh dữ liệu dimension

1. Đặc tả Use case Cấu hình Job Pentaho cập nhật dữ liệu Dimension DW

Bảng 4. 4 Đặc tả cấu hình Job Pentaho cập nhật dữ liệu Dimension DW

|  |  |
| --- | --- |
| Tên Use Case | Cấu hình Job Pentaho cập nhật dữ liệu Dimension DW |
| Tên thực thể | Sales Admin |
| Mô tả | Use Case Cấu hình Job Pentaho cập nhật dữ liệu Dimension DW sẽ cung cấp cho Sales admin màn hình để thực hiện việc chạy đặt lịch cập nhật dữ liệu Dimension |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người dùng có quyền Sales Admin * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống chạy Pentaho để lập lịch cập nhật dữ liệu Dimension của DW * Thông báo thành công |
| Kịch bản | * Sales Admin chọn chức năng cấu hình Job Pentaho cập nhật dữ liệu Dimension DW * Click button Run Job của dimension cần thực hiện |

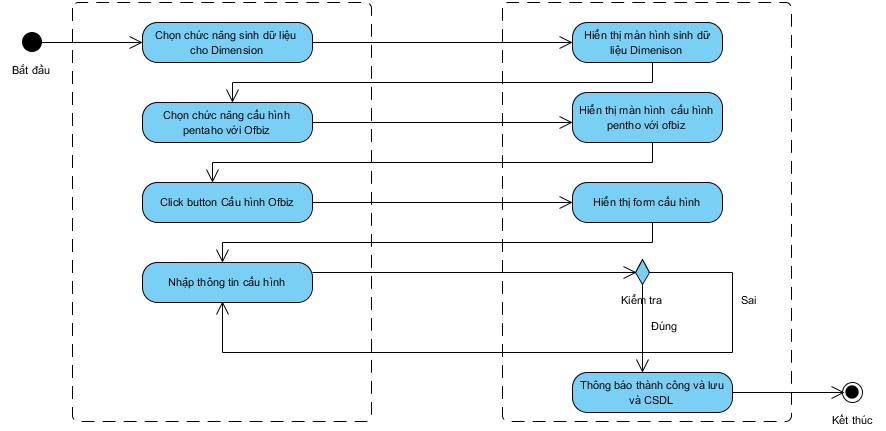


Hình 4. 9 Biểu đồ hoạt động cấu hình Job Pentaho cập nhật dữ liệu Dimension DW

1. Đặc tả Use case Cấu hình Pentaho Dimension với Ofbiz Services

Bảng 4. 5 Đặc tả Use case cấu hình Pentaho dimension với Ofbiz

|  |  |
| --- | --- |
| Tên Use Case | Cấu hình Pentaho Dimension với Ofbiz Services |
| Tên thực thể | Sales Admin |
| Mô tả | Use Case Cấu hình Pentaho Dimension với Ofbiz Services sẽ cung cấp cho Sales admin màn hình Form để thực hiện việc lập lịch chạy file Tranform của pentaho dựa vào tiện ích có sẵn của Ofbiz |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người dùng có quyền Sales Admin * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống cập nhật CSDL và thông báo thành công * Hệ thống sẽ thông báo lỗi nếu không thành công |
| Kịch bản | * Sales Admin chọn chức năng cấu hình Job Pentaho cập nhật dữ liệu Dimension DW * Click button Cấu hình Ofbiz của dimension cần thực hiện * Nhập các thông tin vào form cấu hình |

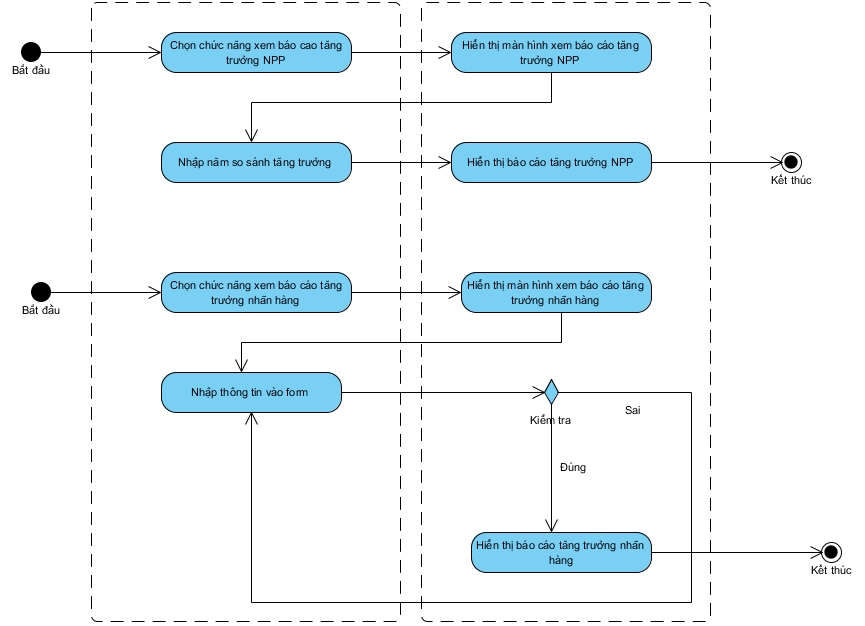


Hình 4. 10 Biểu đồ hoạt động cấu hình Pentaho Dimension với Ofbiz Services

1. Đặc tả Use case Xem báo cáo tăng trưởng NPP, Nhãn hàng

Bảng 4. 6 Đặc tả use case xem báo cáo tăng trưởng

|  |  |
| --- | --- |
| Tên Use Case | Xem báo cáo tăng trưởng NPP |
| Tên thực thể | Người dùng |
| Mô tả | Use Case Xem báo cáo tăng trưởng NPP sẽ cung cấp cho người dùng màn hình Form để thực hiện việc xem báo cáo tăng trưởng của các nhà phân phối hoặc nhãn hàng |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người dùng có quyền xem báo cáo tăng trưởng * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống hiển thị màn hình báo cáo * Hệ thống thông báo lỗi nếu thông tin nhập vào sai |
| Kịch bản | Người dùng chọn chức năng:   * Báo cáo tằng trưởng NPP: * Nhập năm xem báo cáo. * Báo cáo tăng trưởng nhãn hàng: + Nhập nhãn hàng và thời gian |

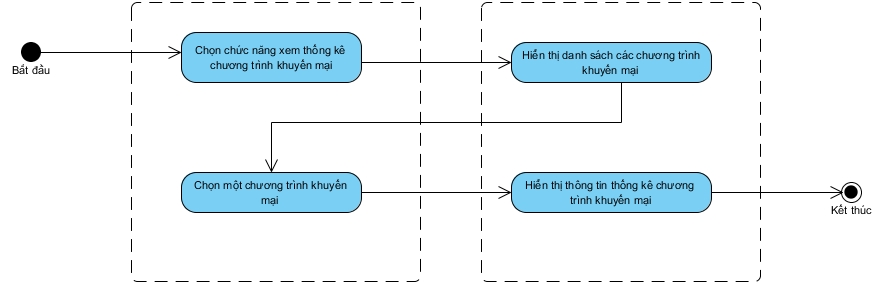


Hình 4. 11 Biểu đồ hoạt động Xem báo cáo tăng trưởng NPP, Nhãn hàng

1. Đặc tả Use case Xem thống kê chương trình khuyến mại

Bảng 4. 7 Đặc tả use case xem thống kê chương trình khuyến mại

|  |  |
| --- | --- |
| Tên Use Case | Xem thống kê chương trình khuyến mại |
| Tên thực thể | Sales Admin |
| Mô tả | Use Case Xem thống kê chương trình khuyến mại sẽ cung cấp cho Sales Admin màn hình để xem báo thống kê chương trình khuyến mại |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người dùng có quyền Sales Admin * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống hiển thị màn hình thống kê |
| Kịch bản | * Người dùng chọn chức năng xem thống kê chương trình khuyến mại * Chọn một chương trình Khuyến mại cụ thể |

