**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN LÝ THUYẾT

Phạm Ngọc Thùy Trang 18127022

Ngô Thị Thanh Thảo 18127053

Hoàng Nguyên Trúc 18127055

**Môn học: Hệ thống thông tin phục vụ trí tuệ kinh doanh**

**Thành phố Hồ Chí Minh**

**Ộ MÔN HỆ**

# Mục lục

[I. Mục lục 1](#_Toc88653279)

[II. Bảng thông tin chi tiết nhóm 2](#_Toc88653280)

[III. ETL từ Source sang Stage 3](#_Toc88653281)

[1. Chiến lược rút trích dữ liệu từ cơ sở dữ liệu: Incremental extract 3](#_Toc88653282)

[2. Sự thay đổi về cấu trúc dữ liệu khi đổ vào Stage 3](#_Toc88653283)

[3. Các bước khi đổ dữ liệu từ Source vào Stage 4](#_Toc88653284)

[4. Flow khi thực hiện đổ dữ liệu từ Source vào Stage trên Visual Studio 4](#_Toc88653285)

[IV. Stage sang NDS 6](#_Toc88653286)

[1. Một vài sự thay đổi dữ liệu chuyển từ giai đoạn Stage sang NDS: 7](#_Toc88653287)

[2. Mô tả flow khi dữ liệu chuyển từ giai đoạn Stage sang NDS 8](#_Toc88653288)

[3. Flow khi thực hiện đổ dữ liệu từ Stage vào NDS trên Visual Studio 9](#_Toc88653289)

[4. Lược đồ NDS dưới dạng mô hình ER 17](#_Toc88653290)

[5. Lược đồ quan hệ dưới dạng physical 20](#_Toc88653291)

[V. ETL dữ liệu từ NDS sang DDS 32](#_Toc88653292)

[1. Lược đồ DDS 32](#_Toc88653293)

[2. Phân tích yêu cầu 33](#_Toc88653294)

[3. Mô hình hoá 33](#_Toc88653295)

[4. Mô hình hoá - Fact table 34](#_Toc88653296)

[5. Mô hình hóa - Thiết kế chiều 34](#_Toc88653297)

[6. Mô hình hoá - Phân cấp dữ liệu 36](#_Toc88653298)

[7. Flow khi đổ dữ liệu từ NDS sang DDS trong Visual Studio Code 36](#_Toc88653299)

[VI. Tài liệu tham khảo 49](#_Toc88653300)

# Bảng thông tin chi tiết nhóm

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã nhóm:** | Nhóm 2 | | |
| **Số lượng:** | 3 | | |
| **MSSV** | **Họ tên** | **Email** | **SĐT** |
| 18127022 | Phạm Ngọc Thùy Trang | 18127022@  student.hcmus.edu.vn |  |
| 18127053 | Ngô Thị Thanh Thảo | 18127053@  student.hcmus.edu.vn |  |
| 18127055 | Hoàng Nguyên Trúc | 18127055@  student.hcmus.edu.vn |  |

# ETL từ Source sang Stage

Theo đồ án lý thuyết, quá trình ETL dữ liệu từ Source sang Stage sử dụng 3 source là Jade, Jupiter và Webtower.

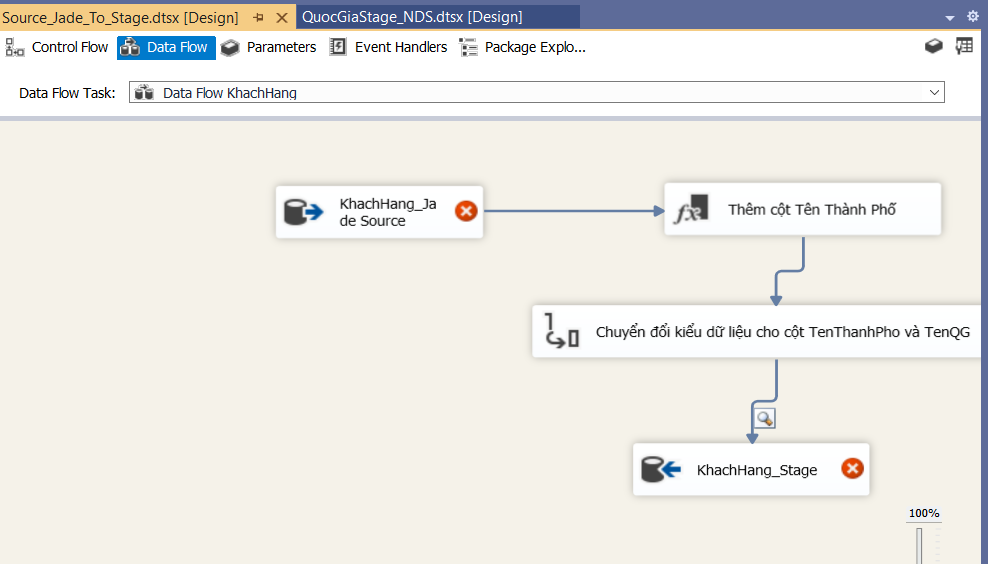
## Chiến lược rút trích dữ liệu từ cơ sở dữ liệu: Incremental extract

Cơ bản về đặc điểm của chiến lược Incremental extract là chỉ rút trích những thay đổi so với lần rút trích gần đây nhất (Dòng mới thêm, dòng mới xoá, dòng được cập nhật) dựa vào thuộc tính nhãn thời gian -> Trong đồ án này sử dụng hai nhãn thời gian là NgayTao (ngày tạo) và NgayCapNhat (ngày cập nhật mới nhất). Mỗi khi có một dòng thay đổi (update, delete) thì cập nhật lại nhãn thời gian.

## Sự thay đổi về cấu trúc dữ liệu khi đổ vào Stage

Khi đổ dữ liệu vào nguồn có sự thay đổi về cấu trúc dữ liệu ở hai bảng KhachHang\_Jade\_Stage và KhachHang\_Webtower9\_Stage:

* Tách cột DiaChi thành 2 cột ThanhPho va QuocGia, khi đổ dữ liệu từ bảng source vào trong bảng stage. Chẳng hạn, với dòng dữ liệu có MaKH là ‘008V’ thì ở cột DiaChi là ‘Kazakhstan - Alabama’, sau khi chuyển từ giai đoạn source sang stage thì phần sau “-” là ‘Alabama’ được bỏ vô cột ThanhPho, phần nằm trước dấu “-” thì nằm trong cột QuocGia là “Kazakhstan”. Dưới đây là flow của tách cột DiaChi thành 2 cột ThanhPho và QuocGia.

