收银系统

Cash register system

该系统可以为用户提供便捷高效的数据处理和管理功能，旨在帮助用户轻松管理销售和商品信息。通过该系统，您可以创建Customer（客户）、Brand（品牌）、ProductType（产品类型）、Product（产品）、ProductProductType（产品与产品类型关联），以及Purchase（购买）表格，建立CustomerInformation（客户信息）、Productinfo（产品信息）、PurchaseInfo（购买信息）、Salesranking（销售排名）、及BrandSales（品牌销售）视图。用户可以通过插入信息记录销售和商品数据，并方便地查看相关信息，从而更好地管理和分析销售业务。

The system provides users with convenient and efficient data processing and management functions, and is designed to help users easily manage sales and product information. With this system, you can create Customer, Brand, ProductType, Product, ProductProductType (product associated with product type), and Purchase tables, create CustomerInformation, Productinfo, PurchaseInfo, Salesranking, and BrandSales views. Users can record sales and product data by inserting information, and conveniently view relevant information to better manage and analyze sales operations.

**一.** E-R 图以及解释性说明。 （E-R diagram as well as explanatory notes.）

**实体（entity）**：Customer，Brand，ProductType，Product

**关系（relationship）**：ProductProductType，Purchase

**Customer**: CustomerID(主键),CustomerName,Phone,LoyaltyPoints,JoinedLoyaltyProgram

**Brand**: BrandID(主键), BrandName

**ProductType**: ProductTypeID(主键), TypeName

**Product**: UPC(主键), ProductName, Price, Quantity, Sale, BrandID(外键)

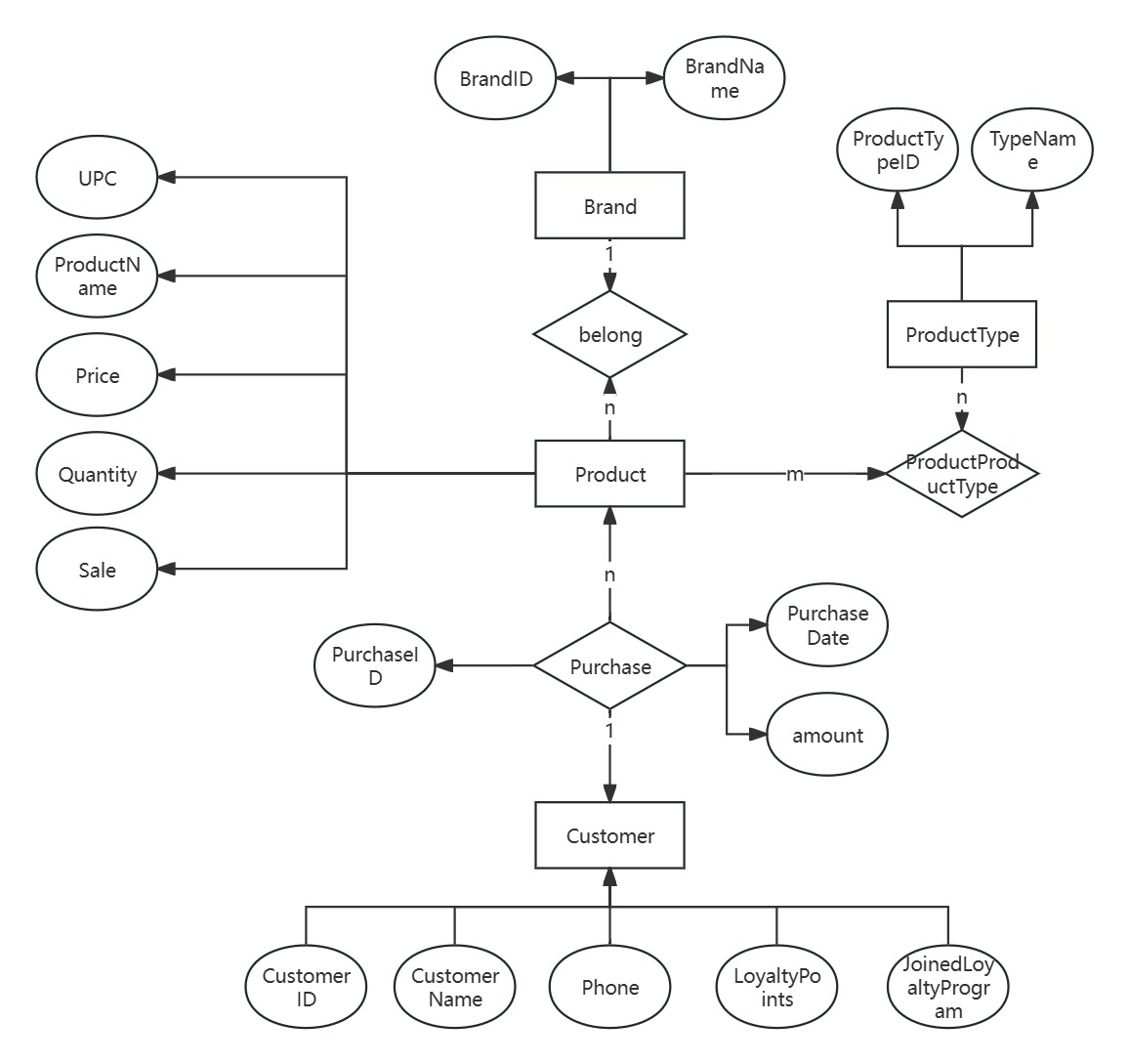
**ProductProductType**: ProductID(外键), ProductTypeID(外键)