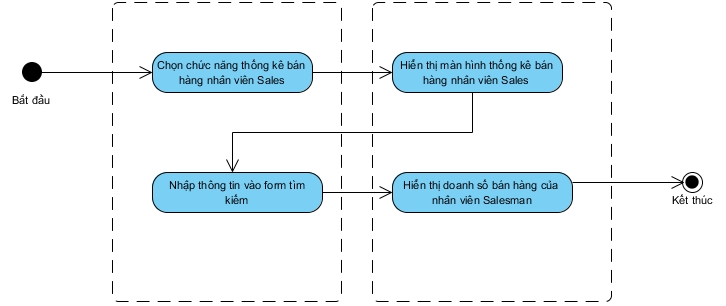


Hình 4. 12 Biểu đồ hoạt động Xem thống kê chương trình khuyến mại

1. Đặc tả use case Xem doanh số bán hàng của Sales man

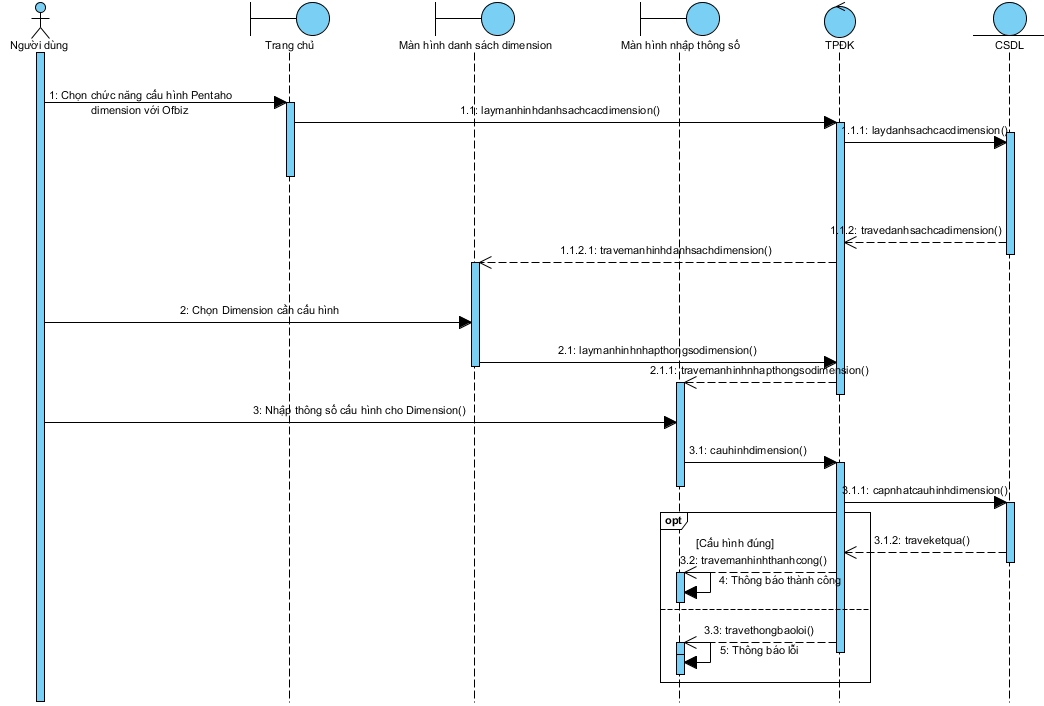
Bảng 4. 8 Đặc tả use case xem doanh số bán hàng Sales man

|  |  |
| --- | --- |
| Tên Use Case | Xem doanh số bán hàng của Sales man |
| Tên thực thể | Trưởng phòng SUP |
| Mô tả | Use Case Xem thống kê chương trình khuyến mại sẽ cung cấp cho Sales Admin màn hình để xem báo thống kê chương trình khuyến mại |
| Tiền điều kiện | * Người dùng đã có tài khoản trong hệ thống * Người là trưởng phòng SUP * Người dùng đăng nhập vào hệ thống |
| Hậu điều kiện | * Hệ thống hiển thị màn hình thống kê bán hàng Sales man * Thông báo lỗi nếu thực hiện không thành công |
| Kịch bản | * Chọn phòng ban mình quản lý * Chọn thời gian xem thống kế |

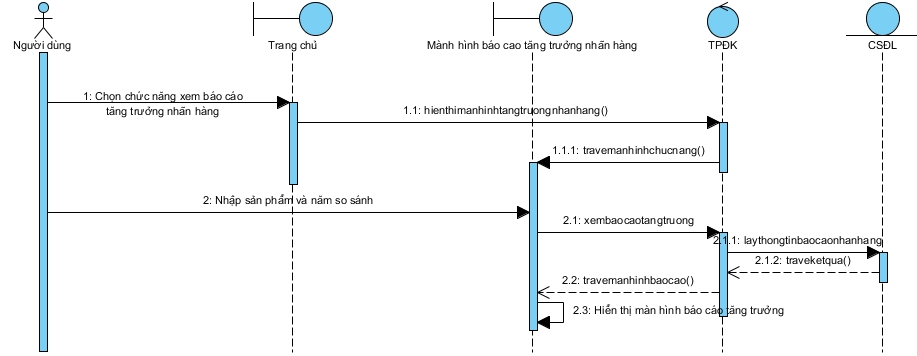


Hình 4. 13 Biểu đồ hoạt động Xem doanh số bán hàng của Sales man

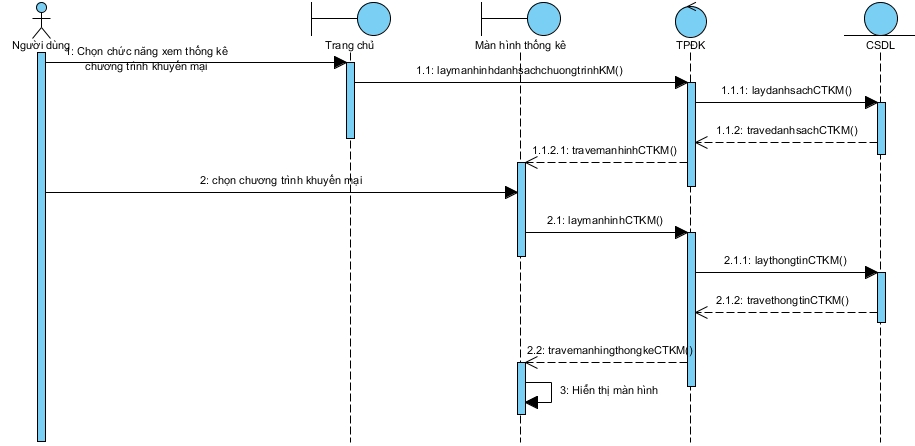
* + 1. **Biểu đồ trình tự**



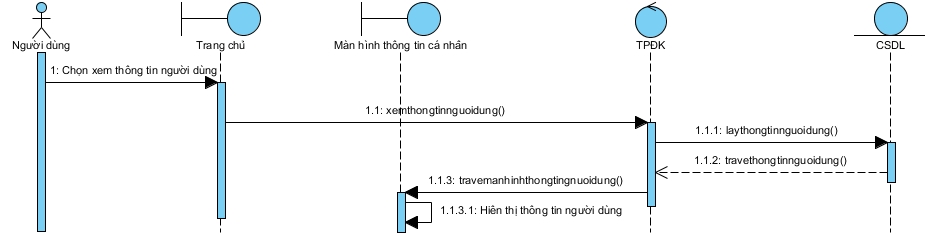
Hình 4. 14 Biểu đồ trình tự cấu hình Pentaho Dimension với Ofbiz Services



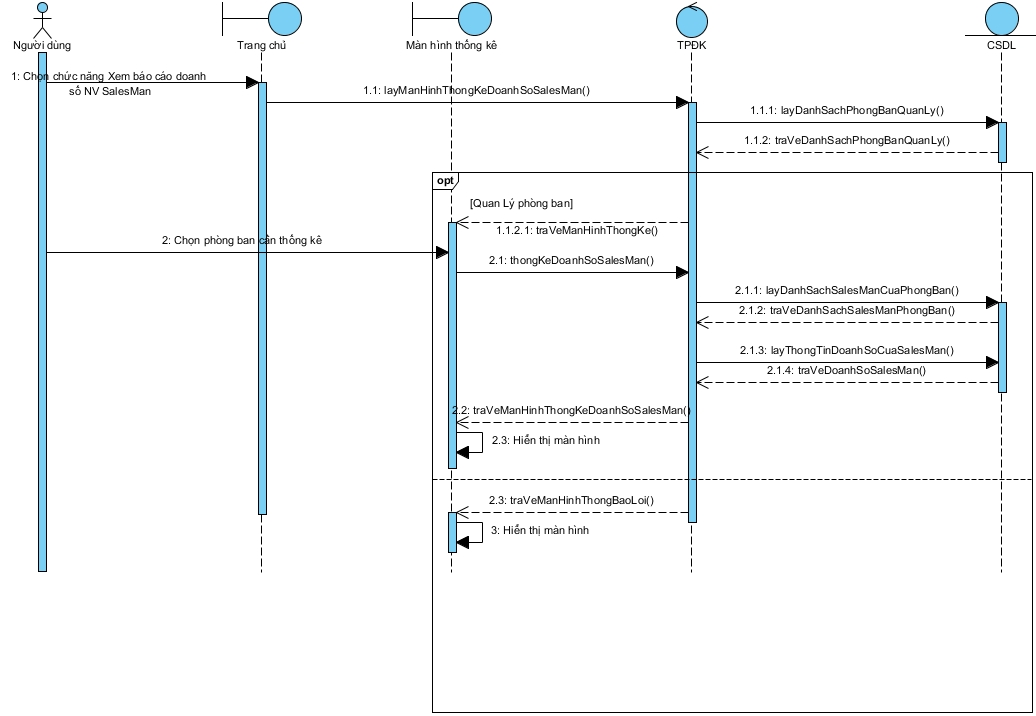
Hình 4. 15 Biểu đồ tình tự xem báo cáo tăng trưởng nhãn hàng



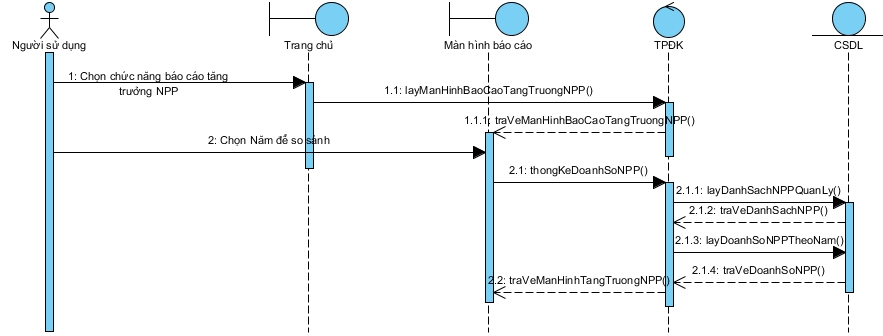
Hình 4. 16 Biểu đồ trình tự xem thống kê chương trình khuyến mại