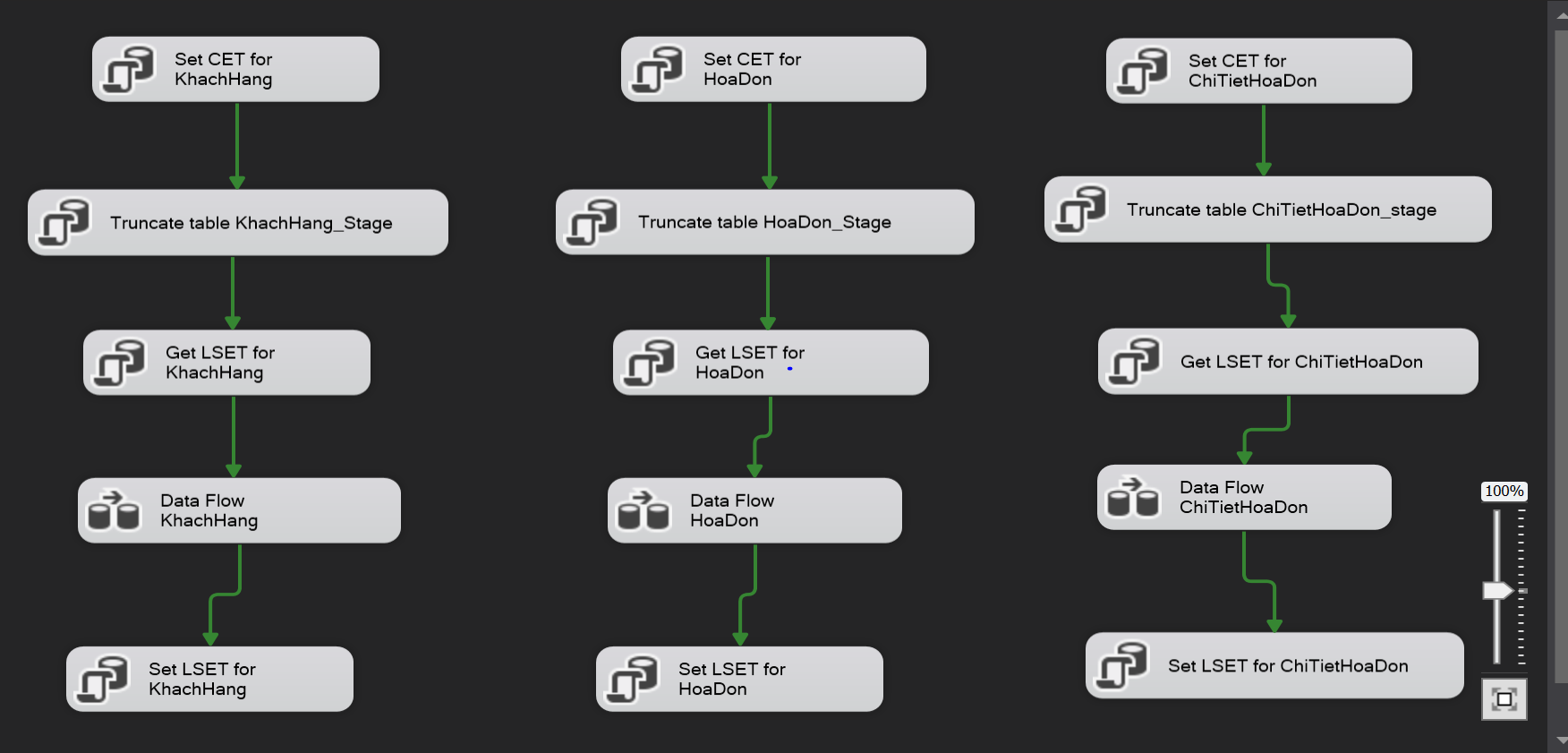
****

## Các bước khi đổ dữ liệu từ Source vào Stage

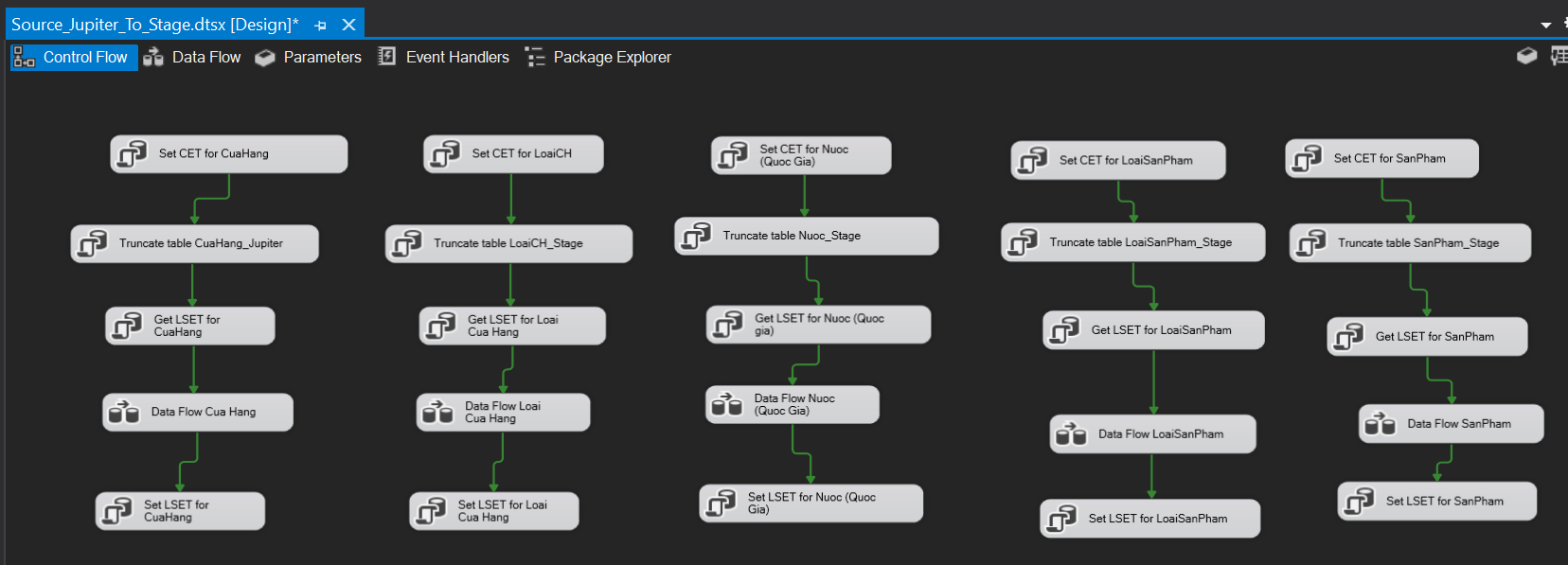
* Tạo một cơ sở dữ liệu metadata gồm bảng có các cột Tên bảng, LSET, CET. Trong đó:
  + LSET: Thời điểm rút trích hiện tại
  + CET: Thời điểm rút trích thành công gần đây nhất
* Tạo một cơ sở dữ liệu Stage có cấu trúc giống với cơ sở dữ liệu nguồn
* Các bước thực hiện đổ nguồn vào stage:
  + **Bước 1**: Lấy thông tin LSET được lưu trong metadata
  + **Bước 2**: Lấy CET bằng thời gian khởi động package
  + **Bước 3:** Truncate bảng Stage hiện tại: TRUNCATE TABLE <TenBang>
  + **Bước 4**: Rút trích dữ liệu từ bảng Source sang bảng Stage: select \* from BangSource where (NgayTao >= LSET and NgayTao < CET) or (NgayCapNhat >= LSET and NgayCapNhat < CET)
  + **Bước 5**: Cập nhật LSET = CET

## Flow khi thực hiện đổ dữ liệu từ Source vào Stage trên Visual Studio

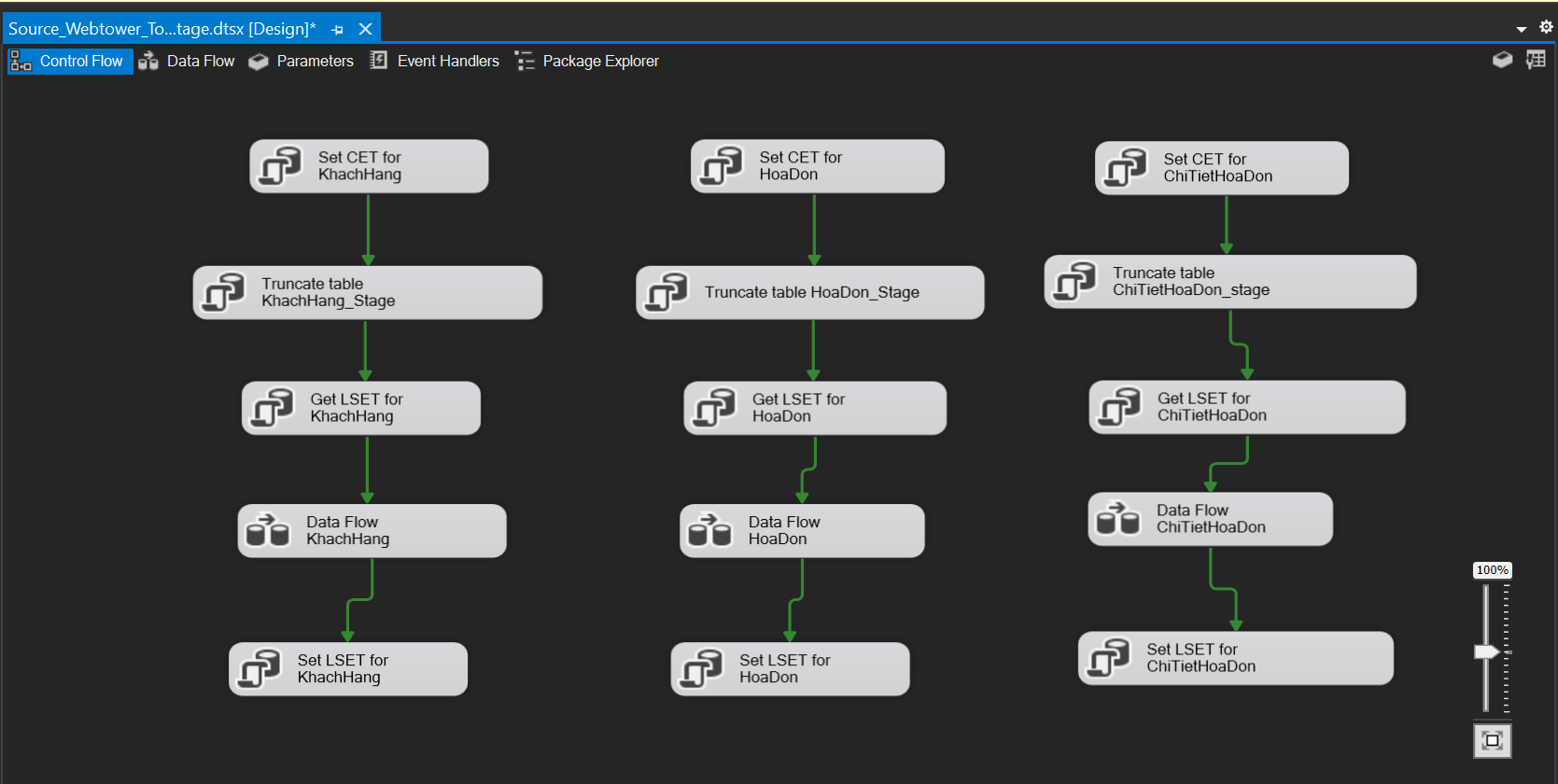
* ETL từ Source Jade sang Stage

**~~~~**

* ETL từ Source Jupiter sang Stage

****

* ETL từ Source Webtower sang Stage

****

# Stage sang NDS

Khi đổ hai nguồn (Jade, Jupiter) vào Stage, dữ liệu trong cơ sở dữ liệu DALT\_Stage sẽ gồm các bảng:

* KhachHang\_Stage
* CuaHang\_Stage
* LoaiCuaHang\_Stage
* HoaDon\_Stage
* ChiTietHoaDon\_Stage
* SanPham\_Stage
* LoaiSanPham\_Stage
* Nuoc\_Stage

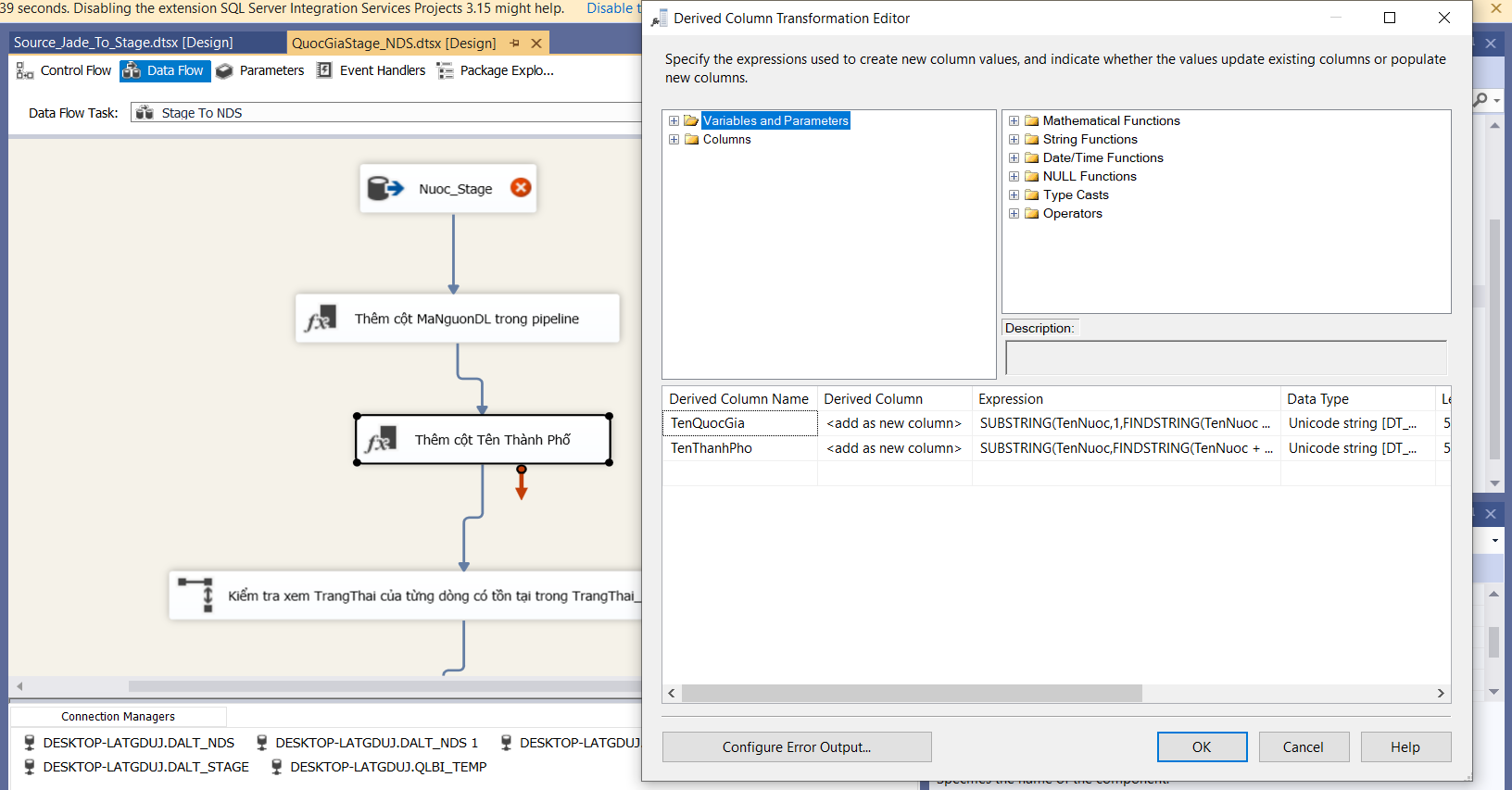
Khi đổ nguồn Webtower vào Stage, dữ liệu trong cơ sở dữ liệu DALT\_STAGE\_Webtower sẽ gồm các bảng. Ở giai đoạn từ source sang stage, nhóm vẫn sẽ chưa trộn dữ liệu của KhachHang, HoaDon và ChiTietHoaDon của 2 nguồn Jade và Webtower lại với nhau vì nếu làm như vậy thì sẽ rất khó để phân biệt được cơ sở dữ liệu nguồn của dữ liệu khi dữ liệu đã được gộp chung.

