1. **Tổng quan về đề tài**
   1. **Mô tả về dữ liệu:**

Bộ dữ liệu Supermarket Sales Date chứa thông tin về các giao dịch bán hạng tại một siêu thị trong một khoảng thời gian nhất định. Dữ liệu bao gồm nhiều thuộc tính khác nhau liên quan đến mỗi giao dịch, như thông tin sản phẩm, số lượng bán, giá cả, chiết khấu, và thôn tin khách hàng. Cụ thể, các thuộc tính bao gồm:

- **Thực thể** ITEM**:**

* **Mô tả**: Thực thể ITem đại diện cho các mặt hàng hoặc sản phẩm có trong siêu thị.
* **Thuộc tính**

+ Item\_Code: Mã duy nhất cho mỗi mặt hàng (mã sản phẩm)

+ Item\_Name: Tên của mặt hàng

+ Category\_Code: Mã danh mục của mặt hàng(dùng để phân loại)

+ Category\_Name: Tên danh mục của mặt hàng.

* **Mối quan hệ:**

**+** Item có mối quan hệ với Transaction để ghi nhận các giao dịch liên quan đến mặt hàng.

+ Item có mối quan hệ với LossRate để xác định tỷ lệ hao hụt của sản phẩm.

+ Item có mối quan hệ với Price để lưu trữ thông tin về giá bán buôn cho mỗi mặt hàng vào từng ngày.

+ Item có mỗi quan hệ với Price để lưu trữ thông tin về giá bán cho mỗi mặt hàng vào từng ngày.

- **Thực thể** Transaction

* **Mô tả:** Thực thể Transaction đại diện cho mỗi giao dịch bán hàng hoặc trả hàng của khách hàng với siêu thị.
* **Thuộc tính**:

+ Date: Ngày giao dịch diễn ra.

+ Time: Thời gian giao dịch diễn ra.

+ Quantity\_Sold\_kilo: Số lượng mặt hàng đã bán (tính bằng kg).

+ Unit\_Selling\_Price\_RMB\_kg: Giá bán lẻ của mặt hàng (tính theo kg, đơn vị RMB).

+ Sale\_Or\_Return: Loại giao dịch (bán hàng hoặc trả lại hàng).

Discount\_Yes\_No: Cho biết giao dịch có được giảm giá hay không.

- **Thực thể** LossRate

* **Mô tả**: Thực thể LossRate lưu trữ thông tin về tỷ lệ hao hụt của từng mặt hàng. Tỷ lệ hao hụt có thể được dùng để quản lý tồn kho và tính toán thiệt hại khi hàng hóa không được bán hoặc bị hỏng hóc.
* **Thuộc tính**:

+ Item\_Name: Tên của mặt hàng.

+Loss\_Rate: Tỷ lệ hao hụt của mặt hàng (thường tính theo % hoặc kg).

* **Mối quan hệ**:

+ LossRate liên kết với Item để xác định tỷ lệ hao hụt của mỗi mặt hàng cụ thể.

- **Thực thể** Price

* **Mô tả**: Thực thể Price lưu trữ thông tin về giá bán buôn cho từng mặt hàng theo từng ngày. Dữ liệu về giá cả theo ngày giúp quản lý lịch sử giá và điều chỉnh chiến lược bán hàng theo biến động giá cả.
* **Thuộc tính**:

+ Date: Ngày áp dụng giá bán buôn.

+Wholesale\_Price\_RMB\_kg: Giá bán buôn của mặt hàng (tính theo kg, đơn vị RMB).

* **Mối quan hệ**:

+ Price liên kết với Item để lưu giữ giá bán buôn theo ngày cho từng mặt hàng.

Dữ liệu này có thể chứa hàng ngàn giao dịch bán hàng và cung cấp bức trang tổng quan về hoạt động bán hàng của siêu thị.

* 1. **Nguồn dữ liệu.**

Bộ dữ liệu Supermarket Sales Data được lấy từ Kaggle, một nền tảng phổ biến chuyên cung cấp các bộ dữ liệu công khai để nghiên cứu và phân tích. Kaggle là một nguồn uy tín và hữu ích cho những người học và nghiên cứu về phân tích dữ liệu, vì các bộ dữ liệu ở đây thường có chất lượng và đang dạng chủ đề.