Hình 4. 17 Biểu đồ trình tự xem thông tin người dùng

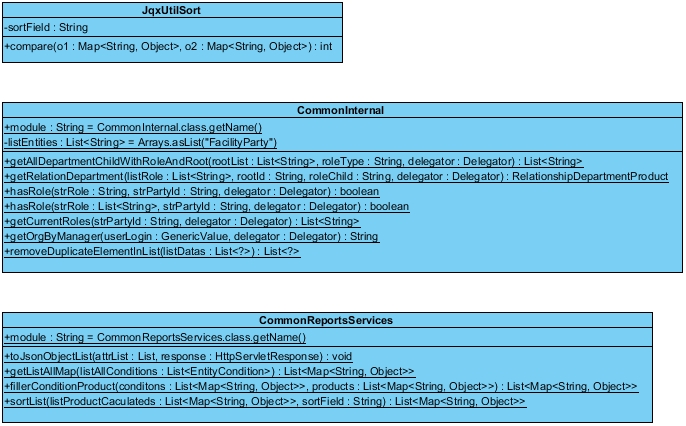


Hình 4. 18 Biểu đồ tuần tự xem thống kê doanh số bán hàng nhân viên SalesMan

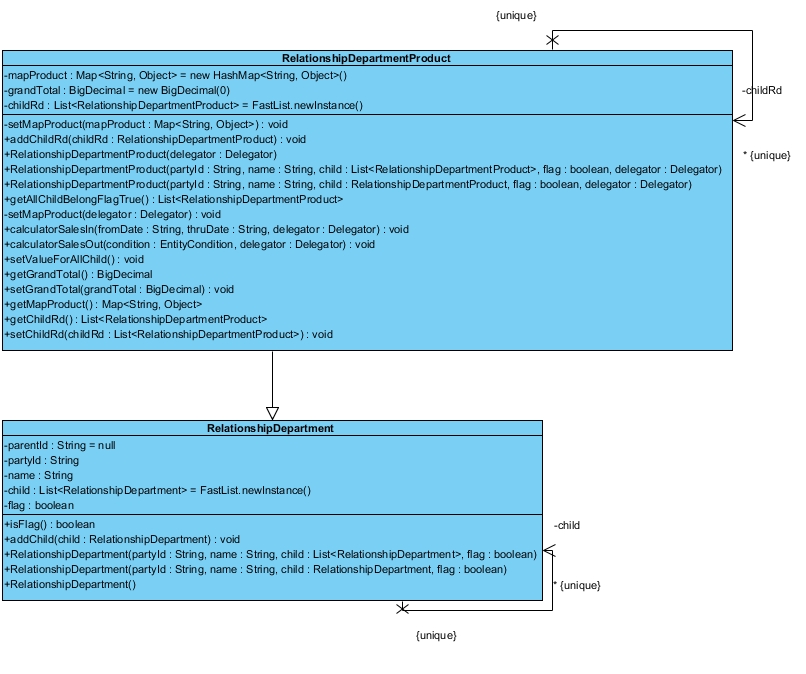


Hình 4. 19 Biểu đồ tuần tự báo cáo tăng trưởng Nhà Phân Phối

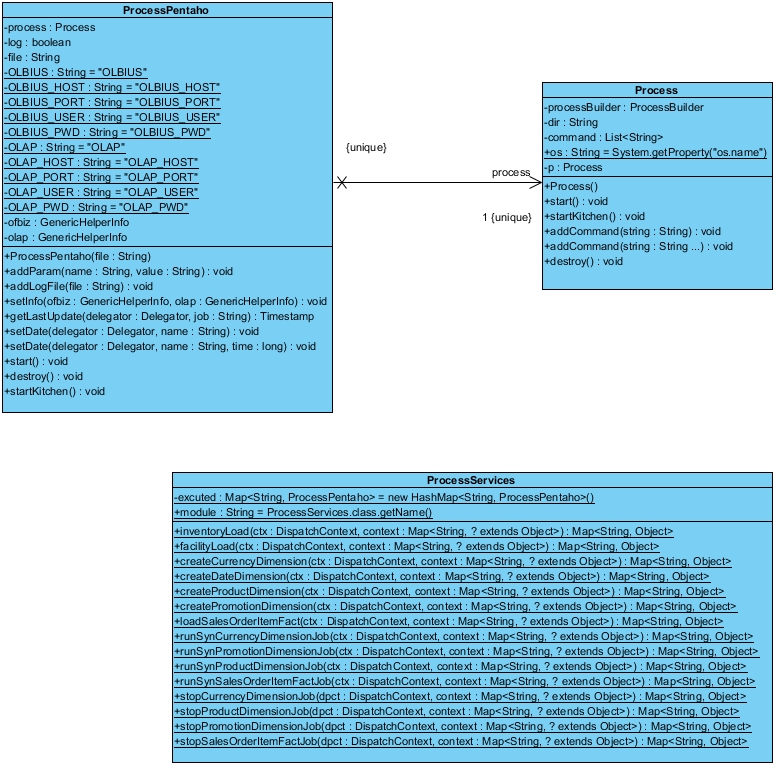
* 1. **Thiết kế hệ thống**
     1. **Thiết kế lớp**



Hình 4. 20 Các lớp dùng chung



Hình 4. 21 Các lớp cấu trúc tổ chức phòng ban

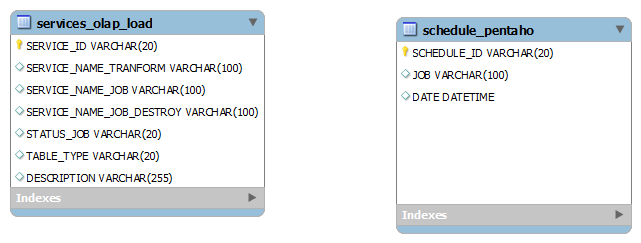


Hình 4. 22 Các lớp xử lý Pentaho

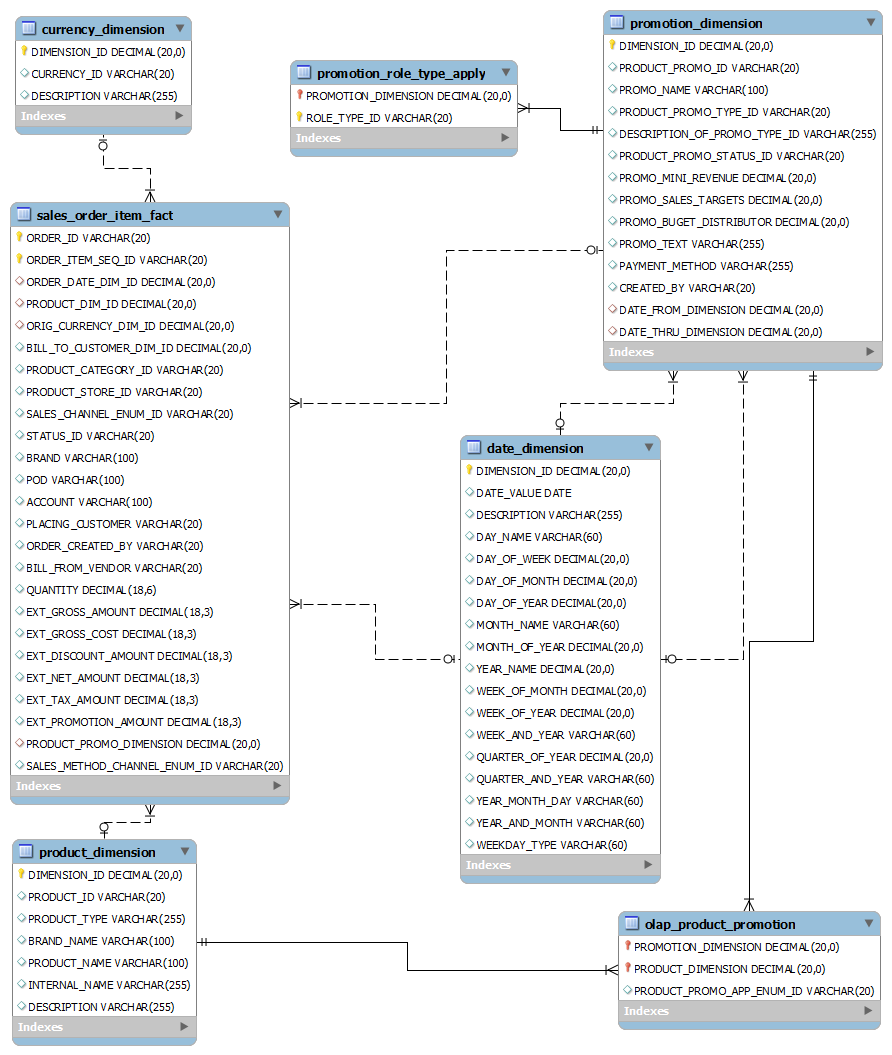
Bảng 4. 9 Chi tiết các lớp

|  |  |  |
| --- | --- | --- |
| STT | Tên lớp | Chức năng |
|  | JqxUtilShort | Sắp xếp các trường trong một cột của JQX |
|  | CommonInternal | Lớp xử lý với các phòng ban |
|  | CommonReportService | Lớp dùng chung trong các service |
|  | RelationshipDepartmentReport | Lớp sử dụng để tính toán, thống kê doanh số sản phẩm theo quan hệ tổ chức của PartyRelationship |
|  | RelationshipDepartment | Lớp xây dựng tổ chức của PartyRelationship |
|  | Process | Lớp làm việc với comman line trong hệ điều hành để diều khiển pentaho |
|  | ProcessPentaho | Lớp cấu hình các tham số pentaho |
|  | ProcessService | Lớp chứa các service chức năng sử dụng pentaho |

