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

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**


# BÁO CÁO ĐỒ ÁN CUỐI KỲ

**Giảng viên hướng dẫn**

Cô Tiết Gia Hồng

Cô Hồ Thị Hoàng Vy

**Thành phố Hồ Chí Minh, ngày 23 tháng 12 năm 2023**

# MỤC LỤC

# I. Thông tin nhóm

## 1. Thông tin chi tiết nhóm

| STT | MSSV | Họ và tên | Email | Ghi chú |
|---|---|---|---|---|
| 1 | 20120542 | Trịnh Thị Tuyết Nhung | 20120542@student.hcmus.edu.vn | |
| 2 | 20120566 | Võ Ngọc Sơn | 20120566@student.hcmus.edu.vn | |
| 3 | 20120577 | Huỳnh Quốc Thái | 20120577@student.hcmus.edu.vn | |
| 4 | 20120590 | Nguyễn Trọng Thuận | 20120590@student.hcmus.edu.vn | Trưởng nhóm |

## 2. Bảng phân công và đánh giá

| Người thực hiện | Công việc thực hiện | Mức độ hoàn thành |
|---|---|---|
| Trịnh Thị Tuyết Nhung | • Nạp dữ liệu từ nguồn vào SQLServer<br>• ETL từ Source vào Stage<br>• Thiết kế, chuẩn hóa NDS<br>• ETL từ Stage vào NDS<br>• Thiết kế DDS<br>• OLAP & Report<br>• Tạo dashboard với PowerBI<br>• Viết và chỉnh sửa báo cáo | 100% |
| Võ Ngọc Sơn | • Mô tả dữ liệu từ Source<br>• Thiết kế, chuẩn hóa NDS<br>• Thiết kế DDS<br>• Tạo database: METADATA, Stage, NDS<br>• ETL từ Stage vào NDS<br>• OLAP & Report<br>• Viết báo cáo | 100% |

| | | |
|---|---|---|
| Huỳnh Quốc Thái | • Profiling Data, Clean Data<br>• Thiết kế, chuẩn hóa NDS<br>• Thiết kế DDS<br>• Tạo database DDS<br>• ETL từ NDS vào DDS<br>• OLAP & Report<br>• Viết báo cáo | 100% |
| Nguyễn Trọng Thuận | • Profiling Data, Clean Data<br>• Thiết kế, chuẩn hóa NDS<br>• Thiết kế DDS<br>• ETL từ NDS vào DDS<br>• OLAP & Report<br>• Tạo dashboard với PowerBI<br>• Data mining<br>• Quay video demo quá trình ETL<br>• Viết báo cáo | 100% |

# II. Nội dung báo cáo

## 1. Mô tả dữ liệu

### 1.1. Dữ liệu từ nguồn

**Bảng supermarket_sales**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | invoice_id | Mã hóa đơn | nvarchar(255) |
| 2 | branch | Chi nhánh | nvarchar(255) |
| 3 | customer_type | Loại khách hàng | nvarchar(255) |
| 4 | gender | Giới tính khách hàng | nvarchar(255) |
| 5 | product_id | Mã sản phẩm | nvarchar(255) |
| 6 | quantity | Số lượng mặt hàng đã mua | float |
| 7 | tax_5 | Thuế 5% | float |
| 8 | total | Tổng hóa đơn | float |
| 9 | date | Ngày mua | datetime |
| 10 | time | Thời gian mua | datetime |
| 11 | payment | Phương thức thanh toán | nvarchar(255) |
| 12 | cogs | Giá vốn hàng bán | float |
| 13 | gross_margin_percentage | Tỷ lệ lợi nhuận gộp | float |
| 14 | gross_income | Thu nhập gộp | float |
| 15 | rating | Đánh giá mua hàng | float |
| 16 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 17 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |

| | | | |
|---|---|---|---|
| 18 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng product_line**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | product_line | Tên dãy hàng | nvarchar(255) |
| 2 | product_line_id | ID dãy hàng | nvarchar(255) |
| 3 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 4 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 5 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng product**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | product_id | Mã sản phẩm | nvarchar(255) |
| 2 | unit_price | Đơn giá | float |
| 3 | product_line | Mã dãy hàng | nvarchar(255) |
| 4 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 5 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 6 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng branch**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | branch | Tên chi nhánh | nvarchar(255) |
| 2 | city | Tên thành phố | nvarchar(255) |
| 3 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 4 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 5 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

## 1.2. Dữ liệu trong NDS

**Bảng invoice**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | id | Khóa đại diện | int |
| 2 | nk_invoice_id | Khóa tự nhiên | varchar(50) |
| 3 | source_id | ID nguồn | int |
| 4 | product_id | Khóa ngoại trỏ đến bảng product | int |
| 5 | branch_id | Khóa ngoại trỏ đến bảng branch | int |
| 6 | payment_type_id | Khóa ngoại trỏ đến bảng payment_type | int |
| 7 | customer_type_id | Khóa ngoại trỏ đến bảng customer_type | int |
| 8 | gender | Giới tính | bit |
| 9 | quantity | Số lượng mặt hàng đã mua | int |
| 10 | tax_5_percent | Thuế 5% | float |
| 11 | total | Tổng hóa đơn | float |
| 12 | date | Ngày mua | datetime |

| | | | |
|---|---|---|---|
| 13 | time | Thời gian mua | datetime |
| 14 | cogs | Giá vốn hàng bán | float |
| 15 | gross_margin_percentage | Tỷ lệ lợi nhuận gộp | float |
| 16 | gross_income | Thu nhập gộp | float |
| 17 | rating | Đánh giá mua hàng | float |
| 18 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 19 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 20 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng branch**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | branch_id | Khóa đại diện | int |
| 2 | nk_branch_code | Khóa tự nhiên | varchar(10) |
| 3 | source_id | ID nguồn | smallint |
| 4 | city_name | Tên thành phố | nvarchar(500) |
| 5 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 6 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 7 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng Payment_type**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | payment_type_id | Khóa đại diện | int |

| | | | |
|---|---|---|---|
| 2 | nk_payment_type_code | Khóa tự nhiên | varchar(50) |
| 3 | source_id | ID nguồn | smallint |
| 4 | payment_type_name | Tên loại thanh toán | nvarchar(100) |
| 5 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 6 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 7 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng Customer_type**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | customer_type_id | Khóa đại diện | int |
| 2 | nk_customer_type_code | Khóa tự nhiên | varchar(50) |
| 3 | source_id | ID nguồn | smallint |
| 4 | customer_type_name | Tên loại khách hàng | nvarchar(100) |
| 5 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 6 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 7 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng Product**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | product_id | Khóa đại diện | int |
| 2 | nk_product_id | Khóa tự nhiên | varchar(50) |
| 3 | source_id | ID nguồn | smallint |

| | | | |
|---|---|---|---|
| 4 | product_line_id | Khóa ngoại trỏ đến bảng product_line | int |
| 5 | unit_price | Đơn giá | float |
| 6 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 7 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 8 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

**Bảng Product_line**

| STT | Thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|---|---|---|---|
| 1 | product_line_id | Khóa đại diện | int |
| 2 | nk_product_line_id | Khóa tự nhiên | varchar(50) |
| 3 | source_id | ID nguồn | smallint |
| 4 | product_line_name | Tên dãy hàng | nvarchar(200) |
| 5 | created_at | Thời gian tạo dữ liệu trong CSDL(*) | datetime |
| 6 | updated_at | Thời gian cập nhật cuối cùng trong CSDL(*) | datetime |
| 7 | is_deleted | Đã xóa mềm trong CSDL(*) | bit |

# 2. Data Profiling

## 2.1. Tổng quát

- Việc thực hiện data Profiling Data sẽ giúp chúng ta kiểm tra, thống kê và phân tích nhanh các dữ liệu trong nguồn. Số dòng trong các bảng, số dòng có dữ liệu khác biệt, dữ liệu rỗng hoặc bị thiếu.
- Sau khi tiến hành Profiling Data chúng ta cần chú ý những điều sau:
    - Ở bảng supermarket_sales có cột invoice_id có thể làm khóa chính. Chúng ta sử dụng invoice_id để làm khóa cho bảng này.

- Ở bảng supermarket_sales có cột gender sẽ có thể mang 1 trong 4 giá trị: Female, Male, F, M.
- Ở bảng supermarket_sales có cột date có 16 trường hợp ngày mua hàng không thuộc không khoảng từ tháng 1/2019 đến tháng 3/2019.

## 2.2. Thông số chi tiết:

**Bảng supermarket_sales**

| **Tiêu chí** | **Kết quả đánh giá** |
|---|---|
| **Đánh giá toàn bảng** | |
| Số dòng dữ liệu | 1016 |
| Số dòng dữ liệu không thỏa mãn | 16 |
| **Đánh giá theo cột** | |
| invoice_id | • Số giá trị duy nhất: 1016<br>• Số giá trị khác biệt: 0 |
| branch | • Các giá trị phân biệt và số lần xuất hiện:<br>  ○ A: 349<br>  ○ B: 338<br>  ○ C: 329 |
| customer_type | • Các giá trị phân biệt và số lần xuất hiện:<br>  ○ Member: 507<br>  ○ Normal: 509 |
| gender | • Các giá trị phân biệt và số lần xuất hiện:<br>  ○ F: 27<br>  ○ Female: 480<br>  ○ M: 25<br>  ○ Male: 484 |
| quantity | • Số dòng có giá trị dương và là số nguyên: 1016 |

| | |
|---|---|
| tax_5 | • Số dòng có giá trị không âm: 1016 |
| total | • Số dòng có giá trị không âm: 1016 |
| date | • Số giá trị thỏa điều kiện nằm trong khoảng thời gian từ tháng 1/2019 đến tháng 3/2019: 1000 |
| time | • Số giá trị thỏa điều kiện nằm trong khoảng từ 10 giờ sáng đến 21 giờ: 1016 |
| payment | • Các giá trị phân biệt và số lần xuất hiện:<br>  ○ Cash: 348<br>  ○ Credit card: 316<br>  ○ Ewallet: 352 |
| cogs, gross_margin_percentage, gross_income | • Số dòng có 1 trong 3 giá trị này nhỏ hơn 0: 0 |
| rating | • Số dòng có rating nằm ngoài khoảng [1, 10] hoặc mang giá trị không phải số: 0 |

**Bảng product**

| **Tiêu chí** | **Kết quả đánh giá** |
|---|---|
| **Đánh giá toàn bảng** | |
| Số dòng dữ liệu | 943 |
| Số dòng dữ liệu không thỏa mãn | 0 |
| **Đánh giá theo cột** | |
| product_id | • Số giá trị duy nhất: 943<br>• Số giá trị khác biệt: 0 |
| unit_price | • Số dòng có unit_price nhỏ hơn 0: 0 |
| product_line | • Số dòng mang giá trị không hợp |

| | |
|---|---|
| | lệ hoặc khác biệt: 0 |

**Bảng product_line**

| **Tiêu chí** | **Kết quả đánh giá** |
|---|---|
| **Đánh giá toàn bảng** | |
| Số dòng dữ liệu | 6 |
| Số dòng dữ liệu không thỏa mãn | 0 |
| **Đánh giá theo cột** | |
| product_line_id | • Số giá trị duy nhất: 6<br>• Số giá trị khác biệt: 0 |
| product_line | • Số dòng mang giá trị không hợp lệ hoặc khác biệt: 0 |

**Bảng branch**

| **Tiêu chí** | **Kết quả đánh giá** |
|---|---|
| **Đánh giá toàn bảng** | |
| Số dòng dữ liệu | 3 |
| Số dòng dữ liệu không thỏa mãn | 0 |
| **Đánh giá theo cột** | |
| branch | • Số giá trị duy nhất: 3<br>• Số giá trị khác biệt: 0 |
| city | • Số dòng mang giá trị không hợp lệ hoặc khác biệt: 0 |

## 3. Clean Data

- Trong quá trình profiling chỉ phát hiện dữ liệu không hợp lệ tại cột **date** của bảng supermarket_salse nên trong quá trình rút trích cần phải thêm điều kiện cho date như sau:
    - Sử dụng Conditional Split với điều kiện “invalid date value” như sau: YEAR(date) != 2019 || MONTH(date) > 3 || MONTH(date) < 1
- Ngoài ra cũng cần tạo điều kiện cho các trường dữ liệu khác nhằm đảm bảo khi có dữ liệu mới thì nếu có các dữ liệu không hợp lệ sẽ được loại bỏ khi ETL vào NDS:
    - Bảng Invoice:

| Order | Output Name | Condition |
|---|---|---|
| 1 | NK is null | ISNULL(invoice_id) |
| 2 | Quantity is null or less than 1 | ISNULL(quantity) \|\| quantity < 1 |
| 3 | Date is null | ISNULL(date) |
| 4 | Time is null | ISNULL(time) |
| 5 | Gross income is null | ISNULL(gross_income) |
| 6 | Invalid gender | !(processed_origin_gender == "female" \|\| processed_origin_gender == "f" \|\| processed_origin_gender == "male" \|\| processed_origin_gender == "m") |
| 7 | invalid date value | YEAR(date) != 2019 \|\| MONTH(date) > 3 \|\| MONTH(date) < 1 |
| 8 | invalid time value | DATEPART("hh",time) < 10 \|\| DATEPART("hh",time) > 21 |

    - Bảng Branch:

| Order | Output Name | Condition |
|---|---|---|
| 1 | NK is null | ISNULL(nk_branch_code) |

    - Bảng Payment Type: N/A
    - Bảng Customer Type: N/A
    - Bảng Product Line:

| Order | Output Name | Condition |
|---|---|---|
| 1 | Invalid Rows | ISNULL(product_line_id) |

    - Bảng Product:

| Order | Output Name | Condition |
|---|---|---|
| 1 | Invalid Rows | ISNULL(product_id) |

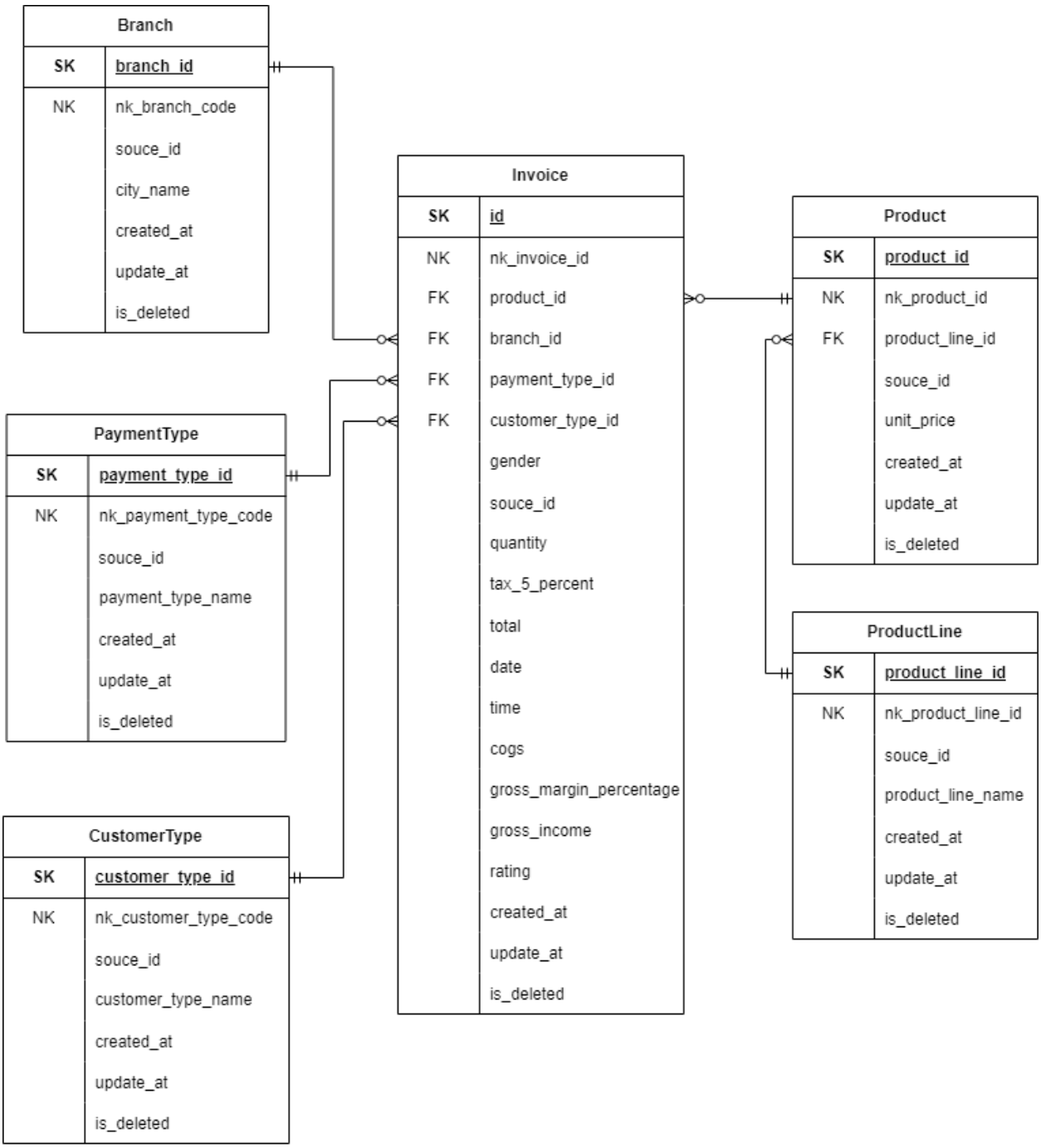
- Đồng thời cũng loại bỏ các dòng bị trùng lặp trên khóa.