* KhachHang\_Webtower9
* HoaDon\_Webtower9
* ChiTietHoaDon\_Webtower9

Theo khái niệm, NDS được chuẩn hoá từ 3NF trở lên. Ta có dữ liệu gồm ba nguồn (Jade, Webtower, Jupiter) chưa được chuẩn hoá vì vậy cần chuẩn hoá dữ liệu để đáp ứng cấu trúc trong NDS.

## Một vài sự thay đổi dữ liệu chuyển từ giai đoạn Stage sang NDS:

* Bảng Nuoc\_Stage:
* Đổi tên thành QuocGia, hai thuộc tính MaNuoc --> MaQG.
* Đồng thời tách cột TenNuoc thành 2 cột riêng lẻ là TenNuoc và TenThanhPho. Lý do là vì ở dữ liệu nguồn, các giá trị ở cột TenNuoc được lưu theo format giống như giá trị ở cột DiaChi của các bảng KhachHang, chẳng hạn như Kazakhstan - Alabama, nên ta cần tách ra thành 2 giá trị riêng biệt cho đúng với ngữ cảnh, đồng thời làm tăng độ mịn của dữ liệu hơn. Dưới đây là hình ảnh minh hoạ khi thực hiện tách cột trên Visual Studio:

****

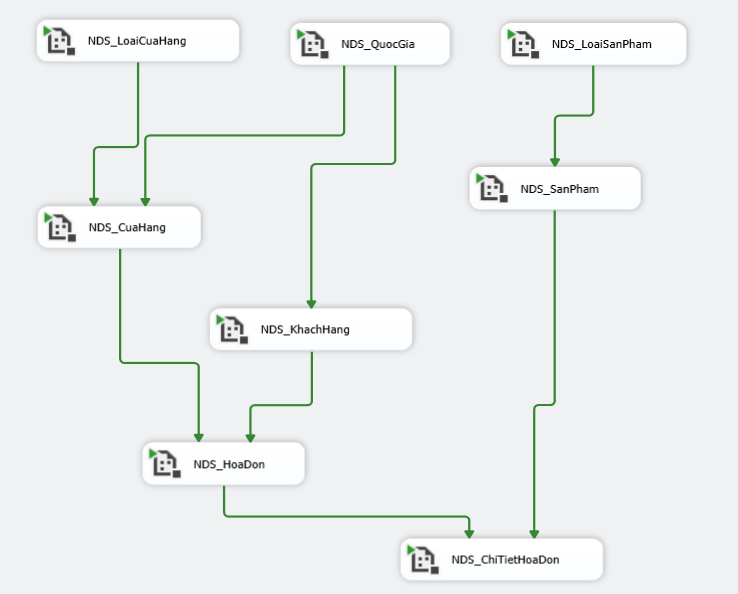
## Mô tả flow khi dữ liệu chuyển từ giai đoạn Stage sang NDS

Ví dụ như khi đổ dữ liệu từ stage sang NDS đối với các bảng KHACHHANG thì sẽ :

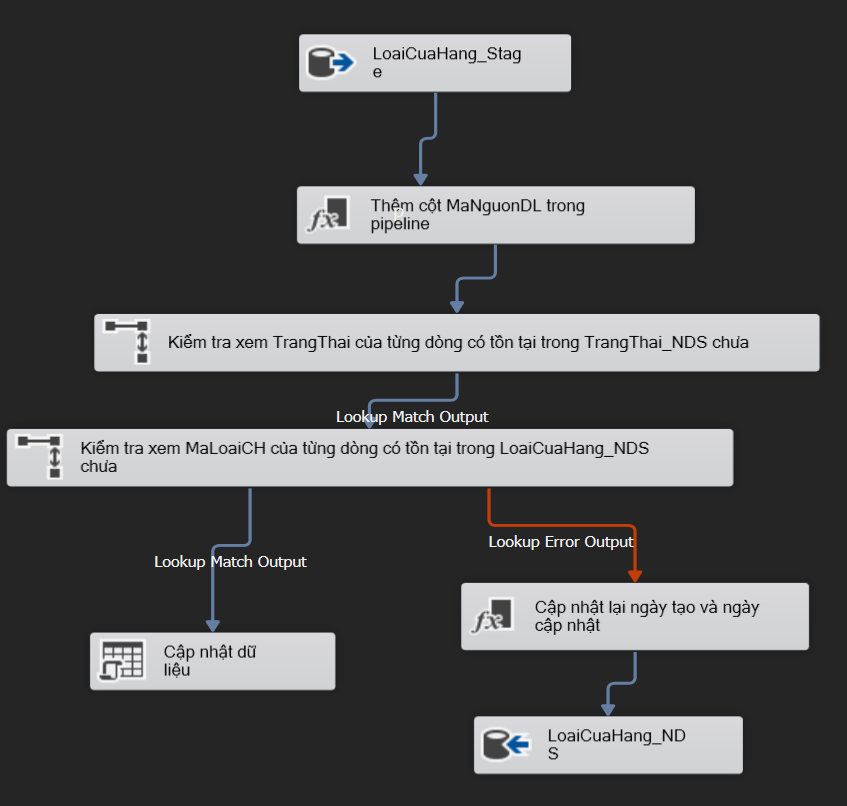
* **Bước 1**: Thêm cột MaNguonDL trong data pipeline dữ liệu. Đối với KhachHang\_Stage đến từ nguồn Jade thì MaNguonDL là 2; đến từ nguồn Webtower thì MaNguonDL là 3.
* **Bước 2:** Kiểm tra trạng thái có từng tồn tại trong trong bảng TrangThai\_NDS chưa. Nếu có thì đi tới bước tiếp theo.
* **Bước 3:** Kiểm tra MaNuoc đã từng tồn tại trong bảng QuocGia\_NDS chưa bằng cách Select tenthanhpho, tenqg From QuocGia. Nếu tồn tại thì trả về khóa tự tăng (skmaqg) của bảng QuocGia. Nếu không tồn tại sẽ thêm dòng dữ liệu và bảng NoMatch-KhachHang để xử lý dữ liệu sau.
* **Bước 4:** Kiểm tra MaKH đã từng tồn tại trong bảng KhachHang\_NDS chưa nếu có thì cập nhật lại dữ liệu. Nếu chưa có thì cập nhật lại ngày tạo và ngày cập nhật bằng thời gian hiện tại và đổ dữ liệu vào bảng KhachHang\_NDS.
* **Bước 5:** Dùng Union All để union hai dataflow đến từ hai nguồn dữ liệu lại với nhau. Lí do mà nhóm không dùng Merge là vì các lý do như sau:
* Một là: nếu muốn dùng merge, trước đó data flow của mỗi nguồn dữ liệu cần phải Sort trước khi merge. Điều này chỉ thực sự cần thiết khi người dùng có yêu cầu dữ liệu đầu ra cần được sắp xếp, nhưng ở trong tình huống hiện tại thì yêu cầu đầu ra không cần được sắp xếp
* Hai là: cả merge và union all đều có thể dùng để trộn 2 input (table hoặc file) thành 1 input. Việc dùng Union All có thể giúp để phòng hờ cho các tình huống cụ thể ở tương lai, khi mà có hơn 2 nguồn cùng cần được trộn để đổ vào NDS. Cả bảng KhachHang ở Jade và bảng KhachHang ở Webtower9 đều có cùng chung số lượng thuộc tính, mỗi thuộc tính lại cùng kiểu dữ liệu nên cũng rất dễ để sử dụng Union All.

Tương tự với bảng KhachHang các bảng khác của tất cả các Stage khi đổ từ Stage sang NDS cũng flow gần giống nhau. Đặc biệt là các data flow cho HoaDon và ChiTietHoaDon sẽ cần dùng thêm Union All.

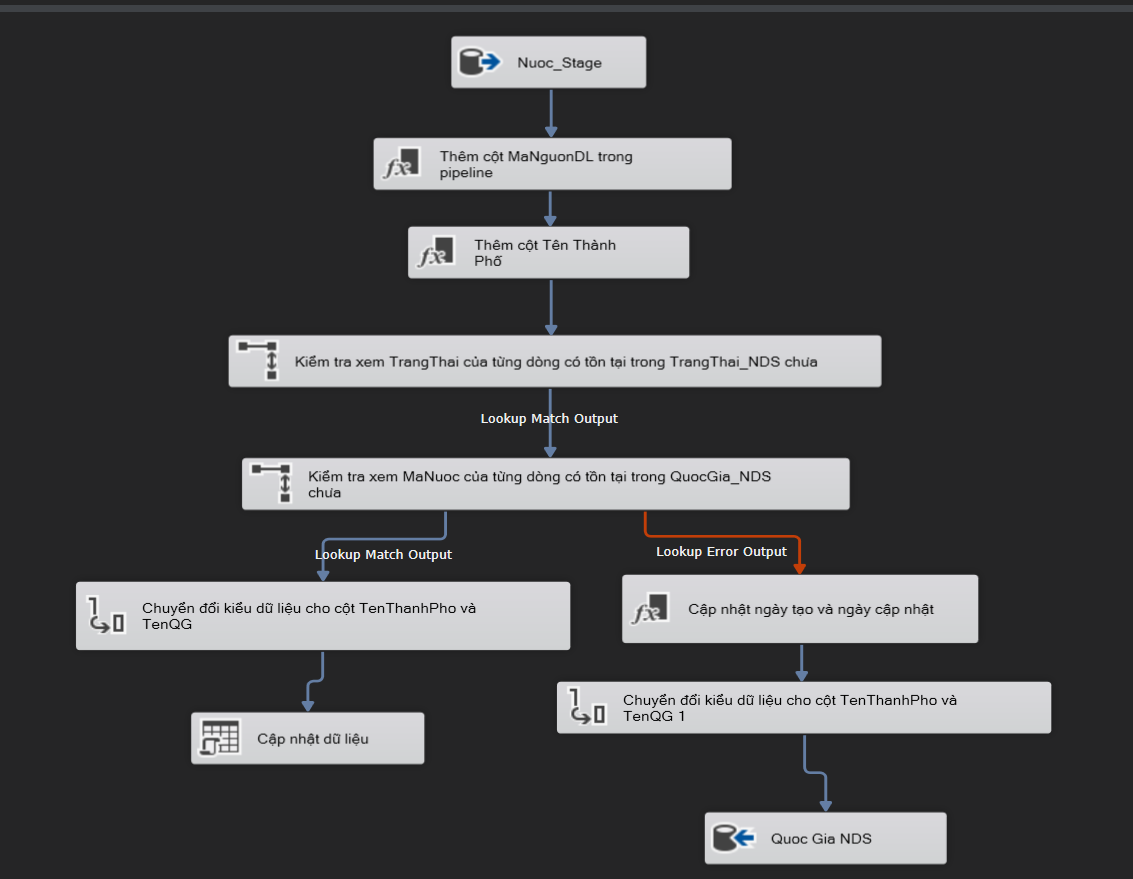
## Flow khi thực hiện đổ dữ liệu từ Stage vào NDS trên Visual Studio