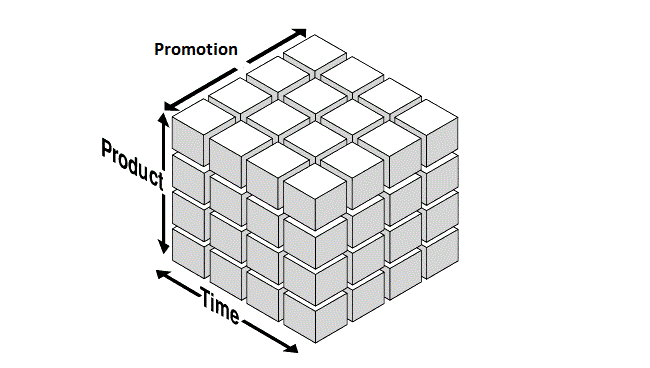
* + 1. **Thiết kế CSDL**



Hình 4. 23 Các bảng cấu hình pentaho



Hình 4. 24 Cấu trúc các bảng trong DW



Hình 4. 25 Thể hiện đa chiều của kho dữ liệu

|  |  |  |
| --- | --- | --- |
| Nhóm tính năng | Tên bảng | Mô tả |
| Cấu hình | Service\_olap\_load | Lưu trữ thông tin về các Sevices cấu hình DW   * ServiceId: Id của Service * Service\_name\_tranform: Tên service chạy file tranform trong pentaho * Service\_name\_job: Tên service chạy file job trong pentaho * Service\_name\_job\_destroy: Tên service tạm dừng việc chạy Job trong pentaho * Status\_job: trạng thái Service job đang chạy hay dừng * Table\_type: Kiểu dữ liệu liên quan đến bảng Fact hay Dimension * Description: mô tả |
| Schedule\_pentaho | Lưu trữ mốc thời gian cuối cùng chạy Service sinh và cập nhật dữ liệu.   * Schedule\_id: Id của bảng * Job: tên công việc * Date: thời gian sử dụng Service |
| Data Warehouse Dimension | Currency\_dimension | Lưu trữ chiều tiền tệ sử dụng.   * dimension\_id: Id của chiều tiền tệ * currency\_id: id của tiền tệ * description: Mô tả |
| Date\_dimension | Lưu trữ chiều thời gian.   * dimension\_id: id của chiều thời gian * date\_value: giá trị thời gian lưu dưới dạng yy-mm-dd * discription: mô tả * day\_name: tên ngày trong tuần * day\_of\_week: vị trí ngày trong tuần (1-7) * day\_of\_month: vị trí ngày trong tháng (1-31) * day\_of\_year: vị trí ngày trong năm (1-366) * month\_name: tên tháng * month\_of\_year: vị trí tháng trong năm (1-12) * year\_name: tên năm * week\_of\_month: vị trí tuần trong tháng * week\_of\_year: vị trí tuần trong năm * quarter\_of\_year: vị trí quý trong năm (1-4) * year\_month\_day: định dạng giá trị date\_value theo kiểu yy-mm-dd * year\_and\_month: định dạng năm và tháng theo kiểu yy-mm * weekday\_type: kiểu ngày trong tuần (weekday hoặc weekend) |
| Promotion\_dimension | Lưu trữ thông tin chiều chương trình khuyến mại.   * dimension\_id: id chiều chương trình khuyến mại * product\_promo\_id: id chương trình khuyến mại * promo\_name: chương trình khuyến mại * product\_promo\_type\_id: Kiểu chương trình khuyến mại * description\_of\_promo\_type\_id: mô tả kiểu chương trình khuyến mại * product\_promo\_status\_id: trạng thái chương trình khuyến mại * promo\_sales\_target: mục tiêu đạt được trong chương trình khuyến mại. * promo\_budget\_distributor: ngân sách phân phối cho chương trình khuyến mại * promo\_text: Thông tin chương trình khuyến mại * created\_by: người tạo chương trình khuyến mại * date\_from\_dimension: Giá trị Id chiều thời gian ngày bắt đầu của chương trình khuyến mại. * date\_thru\_dimension: Giá trị Id chiều thời gian ngày kêt thúc của chương trình khuyến mại |
| Promotion\_role\_type\_apply | Lưu trữ đối tượng được áp dụng chương trình khuyến mại.   * Promotion\_dimension: Id chiều chương trình khuyến mại * Role\_type\_id: kiểu đối tượng áp dụng |
| Olap\_product\_promotion | Lưu trữ sản phẩm được áp dụng KM trong chương trình Khuyến mại   * Promotion\_dimension: Id chiều chương trình khuyến mại * Product\_dimension: chiều sản áp dụng phẩm khuyến mại * Product\_promo\_app\_enum\_id: |
| Product\_dimension | Lưu trữ thông tin chiều sản phẩm   * Dimension\_id: Id chiều sản phẩm * Product\_id: Id sản phẩm * Product\_type: Kiểu sản phẩm * Brand\_name: nhãn hàng * Product\_name: tên sản phẩm * Internal\_name: tên gọi sản phẩm trong nội bộ * Description: mô tả |
| Data warehouse Fact | Sales\_order\_item\_fact | Lưu trữ thông tin về sự kiện của Order item   * order\_id: Id của order * order\_item\_seq\_id: Mã item trong order * order\_date\_dim\_id:giá trị id chiều thời gian của ngày tạo order * product\_promotion\_dimension: giá trị id của chiều khuyến mại * product\_dim\_id: giá trị id chiều sản phẩm của order\_item * orig\_currency\_dim\_id: giá trị id chiều tiền tệ của tiền tệ sử dụng trong Order item * product\_store\_id: mã cửa hàng bán sản phẩm * sales\_chanel\_enum\_id: kênh bán hàng * status\_id: trạng tháu đơn hàng * brand: nhãn hàng * placing\_customer: người đặt hàng * order\_created\_by: người tạo đơn hàng * Bill\_from\_vendor: người, tổ chức xuất hóa đơn * Quantity: số lượng sản phẩm * Ext\_gross\_amount: giá trị đơn hàng = quantiy\*price * Ext\_gross\_cost: giá trị nhập hàng = cost\*quantity * Ext\_discount\_amont: giá trị khấu trừ trên đơn hàng * Ext\_promotion\_acount: giá trị khuyến mại trên đơn hàng * Ext\_net\_amount: giá trị thực tế đơn hàng= ext\_gross\_amount + ext\_discount\_amount + Ext\_promotion\_acount * Ext\_tax\_amount: giá trị tiền thuế của item |

Bảng 4. 10 Chi tiết các bảng trong CSDL

# **CHƯƠNG 5: CÀI ĐẶT VÀ THỬ NGHIỆM HỆ THỐNG**

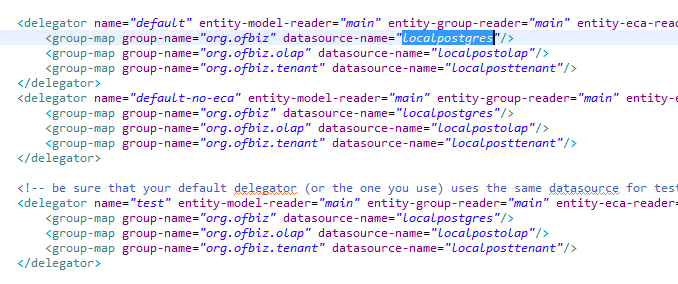
* 1. **Môi trường cài đặt**

Chương trình được cài đặt trên môi trường:

* Web: Localhost.
* Hệ điều hành: Windows 8.0 Professional.
* Ngôn ngữ lập trình: java, Jquery, JavaScript, HTML.
* Hệ cơ sở dữ liệu: Postgres.
  1. **Cài đặt và thử nghiệm**
     1. **Cài đặt ofbiz**

1. Sau khi đã tải thành công ofbiz về máy, ta cần phải tạo cơ sở dữ liệu cho ofbiz để có thể hoạt động:

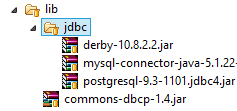
* Bước 1: Tạo cơ sở dữ liệu hoạt động cho ofbiz và khai báo tại đường dẫn: ofbiz/framework/entity/config/entityengine.xm.





