Vietnam National University - Ho Chi Minh City
Ho Chi Minh City University of Technology
Faculty of Computer Science and Engineering

Database Systems

ASSIGNMENT 1

Instructor(s): Ms. Võ Thị Ngọc Châu
Ms. Nguyễn Thị Ái Thảo

Student(s): Lê Quốc Bảo - 2252065 (Leader-Team BK_Shop)
Ngô Phan Khải Tú - 2153951 (Team BK_Shop)
Lê Phan Nhật Minh - 2252478 (Team BK_Shop)
Tiêu Trí Bằng - 2252079 (Team BK_Shop)

Ho Chi Minh City, 10/2024

Contents

1 Detailed Business Description

Our retail business specializes in the sale of laptops and electronic accessories, catering to diverse customer needs with a wide selection of products. The laptop offerings range from basic models for everyday use to high-performance systems tailored for gaming or professional work. In addition to laptops, our store features an extensive accessory lineup, including essential peripherals such as mice, keyboards, and headphones, which enhance the overall computing experience.

To support our operations, we rely on a robust database system. This database helps us manage key areas like employee information, product inventory, sales transactions, and customer details. Through it, we can efficiently monitor stock levels, ensure timely replenishment of products, and analyze purchasing trends to meet customer demands.

Furthermore, the system records customer purchase histories, tracks loyalty or discount programs, and processes customer requests, enhancing the shopping experience. Employee management is also streamlined, with distinct roles like store managers and warehouse managers, who oversee daily store activities and inventory control, respectively. This structure ensures smooth operations and accountability.

By leveraging this database system, we aim to enhance operational efficiency, offer personalized service, and increase overall customer satisfaction, securing a competitive edge in the retail market.

2 Entity relation diagram

Entity relation diagram of our database:

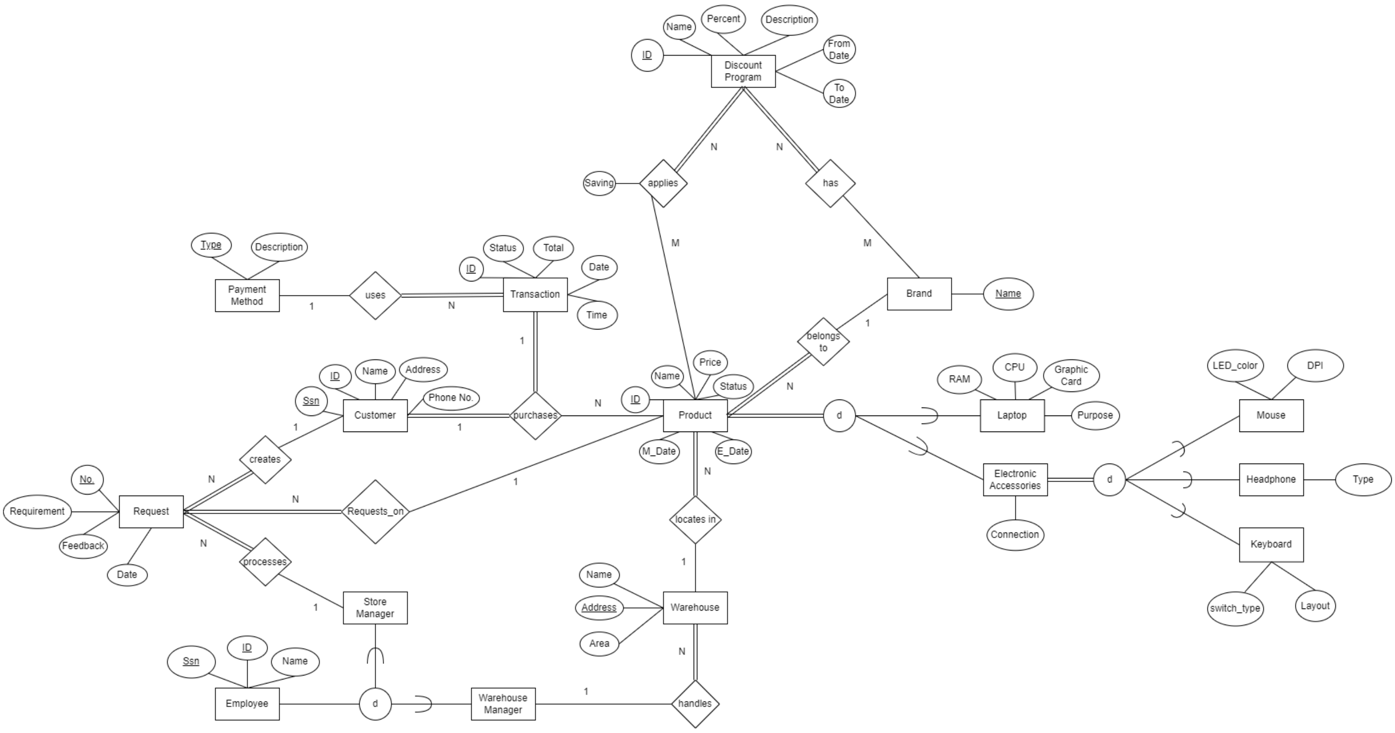

Figure 1: Entity relation diagram

3 Relational database schema

1. **Employee** (SSN, ID, Name)
   Primary Key: ID
   Secondary Key: SSN
   Not null: Name

2. **Store_Manager** (E.ID)
   Primary Key: E.ID
   Foreign Key: E.ID references Employee.ID

3. **Warehouse_Manager** (E.ID)
   Primary Key: E.ID
   Foreign Key: E.ID references Employee.ID

4. **Warehouse** (Name, Address, Area, WM.ID)
   Primary Key: Address
   Foreign Key: WM.ID references Warehouse_Manager.ID
   Not null: WM.ID

5. **Customer** (SSN, ID, Name, Address, Phone)
   Primary Key: ID
   Secondary Key: SSN
   Not null: Name, Address, Phone

6. **Product** (ID, Name, Price, M_Date, E_Date, Status, B.Name, W.Addr, C.ID, T.ID)
   Primary Key: ID
   Foreign Keys: B.Name references Brand.Name, W.Addr references Warehouse.Address, C.ID references Customer.ID, T.ID references Transaction.ID
   Not null: B.Name, W.Addr, Name, Price, Status, M_Date, E_Date

7. **Laptop** (RAM, CPU, Graphic_card, Purpose, P.ID)
   Primary Key:P.ID
   Foreign Key: P.ID references Product.ID
   Not null: RAM, CPU, Graphic_card

8. **Electronic_Accessories** (Connection, P.ID)
   Primary Key: P.ID
   Foreign Key: P.ID references Product.ID
   Not null: Electronic_Accessories

9. **Mouse** (LED_color, DPI, P.ID)
   Primary Key: P.ID
   Foreign Key: P.ID references Product.ID
   Not null: LED_color, DPI

10. **Keyboard** (switch_type, layout, P.ID)
    Primary Key: P.ID
    Foreign Key: P.ID references Product.ID
    Not null: switch_type, layout

11. **Headphone** (Type, P.ID)
    Primary Key: P.ID
    Foreign Key: P.ID references Product.ID
    Not null: Type

12. **Brand** (Name)
    Primary Key: Name

13. **Discount_Program** (ID, Name, Description, FromDate, ToDate, Percent)
    Primary Key: ID
    Not null: Name, FromDate, ToDate, Percent

14. **Transaction** (ID, Date, Time, Status, PM.Type, Total)
    Primary Key: ID
    Foreign Key: PM.Type references Payment_Method.Type
    Not null: PM.Type, Date, Time, Status, Total

15. **Payment_Method** (Type, Description)
    Primary Key: Type

16. **Request** (No., Requirement, Feedback, Date, C.ID, P.ID, SM.ID)
    Primary Key: No.
    Foreign Keys: C.ID references Customer.ID, P.ID references Product.ID, SM.ID references Store_Manager.ID
    Not null: C.ID, P.ID, SM.ID, Date

17. **has** (B.Name, D.ID)
    Primary Key: (B.Name, D.ID)
    Foreign Keys: B.Name references Brand.Name, D.ID references Discount_Program.ID
    Not null: B.Name, D.ID

18. **applies** (P.ID, D.ID, Saving)
    Primary Key: (P.ID, D.ID)
    Foreign Key: P.ID references Product.ID, D.ID references Discount_Program.ID
    Not null: P.ID, D.ID, Saving