****

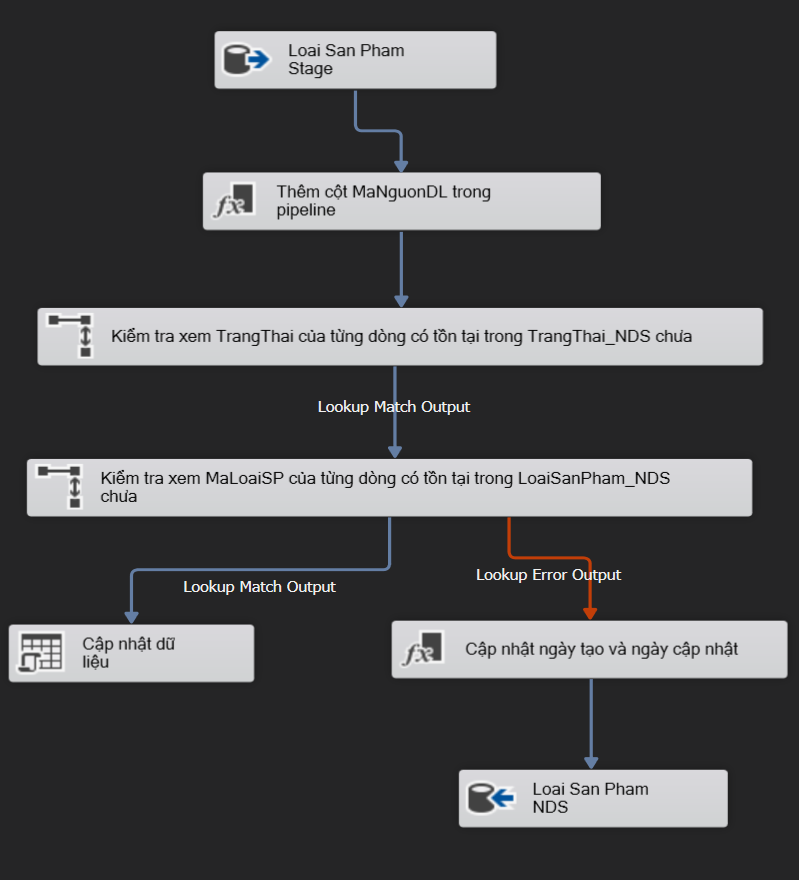
* ETL từ Stage\_LoaiCuaHang sang NDS\_LoaiCuaHang

****

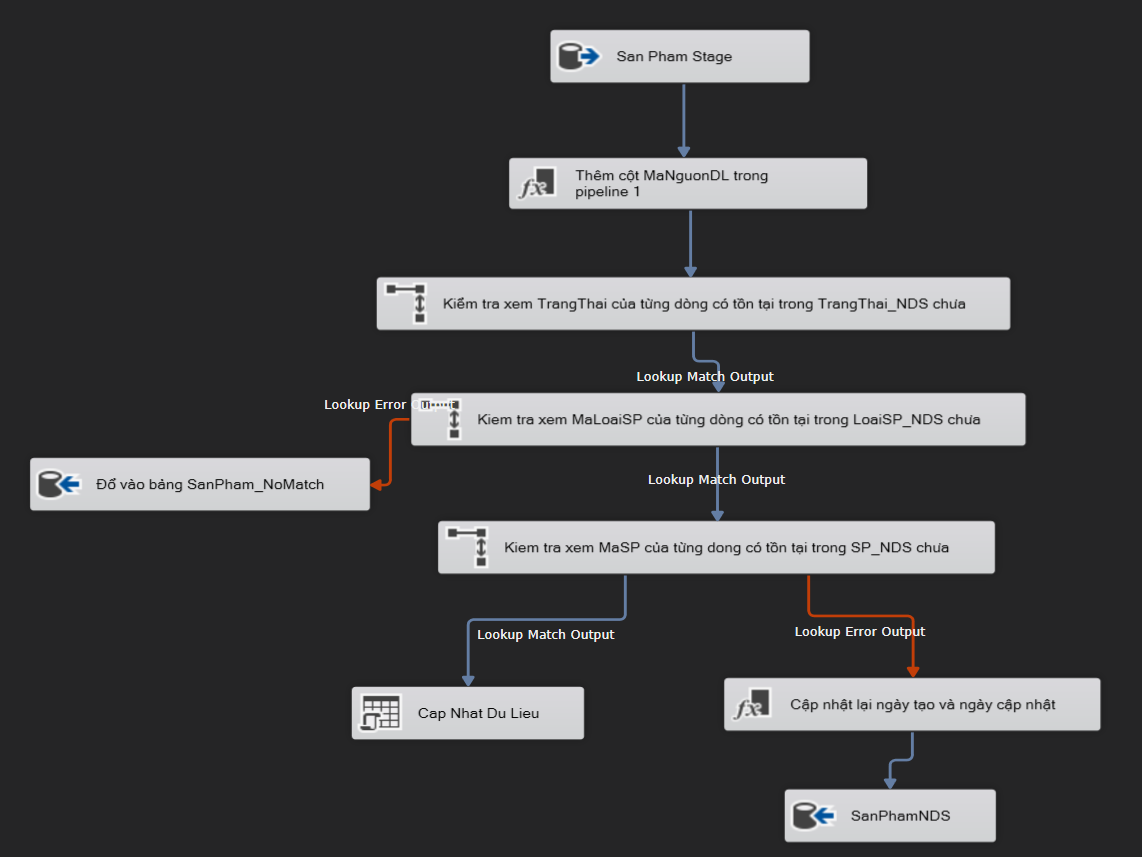
* + ETL từ Stage\_QuocGia sang NDS\_QuocGia

****

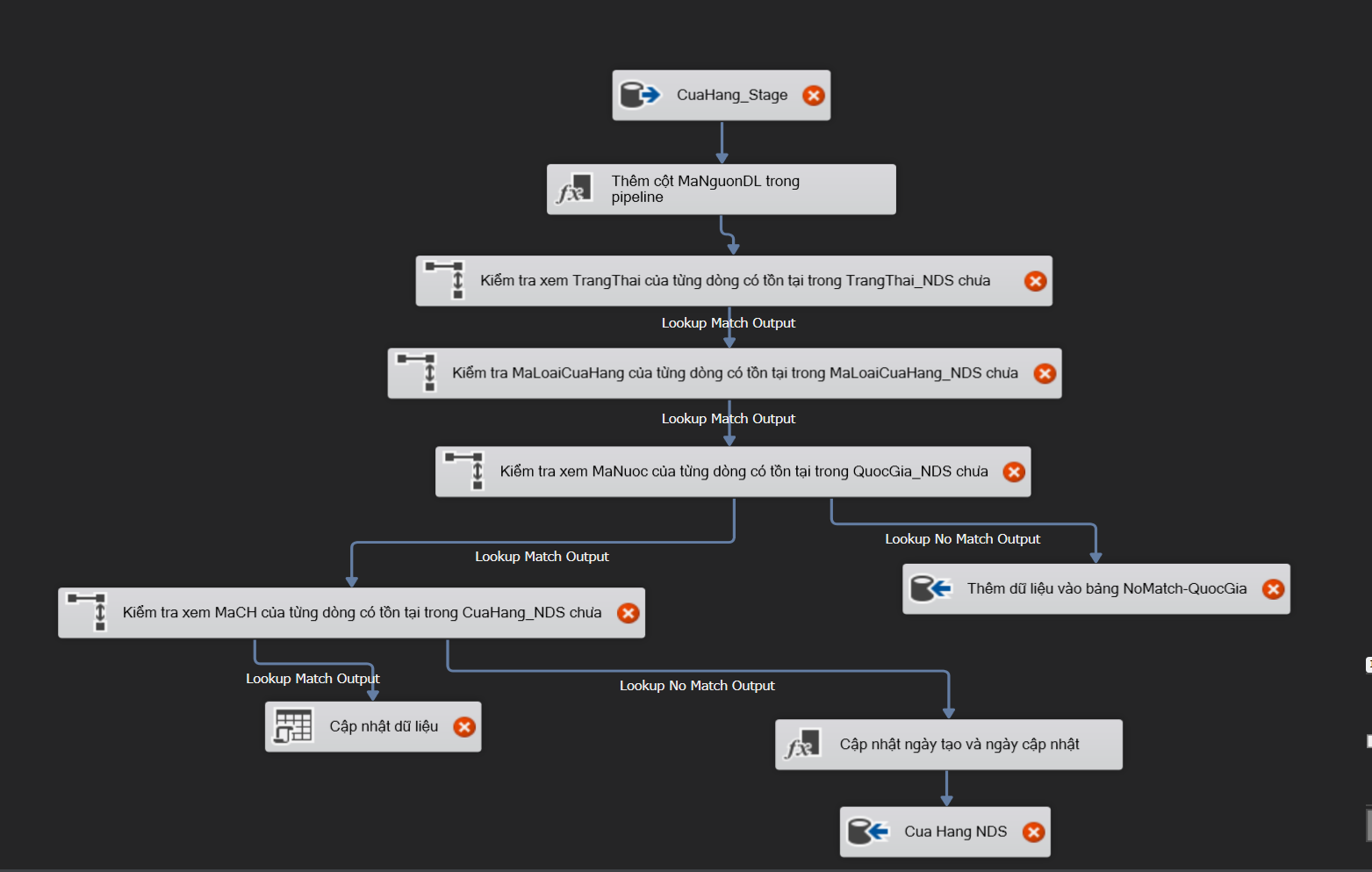
* + ETL từ Stage\_LoaiSanPham sang NDS\_LoaiSanPham