Hình 2. 16 Cấu hình CDSL trong Ofbiz

Ofbiz sử dụng jdbc để kết nối với cơ sở dữ liệu, vì vậy khi làm việc với hệ quản trị cơ sở dữ liệu nào thi cần phải import thư viện jdbc của thư viện đó tại: ofbiz/framework/entity/lib/jdbc:



Hình 2. 17 Lưu trữ driver jdbc trong Ofbiz

* Bước 2: dùng lệnh trên cửa sổ command line để build project cho lần đầu chạy hoặc có bất cứ thay đổi nào đối với các file java trong ofbiz đều phải thực hiện lệnh build để thực hiện việc thay đổi:
  + Từ của sổ command line, di chuyển đến thư mục chứa ofbiz: *cd ofbiz*
  + Sử dụng lệnh: *ant build*
* Bước 3: Chạy ofbiz:
  + Sử dụng lệnh: ant start
  + Lúc này có thể truy cập vào ofbiz trên đường trình duyệt với đường dẫn: <https://localhost:8080/control/main>

b) Tạo mới một component trong ofbiz

- Từ của sổ command line, di chuyển đến thư mục chứa ofbiz: *cd ofbiz*

- Sử dụng lệnh: ./ant create-component. Sau đó điền các thông số

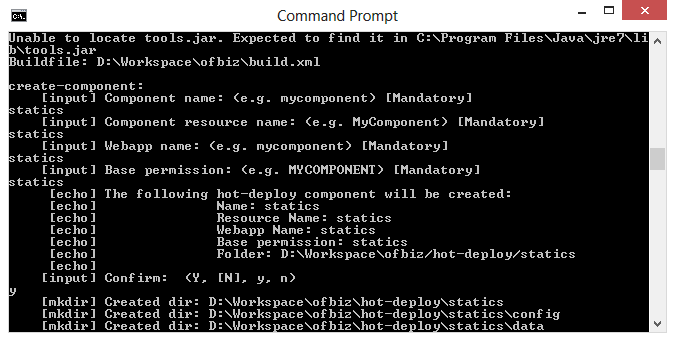
Component name: **statics (***Tên component***)**

Component resource name: **statics** ()

Webapp name: statics (**Tên đường dẫn sẽ được sử dụng để truy cập đến component: https://locahost:8080/hwm**)

Base permission: STM (Quyền được truy cập đến component)

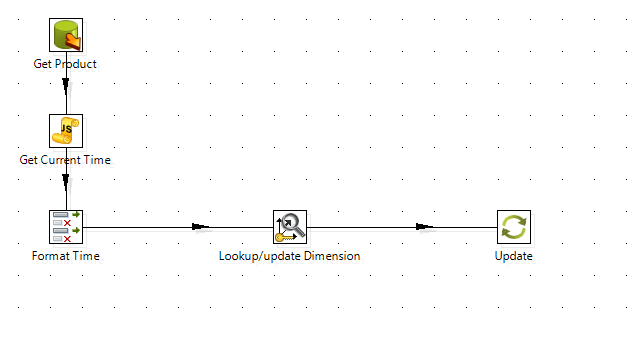
Confirm: Y



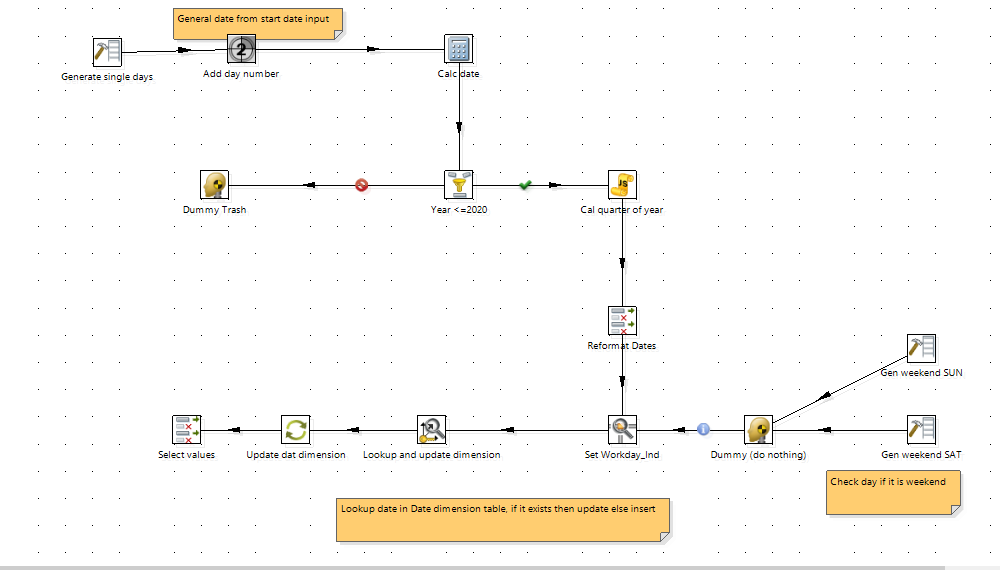
Hình 2. 18 Tạo component trong Ofbiz

* Sau khi tạo thành công, component sẽ được tạo tại đường dẫn: ofbiz/ hot-deploy.
  + 1. **Chức năng ETL trong Data warehouse qua pentaho**

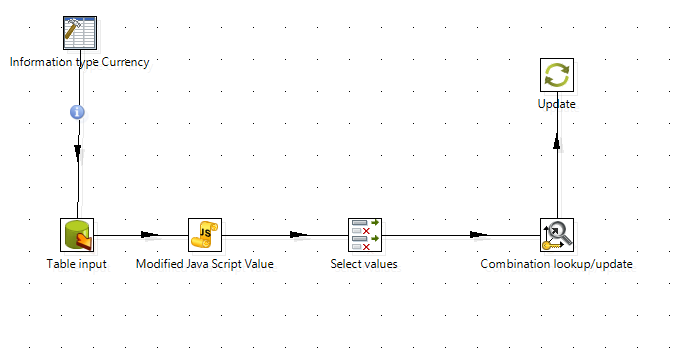
1. Kịch bản file tranform trong pentaho



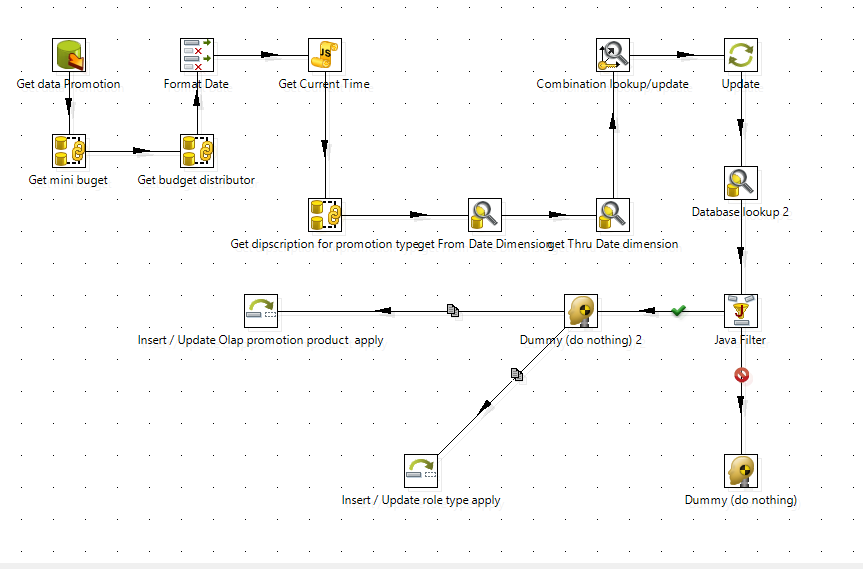
Hình 5. 1 Kịch bản tạo dữ liệu chiểu product



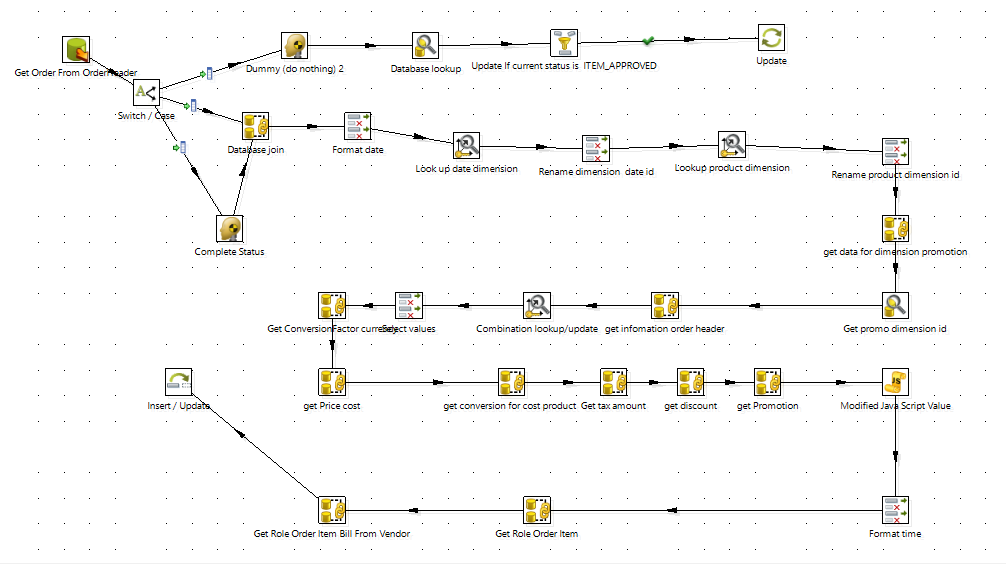
Hình 5. 2 Kịch bản tạo dữ liệu chiều thời gian