* 1. **Lý do chọn đề tài**

Lý do chọn đề tài phân tích dữ liệu bán hàng tại siêu thị dựa trên các điểm sau:

* Phân tích hành vi khách hàng: Thông qua các thông tin như loại khách hàng, giới tính, thành phố và đánh giá của khách hàng, có thể hiểu được hành vi và sở thích của khách hàng trong từng loại sản phẩm, từ đó giúp siêu thị tối ưu hóa chiến lược kinh doanh và cải thiện trải nghiệm mua sắm.
* Tối ưu hóa quản lý tồn kho: Thông qua dữ liệu số lượng bán và dòng sản phẩm, siêu thị có thể nắm bắt được xu hướng tiêu thụ của từng sản phẩm và lên kế hoạch cung ứng hợp lý, từ đó tránh tình trạng thiếu hụt hàng hóa hoặc dư thừa.
* Đánh giá hiệu quả chi nhánh: Dữ liệu từ các chi nhánh và thành phố khác nhau giúp đánh giá hiệu quả hoạt động của từng chi nhánh. Từ đó, quản lý có thể đưa ra các quyết định điều chỉnh hợp lý giữa các chi nhánh.
* Quản lý và tối ưu lợi nhuận: Các thông tin về giá vốn, lợi nhuận gộp và tổng thu nhập từ mỗi giao dịch giúp siêu thị phân tích mức lợi nhuận và hiệu quả chi phí của từng loại sản phẩm. Điều này có thể hỗ trợ đưa ra các quyết định về giá cả, chương trình khuyến mãi và giảm giá để tối ưu hóa lợi nhuận.
* Phân tích xu hướng thời gian: Dữ liệu về ngày và thời gian của từng giao dịch giúp phân tích thói quen mua sắm của khách hàng theo thời gian (giờ trong ngày, ngày trong tuần, tháng trong năm). Điều này hữu ích trong việc lên kế hoạch nhân sự và quản lý khuyến mãi vào các thời điểm cao điểm mua sắm.

1. **Xác định mô hình thực thể liên kết**
   1. **Các kiểu thực thể**:
      1. **Thực thể** Item

- **Các thuộc tính**:

* Item\_Code
* Item\_Name
* Category\_Code
* Category\_Name

- **Thuộc tính khóa**:

* Item\_Code (Khóa chính)
  + 1. **Thực thể** Transaction

- **Các thuộc tính**:

* Date
* Time
* Quantity\_Sold\_kilo
* Unit\_Selling\_RMB\_kg
* Sale\_Or\_Return
* Discount\_Yes\_No

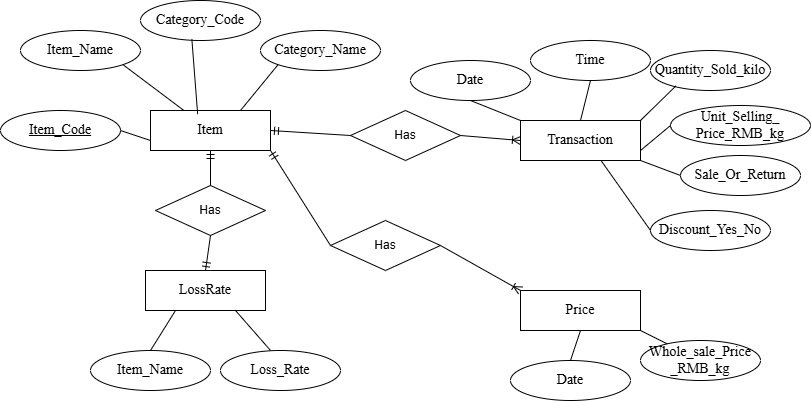
* + 1. **Thực thể** LossRate

- **Các thuộc tính**:

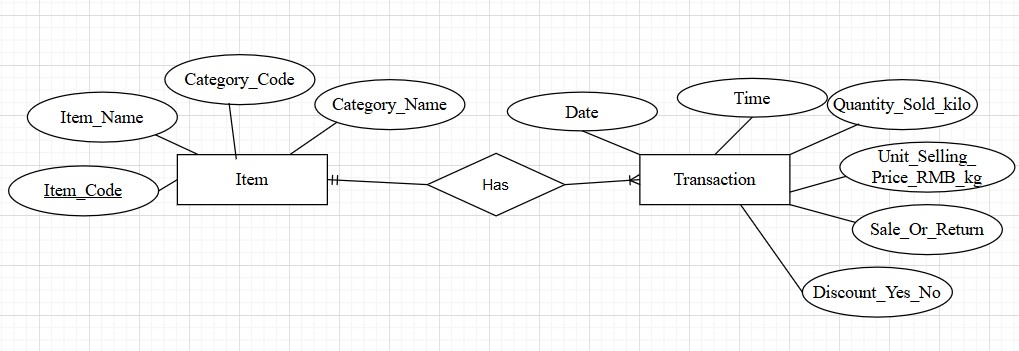
* Loss\_Rate
  + 1. **Thực thể** Price

- **Các thuộc tính**:

* Date
* Whole\_sale\_Price\_RMB\_kg
  1. **Mô hình ER**:



1. **Chuyển từ mô hình thực thể liên kết sang mô hình quan hệ.**

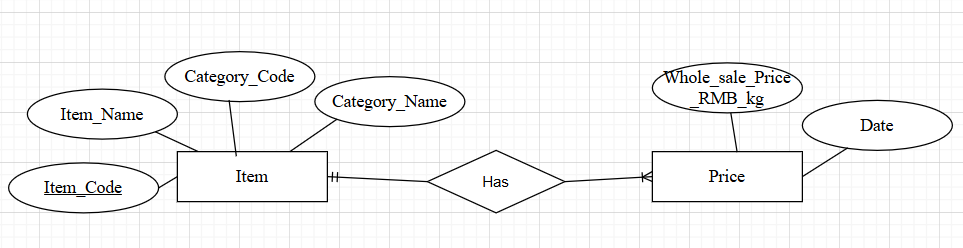
**- Cách chuyển đổi các kiểu thực thể mối liên kết** 

Thực thể Item có mối quan hệ 1-n với thực thể Transaction nên khóa chính của thực thể bên nhánh 1 sẽ làm khóa ngoại bên thực thể bên nhánh nhiều.

Item(ItemCode, Item\_Name, Category\_Code, Category\_Name)

Transaction(Date, Time, Quantity\_Sold\_kilo, Unit\_Selling\_Price\_RMB\_kg

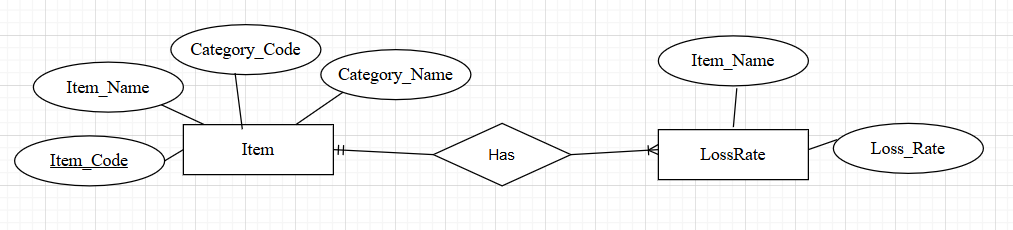
Item\_Code, Discount\_Yes\_No)



Thực thể Item có mối quan hệ 1-n với thực thể Price nên khóa chính của thực thể bên nhánh 1 sẽ làm khóa ngoại bên thực thể bên nhánh nhiều.

Và ở đây trong bảng Price thì Item\_Code và Date sẽ là khóa chính

Price(ItemCode, Date, Whole\_sale\_Price\_RMB\_kg)



Thực thể Item có mối quan hệ 1-n với thực thể LossRate nên khóa chính của thực thể bên nhánh 1 sẽ làm khóa ngoại bên thực thể bên nhánh nhiều.