****

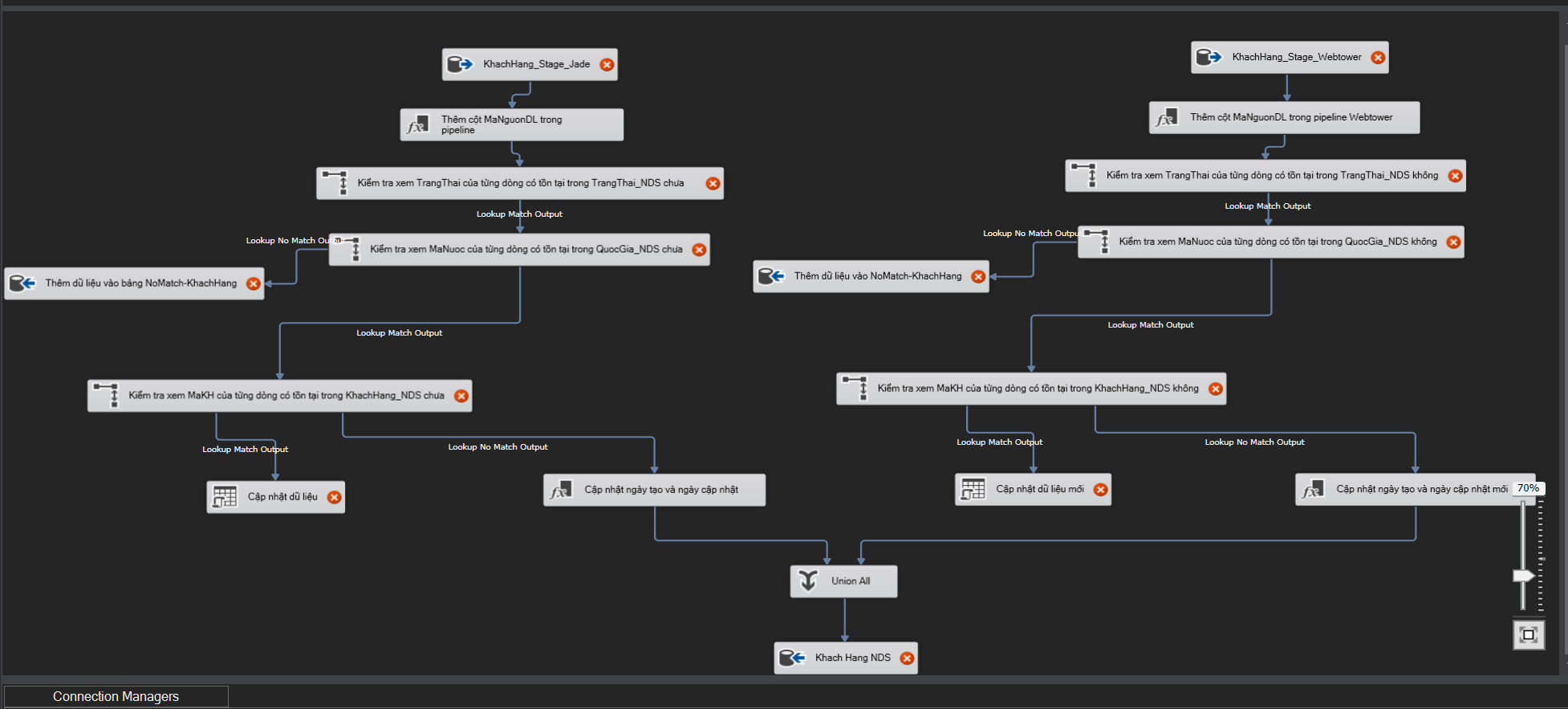
* + ETL từ Stage\_SanPham sang NDS\_SanPham

****

* + ETL từ Stage\_CuaHang sang NDS\_CuaHang

****

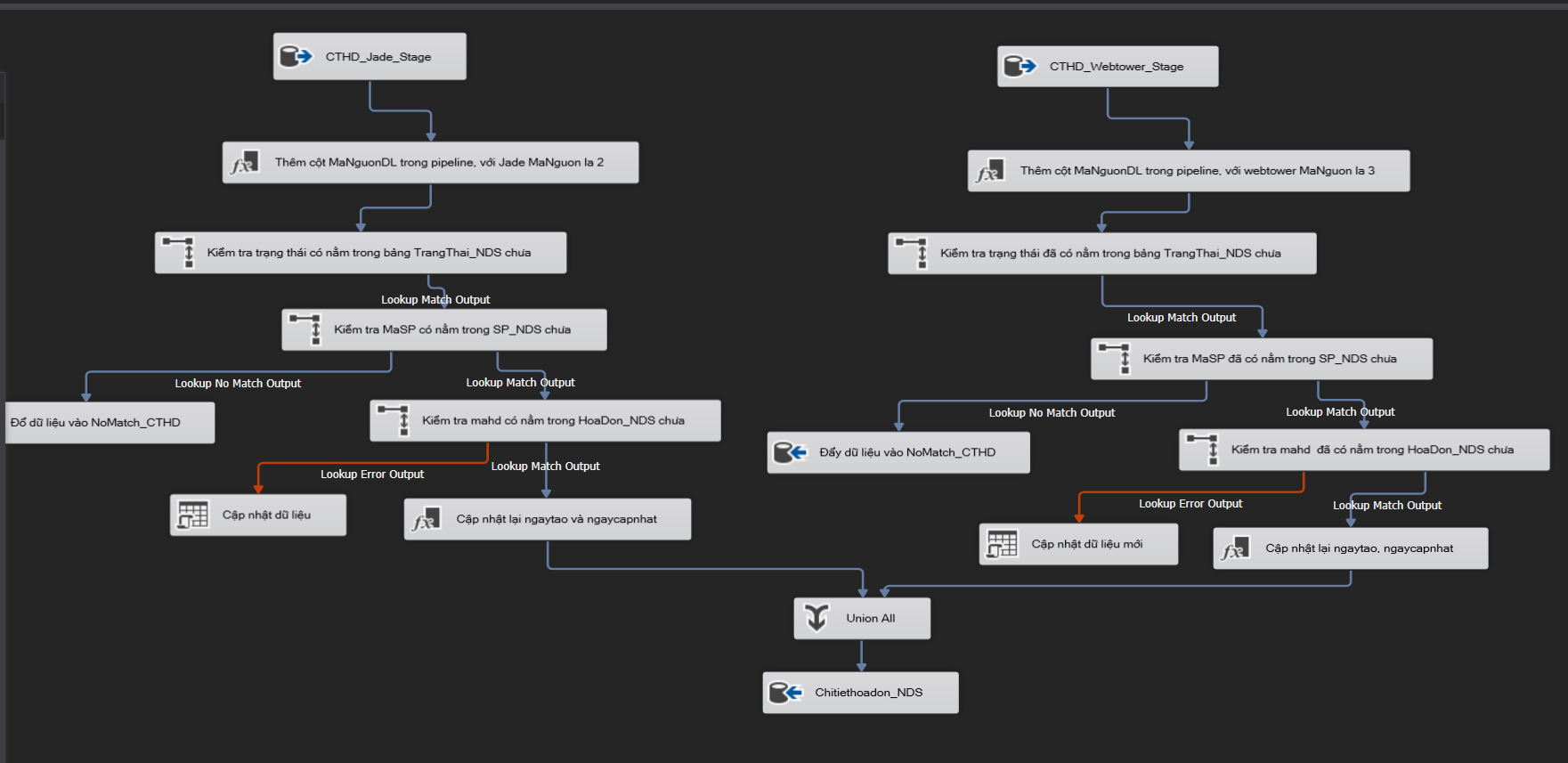
* + ETL từ Stage\_KhachHang sang NDS\_KhachHang

****

* + ETL từ Stage\_HoaDon sang NDS\_HoaDon

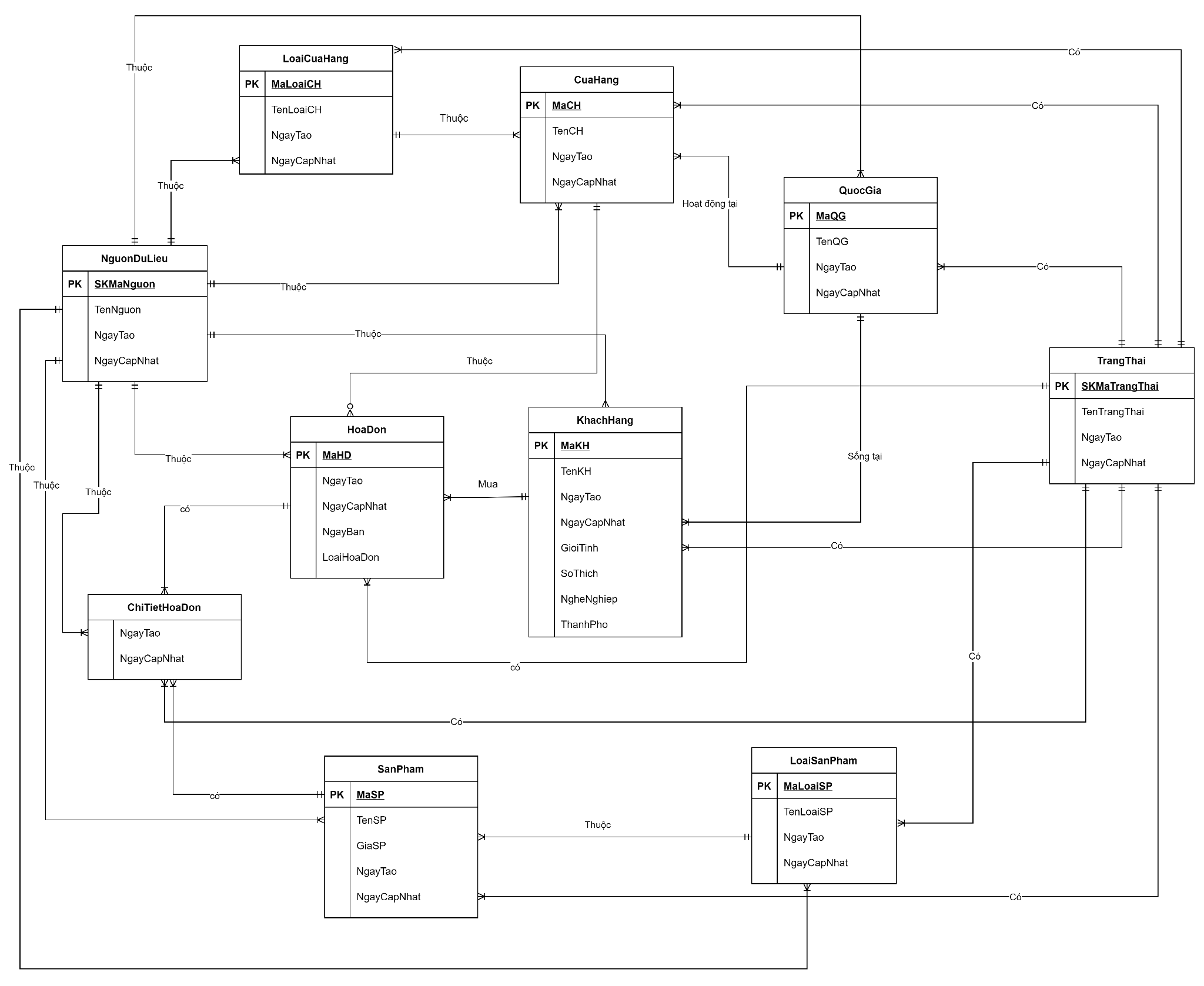
****

* + ETL từ Stage\_ChiTietHoaDon sang NDS\_ChiTietHoaDon

****

## Lược đồ NDS dưới dạng mô hình ER

Hình vẽ dưới thể hiện rõ hơn cơ sở dữ liệu sau khi chuẩn hóa thành NDS dưới dạng mô hình ER, ta sẽ dễ dàng thấy rõ sự thay đổi dữ liệu sau khi chuẩn hóa đã được liệt ở trên.