4 Relational algebra operators

1. Retrieve all products that belong to a specific brand (Brand "Asus"):
$\sigma_{\text{B.Name = "Asus"}} (\text{Product}_{\text{Product.B.Name = Brand.Name}} \bowtie \text{Brand})$

2. Retrieve requirements that include products priced above a specified amount (e.g., \$500):
$\pi_{\text{Require.No., Product.Name, Product.Price}}(\sigma_{\text{Product.Price>500}}(\text{Require} \bowtie_{\text{Require.P.ID=Product.ID}} \text{Product}))$

3. Retrieve the information of all laptops with a specified RAM size of filter attribute.
$\sigma_{\text{Laptop.RAM = "8GB", Laptop.CPU = "i5"}} (\text{Laptop} \bowtie_{\text{Laptop.P.ID = Product.ID}} \text{Product})$

4. List all discount programs available for a brand (Brand "Asus").
$\pi_{\text{Discount\_Program.ID, Name, Percent}} \left(\sigma_{\text{B.Name = "Asus"}} \left(\text{has} \bowtie_{\text{has.D.ID = Discount\_Program.ID}} \text{Discount\_Program}\right)\right)$

5. Retrieve all transactions that happened on a specific date. Assume the date is 1/1/2024.

$\pi_{\text{C.ID, P.ID, Date, Time, Pur.ID}} \left(\sigma_{\text{Date = "1/1/2024"}} \left(\text{Transaction}\right)\right)$

6. Find the total value of transactions by each customer:

$_{\text{C.ID}}\, \Im_{SUM(Transaction.Total)} \left(\text{Transaction} \bowtie_{\text{Transaction.C.ID=Customer.ID}} \text{Customer}\right)$

7. Count the total sales for each laptop over the past month.

$\Im_{\text{SUM(Transaction.ID)}} \left(\sigma_{\text{Date} \geq \text{Today}-30} (\text{Trans}) \bowtie_{\text{Trans.ID=Product.T.ID}} \text{Product} \bowtie_{\text{Product.ID=Laptop.P.ID}} \text{Laptop}\right)$

8. What is the minimum suggested price for each brand.

$_{\text{B.Name}}\, \Im\, _{\text{MINIMUM Price}} \left(\text{Product}\right)$

9. Identify the products that has more than 100 customer feedback ?

$\Im_{\text{COUNT(No.)}>100} \left(\Im_{\text{P.ID, COUNT(No.)}} (\text{Requirement})\right)$

10. What is the total saving cost of a discount program ? Assume discount program is "Christmas".

$\Im_{\text{SUM}}\left(\pi_{\text{Saving}}\left(\text{Applies} \bowtie_{\text{Applies.D\_ID=Discount\_Program.ID}} \sigma_{\text{Name='Christmas'}} (\text{Discount\_Program})\right)\right)$

5 Constraints

1. Attribute Comparison Constraint

$$\text{Manufacture Date} < \text{Expired Date}$$

Explanation: This constraint enforces that the "Manufacture Date" of a product must always be before its "Expired Date". It prevents logical errors in data by ensuring that no product has an expiration date that precedes its manufacture date.

2. Range Constraint

$$\text{Percent} \in (0, 80)$$

Explanation: This constraint restricts the "Percent" attribute (such as a discount percentage) to values within the range of 0 and 80 (exclusive). This means that any percentage value must be greater than 0% and less than 80%.

3. Temporal Constraint

$$\text{FromDate} \leq \text{ToDate}$$

Explanation: This constraint ensures that the "FromDate" (starting date) is always on or before the "ToDate" (ending date). It ensures the logical sequence of dates, preventing situations where a period's starting date is mistakenly set after its ending date.

4. Status Constraint

$$\text{Product Status} = \text{"Second Hand"} \Rightarrow \text{Discount Applied}$$

Explanation: This constraint mandates that if a product's status is marked as "Second Hand," a discount must be applied to its price.

5. Discount Program Validity Constraint

$$\text{Discount Program End Date} < \text{Transaction Date} \Rightarrow \text{Discount Not Applied}$$

Explanation: This constraint ensures that if a discount program's end date is before the transaction date, the discount is not applied to the purchase. This prevents customers from benefiting from discounts that have already expired.