Và khóa ngoại này sẽ làm khóa chính cho bảng LossRate

LossRate(Item\_Code, Item\_Name, Loss\_Rate)

- **Mô hình quan hệ**

**Item**(ItemCode, Item\_Name, Category\_Code, Category\_Name)

**Transaction**(Date, Time, Quantity\_Sold\_kilo, Unit\_Selling\_Price\_RMB\_kg

Item\_Code, Discount\_Yes\_No)

**Price**(ItemCode, Date, Whole\_sale\_Price\_RMB\_kg)

**LossRate**(Item\_Code, Item\_Name, Loss\_Rate)

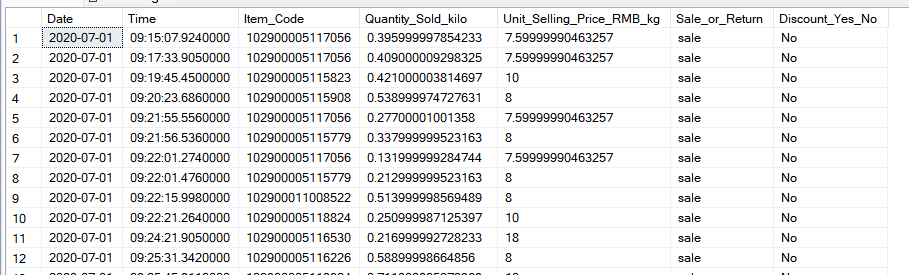
1. **Câu lệnh truy vấn dữ liệu SQL**
   1. 03 lệnh gồm SELECT có điều kiện trên một bảng
      1. Lấy danh sách các transaction có giá trị lớn hơn 7RMB

SELECT \*

FROM [dbo].[Transaction]

WHERE Unit\_Selling\_Price\_RMB\_kg > 7;

Kết quả truy vấn



* + 1. Lấy danh sách các mặt hàng thuộc loại sản phẩm“ Flower/Leaf Vegetables”

|  |
| --- |
| SELECT \*  FROM [dbo].[Item]  WHERE Category\_Name = 'Flower/Leaf Vegetables';  Kết quả thực thi |

* 1. 02 lệnh gồm SELECT có điều kiện trên hai bảng
     1. Lấy các mặt hàng có số lượng bán lớn hơn 10 kg trong giao dịch

|  |
| --- |
| SELECT Item.Item\_Name, t.Quantity\_Sold\_kilo  FROM [dbo].[Transaction] as t  JOIN Item ON t.Item\_Code = Item.Item\_Code  WHERE t.Quantity\_Sold\_kilo > 10  Kết quả truy vấn |

**4.2.2**  Lấy danh sách các mặt hàng và giá bán buôn của chúng vào ngày

‘2020-01-01’

|  |
| --- |
| SELECT i.Item\_Name, p.Wholesale\_Price\_RMB\_kg  FROM Item as i  JOIN Price as p ON i.Item\_Code = p.Item\_Code  WHERE p.Date = '2020-07-01';  Kết quả truy vấn |

* 1. 2 câu lệnh SELECT có WHERE và GROUP BY
     1. Truy vấn tổng số lượng bán theo từng mặt hàng

|  |
| --- |
| SELECT Item\_Code, SUM(Quantity\_Sold\_kilo) AS Total\_Quantity\_Sold  FROM [dbo].[Transaction]  WHERE Quantity\_Sold\_kilo > 0  GROUP BY Item\_Code;  Kết quả truy vấn |

* + 1. Tính tổng doanh thu theo loại sản phẩm

|  |
| --- |
| SELECT Item.Category\_Name, SUM(t.Quantity\_Sold\_kilo \* t.Unit\_Selling\_Price\_RMB\_kg) AS Total\_Revenue  FROM [dbo].[Transaction] as t  JOIN Item ON t.Item\_Code = Item.Item\_Code  WHERE t.Quantity\_Sold\_kilo > 0  GROUP BY Item.Category\_Name;  Kết quả truy vấn |