****

* Mô tả dữ liệu:

1. CuaHang (MaCH, TenCH, NgayTao, NgayCapNhat, MaLoaiCH, MaQG, SKMaNguon, SKMaTrangThai)

PHỤ THUỘC HÀM F = { MaCH -> TenCH, NgayTao, NgayCapNhat, MaLoaiCH, MaQG, SKMaNguon, SKMaTrangThai }

DẠNG CHUẨN: BCK

1. LoaiCuaHang (MaLoaiCH, TenLoaiCH, NgayTao, NgayCapNhat, SKMaTrangThai, SKMaNguon)

PHỤ THUỘC HÀM F = {MaLoaiCH -> TenLoaiCH, NgayTao, NgayCapNhat, SKMaTrangThai, SKMaNguon}

DẠNG CHUẨN: BCK

1. QuocGia (MaQG, TenQG, NgayTao, NgayCapNhat, SKMaTrangThai, SKMaNguon)

PHỤ THUỘC HÀM F = {MaQG -> TenQG, NgayTao, NgayCapNhat, SKMaTrangThai, SKMaNguon}

DẠNG CHUẨN: BCK

1. KhachHang ( MaKH, TenKH, NgayTao, NgayCapNhat, GioiTinh, SoThich, NgheNghiep, ThanhPho, MaQG, SKMaNguon, SKMaTrangThai)

PHỤ THUỘC HÀM F = {MaKH -> TenKH, NgayTao, NgayCapNhat, GioiTinh, SoThich, NgheNghiep, ThanhPho, MaQG, SKMaNguon, SKMaTrangThai }

DẠNG CHUẨN: BCK

1. SanPham ( MaSP, TenSP, GiaSP, NgayTao, NgayCapNhat, MaLoaiSP, SKMaNguon, SKMaTrangThai)

PHỤ THUỘC HÀM F = {MaSP -> TenSP, GiaSP, NgayTao, NgayCapNhat, MaLoaiSP, SKMaNguon, SKMaTrangThai}

DẠNG CHUẨN: BCK

1. LoaiSanPham (MaLoaiSP, TenLoaiSP, NgayTao, NgayCapNhat, SKMaNguon, SKMaTrangThai)

PHỤ THUỘC HÀM F = { MaLoaiSP > TenLoaiSP, NgayTao, NgayCapNhat, SKMaNguon, SKMaTrangThai}

DẠNG CHUẨN: BCK

1. HoaDon (MaHD, NgayBan, NgayCapNhat, NgayTao, LoaiHoaDon, MaCH, MaKH, SKMaNguon, SKMaTrangThai )

PHỤ THUỘC HÀM F = {MaHD -> NgayBan, NgayCapNhat, NgayTao, LoaiHoaDon MaCH, MaKH, SKMaNguon, SKMaTrangThai }

DẠNG CHUẨN: BCK

1. ChiTietHoaDon (MaHD, MaSP, NgayTao, NgayCapNhat, SKMaNguon, SKMaTrangThai)

PHỤ THUỘC HÀM F = {MaHD, MaSP -> NgayTao, NgayCapNhat, SKMaNguon, SKMaTrangThai}

DẠNG CHUẨN: BCK

1. NguonDuLieu(SKMaNguon, TenNguon, NgayTao, NgayCapNhat)

PHỤ THUỘC HÀM F = {SKMaNguon -> TenNguon, NgayTao, NgayCapNhat}

DẠNG CHUẨN: BCK

1. TrangThai(SKMaTrangThai, TenTrangThai, NgayTao, NgayCapNhat)

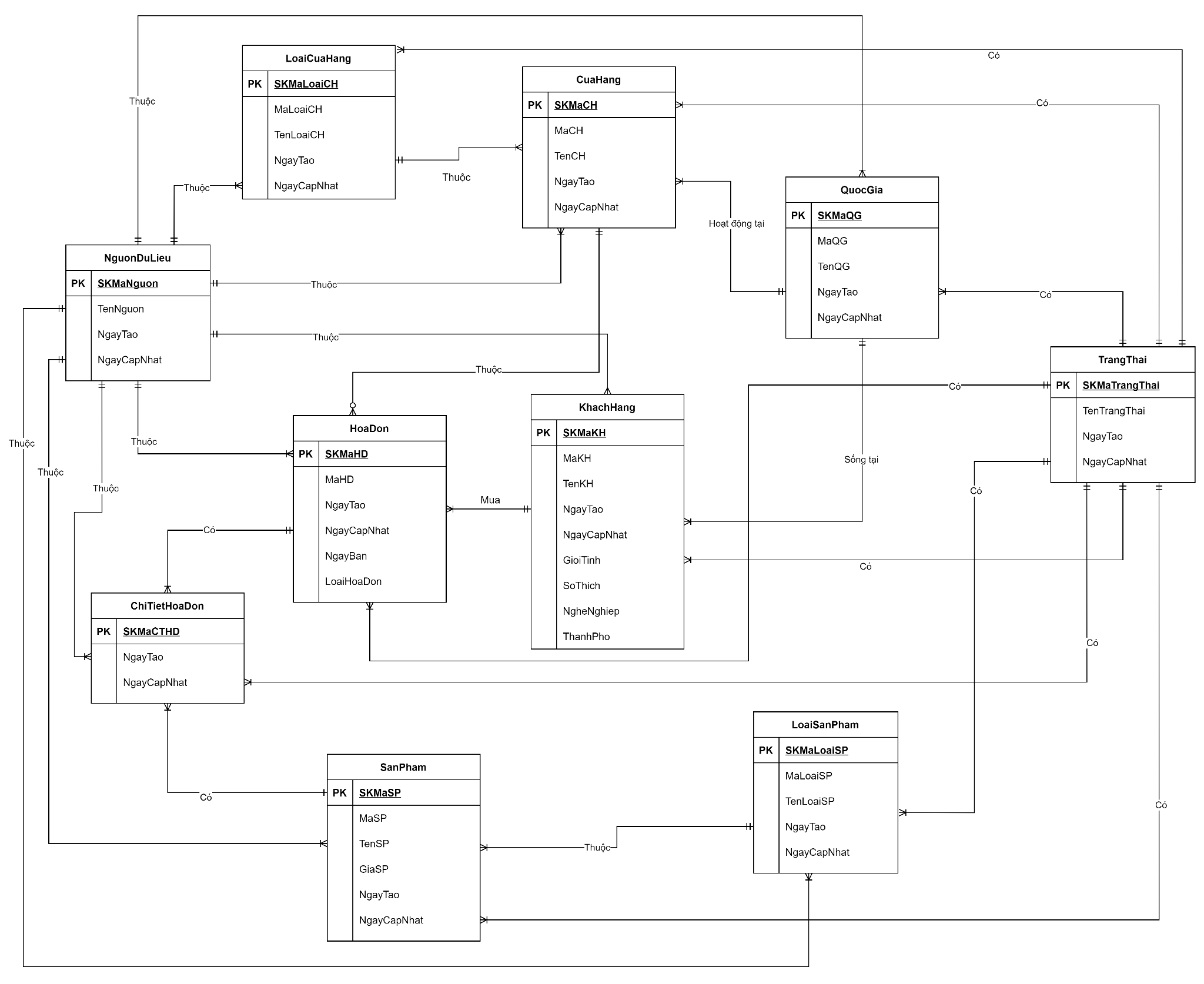
PHỤ THUỘC HÀM F = {SKMaTrangThai -> TenTrangThai, NgayTao, NgayCapNhat}

DẠNG CHUẨN: BCK

=> Kết luận: Như vậy, lược đồ tổng thể đạt dạng chuẩn BCK, đáp ứng cấu trúc trong NDS

## Lược đồ quan hệ dưới dạng physical

Khi cài đặt ở mức vật lý, để tránh tình trạng sai sót và thất thoát dữ liệu trong tương lai khi các mã cửa hàng, mã quốc gia hay mã loại cửa hàng bị thay đổi sẽ ảnh hưởng đến tham chiếu khóa ngoại. Nhóm chúng em quyết định tạo thêm một trường khóa tự tăng có tiền tố là SK ở mỗi bảng với kiểu dữ liệu là int. Khi đó, các bảng khác nếu có tham chiếu sẽ tham chiếu tới các giá trị có kiểu dữ liệu là int này, và ta cũng không cần phải lo lắng nếu trong tương lai các mã thực thể kia có thay đổi.