Hình 5. 3 Kịch bản tạo dữ liệu chiều tiền tệ

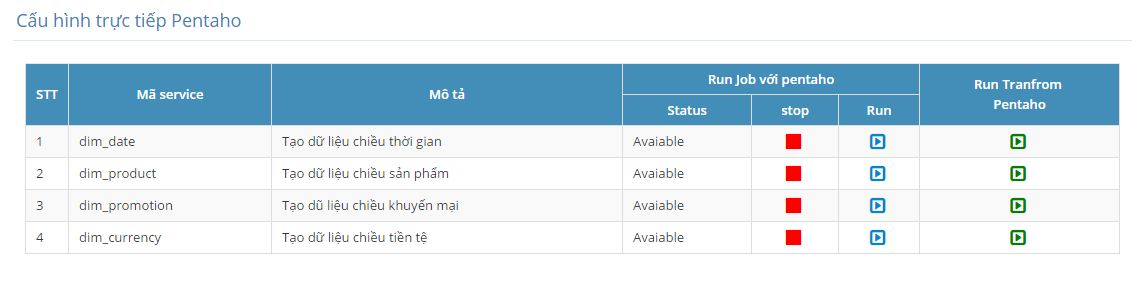


Hình 5. 4 Kịch bản tạo dữ liệu chiều chương trình Khuyến mại

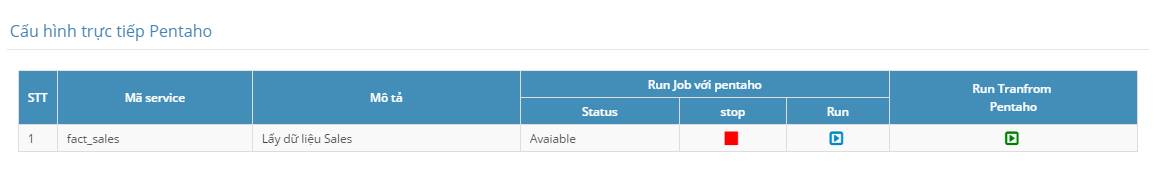


Hình 5. 5 Kịch bản tọa dữ liệu cho bảng fact Sales

1. Giao diện người dùng

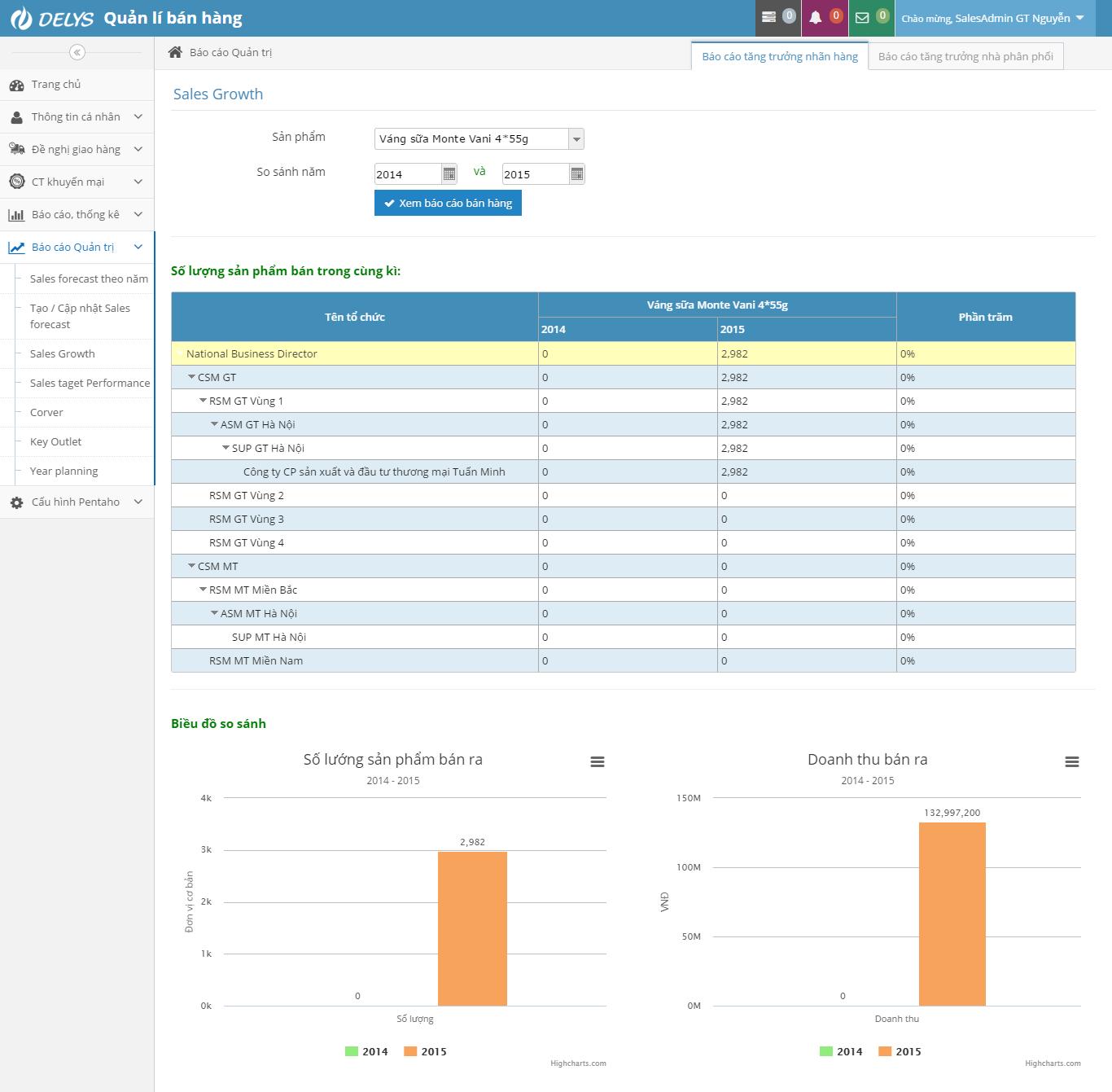


Hình 5. 6 Giao diện cấu hình Pentaho cho dimension



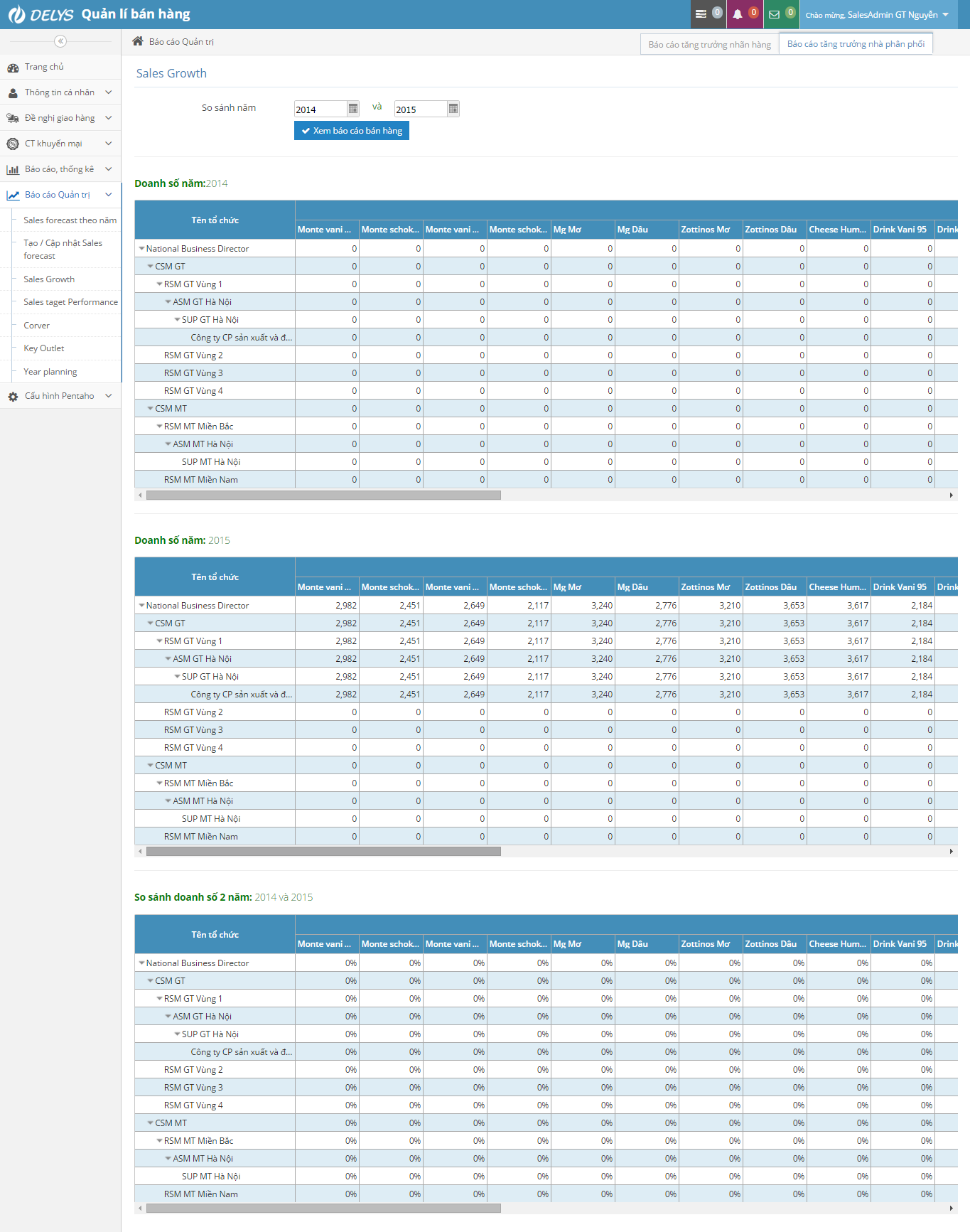
Hình 5. 7 Giao diện cấu hình Pentaho cho bảng fact