## 4. Thiết kế NDS

"Tư duy" thiết kế:

- Dựa tên tập dữ liệu trong bảng supermaket_sales tại Stage ta rút ra được các điều sau:
  - Cột "branch" tồn tại các giá trị trùng lắp, về mặt ý nghĩa "branch" thể hiện thông tin về chi nhánh – là nơi mà sự kiện mua bán xảy ra.
  - Cột "customer_type" tồn tài các giá trị trùng lắp, về mặt ý nghĩa "customer_type" thể hiện thông tin về loại khách hàng đã thanh toán hóa đơn đó.
  - Tương tự, "gender" thể hiện giới tính của khách hàng.
  - "product_id" biểu thị thông tin sản phẩm được mua bởi hóa đơn đó. Ngoài ra, còn có bảng "product" thể hiện thông tin chi tiết của các sản phẩm.
  - "payment" cho biết hóa đơn đó được thanh toán bằng hình thức nào.
- Dựa trên các thông tin vừa rút ra, ta có thể tách "branch", "customer_type", "product", "payment" thành các bảng con được tham chiếu bởi bảng "invoice".
- Riêng "gender" cũng có thể tách thành một bảng riêng, tuy nhiên trong lưu trữ thực tế "gender" cũng chỉ chưa một trong các cặp giá trị (0, 1), (f, m),… thể hiện giới tính nam hoặc nữ. Cho nên ta có thể coi "gender" là một cột trong bảng "invoice" chỉ chưa giá trị true hoặc false.

Ta có được lược đồ NDS sau:




# 5. Thiết kế DDS

## 5.1. Phân tích yêu cầu

### a. Yêu cầu 1

Yêu cầu 1: Thống kê số lượt mua hàng theo ngày, tháng, năm

- Sự kiện: Khi có thêm 1 hóa đơn.

- Bối cảnh:
    - Khi nào: Ngày, tháng, năm.
  - Đo lường (dữ kiện): Số thành viên mua hàng.

### b. Yêu cầu 2

Yêu cầu 2: Thống kê tổng doanh thu của khách hàng (member, normal) theo ngày, tháng, năm và theo chi nhánh.

- Sự kiện: Khi có thêm 1 hóa đơn.
- Bối cảnh:
    - Ở đâu: Chi nhánh.
    - Khi nào: Ngày, tháng, năm.
    - Loại khách hàng nào: Loại khách hàng.
- Đo lường (dữ kiện): Doanh thu.

### c. Yêu cầu 3

Yêu cầu 3: Thống kê số lượt khách thanh toán theo cash/ debit/... ở từng chi nhánh theo từng tháng trong năm.

- Sự kiện: Khi có thêm 1 hóa đơn.
- Bối cảnh:
    - Ở đâu: Chi nhánh.
    - Khi nào: Tháng, năm.
    - Thanh toán bằng phương thức nào: Phương thức thanh toán
- Đo lường (dữ kiện): Số lượng khách thanh toán.

### d. Yêu cầu 4

Yêu cầu 4: Thống kê lượng rating của khách hàng (member, normal) theo từng loại sản phẩm (ProductLine)

- Sự kiện: Khi có thêm 1 hóa đơn.
- Bối cảnh:
    - Dòng sản phẩm nào: Dòng sản phẩm
    - Loại khách hàng nào: Loại khách hàng.

- Đo lường (dữ kiện): Lượng rating.

### e. Yêu cầu 5

Yêu cầu 5: Thống kê số lượng sản phẩm bán được theo từng thời điểm (time / date)

- Sự kiện: Khi có thêm 1 hóa đơn.
- Bối cảnh:
  - Khi nào: Ngày
- Đo lường (dữ kiện): Số lượng sản phẩm bán được.

### f. Yêu cầu 6

Yêu cầu 6: Thống kê số lượt khách hàng nữ đã mua theo từng loại sản phẩm

- Sự kiện: Khi có thêm 1 hóa đơn.
- Bối cảnh:
  - Loại sản phẩm nào: Loại sản phẩm.
- Đo lường (dữ kiện): Số lượng khách hàng nữ đã mua.

## 5.2. Mô hình hóa

- Fact table: Fact_Purchase
  - Khóa:
    - product id (PK,FK)
    - branch id (PK,FK)
    - customer type id (PK,FK)
    - payment type id (PK,FK)
    - datetime id (PK,FK)
  - Giá trị có sẵn từ nguồn: quantity, unit price, COGS.
  - Giá trị cần tính toán:
    - Tổng số sản phẩm đã bán
    - Tổng số lần mua hàng
    - Tổng thu nhập

        - Tổng điểm đánh giá
        - Tổng số khách hàng nữ
- Dimension table:
    - Dim_Product
    - Dim_ProductLine
    - Dim_Branch
    - Dim_CustomerType
    - Dim_PaymentType
    - Dim_Datetime
    - Dim_Day
    - Dim_Month
    - Dim_Year

## 5.3. Chiều thoái hóa

- Chiều 1: Dim_Datetime → Dim_Day →Dim_Month → Dim_Year
- Chiều 2: Dim_Product → Dim_ProductLine

## 5.4. Thiết kế chiều

### a. Dim_Product

- Sử dụng SCD 1 - Ghi chồng giá trị cũ vì không cần lưu giá trị cũ khi có thay đổi dữ liệu chỉ cần ghi đè lên.
- Chuyển product từ bảng product của NDS thêm vào Dim_Product

### b. Dim_ProductLine

- Sử dụng SCD 1 - Ghi chồng giá trị cũ vì không cần lưu giá trị cũ khi có thay đổi dữ liệu chỉ cần ghi đè lên.
- Chuyển Product Line từ bảng product_line của NDS thêm vào Dim_ProductLine

### c. Dim_Branch

- Sử dụng SCD 1 - Ghi chồng giá trị cũ vì không cần lưu giá trị cũ khi có thay đổi dữ liệu chỉ cần ghi đè lên.
- Chuyển Branch từ bảng branch của NDS thêm vào Dim_Branch

### d. Dim_CustomeType

- Sử dụng SCD 1 - Ghi chồng giá trị cũ vì không cần lưu giá trị cũ khi có thay đổi dữ liệu chỉ cần ghi đè lên.
- Chuyển Custome Type từ bảng custome_type của NDS thêm vào Dim_ CustomeType

### e. Dim_PaymentType

- Sử dụng SCD 1 - Ghi chồng giá trị cũ vì không cần lưu giá trị cũ khi có thay đổi dữ liệu chỉ cần ghi đè lên.
- Chuyển PaymentType từ bảng payment_type của NDS thêm vào Dim_ PaymentType

### f. Dim_Datetime

- Không dùng chiều thay đổi chậm
- Rút trích hour từ cột time của bảng invoice thêm vào Dim_Datetime

### g. Dim_Date

- Không dùng chiều thay đổi chậm
- Rút trích date từ cột date của bảng invoice thêm vào Dim_Date

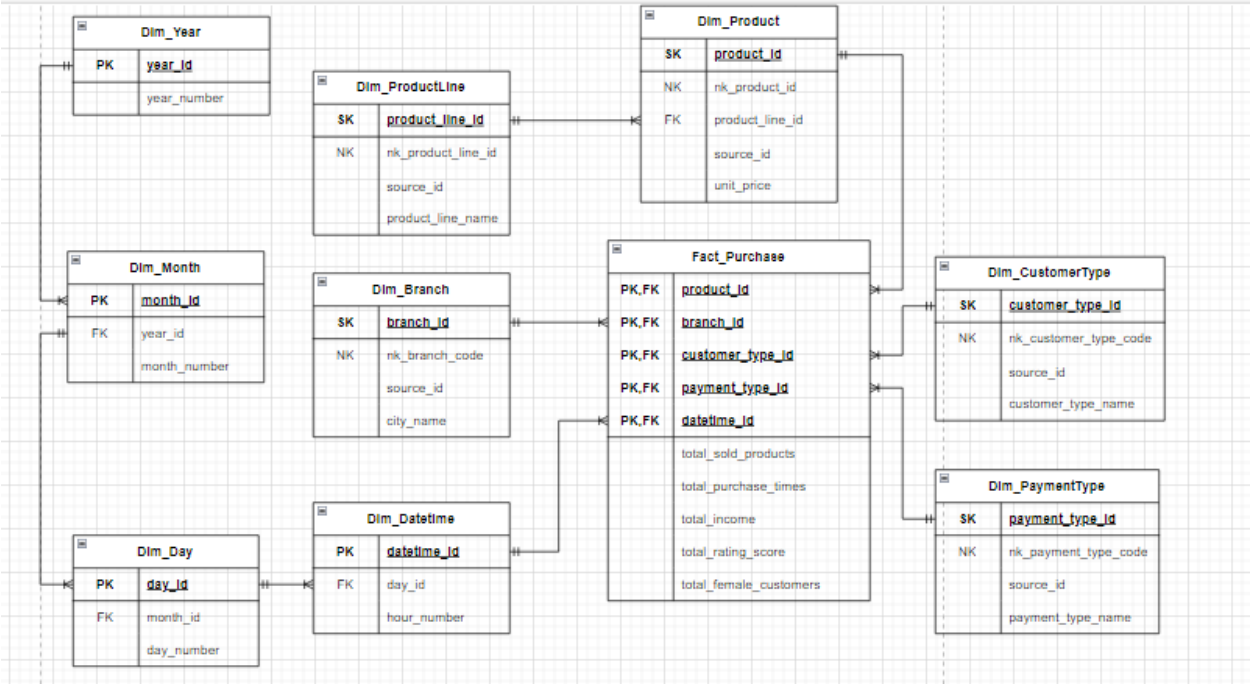
### h. Dim_Month

- Không dùng chiều thay đổi chậm
- Rút trích month từ cột date của bảng invoice thêm vào Dim_Month

### i. Dim_Year

- Không dùng chiều thay đổi chậm
- Rút trích year từ cột date của bảng invoice thêm vào DimYear