* 1. 01 câu lệnh SELECT có WHERE, GROUP BY, HAVING
     1. Lấy danh sách các mặt hàng có tổng số lượng bán ra lớn hơn 50k.

|  |
| --- |
| SELECT Item\_Code, SUM(Quantity\_Sold\_kilo) AS Total\_Quantity\_Sold  FROM [dbo].[Transaction] as t  WHERE Quantity\_Sold\_kilo > 0  GROUP BY Item\_Code  HAVING SUM(Quantity\_Sold\_kilo) > 50;  Kết quả truy vấn |

* 1. 07 câu lệnh gồm SELECT có WHERE, GROUP BY, HAVING và ORDER BY
     1. Lấy danh sách các loại sản phẩm có tổng doanh thu lớn hơn 5000,

Sắp xếp theo doanh thu giảm dần

|  |
| --- |
| SELECT i.Category\_Name, SUM(t.Quantity\_Sold\_kilo \* t.Unit\_Selling\_Price\_RMB\_kg) AS Total\_Revenue  FROM [dbo].[Transaction] as t  JOIN Item as i ON t.Item\_Code = t.Item\_Code  WHERE t.Quantity\_Sold\_kilo > 0  GROUP BY i.Category\_Name  HAVING SUM(t.Quantity\_Sold\_kilo \* t.Unit\_Selling\_Price\_RMB\_kg) > 5000  ORDER BY Total\_Revenue DESC;  Kết quả truy vấn |

* + 1. Câu truy vấn thông kế tổng doanh thu theo loại sản phẩm và thời gian (năm tháng).

|  |
| --- |
| SELECT Item.Category\_Name,  DATEPART(YEAR, t.Date) AS Year,  DATEPART(MONTH, t.Date) AS Month,  SUM(t.Quantity\_Sold\_kilo \* t.Unit\_Selling\_Price\_RMB\_kg) AS Total\_Revenue  FROM [dbo].[Transaction] as t  JOIN Item ON t.Item\_Code = Item.Item\_Code  WHERE t.Sale\_Or\_Return = 'Sale' -- Chỉ tính các giao dịch bán  GROUP BY Item.Category\_Name, DATEPART(YEAR, t.Date), DATEPART(MONTH, t.Date)  ORDER BY Year, Month;  Kết quả truy vấn |

* + 1. Câu truy vấn thống kế lợi nhuận gộp theo loại sản phẩm và thời gian

|  |
| --- |
| SELECT Item.Category\_Name,  DATEPART(YEAR, t.Date) AS Yea  DATEPART(MONTH, t.Date) AS Month,  SUM(t.Quantity\_Sold\_kilo \* (t.Unit\_Selling\_Price\_RMB\_kg - Price.[Wholesale\_Price\_RMB\_kg])) AS Gross\_Profit  FROM [dbo].[Transaction] as t  JOIN Item ON t.Item\_Code = Item.Item\_Code  JOIN Price ON t.Item\_Code = Price.Item\_Code AND t.Date = Price.Date  WHERE t.Sale\_Or\_Return = 'Sale' -- Chỉ tính các giao dịch bán  GROUP BY Item.Category\_Name, DATEPART(YEAR, t.Date), DATEPART(MONTH, t.Date)  ORDER BY Year, Month;  Kết quả truy vấn |

* + 1. Tổng doanh thu và số lượng bán ra của các sản phẩm được giảm giá.

|  |
| --- |
| SELECT i.Item\_Name,  SUM(t.Quantity\_Sold\_kilo) AS Total\_Quantity\_Sold,  SUM(t.Quantity\_Sold\_kilo \* t.Unit\_Selling\_Price\_RMB\_kg) AS Total\_Revenue  FROM [dbo].[Transaction] as t  JOIN Item as i ON t.Item\_Code = i.Item\_Code  WHERE t.Discount\_Yes\_No = 'YES'-- Chỉ tính các giao dịch có giảm giá  GROUP BY i.Item\_Name  ORDER BY Total\_Revenue DESC;  Kết quả truy vấn: |