****

* + Cấu trúc bảng CuaHang (WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaCH | Khóa tự tăng của bảng CuaHang |  |  |
| MaCH | Mã cửa hàng trong source system | Ja.HoaDon\_Stage.MaCH  Wt.HoaDon\_STAGE\_Webtower.MaCH  Ju.CuaHang\_Stage.MaCH |  |
| TenCH | Tên của cửa hàng | Ju.CuaHang\_Stage.TenCH |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| SKMaLoaiCH | Mã loại cửa hàng cho biết cửa hàng thuộc loại cửa hàng nào | Tham chiếu khóa ngoại từ bảng LoaiCuaHang |  |
| SKMaQG | Mã quốc gia mà khách hàng đang sống | Tham chiếu khóa ngoại từ bảng QuocGia |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng LoaiCuaHang (WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaLoaiCH | Khóa tự tăng của bảng LoaiCuaHang |  |  |
| MaLoaiCH | Mã loại cửa hàng trong source system | Ju.LoaiCuaHang\_Stage.MaLoaiCH |  |
| TenLoaiCH | Tên của loại cửa hàng | Ju.LoaiCuaHang\_Stage.TenLoaiCH |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng KhachHang(WebTower (Wt), Jade (Ja), Jupiter (Ju) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaKH | Khoá tự tăng của bảng KhachHang |  |  |
| MaKH | Mã khách hàng trong source system | Ja.KhachHang\_Stage.MaKH  Wt.KhachHang\_STAGE\_Webtower.MaKH |  |
| TenKH | Tên của khách hàng | Ja.KhachHang\_Stage.TenKH  Wt.KhachHang\_STAGE\_Webtower.TenKH |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| GioiTinh | Giới tính của khách hàng | Ja.KhachHang\_Stage.GioiTinh  Wt.KhachHang\_STAGE\_Webtower.GioiTinh |  |
| SoThich | Sở thích của khách hàng | Ja.KhachHang\_Stage.SoThich  Wt.KhachHang\_STAGE\_Webtower.SoThich |  |
| NgheNghiep | Nghề nghiệp của khách hàng | Ja.KhachHang\_Stage.NgheNghiep  Wt.KhachHang\_STAGE\_Webtower.NgheNghiep |  |
| ThanhPho | Thành phố mà khách hàng đang sống | Ja.KhachHang\_Stage.DiaChi  Wt.KhachHang\_STAGE\_Webtower.DiaChi | Đổi từ fomat “Nuoc-ThanhPho” thành fomat “ThanhPho” |
| SKMaQG | Mã quốc gia mà khách hàng đang sống | Tham chiếu khóa ngoại từ bảng QuocGia |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng QuocGia(WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaQG | Khoá tự tăng của bảng QuocGia |  |  |
| MaQG | Mã Quốc Gia trong source system | Ju.Nuoc\_Stage.MaNuoc |  |
| TenQG | Tên của quốc gia | Ju.Nuoc\_Stage.TenNuoc |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng SanPham(WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaSP | Khoá tự tăng của bảng SanPham |  |  |
| MaSP | Mã sản phẩm trong source system | Ju.SanPham\_Stage.MaSP |  |
| TenSP | Tên của sản phẩm | Ju.SanPham\_Stage.TenSP |  |
| GiaSP | Giá của sản phẩm | Ju.SanPham\_Stage.GiaSP |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| SKMaLoaiSP | Mã loại sản phẩm cho biết sản phẩm thuộc loại sản phẩm nào | Tham chiếu khóa ngoại từ bảng LoaiSanPham |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng LoaiSanPham(WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaLoaiSP | Khoá tự tăng của bảng LoaiSanPham |  |  |
| MaLoaiSP | Mã loại sản phẩm trong source system | Ju.LoaiSanPham\_Stage.MaLoaiSP |  |
| TenLoaiSP | Tên của loại sản phẩm | Ju.LoaiSanPham\_Stage.TenLoaiSP |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng HoaDon (WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaHD | Khoá tự tăng của bảng HoaDon |  |  |
| MaHD | Mã hóa đơn trong source system | Ja.HoaDon\_Stage.MaHD  Wt.HoaDon\_STAGE\_Webtower.MaHD |  |
| Ngày bán | Ngày lập hóa đơn bán hàng | Ja.HoaDon\_Stage.NgayBan  Wt.HoaDon\_STAGE\_Webtower.NgayBan |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| LoaiHoaDon | Mã của loại hóa đơn | Tham chiếu khóa ngoại từ bảng LoaiHoaDon |  |
| SKMaCH | Mã của cửa hàng tạo hóa đơn | Tham chiếu khóa ngoại từ bảng CuaHang |  |
| SKMaKH | Mã của khách hàng mua | Tham chiếu khóa ngoại từ bảng KhachHang |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng ChiTietHoaDon(WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaHD | Khoá tự tăng của bảng HoaDon |  |  |
| SKMaSP | Khoá tự tăng của bảng SanPham |  |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |
| SKMaNguon | Nguồn chứa record | 1 đối với Jupiter, 2 đối với Jade, 3 đối với WebTower |  |
| SKMaTrangThai | Trạng thái của dòng record | Tham chiếu khóa ngoại từ bảng TrangThai |  |

* + Cấu trúc bảng NguonDuLieu (WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaNguon | Khóa tự tăng của bảng NguonDuLieu |  |  |
| TenNguon | Tên của nguồn dữ liệu |  |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |

* + Cấu trúc bảng TrangThai (WebTower (Wt), Jade (Ja), Jupiter (Jupiter) trong NDS

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Mô tả | Nguồn | Transformation |
| SKMaTrangThai | Khóa tự tăng của bảng TrangThai |  |  |
| TenTrangThai | Tên của trạng thái |  |  |
| NgayTao | Thời gian mà record được tạo trong Dimension Data Store |  |  |
| NgayCapNhat | Thời gian mà record được cập nhật trong Dimension Data Store |  |  |

# ETL dữ liệu từ NDS sang DDS

## Lược đồ DDS

Diagram, schematic

Description automatically generated

**Link ảnh:** [**https://app.diagrams.net/#G1G9qdVZ8tlg-vpB-UTrtkeq2rC4oepJq**](https://app.diagrams.net/#G1G9qdVZ8tlg-vpB-UTrtkeq2rC4oepJq)

## Phân tích yêu cầu

Người dùng nghiệp vụ cần phân tích doanh số bán hàng theo thời gian trên các tiêu chí: vùng miền của khách hàng mua hàng, cửa hàng, nhóm sản phẩm.

* Sự kiện: Khi 1 khách hàng mua sản phẩm online hoặc offline tại một trong các vùng miền của các nước Mỹ, Đức, Pháp, Anh.
* Bối cảnh sự kiện:
* Ai: Khách hàng
* Ở đâu: Cửa hàng
* Cái gì: Sản phẩm
* Khi nào: Ngày mua hàng
* Đo lường (dữ kiện): đơn giá của sản phẩm

## Mô hình hoá

* Đo lường -> Fact Table : Một dòng trong bảng Fact\_ABCDoanhThu tương ứng với mỗi item được bán cho một khách hàng nhất định tại một cửa hàng cụ thể vào thời gian cụ thể.
* Bối cảnh -> Dimension table: gồm các bảng Dim\_LoaiCuaHang, Dim\_QuocGia, Dim\_CuaHang, Dim\_KhachHang, Dim\_LoaiSanPham, Dim\_SanPham, Dim\_HoaDon, DimDate. Trong đó DimDate là bảng duy nhất được tạo sẵn và không có ETL flow theo yêu cầu của giáo viên.
* Mô hình liên kết các dimension table và fact table: lược đồ bông tuyết (Snowflake Schema).
* Lý do chọn mô hình này :Việc nhóm lựa chọn sử dụng lược đồ bông tuyết là vì lược đồ bông tuyết chuẩn hoá tất cả các bảng chiều giúp tránh được việc dư thừa dữ liệu, đảm bảo tính toàn vẹn của dữ liệu, dễ thiết lập, bảo trì và ít tốn dung lượng. Các lược đồ hình sao có thể chạy các truy vấn nhanh hơn nhưng chúng yêu cầu nhiều không gian hơn so với lược đồ bông thuyết do dư thừa dữ liệu ở lược đồ sao. Tính toàn vẹn dữ liệu trong lược đồ hình sao có nhiều rủi ro hơn so với lược đồ bông tuyết. Vì dữ liệu ở lược đồ hình sao được lưu trữ dư thừa nên nhiều bản sao của cùng một dữ liệu tồn tại trong các bảng chiều của lược đồ hình sao, điều này có nghĩa là  khi thêm, cập nhật hay xoá mới có thể ảnh hưởng đến tính toàn vẹn của dữ liệu. Ngược lại, lược đồ bông tuyết gặp ít vấn đề về tính toàn vẹn dữ liệu hơn vì nó được chuẩn hoá hoàn toàn các bảng chiều, lưu dữ liệu chiều chỉ một lần trong bảng thích hợp. Ngoài ra câu hỏi  bài toán đặt ra theo yêu cầu có phân cấp dữ liệu như theo vùng miền, theo thời gian, loại sản phẩm mà lược đồ bông tuyết có sự phân cấp nên nhóm quyết định lựa chọn lược đồ bông tuyết.
* Có sự phân cấp dữ liệu:
* Phân cấp theo thời gian đánh giá: ngày, tháng, quý, năm.
* Phân cấp theo khu vực khách hàng mua hàng: QuocGia -> ThanhPho.
* Phân cấp theo nhóm sản phẩm: LoaiSanPham -> SanPham.

## Mô hình hoá - Fact table

* **Các giá trị có sẵn từ nguồn:** giá của sản phẩm (GiaSP)
* **Các giá trị phải tính toán:** Không có
* **Cấp chi tiết dữ liệu (độ mịn)**: Đơn vị nhỏ nhất xảy ra sự kiện: Một dòng trong bảng Fact\_ABCDoanhThu tương ứng với mỗi item được bán
* **Chiều thái hóa:** Không
* **Dữ liệu đến từ nguồn nào**: ở đây dữ liệu đến từ các nguồn của các hệ thống WebTower9, Jade, Jupiter
* **Nhãn thời gian:**
* Create\_timestamp: NgayTao trong bảng Fact\_ABCDoanhThu để theo dõi ngày , tạo dòng dữ liệu, thời gian sinh viên điền form đánh giá.
* Update\_timestamp: NgayCapNhat trong bảng Fact\_ABCDoanhThu để theo dõi ngày cập nhật dòng dữ liệu, thời gian chỉnh sửa dòng đánh giá.
* **Khóa tự nhiên của bảng Fact:**
* SKDoanhThu
* SKMaCH
* SKMaKH
* SKMaHD
* SKMaSP
* SKDate

## Mô hình hóa - Thiết kế chiều

* Các chiều liên quan đến sự kiện phân tích:
* Dim\_LoaiCuaHang dimension
* Dim\_QuocGia dimension
* Dim\_CuaHang dimension
* Dim\_LoaiSanPham dimension
* Dim\_SanPham dimension
* Dim\_HoaDon dimension
* DimDate dimension
* Dim\_KhachHang dimension
* Lưu giá trị cũ của chiều:
* Dim\_LoaiCuaHang dimension: Không yêu cầu lưu giá trị cũ => SCD1- Ghi chồng giá trị cũ; chỉ có một nguồn là từ cơ sở dữ liệu NDS nên chuyển LoaiCuaHang\_NDS qua Dim\_LoaiCuaHang.
* Dim\_QuocGia dimension: Không yêu cầu lưu giá trị cũ => SCD1- Ghi chồng giá trị cũ; chỉ có một nguồn là từ cơ sở dữ liệu NDS nên chuyển QuocGia\_NDS qua Dim\_QuocGia.
* Dim\_CuaHang dimension: Không yêu cầu lưu giá trị cũ => SCD1- Ghi chồng giá trị cũ; chỉ có một nguồn là từ cơ sở dữ liệu NDS nên chuyển CuaHang\_NDS qua Dim\_CuaHang.
* Dim\_LoaiSanPham dimension: Không yêu cầu lưu giá trị cũ => SCD1- Ghi chồng giá trị cũ; chỉ có một nguồn là từ cơ sở dữ liệu NDS nên chuyển LoaiSanPham\_NDS qua Dim\_LoaiSanPham.
* Dim\_SanPham dimension: Không yêu cầu lưu giá trị cũ => SCD1- Ghi chồng giá trị cũ; chỉ có một nguồn là từ cơ sở dữ liệu NDS nên chuyển SanPham\_NDS qua Dim\_SanPham.
* Dim\_HoaDon dimension:Không yêu cầu lưu giá trị cũ => SCD1- Ghi chồng giá trị cũ; chỉ có một nguồn là từ cơ sở dữ liệu NDS nên chuyển HoaDon\_NDS qua Dim\_HoaDon.
* DimDate dimension: lấy dữ liệu từ bảng DimDate mới tạo trong DDS bằng cách Random lấy mã khoá bất kỳ của DimDate.
* Dim\_KhachHang dimension: Không yêu cầu lưu giá trị cũ => SCD1- Ghi chồng giá trị cũ; chỉ có một nguồn là từ cơ sở dữ liệu NDS nên chuyển KhachHang\_NDS qua Dim\_KhachHang.