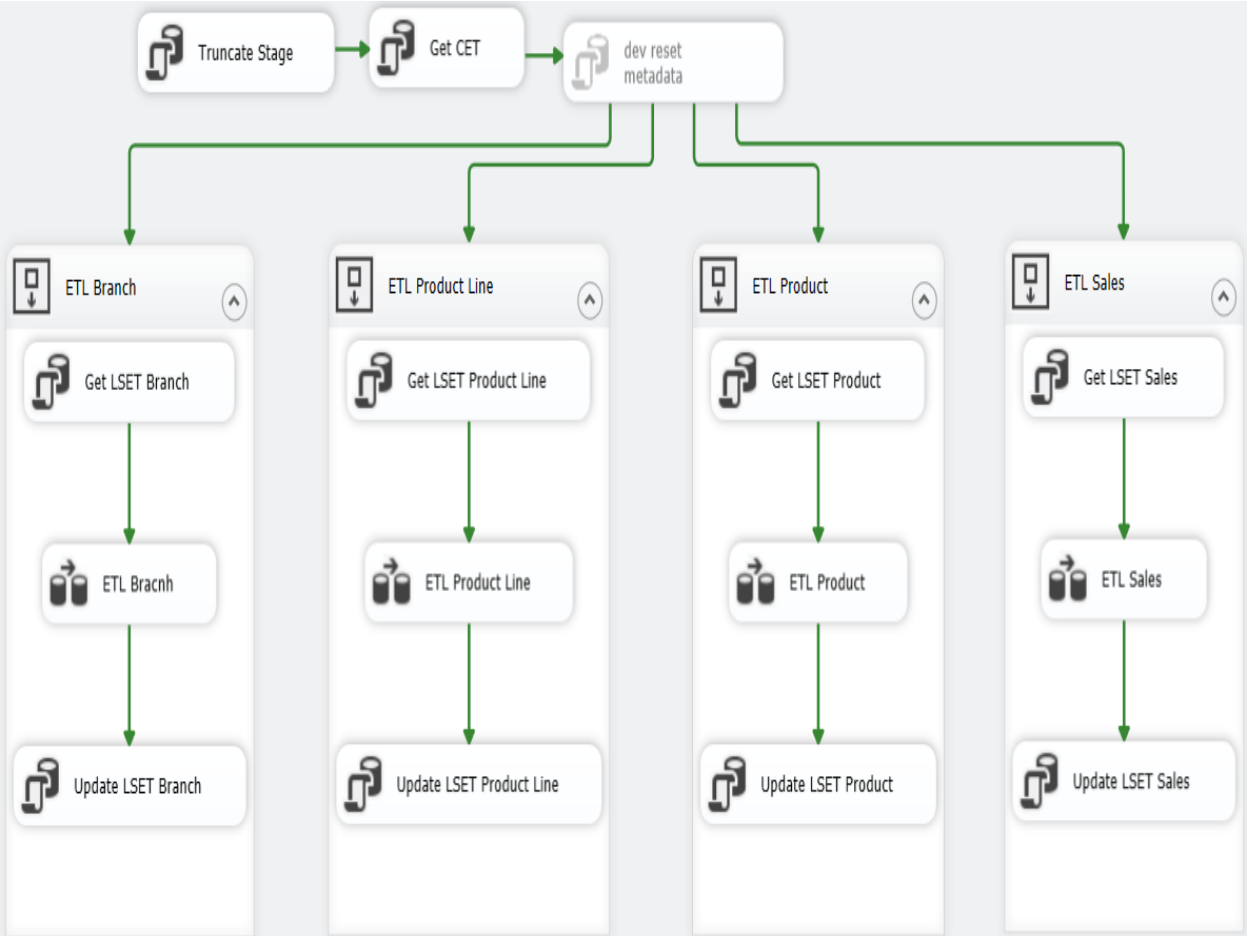
### 5.5. Lược đồ DDS bông tuyết




# 6. Quá trình ETL

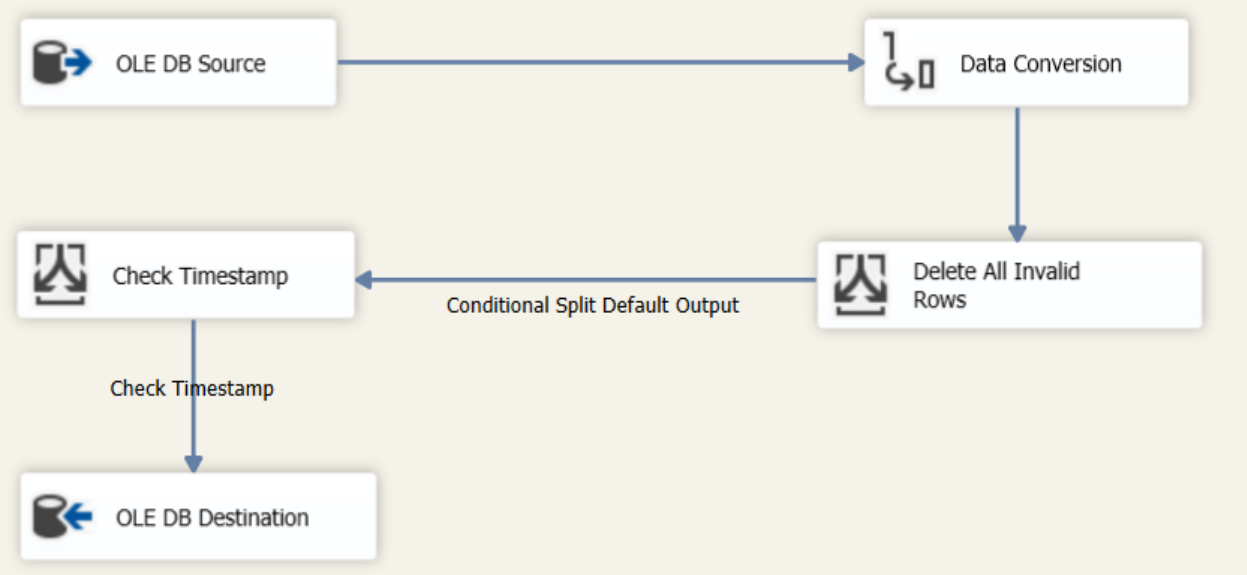
## 6.1. ETL từ Source vào Stage

- Quá trình thực hiện ETL từ Source vào Stage của các bảng là tương tự nhau.
- Dưới đây là mô tả của quá trình ETL từ Source vào Stage:
  - Đầu tiên, thực hiện truncate tất cả các bảng trong Stage.
  - Sau đó lấy CET từ CSDL Metadata.
  - Cuối cùng, thực hiện quy trình ETL dữ liệu từ source vào stage.



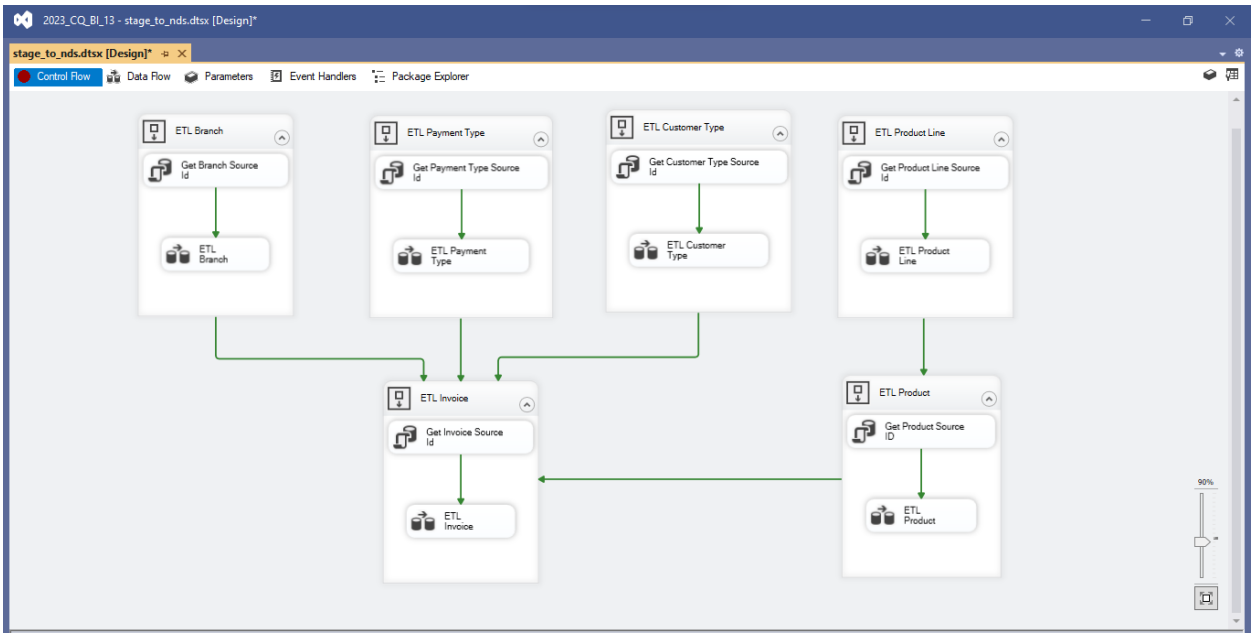

- Quá trình ETL các bảng có cơ chế tương tự nhau:
    - Đầu tiên, lấy LSET của bảng tương ứng từ Metadata và lưu vào biến đã tạo sẵn trong SSIS: SELECT lset AS LSET FROM stage_etl_info WHERE table_name = <table_name>.
    - Trong bước ETL sẽ thực hiện lần lượt như sau:
        - Rút dữ liệu từ nguồn tương ứng.
        - Chuyển đổi kiểu dữ liệu, tên biến.
        - Xóa các dòng không hợp lệ.
        - Chọn các dòng thỏa điều kiện rút trích, ở đây sử dụng incremental extract nên điều kiện là: (_created_at < @[User::CET] && _created_at >= @[User::LSET]) || (_updated_at < @[User::CET] && _updated_at >= @[User::LSET]).

- Cuối cùng, load dữ liệu vào bảng tương ứng trong stage và update lại LSET của bảng đó trong metadata.



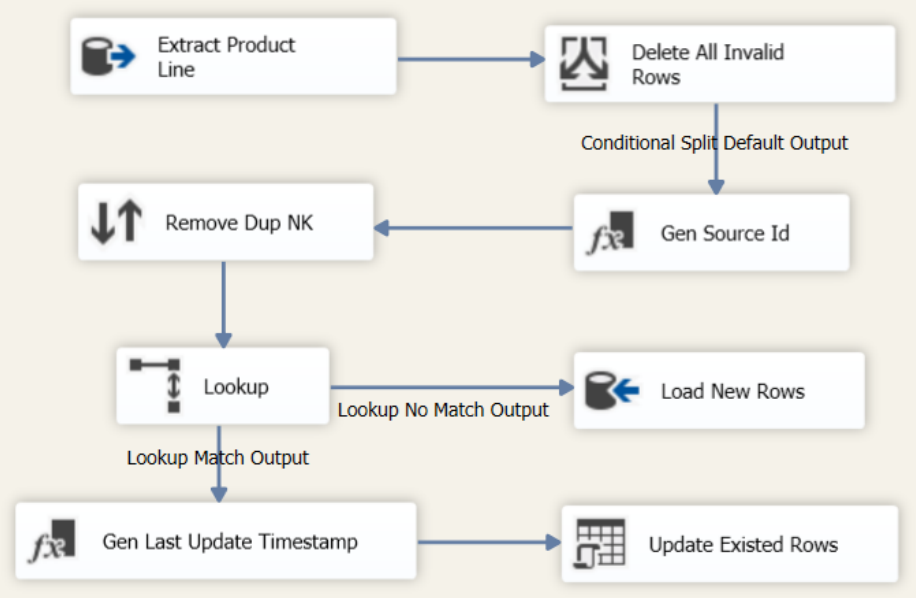

## 6.2. ETL từ Stage vào NDS

- Quá trình ETL từ Stage vào NDS sẽ được thực hiện ở các bảng không chứa khóa ngoại trước, các bảng chứa khóa ngoại sẽ thực hiện sau khi đã hoàn thành với bảng tham chiếu tới.



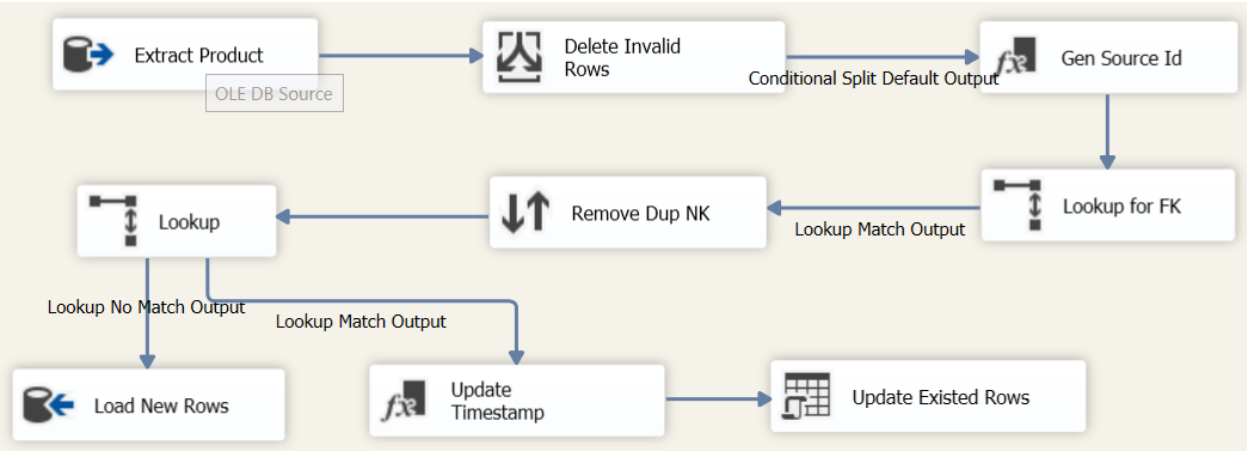
- ETL một bảng từ Stage vào NDS sẽ xảy ra một trong 2 trường hợp:
    - Bảng không có khóa ngoại.
    - Bảng có khóa ngoại.
- Lấy ví dụ ETL bảng product_line cho trường hợp bảng không có khóa ngoại:
    - Đầu tiên ta cần lấy ra source ID của bảng từ Metadata và lưu vào biến đã tạo sẵn: SELECT id AS SourceId FROM nds_etl_info WHERE table_name = 'product_line'.
    - Sau đó đến quá trình ETL:
        - Ta sẽ rút dữ liệu từ nguồn tương ứng.
        - Sau đó xóa tất cả các cột không hợp lệ.
        - Tạo một cột mới là source_id có giá trị chính là source ID vừa lấy.
        - Xóa các dòng bị trùng lặp.
        - Tiếp theo sử dụng Natural key và Source ID (nếu ETL từ nhiều nguồn) để tìm kiếm lịch sử của dòng đó trong NDS:
            - Nếu đã tồn tại dòng dữ liệu tương ứng thì ta phát sinh cột _last_update_time cho biết thời gian cập nhật cuối cùng của dòng đó và cập nhật dòng đó trong NDS.

- Nếu chưa tồn tại thì ta tạo một dòng mới trong NDS.




- Lấy ví dụ bảng product cho trường hợp bảng có khóa ngoại:
  - Tương tự như trên, đầu tiên ta cần lấy ra source ID của bảng từ Metadata và lưu vào biến đã tạo sẵn: SELECT id AS SourceId FROM nds_etl_info WHERE table_name = 'product_line'.
  - Sau đó đến quá trình ETL:
    - Ta sẽ rút dữ liệu từ nguồn tương ứng.
    - Sau đó xóa tất cả các cột không hợp lệ.
    - Tạo một cột mới là source_id có giá trị chính là source ID vừa lấy.
    - Đây là bước khác như với bảng product_line, tại đây ta cần lookup tất cả các khóa ngoại của bảng. Output sẽ nhận các dòng thỏa mãn rằng các khóa ngoại của nó đều được tìm thấy trong NDS.
    - Xóa các dòng bị trùng lặp.
    - Tiếp theo sử dụng Natural key và Source ID (nếu ETL từ nhiều nguồn) để tìm kiếm lịch sử của dòng đó trong NDS:

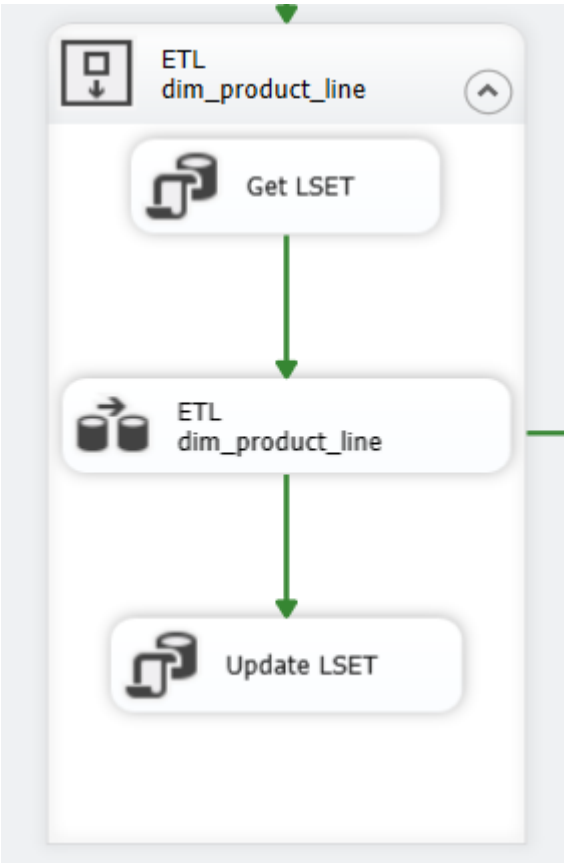
- Nếu đã tồn tại dòng dữ liệu tương ứng thì ta phát sinh cột _last_update_time cho biết thời gian cập nhật cuối cùng của dòng đó và cập nhật dòng đó trong NDS.
- Nếu chưa tồn tại thì ta tạo một dòng mới trong NDS.