* + 1. Giá vốn hàng bán (COGS) theo loại sản phẩm và thời gian

|  |
| --- |
| SELECT Item.Category\_Name,  DATEPART(YEAR, t.Date) AS Year,  DATEPART(MONTH, t.Date) AS Month,  SUM(t.Quantity\_Sold\_kilo \* Price.Wholesale\_Price\_RMB\_kg) AS COGS  FROM [dbo].[Transaction] as t  JOIN Item ON t.Item\_Code = Item.Item\_Code  JOIN Price ON t.Item\_Code = Price.Item\_Code AND t.Date = Price.Date  WHERE t.Sale\_Or\_Return = 'Sale' -- Chỉ tính các giao dịch bán  GROUP BY Item.Category\_Name, DATEPART(YEAR, t.Date), DATEPART(MONTH, t.Date)  ORDER BY Year, Month;  Kết quả truy vấn |

* + 1. Phân tích tỷ lệ hao hụt của các sản phẩm

|  |
| --- |
| SELECT Item.Category\_Name,  AVG(LossRate.Loss\_Rate) AS Average\_Loss\_Rate  FROM Item  JOIN LossRate ON Item.Item\_Code = LossRate.Item\_Code  GROUP BY Item.Category\_Name  ORDER BY Average\_Loss\_Rate DESC;  Kết quả truy vấn |

* + 1. Sản phẩm có lợi nhuận gộp cao nhất cho các sản phẩm giảm giá

|  |
| --- |
| SELECT Item.Item\_Name, t.Discount\_Yes\_No,  SUM(t.Quantity\_Sold\_kilo \* (t.Unit\_Selling\_Price\_RMB\_kg - Price.Wholesale\_Price\_RMB\_kg)) AS Gross\_Profit  FROM [dbo].[Transaction] as t  JOIN Item ON t.Item\_Code = Item.Item\_Code  JOIN Price ON t.Item\_Code = Price.Item\_Code AND t.Date = Price.Date  WHERE t.Discount\_Yes\_No = 'YES'  GROUP BY Item.Item\_Name, t.Discount\_Yes\_No  HAVING SUM(t.Quantity\_Sold\_kilo \* (t.Unit\_Selling\_Price\_RMB\_kg - Price.Wholesale\_Price\_RMB\_kg)) > 0  ORDER BY Gross\_Profit DESC; |

* + 1. 01 câu lệnh gồm SELECT có WHERE, GORUP BY, HAVING và truy vấn con
    2. Lấy danh sách các mặt hàng có tổng doanh thu cao hơn mức trung bình của tất cả các mặt hàng.

|  |
| --- |
| SELECT Item\_Code, SUM(Quantity\_Sold\_kilo \* Unit\_Selling\_Price\_RMB\_kg) AS Total\_Revenue  FROM [dbo].[Transaction] as t  WHERE Quantity\_Sold\_kilo > 0  GROUP BY Item\_Code  HAVING SUM(Quantity\_Sold\_kilo \* Unit\_Selling\_Price\_RMB\_kg) >  (SELECT AVG(Quantity\_Sold\_kilo \* Unit\_Selling\_Price\_RMB\_kg) FROM [dbo].[Transaction]); |

* 1. 01 câu lệnh INSERT có điều kiện

|  |
| --- |
| INSERT INTO Item (Item\_Code, Item\_Name, Category\_Code, Category\_Name)  SELECT 106973990980124, 'New Product', 1011010101, 'Electronics'  WHERE NOT EXISTS (SELECT 1 FROM Item WHERE Item\_Code = 106973990980124);  Kết quả thực thi |

* 1. 01 câu lệnh UPDATE có điều kiện

|  |
| --- |
| UPDATE Price  SET [Wholesale\_Price\_RMB\_kg] = 10  WHERE Item\_Code = '102900005115762' AND [Wholesale\_Price\_RMB\_kg] < 5;  Kết quả thực thi |

* 1. 01 câu lệnh DELETE có điều kiện

|  |
| --- |
| DELETE FROM Price  WHERE [Wholesale\_Price\_RMB\_kg] < 2  Kết quả thực thi |