## Mô hình hoá - Phân cấp dữ liệu

Diagram

Description automatically generated

## Flow khi đổ dữ liệu từ NDS sang DDS trong Visual Studio Code

* Điểm lưu ý khi đổ dữ liệu từ NDS vào DDS:

Nhóm đã quy định nếu matrangthai=1 tức là dòng đó đang hoạt động (active), matrangthai=2 tức là dòng đó đã không còn hoạt động nữa (disabled). Chính vì lý do này mà khi đổ dữ liệu từ NDS sang DDS, nhóm chỉ lấy những dòng dữ liệu có trạng thái đang hoạt động.

* Tuy nhiên theo dữ liệu có sẵn, khi lọc tất cả các dòng dữ liệu có matrangthai=1 ở bảng QuocGia, các KhachHang có matrangthai=1 thì vẫn có dòng tham chiếu khoá ngoại tới dòng của bảng QuocGia có matrangthai=2 dẫn đến KhachHang rep tới bảng QuocGia sẽ có giá trị bằng NULL ở trường SKMaQG. Nhóm đưa ra giải pháp là sử dụng ignore failure ở mỗi component lookup để những dòng không có giá trị để tham chiếu tại trường SKMaQG sẽ gán bằng NULL và đi đến component tiếp theo, sau đó nhóm sử dụng Conditional Split trong SISS để chỉ lấy những dòng khi lookup cột dữ liệu không có giá trị bằng NULL để đảm bảo rằng dữ liệu trong DDS luôn là dữ liệu đầy đủ.

Graphical user interface, application, Word

Description automatically generated

Diagram

Description automatically generated

* ETL từ LoaiSanPham\_NDS sang  Dim\_LoaiSanPham trong DDS

Graphical user interface, text

Description automatically generatedGraphical user interface

Description automatically generated

* **Bước 1**: Gán CET bằng thời gian khởi động package.
* **Bước 2:** Lấy LSET, CET từ dòng có Name\_Dataflow =  ‘LoaiSanPham\_NDStoDDS’ trong bảng Metadata
* **Bước 3**: Rút trích dữ liệu bảng LoaiSanPham trong cơ sở dữ liệu NDS sang DDS: **Select \* from LoaiSanPham where MaTrangThai=1 and (NgayTao >= LSET and NgayTao < CET) or (NgayCapNhat >= LSET and NgayCapNhat < CET)**.
* **Bước 4**: Cập nhật LSET = CET

Tương tự các flow khi đổ dữ liệu  từ các bảng QuocGia\_NDS, LoaiCuaHang\_NDS, HoaDon\_NDS sang DDS cũng giống flow khi đổ dữ liệu từ LoaiSanPham\_NDS vào DDS

* Flow ETL từ QuocGia\_NDS sang  Dim\_QuocGia trong DDS

Text

Description automatically generatedA screenshot of a phone

Description automatically generated with medium confidence

* Flow ETL từ LoaiCuaHang\_NDS sang  Dim\_LoaiCuaHang trong DDS

Diagram

Description automatically generatedGraphical user interface

Description automatically generated

* Flow ETL từ HoaDon\_NDS sang Dim\_HoaDon trong DDS

Graphical user interface, text, application

Description automatically generated with medium confidenceGraphical user interface, diagram, application

Description automatically generated

* Flow ETL từ SanPham\_NDS sang  Dim\_SanPham trong DDS

Graphical user interface, text, application

Description automatically generatedA screenshot of a phone

Description automatically generated with medium confidence

* **Bước 1**: Gán CET bằng thời gian khởi động package.
* **Bước 2:** Lấy LSET, CET từ dòng có Name\_Dataflow =  ‘Fact\_SanPham\_NDStoDDS’ trong bảng Metadata
* **Bước 3**: Rút trích dữ liệu bảng SanPham trong cơ sở dữ liệu NDS sang DDS: **Select \* from SanPham where MaTrangThai=1 and (NgayTao >= LSET and NgayTao < CET) or (NgayCapNhat >= LSET and NgayCapNhat < CET)**. Trong quá trình đổ sẽ xử lý dữ liệu của SanPham\_NDS để match với bảng Dim\_SanPham trong DDS, các bước xử lý như sau:
  + Kiểm tra xem Mã loại sản phẩm(MaLoaiSP) đã có trong bảng Dim\_LoaiSanPham trong cơ sở dữ liệu DDS chưa, nếu có thì đi tới bước tiếp theo. Nếu không có thì phần dữ liệu (tức là cột MaLoaiSP) sẽ được bỏ qua và hàng dữ liệu được chuyển đến đầu ra của phép chuyển đổi. Trong SISS sử dụng **Ignore failure** để có thể thực hiện được xử lý lúc không có MaLoaiSP trong bảng Dim\_LoaiSanPham.

Graphical user interface, application

Description automatically generated

* Kiểm tra xem Mã loại sản phẩm(MaLoaiSP) đã có trong bảng Dim\_LoaiSanPham trong cơ sở dữ liệu DDS chưa, nếu có thì đi tới bước tiếp theo. Nếu không có thì phần dữ liệu (tức là cột MaLoaiSP) sẽ được bỏ qua và hàng dữ liệu được chuyển đến đầu ra của phép chuyển đổi. Trong SISS sử dụng **Ignore failure** để có thể thực hiện được xử lý lúc không có MaLoaiSP trong bảng Dim\_LoaiSanPham.
* Do ở component trước tìm kiếm (Lookup) đều lựa chọn là “Ignore failure” nên dẫn đến việc giá trị các trường Lookup sẽ bị NULL nếu không thấy MaLoaiSP trong bảng Dim\_SanPham vì vậy nhóm quyết định sẽ sử dụng Conditional Split để lọc các dòng có giá trị null và đổ dữ liệu không có NULL vô bảng Dim\_SanPham trong DDS
* **Bước 4**: Cập nhật LSET = CET
* Tương tự các flow khi đổ dữ liệu  từ các bảng KhachHang\_NDS, CuaHang\_NDS cũng giống flow khi đổ dữ liệu từ SanPham\_NDS vào DDS
* Flow ETL từ KhachHang\_NDS sang Dim\_KhachHang trong DDS

Graphical user interface, text, application

Description automatically generatedA screenshot of a phone

Description automatically generated with medium confidence

* Flow ETL từ CuaHang\_NDS sang Dim\_CuaHang trong DDS

Text, application, chat or text message

Description automatically generatedA screenshot of a phone

Description automatically generated with medium confidence

* Flow ETL Fact\_DoanhThu trong DDS

A screenshot of a text message

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

* **Bước 1**: Gán CET bằng thời gian khởi động package.
* **Bước 2:** Lấy LSET, CET từ dòng có Name\_Dataflow =  ‘Fact\_CTHD\_NDStoDDS’ trong bảng Metadata
* **Bước 3**: Rút trích dữ liệu bảng ChiTietHoaDon trong cơ sở dữ liệu NDS sang DDS: **Select \*, 1+ABS(CHECKSUM(NEWID())) % 8037 as randomdate from ChiTietHoaDon where MaTrangThai=1 and (NgayTao >= LSET and NgayTao < CET) or (NgayCapNhat >= LSET and NgayCapNhat < CET)**. Câu truy vấn sau select \* còn có thêm “**1+ABS(CHECKSUM(NEWID())) % 8037 as randomdate”** để tạo giá trị khóa ngoại ngẫu nhiên tại cột SKDate để tham chiếu tới bảng DimDate (do trước đó nguồn dữ liệu không có sẵn các giá trị này trừ trường timestamp)**.** Trong quá trình ETL sẽ xử lý dữ liệu của ChiTietHoaDon\_NDS để match với bảng Fact\_DoanhThu trong DDS, các bước xử lý như sau:
* Kiểm tra xem Mã hoá đơn(MaHD) đã có trong bảng Dim\_HoaDon trong cơ sở dữ liệu DDS chưa, nếu có thì đi tới bước tiếp theo. Nếu không có thì phần dữ liệu (tức là cột MaHD) sẽ được bỏ qua và hàng dữ liệu được chuyển đến đầu ra của phép chuyển đổi. Trong SISS sử dụng **Ignore failure** để có thể thực hiện được xử lý lúc không có Mã hoá đơn trong bảng HoaDon\_DDS.

Graphical user interface, application

Description automatically generated

* Kiểm tra xem Mã sản phẩm(MaSP) đã có trong bảng Dim\_SanPham trong cơ sở dữ liệu DDS chưa, nếu có thì đi tới bước tiếp theo. Nếu không có thì phần dữ liệu (tức là cột MaSP) sẽ được bỏ qua và hàng dữ liệu được chuyển đến đầu ra của phép chuyển đổi.
* Tìm kiếm giá sản phẩm bằng cách truy vấn tới bảng SanPham\_NDS.
* Tìm kiếm Mã khách hàng và mã cửa hàng bằng cách truy vấn từ bảng HoaDon\_NDS
* Kiểm tra xem Mã cửa hàng(MaCH) đã có trong bảng Dim\_CuaHang trong cơ sở dữ liệu DDS chưa, nếu có thì đi tới bước tiếp theo. Nếu không có thì phần dữ liệu (tức là cột MaCH) sẽ được bỏ qua và hàng dữ liệu được chuyển đến đầu ra của phép chuyển đổi.
* Kiểm tra xem Mã khách hàng(MaKH) đã có trong bảng Dim\_KhachHang trong cơ sở dữ liệu DDS chưa, nếu có thì đi tới bước tiếp theo. Nếu không có thì phần dữ liệu (tức là cột MaKH) sẽ được bỏ qua và hàng dữ liệu được chuyển đến đầu ra của phép chuyển đổi.
* Do ở mỗi component tìm kiếm (Lookup) đều lựa chọn là “Ignore failure” nên dẫn đến việc giá trị các trường Lookup sẽ bị NULL nếu không thấy trong bảng cần tìm kiếm vì vậy nhóm quyết định sẽ sử dụng Conditional Split để lọc các dòng có  giá trị null ở các cột MaCH, MaKH, MaSP, MaHD và đổ dữ liệu không có NULL vô bảng Fact\_DoanhThu trong DDS
* **Bước 4**: Cập nhật LSET = CET

# Tài liệu tham khảo

<https://drive.google.com/file/d/1N6BIL4k77qrnY2HxsUee2-sBJkfq1wun/view?usp=sharing>