## 6.3. ETL từ NDS vào DDS

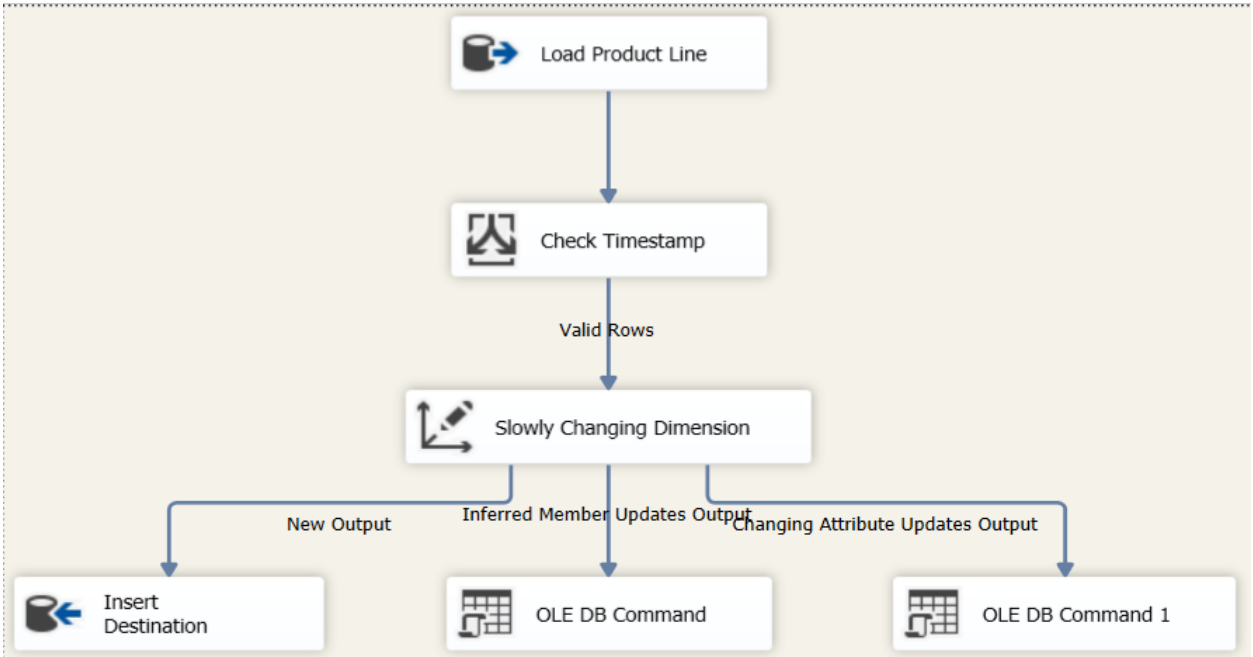
### a. ETL các bảng chiều không có phân cấp chiều (Dim)

- Các bước tương tự như từ Source vào Stage (Không có bước Truncate table).




- Cả quá trình như sau:
  - Đầu tiên, lấy CET từ Metadata vào lưu vào biến đã tạo sẵn. Đây là bước toàn cục của quá trình ETL từ NDS và DDS.
  - Tiếp theo lấy LSET của bảng tương ứng từ Metadata lưu vào biến đã tạo sẵn: SELECT lset AS LSET FROM dds_etl_info WHERE table_name = 'dim_product'
  - Sau đó ta rút dữ liệu từ bảng tương ứng trong NDS.
  - Kiểm tra và lấy ra các dòng thỏa điều kiện rút trích.
  - Chuyển đổi kiểu dữ liệu, tên cột, phát sinh cột,... nếu có.

- Xác định chiều thay đổi của dữ liệu để có cách Load dữ liệu tương ứng.



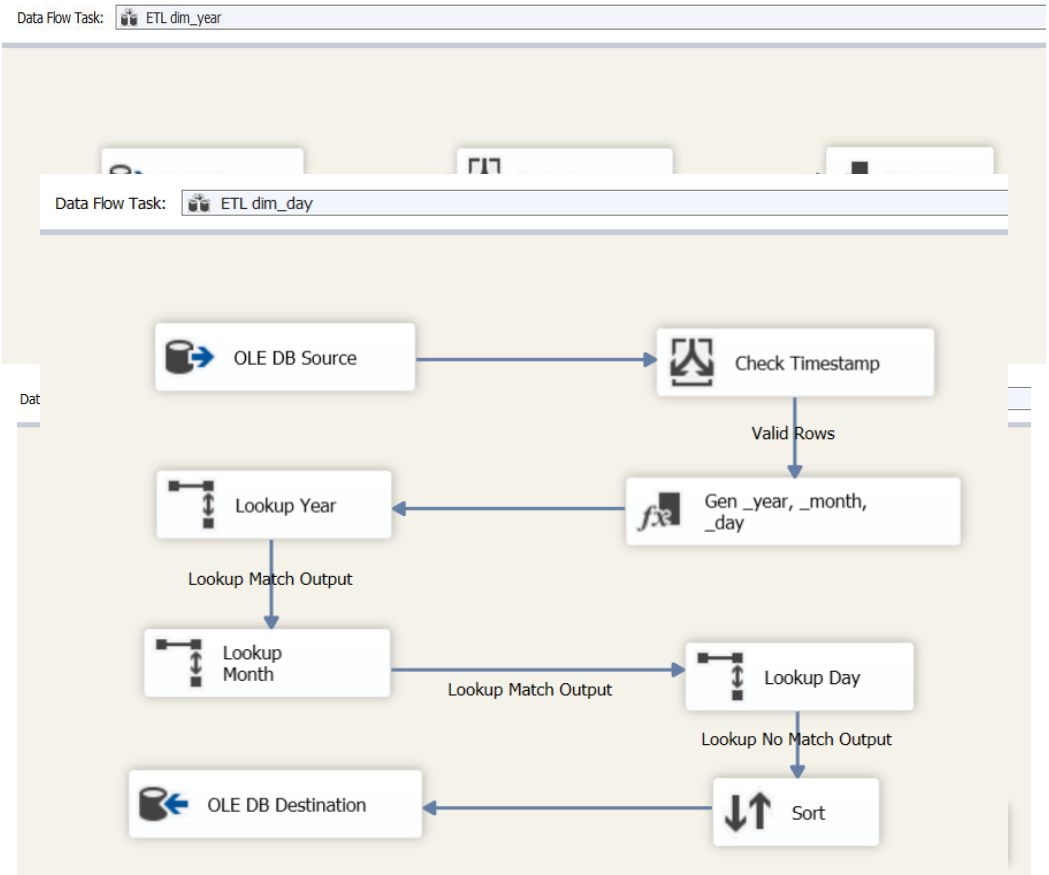

- Update lại LSET.

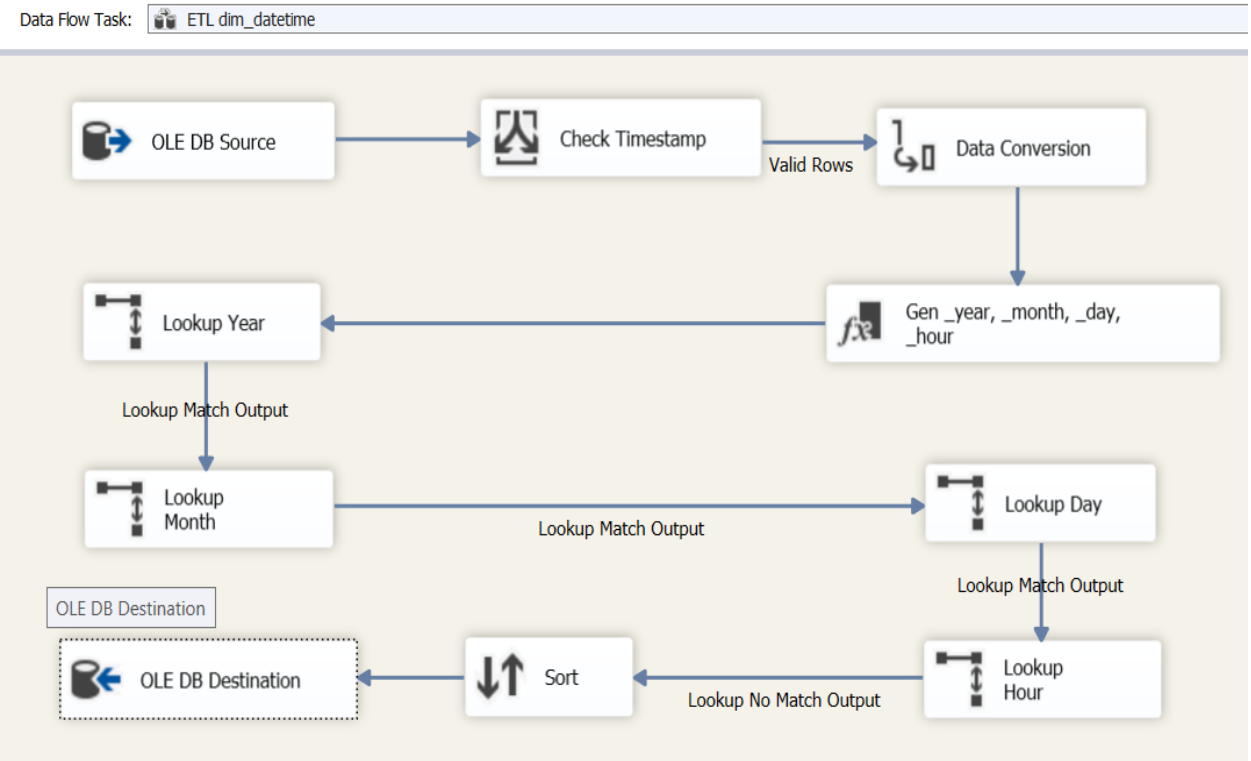
### b. ETL các bảng chiều có phân cấp (chiều thời gian, địa lý)

- Thực hiện ETL chiều theo thứ tự có phân cấp từ cao nhất đến thấp nhất: dim_year → dim_month → dim_day → dim_datetime
- Quá trình ETL các chiều thời gian:
  - Đầu tiên, lấy CET từ Metadata vào lưu vào biến đã tạo sẵn.
  - Tiếp theo lấy LSET của bảng tương ứng từ Metadata lưu vào biến đã tạo sẵn: SELECT lset AS LSET FROM dds_etl_info WHERE table_name = <table_name>
  - Rút trích dữ liệu từ nguồn tương ứng (cột date và time trong bảng invoice).
  - Kiểm tra và lấy ra các dòng thỏa dấu thời gian theo cơ chế rút trích.
  - Tạo các cột biểu thị thời gian cần thiết:
    - dim_year: tạo cột _year sử dụng Derived Column với hàm YEAR((DT_DBDATE)date)

- dim_month: tạo cột _year, _month sử dụng Derived Column với hàm YEAR((DT_DBDATE)date), MONTH((DT_DBDATE)date)
- dim_day: tạo cột _year, _month, _day sử dụng Derived Column với hàm YEAR((DT_DBDATE)date), MONTH((DT_DBDATE)date), DAY((DT_DBDATE)date)
- dim_datetime: tạo cột _year, _month, _day, _hour sử dụng Derived Column với hàm YEAR((DT_DBDATE)date), MONTH((DT_DBDATE)date), DAY((DT_DBDATE)date), DATEPART("hh",(DT_DBTIME)updated_time)

- Look up theo các cột thời gian vừa phát sinh. Output nhận là các dòng thỏa mãn rằng các phép loop uk là có dữ liệu trong DDS.
- Sử dụng Soft component để xóa các cột bị trùng lặp.
- Cuối cùng, Load dữ liệu vào DDS và cập nhật lại LSET.

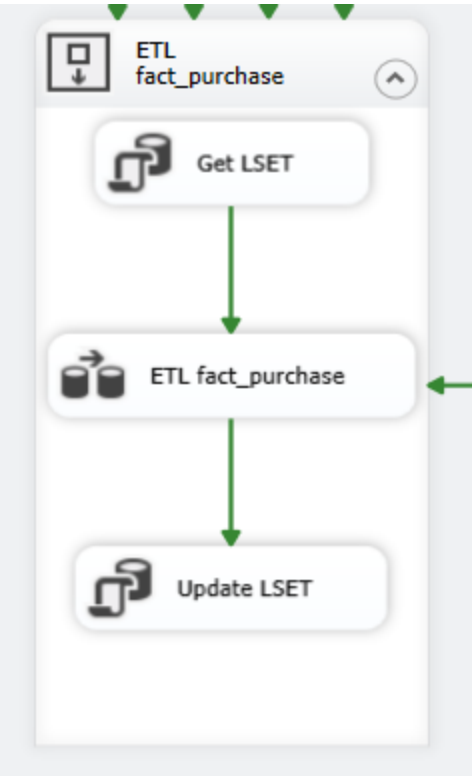





### c. ETL các bảng Fact

- Sau khi ETL xong các bảng chiều cần thiết sẽ đến quá trình ETL bảng Fact.
- Tương tự như các quá trình ETL các bảng chiều:

- Ta cũng bắt đầu bằng việc lấy LSET, CET từ Metadata và lưu vào các biến đã tạo.



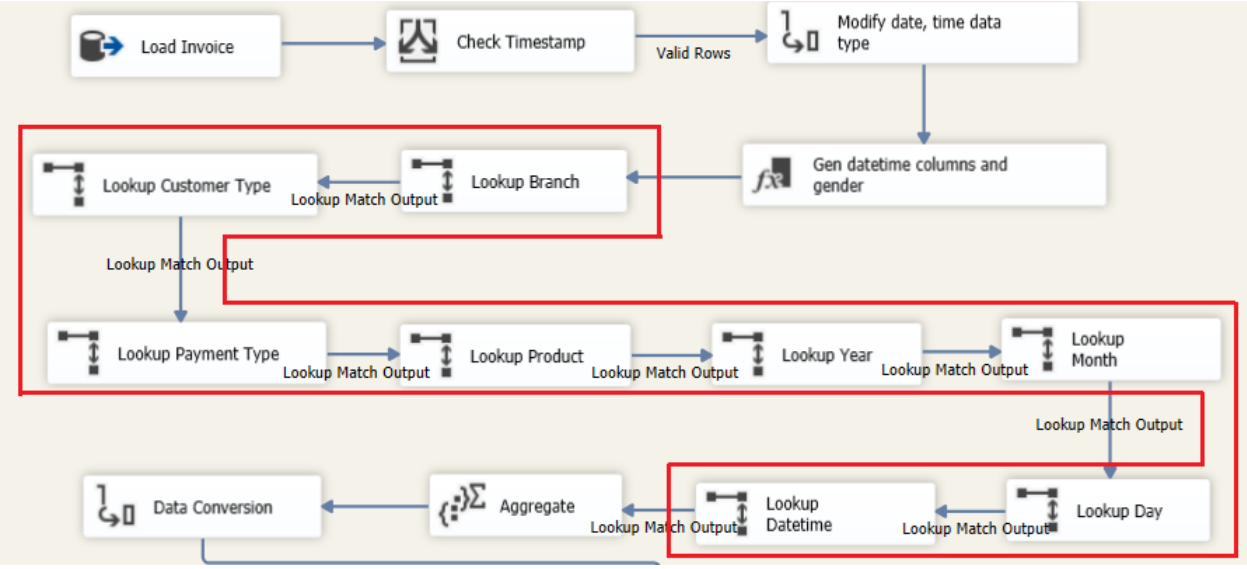

- Sau đó là đến bước Load dữ liệu từ bảng tương ứng trong NDS.
- Tiếp theo là kiểm tra dấu thời gian và lấy ra các dòng thỏa mãn theo cơ chế rút trích: (_created_at < @[User::CET] && _created_at >= @[User::LSET]) || (_updated_at < @[User::CET] && _updated_at >= @[User::LSET])
- Thực hiện biến đổi kiểu dữ liệu, tên cột nếu cần.

| Input Column | Output Alias | Data Type | L |
|---|---|---|---|
| date | _date | date [DT_DATE] | |
| time | _time | database time with pr... | |

- Tạo các cột _year, _month, _day, _hour cho mục đích lookup theo thời gian mua hàng, tạo cột _is_female cho mục đích đếm số khách hàng nữ.

| Derived Column Name | Derived Column | Expression |
|---|---|---|
| _year | <add as new column> | YEAR(_date) |
| _month | <add as new column> | MONTH(_date) |
| _day | <add as new column> | DAY(_date) |
| _hour | <add as new column> | DATEPART("hh",(DT_DBTIME)_time) |
| _is_female | <add as new column> | !gender ? 1 : 0 |

- Tiếp đến ta look up theo tất cả các khóa ngoại và chiều thời gian.