* + 1. **Báo cáo tăng trưởng nhãn hàng**



Hình 5. 8 Giao diện báo cáo tăng trưởng nhãn hàng

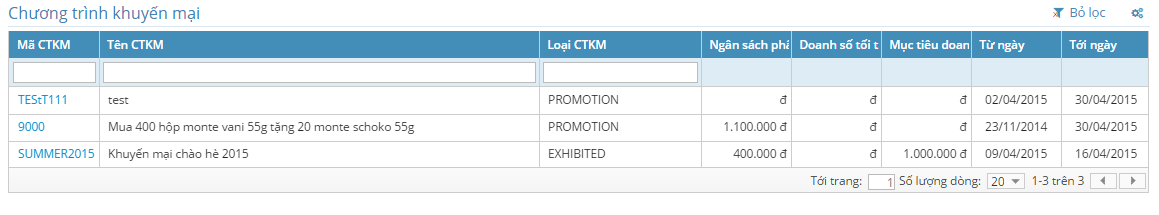
* + 1. **Báo cáo tăng trưởng nhà phân phối**



Hình 5. 9 Báo cáo tăng trưởng nhà phân phối

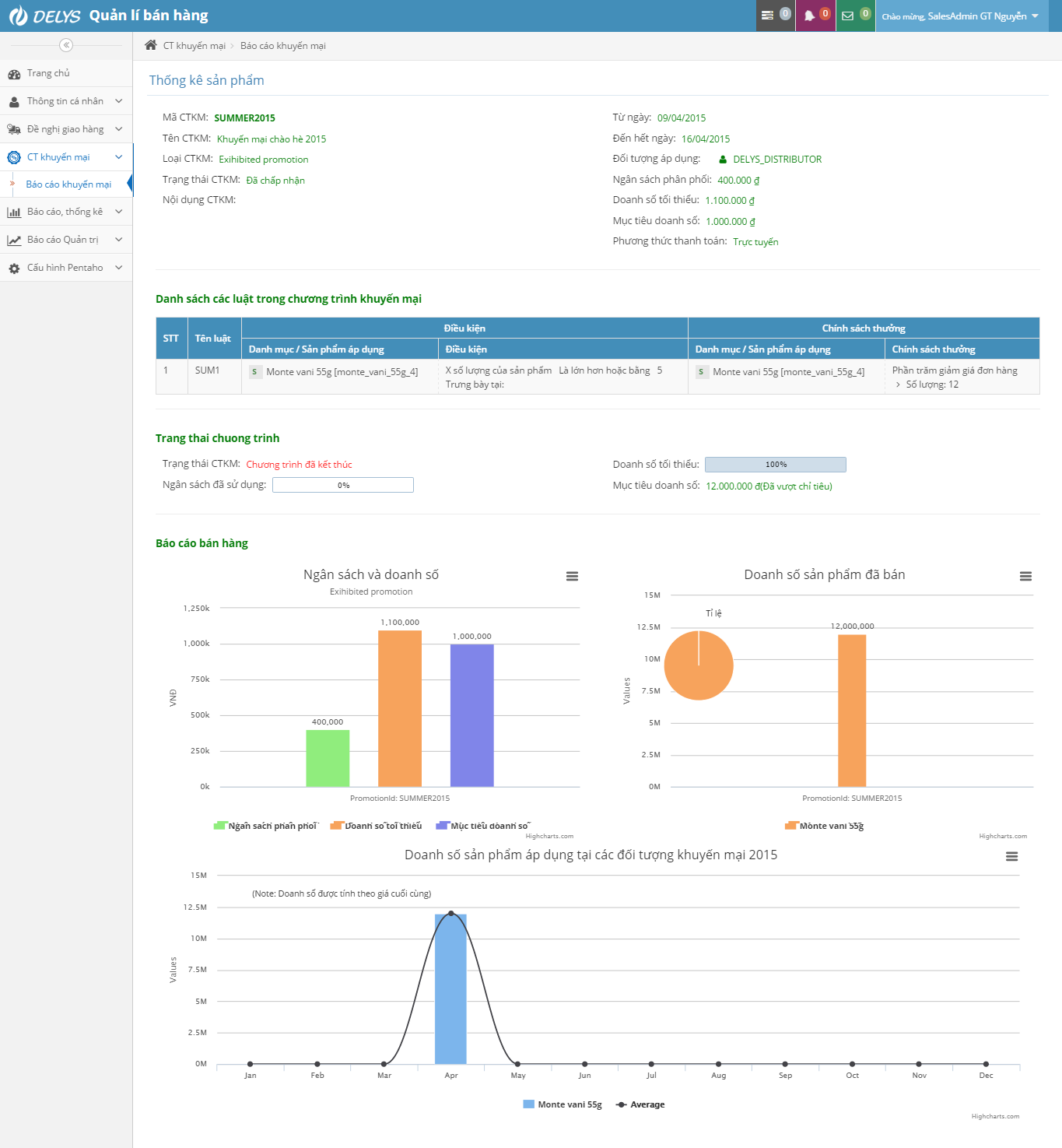
* + 1. **Báo cáo chương trình khuyến mại**

1. Danh sách chương trình khuyến mại



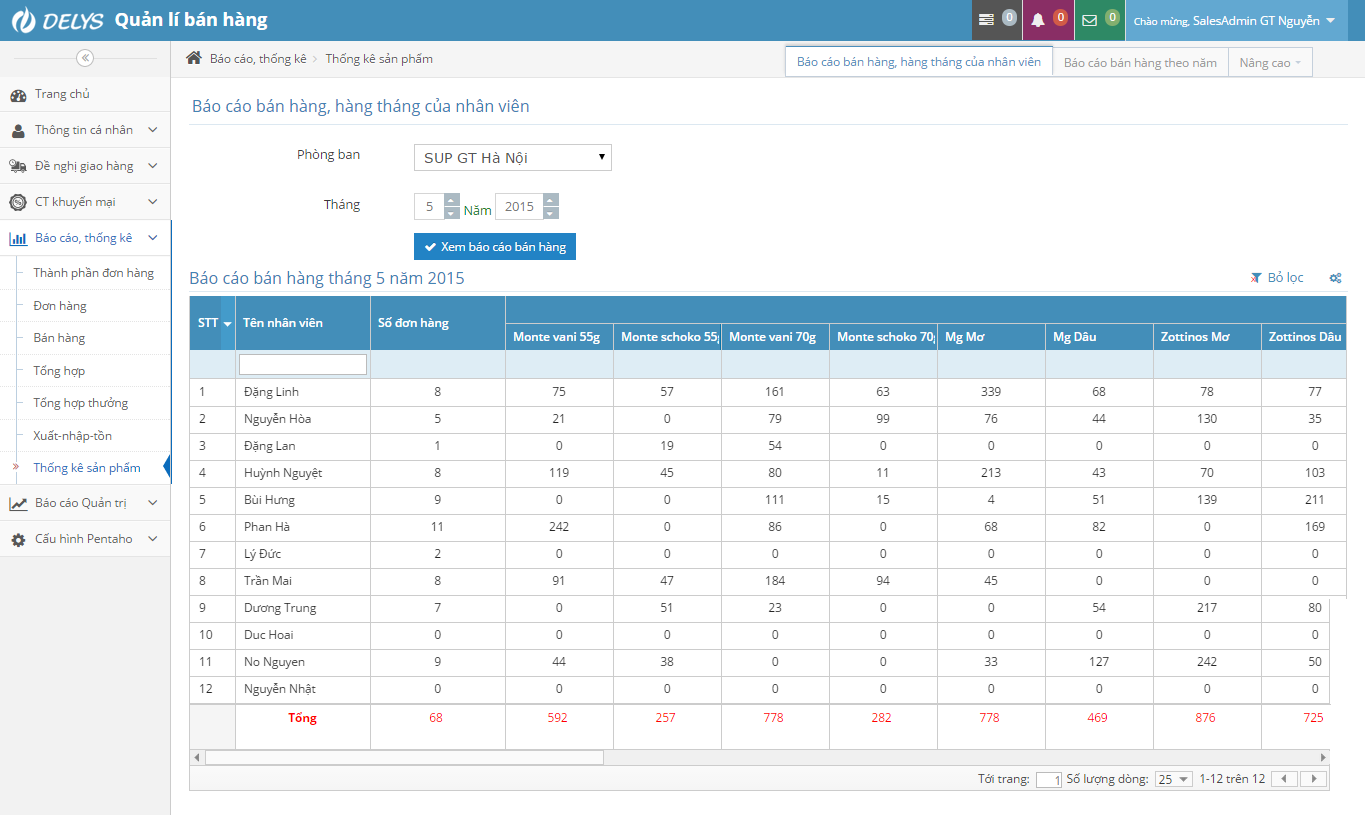
Hình 5. 10 Giao diện danh sách chương trình khuyến mại