- Tiếp đến là bước tạo ra các cột Measure, ta sử dụng Aggregate component, group by theo các khóa ngoại:
  - total_sold_products: Được tính toán dựa trên việc cộng quantity ở tất cả các hóa đơn.
  - total_purchase_times: Được tính toán dựa trên việc đến số hóa đơn (ID hóa đơn) riêng biệt.
  - total_income: Được tính toán dựa trên việc cộng gross_income ở tất cả các hóa đơn.

- total_rating_score: Được tính toán dựa trên tính tổng rating ở tất cả các hóa đơn.
- total_female_customers: Như có đề cập ở trên, hóa đơn đơn nào được mua bởi khách hàng nữ thì sẽ được phát sinh cột _is_female với giá trị là 1, còn lại sẽ là 0. Vì vậy, để tính số khách hàng là nữ ta chỉ cần tính tổng các cột _is_famle.

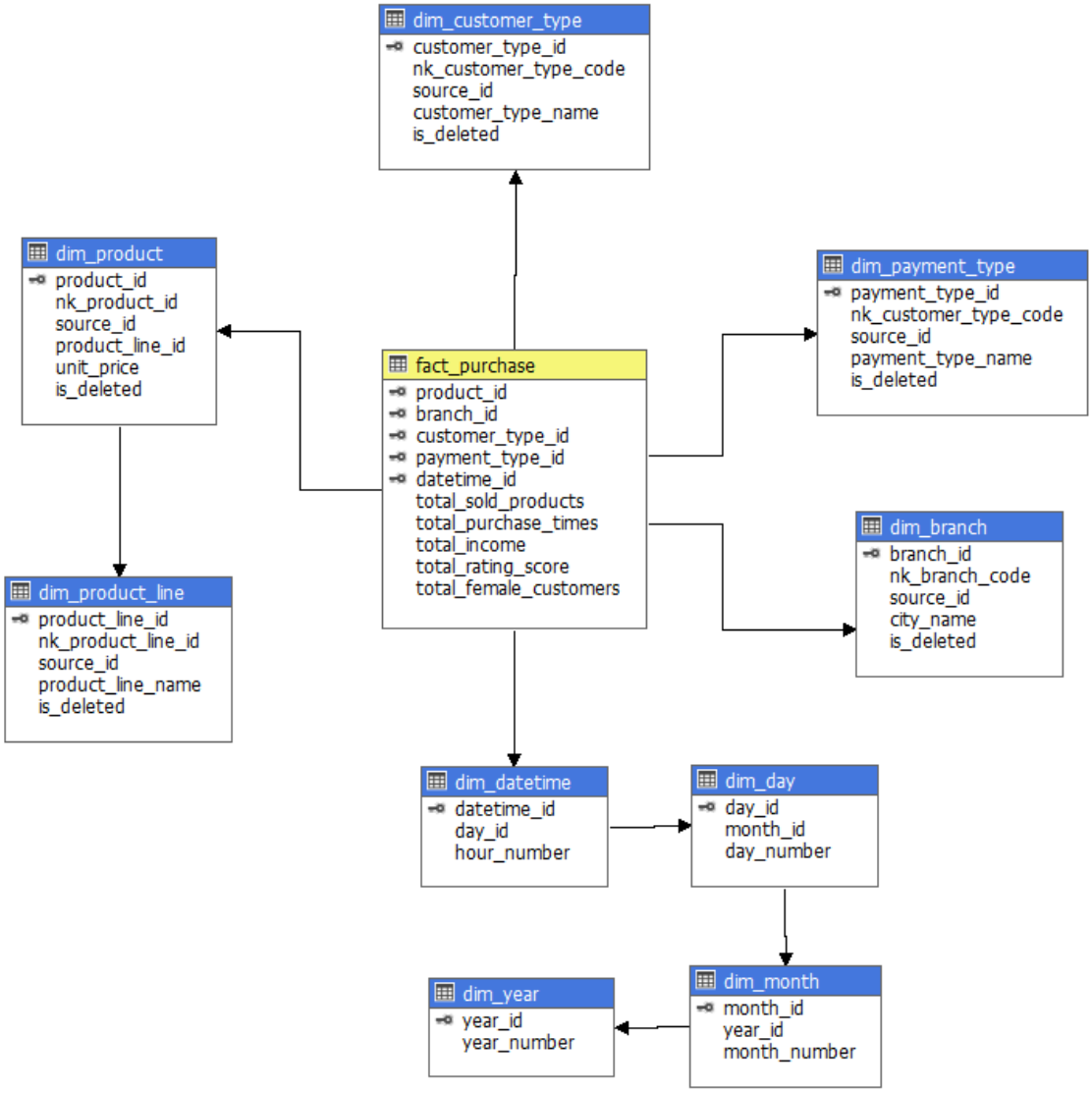
| Input Column | Output Alias | Operation |
|---|---|---|
| datetime_id | datetime_id | Group by |
| product_id | product_id | Group by |
| branch_id | branch_id | Group by |
| payment_type_id | payment_type_id | Group by |
| customer_type_id | customer_type_id | Group by |
| quantity | total_sold_products | Sum |
| id | total_purchase_times | Count distinct |
| gross_income | total_income | Sum |
| rating | total_rating_score | Sum |
| _is_female | total_female_customers | Sum |

- Tương tự, ta xác định loại chiều thay đổi sao cho phù hợp.
- Cuối cùng, Load dữ liệu vào DDS và update lại LSET.

# 7. Khai thác Kho dữ liệu

## 7.1. OLAP

Nhóm sử dụng một CUBE duy nhất cho tất các các yêu cầu phân tích:




Trong đó có 2 phân cấp chiều:

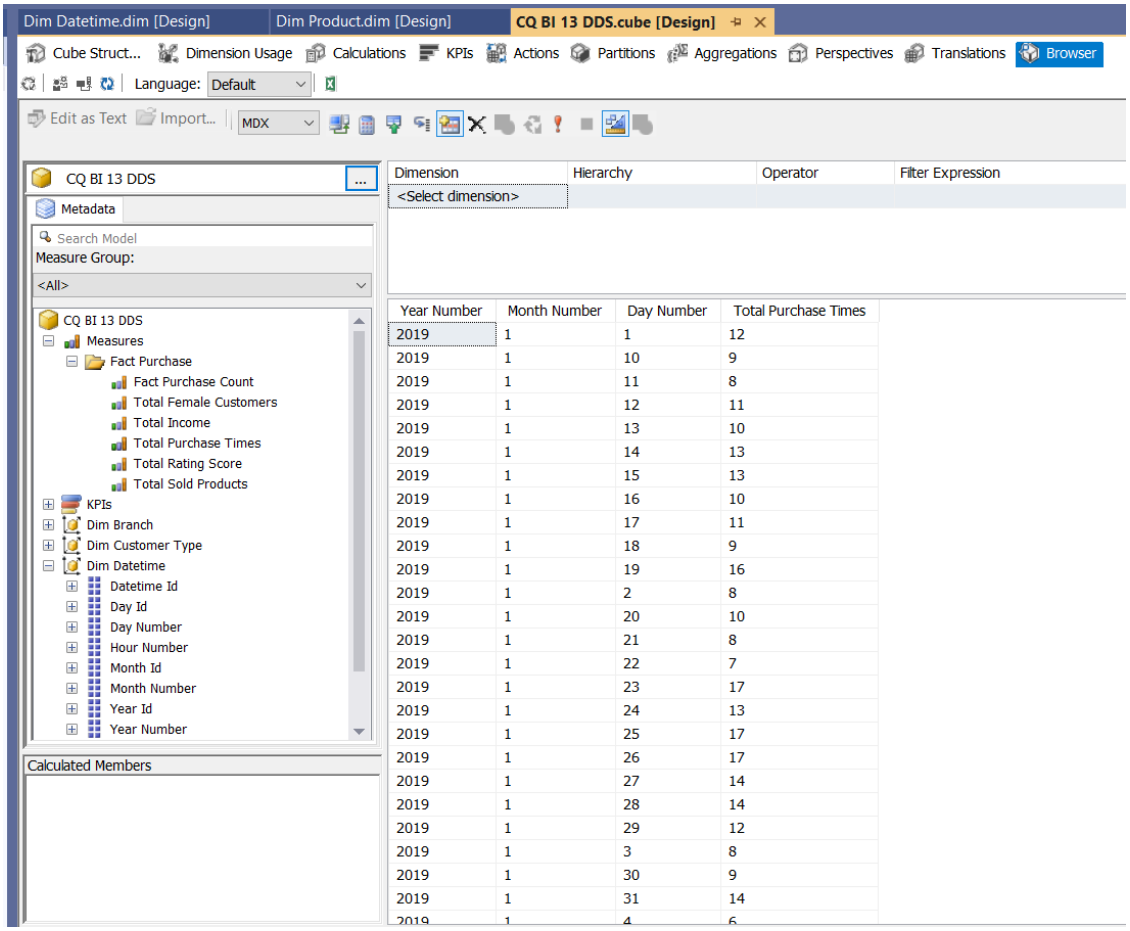
- dim_year → dim_month → dim_day → dim_datetime
- dim_product_line → dim_product

Kết quả phân tích cho các yêu cầu như sau:

- **Yêu cầu 1**: Thống kê số lượt mua hàng theo ngày, tháng, năm
    - Giải thích:

- Sử dụng các chiều **dim_day**, **dim_month**, và **dim_year** để phân tích dữ liệu mua hàng theo thời gian.
- Sử dụng đo lường **total_purchase_times** để hiển thị số lượt mua.

○ OLAP Browser:




○ MDX:

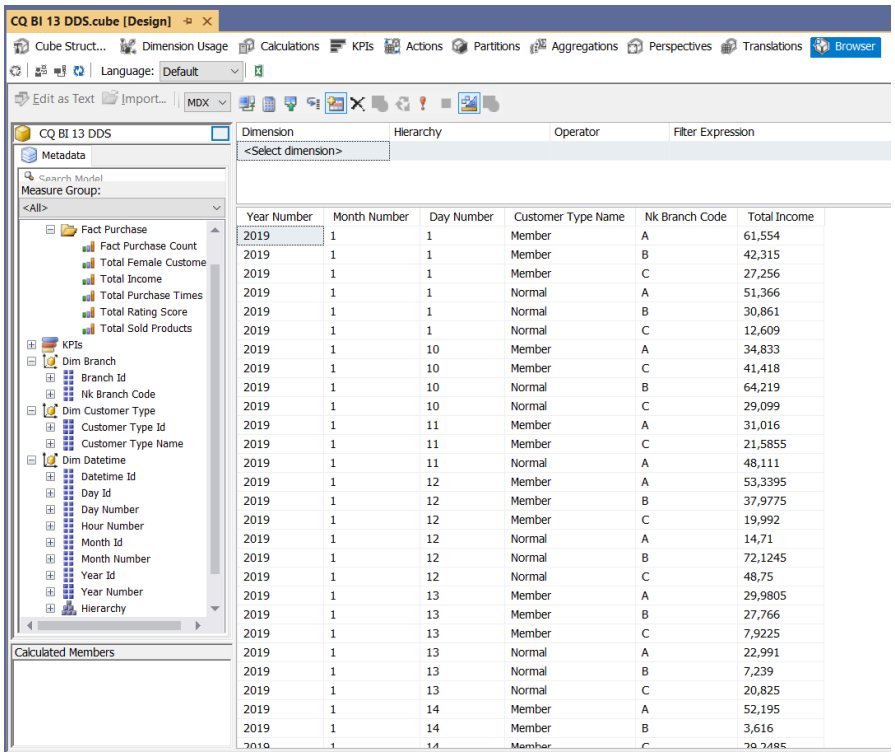
```
SELECT NON EMPTY {
    [Measures].[Total Purchase Times]
} ON COLUMNS,
NON EMPTY {
    (
        [Dim Datetime].[Year Number].[Year Number].ALLMEMBERS
        * [Dim Datetime].[Month Number].[Month Number].ALLMEMBERS
        * [Dim Datetime].[Day Number].[Day Number].ALLMEMBERS
    )
} DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
 FROM [CQ BI 13 DDS]
```

110 %

Messages | Results

| | | | Total Purchase Times |
|---|---|---|---|
| 2019 | 1 | 1 | 12 |
| 2019 | 1 | 10 | 9 |
| 2019 | 1 | 11 | 8 |
| 2019 | 1 | 12 | 11 |
| 2019 | 1 | 13 | 10 |
| 2019 | 1 | 14 | 13 |
| 2019 | 1 | 15 | 13 |
| 2019 | 1 | 16 | 10 |
| 2019 | 1 | 17 | 11 |
| 2019 | 1 | 18 | 9 |
| 2019 | 1 | 19 | 16 |
| 2019 | 1 | 2 | 8 |
| 2019 | 1 | 20 | 10 |
| 2019 | 1 | 21 | 8 |
| 2019 | 1 | 22 | 7 |
| 2019 | 1 | 23 | 17 |
| 2019 | 1 | 24 | 13 |
| 2019 | 1 | 25 | 17 |
| 2019 | 1 | 26 | 17 |
| 2019 | 1 | 27 | 14 |
| 2019 | 1 | 28 | 14 |

- **Yêu cầu 2**: Thống kê tổng doanh thu của khách hàng (member, normal) theo ngày, tháng, năm và theo chi nhánh.
    - Giải thích:
        - Sử dụng các chiều **dim_datetime**, **dim_branch** và **dim_customer_type** để thông kê doanh thu theo thời gian, theo chi nhánh và theo loại khách hàng.
        - Sử dụng đo lường **total_incomes** để hiển thị số lượt mua.
    - OLAP Browser:




| Year Number | Month Number | Day Number | Customer Type Name | Nk Branch Code | Total Income |
|---|---|---|---|---|---|
| 2019 | 1 | 1 | Member | A | 61,554 |
| 2019 | 1 | 1 | Member | B | 42,315 |
| 2019 | 1 | 1 | Member | C | 27,256 |
| 2019 | 1 | 1 | Normal | A | 51,366 |
| 2019 | 1 | 1 | Normal | B | 30,861 |
| 2019 | 1 | 1 | Normal | C | 12,609 |
| 2019 | 1 | 10 | Member | A | 34,833 |
| 2019 | 1 | 10 | Member | C | 41,418 |
| 2019 | 1 | 10 | Normal | B | 64,219 |
| 2019 | 1 | 10 | Normal | C | 29,099 |
| 2019 | 1 | 11 | Member | A | 31,016 |
| 2019 | 1 | 11 | Member | C | 21,5855 |
| 2019 | 1 | 11 | Normal | A | 48,111 |
| 2019 | 1 | 12 | Member | A | 53,3395 |
| 2019 | 1 | 12 | Member | B | 37,9775 |
| 2019 | 1 | 12 | Member | C | 19,992 |
| 2019 | 1 | 12 | Normal | A | 14,71 |
| 2019 | 1 | 12 | Normal | B | 72,1245 |
| 2019 | 1 | 12 | Normal | C | 48,75 |
| 2019 | 1 | 13 | Member | A | 29,9805 |
| 2019 | 1 | 13 | Member | B | 27,766 |
| 2019 | 1 | 13 | Member | C | 7,9225 |
| 2019 | 1 | 13 | Normal | A | 22,991 |
| 2019 | 1 | 13 | Normal | B | 7,239 |
| 2019 | 1 | 13 | Normal | C | 20,825 |
| 2019 | 1 | 14 | Member | A | 52,195 |
| 2019 | 1 | 14 | Member | B | 3,616 |

- MDX:

```
SELECT NON EMPTY {
    [Measures].[Total Income]
} ON COLUMNS,
NON EMPTY {
    (
        [Dim Datetime].[Year Number].[Year Number].ALLMEMBERS
        * [Dim Datetime].[Month Number].[Month Number].ALLMEMBERS
        * [Dim Datetime].[Day Number].[Day Number].ALLMEMBERS
        * [Dim Customer Type].[Customer Type Name].[Customer Type Name].ALLMEMBERS
        * [Dim Branch].[Nk Branch Code].[Nk Branch Code].ALLMEMBERS
    )
} DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
 FROM [CQ BI 13 DDS]
```

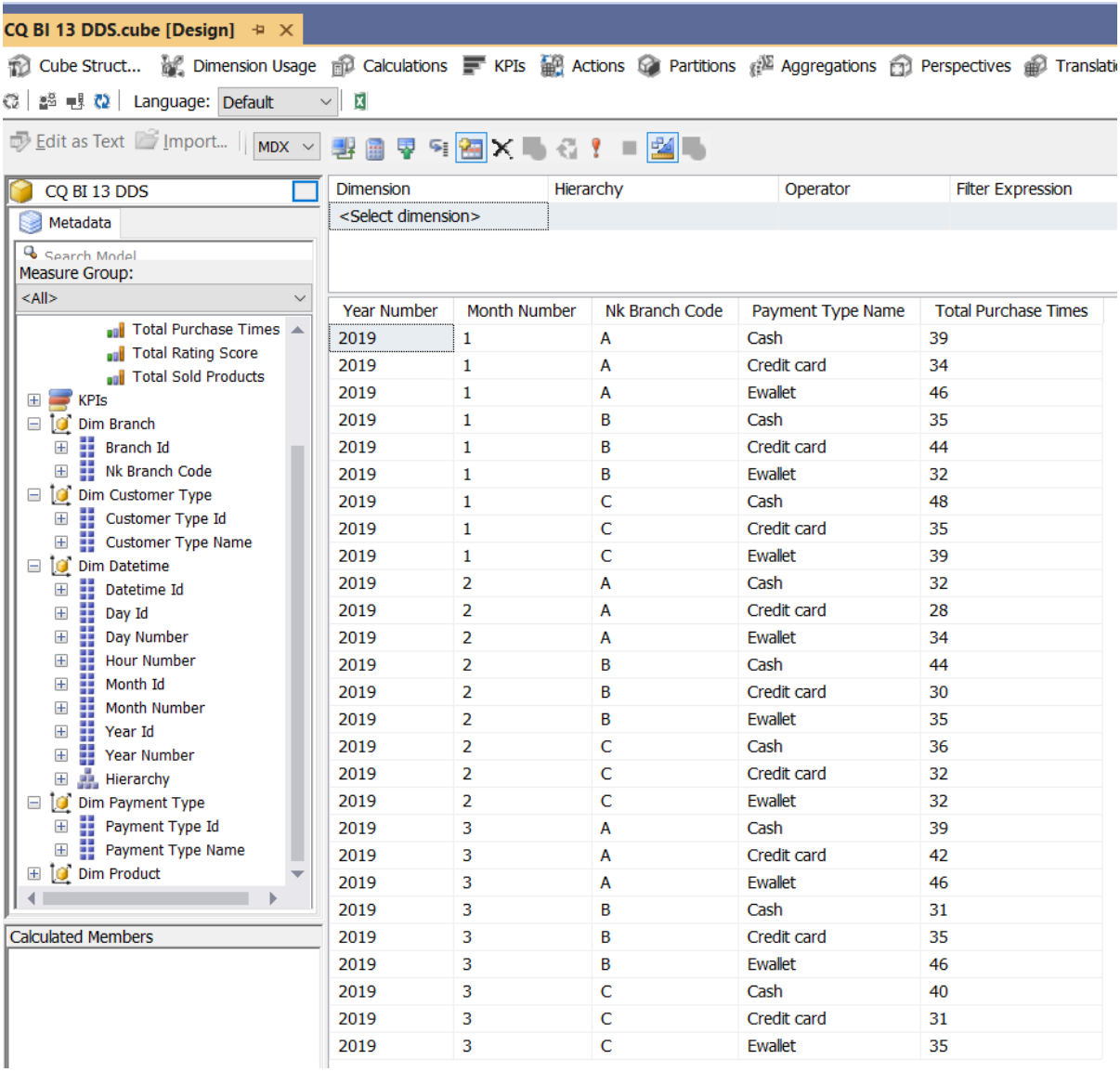
110 %

Messages | Results

| | | | | | Total Income |
|---|---|---|---|---|---|
| 2019 | 1 | 1 | Member | A | 61.554 |
| 2019 | 1 | 1 | Member | B | 42.315 |
| 2019 | 1 | 1 | Member | C | 27.256 |
| 2019 | 1 | 1 | Normal | A | 51.366 |
| 2019 | 1 | 1 | Normal | B | 30.861 |
| 2019 | 1 | 1 | Normal | C | 12.609 |
| 2019 | 1 | 10 | Member | A | 34.833 |
| 2019 | 1 | 10 | Member | C | 41.418 |
| 2019 | 1 | 10 | Normal | B | 64.219 |
| 2019 | 1 | 10 | Normal | C | 29.099 |
| 2019 | 1 | 11 | Member | A | 31.016 |
| 2019 | 1 | 11 | Member | C | 21.5855 |
| 2019 | 1 | 11 | Normal | A | 48.111 |
| 2019 | 1 | 12 | Member | A | 53.3395 |
| 2019 | 1 | 12 | Member | B | 37.9775 |
| 2019 | 1 | 12 | Member | C | 19.992 |
| 2019 | 1 | 12 | Normal | A | 14.71 |
| 2019 | 1 | 12 | Normal | B | 72.1245 |
| 2019 | 1 | 12 | Normal | C | 48.75 |
| 2019 | 1 | 13 | Member | A | 29.9805 |
| 2019 | 1 | 13 | Member | B | 27.766 |

- **Yêu cầu 3**: Thống kê số lượt khách thanh toán theo cash/ debit/... ở từng chi nhánh theo từng tháng trong năm.
  - Giải thích**:**
    - Sử dụng các chiều **dim_datetime**, **dim_customer_type, dim_branch** và **dim_product_line** để hiển thị số lượt thanh toán của khách hàng theo thời gian, theo chi nhánh, theo loại khách hàng, theo chi nhánh và theo hình thức thanh toán.
    - Sử dụng đo lường **total_purchase_times** để hiển thị số lượt mua.

- OLAP Browser:




- MDX:

```
SELECT NON EMPTY {
    [Measures].[Total Purchase Times]
} ON COLUMNS,
NON EMPTY {
    (
        [Dim Datetime].[Year Number].[Year Number].ALLMEMBERS
        * [Dim Datetime].[Month Number].[Month Number].ALLMEMBERS
        * [Dim Branch].[Nk Branch Code].[Nk Branch Code].ALLMEMBERS
        * [Dim Payment Type].[Payment Type Name].[Payment Type Name].ALLMEMBERS
    )
} DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
 FROM [CQ BI 13 DDS]
```

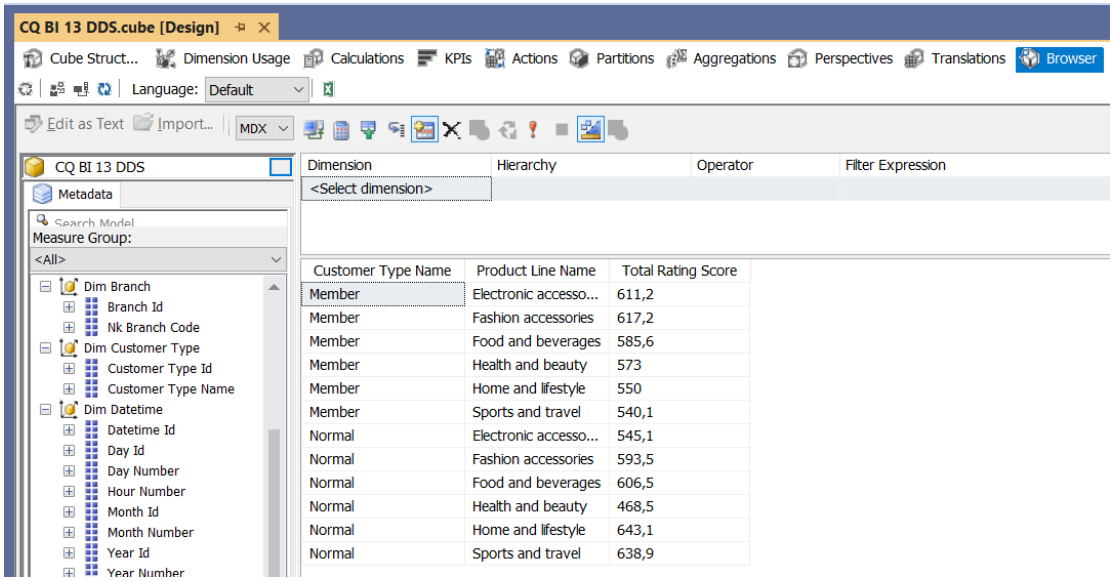
110 %

Messages | Results

| | | | | Total Purchase Times |
|---|---|---|---|---|
| 2019 | 1 | A | Cash | 39 |
| 2019 | 1 | A | Credit card | 34 |
| 2019 | 1 | A | Ewallet | 46 |
| 2019 | 1 | B | Cash | 35 |
| 2019 | 1 | B | Credit card | 44 |
| 2019 | 1 | B | Ewallet | 32 |
| 2019 | 1 | C | Cash | 48 |
| 2019 | 1 | C | Credit card | 35 |
| 2019 | 1 | C | Ewallet | 39 |
| 2019 | 2 | A | Cash | 32 |
| 2019 | 2 | A | Credit card | 28 |
| 2019 | 2 | A | Ewallet | 34 |
| 2019 | 2 | B | Cash | 44 |
| 2019 | 2 | B | Credit card | 30 |
| 2019 | 2 | B | Ewallet | 35 |
| 2019 | 2 | C | Cash | 36 |
| 2019 | 2 | C | Credit card | 32 |
| 2019 | 2 | C | Ewallet | 32 |
| 2019 | 3 | A | Cash | 39 |
| 2019 | 3 | A | Credit card | 42 |
| 2019 | 3 | A | Ewallet | 46 |

- **Yêu cầu 4**: Thống kê lượng rating của khách hàng (member, normal) theo từng loại sản phẩm (ProductLine)
    - Giải thích**:**
        - Sử dụng các chiều **dim_customer_type, dim_product** để phân tích dữ liệu rating của khách hàng theo từng loại sản phẩm.
        - Sử dụng đo lường **total_rating_score** để hiển thị số lượng rating.

- OLAP Browser:




- MDX:

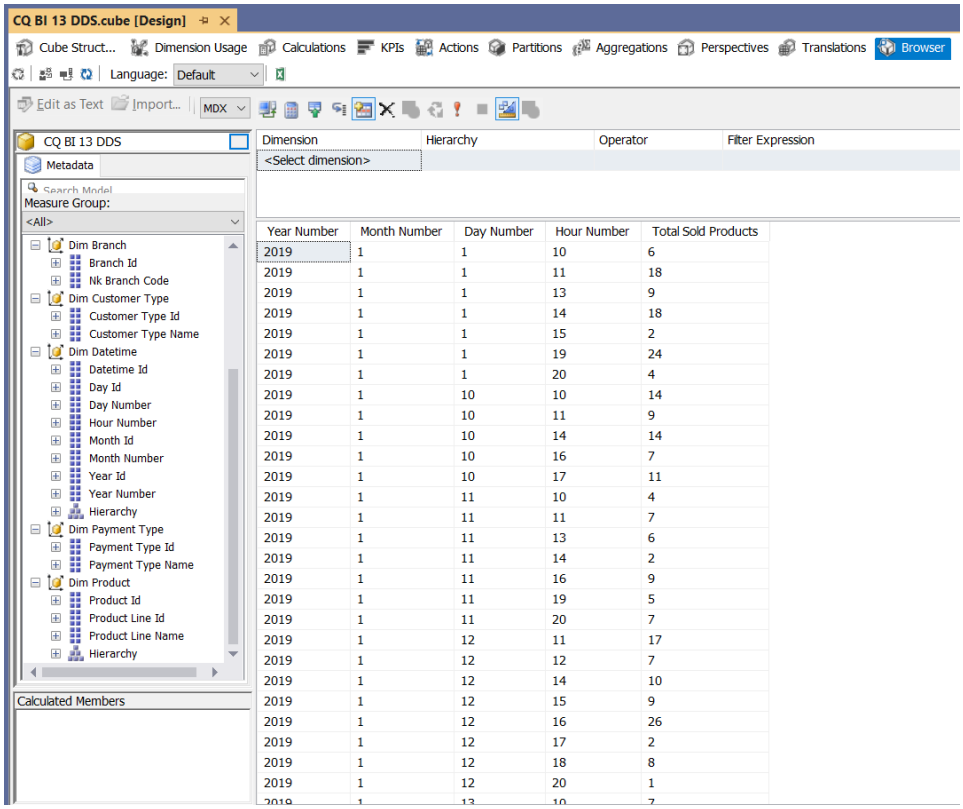
```
SELECT NON EMPTY {
    [Measures].[Total Rating Score]
} ON COLUMNS,
NON EMPTY {
    (
        [Dim Customer Type].[Customer Type Name].[Customer Type Name].ALLMEMBERS
        * [Dim Product].[Product Line Name].[Product Line Name].ALLMEMBERS
    )
} DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
 FROM [CQ BI 13 DDS]
```

110 %

Messages | Results

| | | Total Rating Score |
|---|---|---|
| Member | Electronic accessories | 611.2 |
| Member | Fashion accessories | 617.2 |
| Member | Food and beverages | 585.6 |
| Member | Health and beauty | 573 |
| Member | Home and lifestyle | 550 |
| Member | Sports and travel | 540.1 |
| Normal | Electronic accessories | 545.1 |
| Normal | Fashion accessories | 593.5 |
| Normal | Food and beverages | 606.5 |
| Normal | Health and beauty | 468.5 |
| Normal | Home and lifestyle | 643.1 |
| Normal | Sports and travel | 638.9 |

- **Yêu cầu 5**: Thống kê số lượng sản phẩm bán được theo từng thời điểm (time / date)
  - Giải thích:
    - Sử dụng các chiều **dim_datetime** để phân tích dữ liệu sản phẩm bán được theo thời gian.
    - Sử dụng đo lường **total_sold_products** để hiển thị số lượng sản phẩm bán được.
  - OLAP Browser:


CQ BI 13 DDS.cube [Design]
Cube Struct...
Dimension Usage
Calculations
KPIs
Actions
Partitions
Aggregations
Perspectives
Translations
Browser
Language: Default
Edit as Text
Import...
MDX
CQ BI 13 DDS
Metadata
Measure Group:
<All>
Dim Branch
Branch Id
Nk Branch Code
Dim Customer Type
Customer Type Id
Customer Type Name
Dim Datetime
Datetime Id
Day Id
Day Number
Hour Number
Month Id
Month Number
Year Id
Year Number
Hierarchy
Dim Payment Type
Payment Type Id
Payment Type Name
Dim Product
Product Id
Product Line Id
Product Line Name
Hierarchy
Calculated Members
Dimension
Hierarchy
Operator
Filter Expression
<Select dimension>
Year Number
Month Number
Day Number
Hour Number
Total Sold Products
2019 1 1 10 6
2019 1 1 11 18
2019 1 1 13 9
2019 1 1 14 18
2019 1 1 15 2
2019 1 1 19 24
2019 1 1 20 4
2019 1 10 10 14
2019 1 10 11 9
2019 1 10 14 14
2019 1 10 16 7
2019 1 10 17 11
2019 1 11 10 4
2019 1 11 11 7
2019 1 11 13 6
2019 1 11 14 2
2019 1 11 16 9
2019 1 11 19 5
2019 1 11 20 7
2019 1 12 11 17
2019 1 12 12 7
2019 1 12 14 10
2019 1 12 15 9
2019 1 12 16 26
2019 1 12 17 2
2019 1 12 18 8
2019 1 12 20 1

- MDX:

```
SELECT NON EMPTY {
    [Measures].[Total Sold Products]
} ON COLUMNS,
NON EMPTY {
    (
        [Dim Datetime].[Year Number].[Year Number].ALLMEMBERS
        * [Dim Datetime].[Month Number].[Month Number].ALLMEMBERS
        * [Dim Datetime].[Day Number].[Day Number].ALLMEMBERS
        * [Dim Datetime].[Hour Number].[Hour Number].ALLMEMBERS
    )
} DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
 FROM [CQ BI 13 DDS]
```

110 %

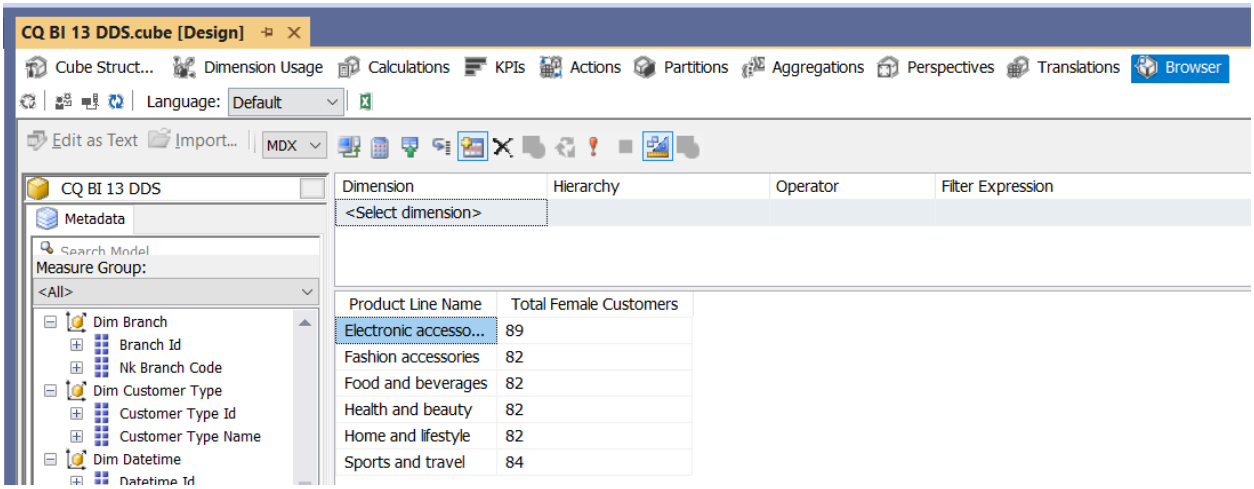
Messages | Results

| | | | | Total Sold Products |
|---|---|---|---|---|
| 2019 | 1 | 1 | 10 | 6 |
| 2019 | 1 | 1 | 11 | 18 |
| 2019 | 1 | 1 | 13 | 9 |
| 2019 | 1 | 1 | 14 | 18 |
| 2019 | 1 | 1 | 15 | 2 |
| 2019 | 1 | 1 | 19 | 24 |
| 2019 | 1 | 1 | 20 | 4 |
| 2019 | 1 | 10 | 10 | 14 |
| 2019 | 1 | 10 | 11 | 9 |
| 2019 | 1 | 10 | 14 | 14 |
| 2019 | 1 | 10 | 16 | 7 |
| 2019 | 1 | 10 | 17 | 11 |
| 2019 | 1 | 11 | 10 | 4 |
| 2019 | 1 | 11 | 11 | 7 |
| 2019 | 1 | 11 | 13 | 6 |
| 2019 | 1 | 11 | 14 | 2 |
| 2019 | 1 | 11 | 16 | 9 |
| 2019 | 1 | 11 | 19 | 5 |
| 2019 | 1 | 11 | 20 | 7 |
| 2019 | 1 | 12 | 11 | 17 |
| 2019 | 1 | 12 | 12 | 7 |

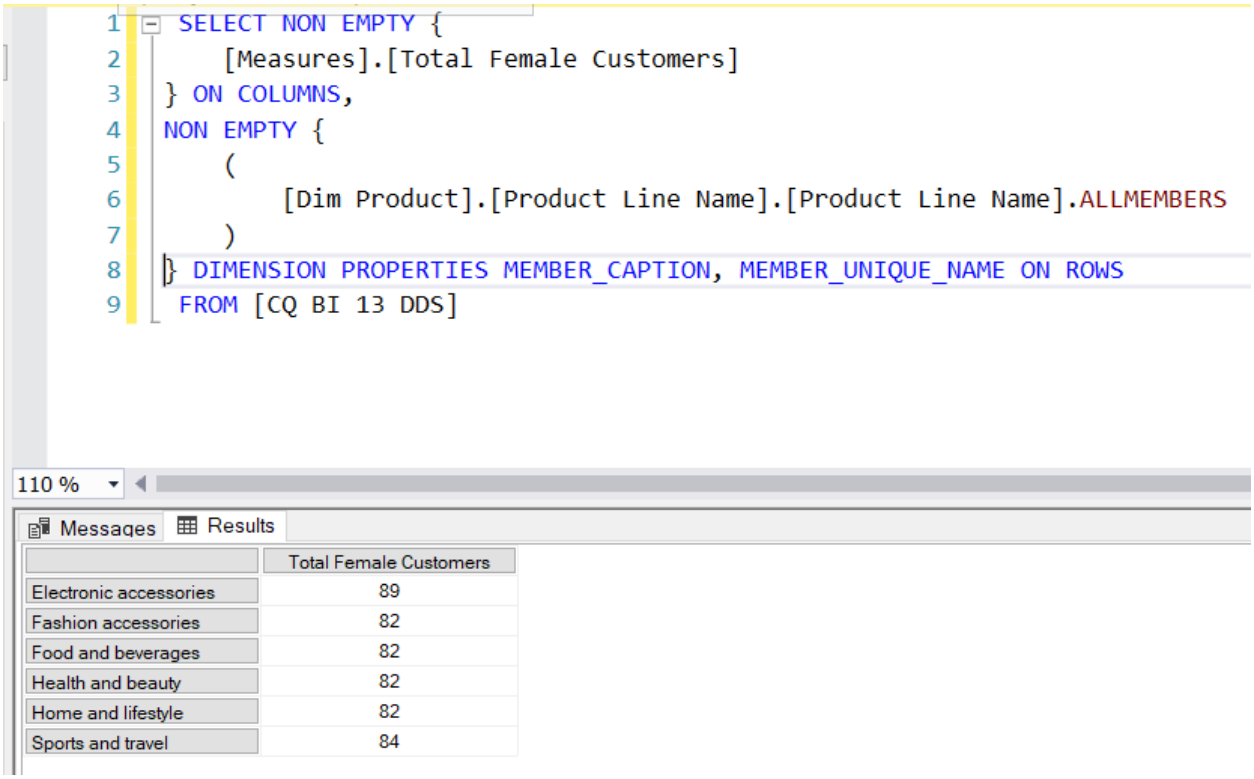
- **Yêu cầu 6**: Thống kê số lượt khách hàng nữ đã mua theo từng loại sản phẩm
    - Giải thích:
        - Sử dụng các chiều **dim_product** để phân tích dữ liệu khách hàng nữ mua theo từng loại sản phẩm.

- Sử dụng đo lường **total_female_customers** để hiển thị số lượt khách hàng nữ đã mua.

  - OLAP Browser:

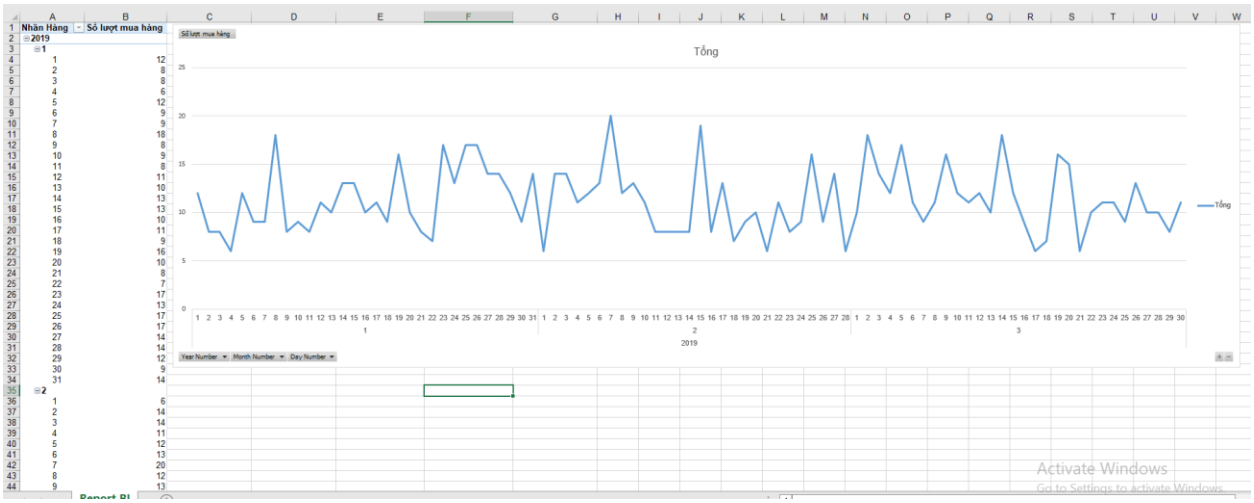



  - MDX:



## 7.2. Tạo Report và Visualize

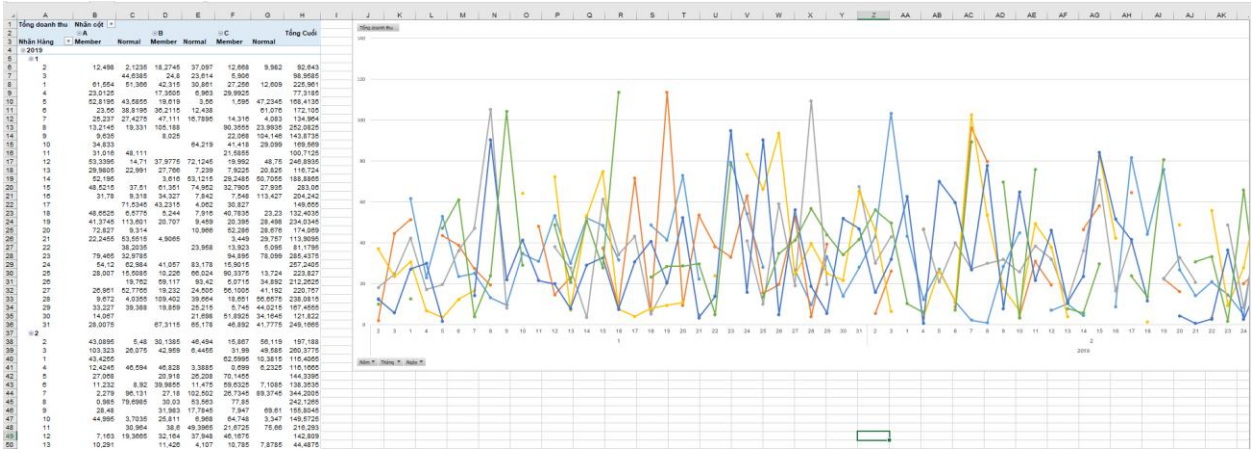
- Với report, nhóm sử dụng Excel để kết nối với SSAS Server (chi tiết trong file BI Report.xlsx):
    - **Yêu cầu 1**: Biểu đồ đường thể hiện sự biến thiên số lượt mua hàng được bán theo từng thời điểm.




**Nhận xét:** Dựa vào biểu đồ ta thấy:

- Số lượt mua hàng theo ngày cao nhất (20 lượt) vào ngày 7/2 và ngày thấp nhất (6 lượt) rơi vào các ngày 4/1, 1/2, 21/2, 28/2, 17/3 và 21/3.
- Số lượt mua hàng theo tháng cao nhất rơi vào tháng 1 với tổng lượt mua là 352 lượt mua hàng.

- **Yêu cầu 2**: Biểu đồ đường đa trục thể hiện sự tương quan tổng doanh thu theo chi nhánh, loại khách hàng tại các thời điểm.



**Nhận xét:** Dựa vào biểu đồ ta thấy:

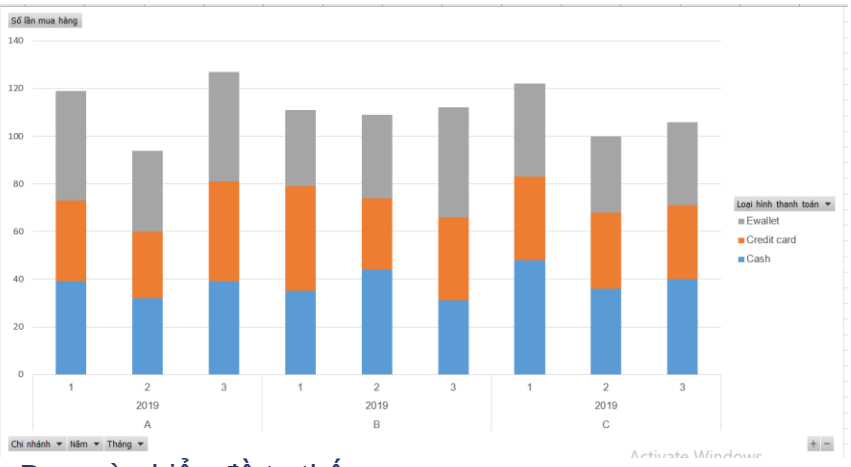
- Đối với Khách hàng là Member:
  - Chi nhánh A:
    - Tổng doanh thu theo ngày cao nhất (103.323) rơi vào ngày 3/2 và thấp nhất (0.985) vào ngày 8/2.
    - Tổng doanh thu cao nhất theo tháng rơi vào tháng 1 (961.81).
  - Chi nhánh B:
    - Tổng doanh thu theo ngày cao nhất (109.727) rơi vào ngày 9/3 và thấp nhất (3.616) vào ngày 14/1.
    - Tổng doanh thu cao nhất theo tháng rơi vào tháng 1 (884.23).
  - Chi nhánh C:
    - Tổng doanh thu theo ngày cao nhất (94.895) rơi vào ngày 23/1 và thấp nhất (0.627) vào ngày 21/2.
    - Tổng doanh thu cao nhất theo tháng 3(979.41).
- Đối với Khách hàng là Normal:
  - Chi nhánh A:
    - Tổng doanh thu theo ngày cao nhất (113.6) rơi vào ngày 19/1 và thấp nhất (2.1235) vào ngày 2/1.