1. Chi tiết chương trình khuyến mại

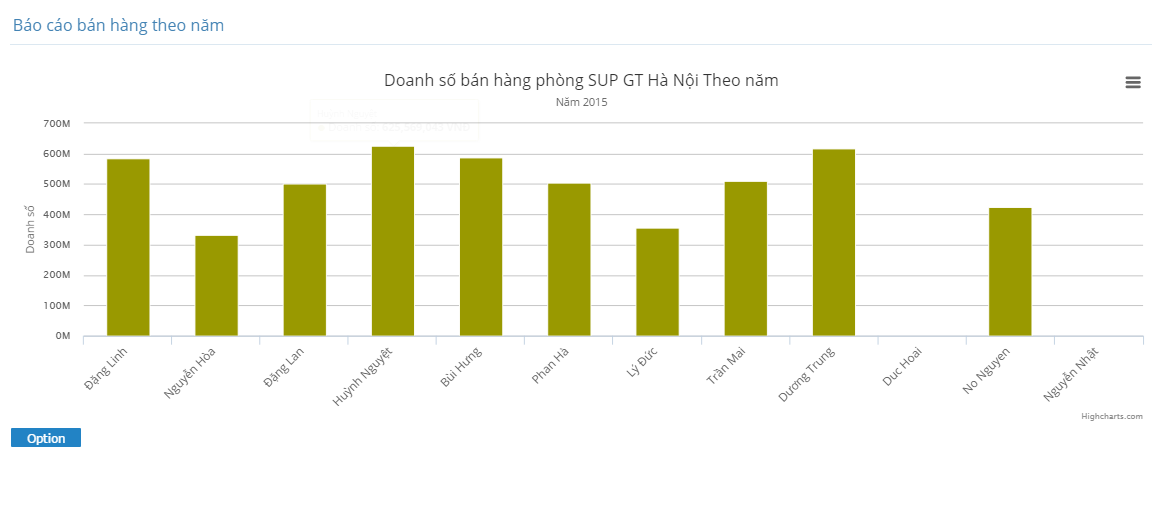


Hình 5. 11 Giao diện chi tiết báo cao chương trình khuyến mại

* + 1. **Báo cáo doanh số bán hàng nhân viên**



Hình 5. 12 Giao diện báo cáo bán hàng của nhân viên



Hình 5. 13 Biểu đồ doanh số bán hàng nhân viên

# **CHƯƠNG 6: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN**

* 1. **Kết luận**

Đồ án đã đạt được những mục tiêu cơ bản được đặt ra trước khi thực hiện:

* Hiểu được cấu trúc kho dữ liệu, luồng hoạt động nghiệp vụ bán hàng trong ofbiz
* Xây dựng hệ thống Data warehouse có đầy đủ các bước thực hiện
* Xây dựng được kịch bản cho pentaho
* Đưa ra hệ thống báo cáo sinh động hỗ trợ người sử dụng

Bên cạnh đó, đồ án còn mắc phải một số thiếu sót:

* Kịch bản chạy pentaho còn chưa tối ưu do chưa có nhiều kinh nghiệm
* Số lượng báo cáo còn ít do hạn chế về thời gian
* Do kinh nghiệm của bản thân không cao nên các nghiệp vụ trong hệ thống còn tương đối chậm.
* Giao diện đôi chỗ còn chưa được thân thiện.
  1. **Hướng phát triển**

Do khả năng thực hiện cũng như khả năng còn hạn chế nên hệ thống cần được bổ sung và nâng cấp nhiều hơn. Sau đây đồ án xin đề xuất một số hướng để phát triển hệ thống:

* Bổ sung một số chiều dữ liệu trong hệ thống như: Vị trí địa lý, Khách hàng….
* Áp dụng một số thuật toán để hỗ trợ người dùng đưa ra quyết định
* Xây dựng giao diện đẹp mắt hơn, thân thiện hơn

# **TÀI LIỆU THAM KHẢO**

1. Apache OFBiz Development: The Beginner's Tutorial - Rupert Howell, Jonathon Wong

2. Apache OFBiz Cookbook - Ruth Hoffman

3. Trang chủ ofbiz: [*ofbiz.apache.org*](https://ofbiz.apache.org/)

4. Phân tích và thiết kế hệ thống thông tin – Nguyễn Văn Ba

5. Pentaho Data Integration 4 Cookbook - Adrián Sergio Pulvirenti,María Carina Roldán

6. Diễn đàn Business Intelligence Solutions: [*bsdinsight.com*](bsdinsight.com)

*7.* The Data Model Resource Book - Len Silverston

8. Kho dữ liệu và Hệ hỗ trợ quyết định – Nguyễn Thanh Bình (Đại học huế)

9. The Data Warehouse Toolkit Second Edition- Kimball

10. Định nghĩa kho dữu liệu : [http://vi.wikipedia.org/wiki/Kho\_d%E1%BB%AF\_li%E1%BB%87u#cite\_note-1](http://vi.wikipedia.org/wiki/Kho_d%E1%BB%AF_li%E1%BB%87u%23cite_note-1%20)

11. Một số diễn đàn trên internet khác…

# **PHỤ LỤC**

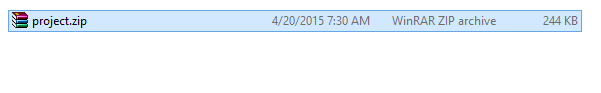
1. **Hướng dẫn cài đặt chương trình**
2. **Công cụ:**

* Eclipse
* Hệ quản trị cơ sở dữ liệu Postgres,..

1. **Các bước cài đặt**

*Bước 1: quản lý mã nguồn trên IDE Eclipse*

* Giải nén file project.zip được thư mục chứa toàn bộ source của chương trình



* Sử dụng Eclipse: File-> Import project-> chọn đường dẫn đên thư mục chứa source code.

*Bước 2: Cấu hình Ofbiz*

* Tạo cơ sở dữ liệu cho ofbiz: cần tạo 2 cở sở dữ liệu cho ofbiz bao gồm cơ sở dữ liệu hoạt động và cơ sở dữ liệu của kho dữ liệu (OLAP), bắt buộc trong ofbiz các cơ sở dữ liệu phải sử dụng mật khẩu.
* Cấu hình thông số cơ sở dữ liệu cho Ofbiz: tại đường dẫn ../ofbiz/framework/entity/config/entityengine.xml
* Khai báo 2 cơ sở dữ liệu mới tạo trong Ofbiz: Khi sử dụng hệ quản trị cơ sở nào sẽ phải khai báo đúng thẻ được quy định trong file entityengine.xml của ofbiz. Hiện tại ofbiz hỗ trợ một số hệ quản trị CSDL như: mysql, sql server, postgres,firebird….Vi dụ: Khi sử dụng Hệ quản trị CSDL postgres sẽ phải khai báo thông số tại thẻ datasource có name=”localpostgres” :

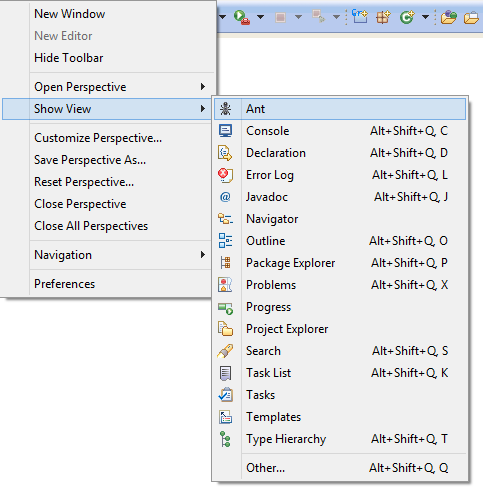


* Trỏ ofbiz đến DB sử dụng:

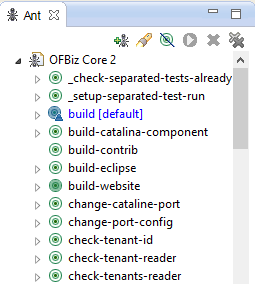
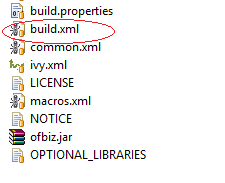


*Bước 3: Chạy dữ liệu demo*

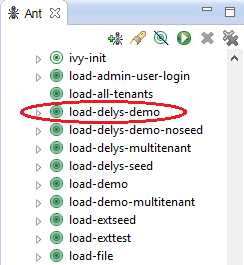
* Mở cửa sổ hiển thị file ant trong eclipse window->showview->Ant:



* Sau đó kéo file build.xml vào của sổ view Ant

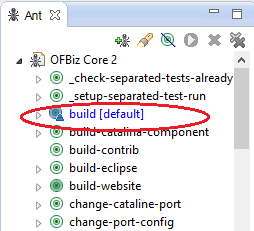


* Click đúp vào task load-delys-demo để chạy các dữ liệu sử dụng cho việc demo chương trình.

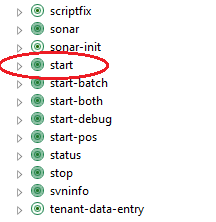


*Bước 4: Chạy Ofbiz*

* Sau khi đã thực hiện hoàn thành các bước khởi tạo dữ liệu cho Ofbiz, chúng ta thực hiện build các class trong Ofbiz bằng cách click đúp vào task : build[default].



* Sau khi đã thực hiện build thành công các class, bắt đầu chạy chương trình bằng cách: click đúp vào task Start trong file build.xml



* Truy cập tại địa chỉ <https://localhost:8443/delys/control/main> với tài khoản: salesadmingt, mật khẩu: ofbiz để truy cập hệ thống.