- Tổng doanh thu cao nhất theo tháng rơi vào tháng 3 (961.58) .

- Chi nhánh B:
  - Tổng doanh thu theo ngày cao nhất (111.06) rơi vào ngày 5/3 và thấp nhất (1.476) vào ngày 18/2.
  - Tổng doanh thu cao nhất theo tháng rơi vào tháng 1 (886.06) .
- Chi nhánh C:
  - Tổng doanh thu theo ngày cao nhất (129.554) rơi vào ngày 14/3 và thấp nhất (1.512) vào ngày 4/3.
  - Tổng doanh thu cao nhất theo tháng rơi vào tháng 1 (1013.6).

- **Yêu cầu 3**: Biểu đồ cột xếp trồng thể hiện số lượt mua hàng tại các chi nhánh theo từng thời điểm. Mỗi cột được chia thành các vùng màu thể hiện sự tương quan số lượng mua thanh theo từng hình thức thanh toán.

| Số lần mua hàng | Nhãn cột | | | |
|---|---|---|---|---|
| Nhãn Hàng | Cash | Credit card | Ewallet | Tổng Cuối |
| A | | | | |
| 2019 | | | | |
| 1 | 39 | 34 | 46 | 119 |
| 2 | 32 | 28 | 34 | 94 |
| 3 | 39 | 42 | 46 | 127 |
| B | | | | |
| 2019 | | | | |
| 1 | 35 | 44 | 32 | 111 |
| 2 | 44 | 30 | 35 | 109 |
| 3 | 31 | 35 | 46 | 112 |
| C | | | | |
| 2019 | | | | |
| 1 | 48 | 35 | 39 | 122 |
| 2 | 36 | 32 | 32 | 100 |
| 3 | 40 | 31 | 35 | 106 |
| Tổng Cuối | 344 | 311 | 345 | 1000 |

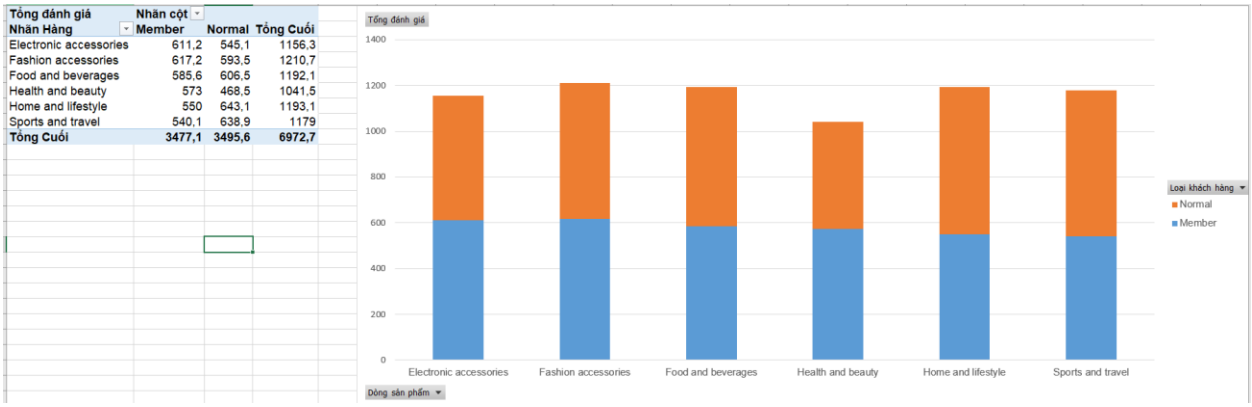



**Nhận xét:** Dựa vào biểu đồ ta thấy:

- Đối với chi nhánh A số lượt khách thanh toán theo cả 3 phương thức tại tháng 3 là cao nhất (127 lượt).
- Đối với chi nhánh B số lượt khách thanh toán theo cả 3 phương thức tại tháng 3 là cao nhất (112 lượt).

- Đối với chi nhánh C số lượt khách thanh toán theo cả 3 phương thức tại tháng 1 là cao nhất (122 lượt).
- Trong đó hình thức thanh toán bằng Ewallet là cao nhất với 345 lượt.
- Chi nhánh có lượt mua cao nhất là chi nhánh A rơi vào tháng 3.

- **Yêu cầu 4**: Tương tự như yêu cầu 3, biểu đồ này cũng thể hiện tổng rating của từng loại khách hàng theo từng loại sản phẩm.



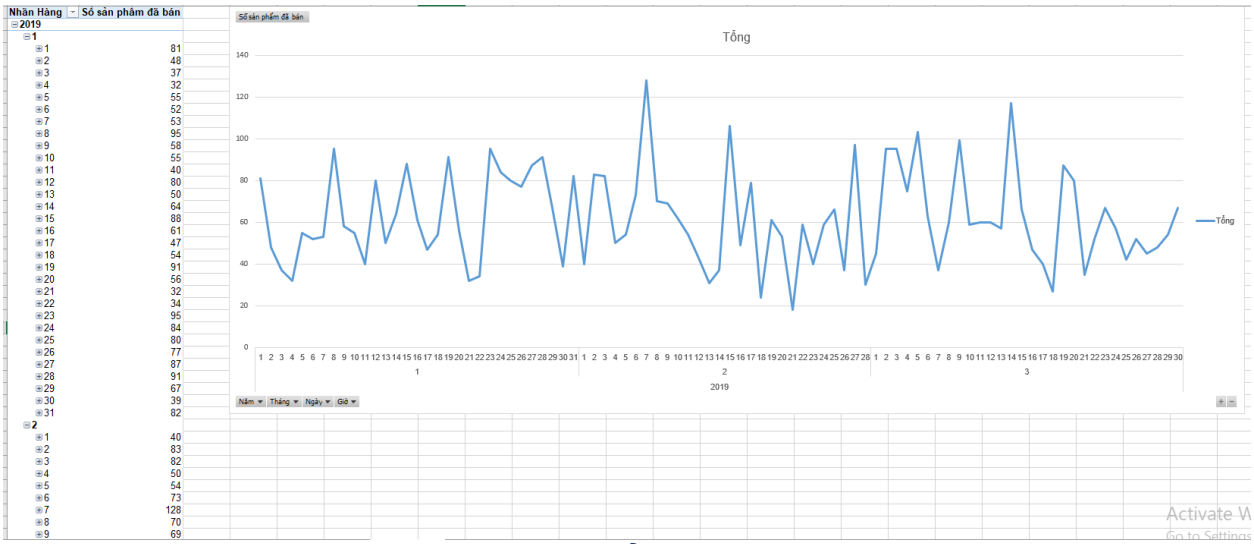
| Tổng đánh giá | Nhãn cột | | |
|---|---|---|---|
| Nhãn Hàng | Member | Normal | Tổng Cuối |
| Electronic accessories | 611,2 | 545,1 | 1156,3 |
| Fashion accessories | 617,2 | 593,5 | 1210,7 |
| Food and beverages | 585,6 | 606,5 | 1192,1 |
| Health and beauty | 573 | 468,5 | 1041,5 |
| Home and lifestyle | 550 | 643,1 | 1193,1 |
| Sports and travel | 540,1 | 638,9 | 1179 |
| **Tổng Cuối** | **3477,1** | **3495,6** | **6972,7** |



**Nhận xét:** Dựa vào biểu đồ ta thấy:

- Đối với Khách hàng Member sản phẩm Fashion accessories có lượng rating cao nhất (617.2) và sản phẩm Sports and travel có lượng rating thấp nhất (540.1).
- Đối với Khách hàng Normal sản phẩm Home and lifestyle có lượng rating cao nhất (643.1) và sản phẩm Health and beauty có lượng rating thấp nhất (468.5).
- Khách hàng Normal có lượng rating sản phẩm cao hơn khách hàng Member.
- Sản phẩm có lượng rating cao nhất là Fashion accessories (1210.7).
- Sản phẩm có lượng rating thấp nhất là Electronic accessories (1156.3).

- **Yêu cầu 5**: Biểu đồ đường thể hiện sự biến thiên trong số lượng



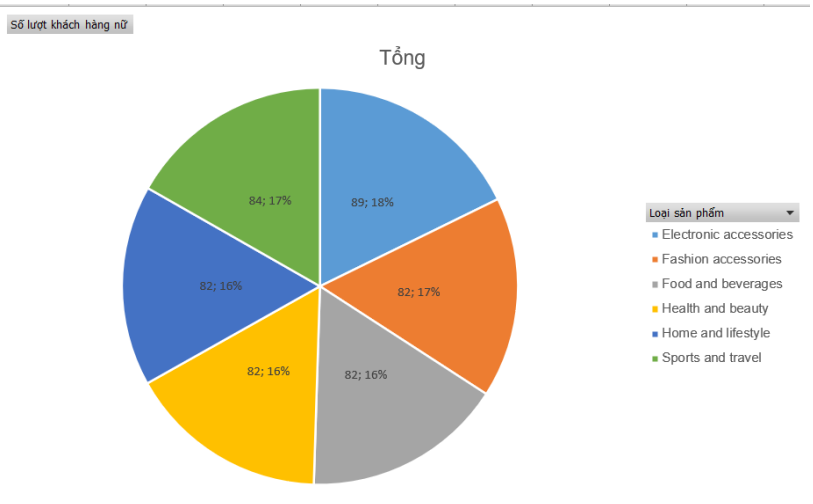

mua hàng tại từng thời điểm.

**Nhận xét:** Dựa vào biểu đồ ta thấy:

- Số lượng sản phẩm cao nhất được bán theo ngày rơi vào ngày 7/2 (128 lượt).
- Số lượng sản phẩm thấp nhất được bán theo ngày rơi vào ngày 21/2 (18 lượt).
- Trong đó tháng 1 có số lượng sản phẩm bán cao nhất (1965 lượt).

- **Yêu cầu 6**: Biểu đồ tròn thể hiện phần trăm/sự tương quan trong số lượt mua hàng với từng nhãn hàng của các lượt khách hàng là nữ.

| Nhãn Hàng | Số lượt khách hàng nữ |
|---|---|
| Electronic accessories | 89 |
| Fashion accessories | 82 |
| Food and beverages | 82 |
| Health and beauty | 82 |
| Home and lifestyle | 82 |
| Sports and travel | 84 |
| **Tổng Cuối** | **501** |




**Nhận xét:** Dựa vào biểu đồ ta thấy:

- Số lượt khách hàng nữ mua sản phẩm Electronic accessories là cao nhất (89 lượt) chiếm 89,18%.
- Đối với các sản phẩm Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle có số lượt mua bằng nhau là 82 lượt chiếm 82,16%.

- Visualize nhóm sử dụng Power BI để làm dashboard theo dõi tình trạng mua bán (chi tiết trong file power_bi.pbix):
  - Biểu đồ đường đa trục ở góc trên bên trái cho thấy sự biến thiên trong doanh thu của các chi nhánh tại các thời điểm.
  - Biểu đồ cột nằm ngang góc trên bên phải thể hiện top 10 sản phẩm bán chạy nhất kèm theo đó là số lượng hàng được bán của mỗi sản phẩm.
  - Biểu đồ cột nằm ngang góc dưới bên trái thể hiện các chi nhanh có doanh thu cao nhất và tổng doanh thu của chúng.

- Biểu đồ cột xếp chồng góc dưới bên phải thể hiện số lượng mua hàng tại các chi nhánh. Các cột được chia thành các vùng màu thể hiện sự tương quan giữa các loại khách hàng.


## 8. Data Mining

**Bài toán:** Dự đoán số mặt hàng sẽ bắn trong tương lai dự trên các yếu tố về sản phẩm, chi nhánh, ngày, loại khách hàng và hình thức thanh toán.

**Công cụ:** Nhóm sử dụng ngôn ngữ Python với sự hỗ trợ của các thư viện:

- Adodbapi: Kết nối đến SSAS Server và truy vấn dữ liệu.
- Pandas: Xử lý dữ liệu.
- Scikit-learn: Xây dựng mô hình dự đoán.

**Cách thực hiện:**

- Sử dụng môi trường làm việc là jupiter notebook.
- Đầu tiên tạo môi trường ảo bằng câu lệnh: python -m venv venv
- Sau đó kích hoạt môi trường ảo: venv\Scripts\activate
- Cài đặt các package cần thiết: pip install pyodbc pandas adodbapi

- Import các module cần cho việc huấn luyện mô hình

```python
import adodbapi
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
```

- Khai báo thông tin server và connection string

```python
server = "REVISION-PC"
catalog = "2023_CQ_BI_13_OLAP"
cube = "CQ BI 13 DDS"

conn_str = f"Provider=MSOLAP;Data Source={server};Initial Catalog={catalog};"
```

- Kết nối đến SSAS Server là truy vấn dữ liệu cần cho việc huấn luyện

```python
try:
    with adodbapi.connect(conn_str) as conn:
        cursor = conn.cursor()

        sql_stmt = """
            SELECT NON EMPTY { [Measures].[Total Sold Products] } ON COLUMNS,
            NON EMPTY {
                (
                    [Dim Product].[Product Id].[Product Id].ALLMEMBERS
                    * [Dim Branch].[Branch Id].[Branch Id].ALLMEMBERS
                    * [Dim Payment Type].[Payment Type Id].[Payment Type Id].ALLMEMBERS
                    * [Dim Customer Type].[Customer Type Id].[Customer Type Id].ALLMEMBERS
                    * [Dim Datetime].[Day Number].[Day Number].ALLMEMBERS
                    * [Dim Datetime].[Month Number].[Month Number].ALLMEMBERS
                    * [Dim Datetime].[Year Number].[Year Number].ALLMEMBERS
                )
            } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
            FROM [CQ BI 13 DDS]
        """

        cursor.execute(sql_stmt)
        rows = cursor.fetchall()
```

- Biến đổi dữ liệu về dạng Pandas Dataframe để dễ làm việc hơn

```
result = [
    {
        "product_id": row[0],
        "branch_id": row[2],
        "payment_type_id": row[4],
        "customer_type_id": row[6],
        "day": row[8],
        "month": row[10],
        "year": row[12],
        "total_sold_products": row[14]
    } for row in rows
]

df = pd.DataFrame(result)
```

- Sau đó chia tập dữ liệu thành 2 phần, trong đó 80% dùng để train và 20% còn lại dùng làm tập test.

```
df = pd.DataFrame(result)
y = df["total_sold_products"]
X = df.drop(["total_sold_products"], axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
print(f"Mean Squared Error: {mse}")
```

- Sử dụng mô hình Linear Regression để train trên tập dữ liệu ở trên.
- Kết quả train cho thấy chỉ số MSE xấp xỉ 10.0. Đây là **độ lệch khá lớn**, nguyên nhân là **do tập dữ liệu không đủ lớn** và có những **giá trị oulier** khiến cho mô hình có độ sai số khi dự đoán lớn.