**Purchase**: PurchaseID(主键), CustomerID(外键), UPC(外键)，PurchaseDate，amount

Product对Brand多对一

Product对ProductType多对多

Customer对Product一对多



**二**. 关系模式及设计（Relational schema and design）

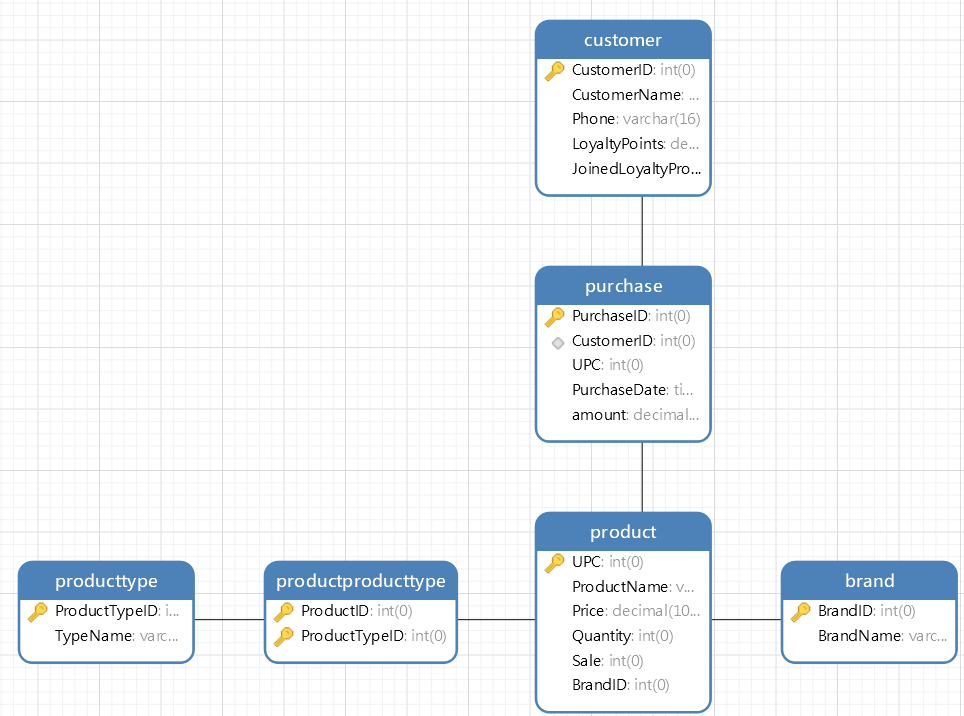
**Customer**: CustomerID,CustomerName,Phone,LoyaltyPoints,JoinedLoyaltyProgram

**Brand**: BrandID, BrandName

**ProductType**: ProductTypeID, TypeName

**Product**: UPC, ProductName, Price, Quantity, Sale, BrandID，PurchaseID，CustomerID，PurchaseDate，

**ProductProductType**: ProductID, ProductTypeID



**约束（restraint）**：

primary key主键约束

CONSTRAINT CheckQuantity CHECK (Quantity >= 0)确保商品数量不小于0

Purchase(PurchaseDate)通过DEFAULT CURRENT\_TIMESTAMP设置默认值-当前时间

**索引（index）**：

Product(BrandID)

ProductProductType(ProductTypeID)

Purchase(CustomerID, UPC)

**触发器（trigger）**：

**fillPriceBeforeInsert**: 将新记录插入“购买”表之前自动填写购买价格。（The purchase price is automatically populated before a new record is inserted into the Purchase table.）

**updateLoyaltyPoints**: 将新记录插入“购买”表之后自动将金额等比转换为顾客积分。（When a new record is inserted into the "Purchases" table, the proportional amount is automatically converted into customer points.）

**update\_product\_quantity**: 将新记录插入“购买”表之后自动将产品的数量减一。（When a new record is inserted into the Purchase table, the quantity of the product is automatically reduced by one.）

**UpdateProductSalesTrigger**: 将新记录插入“购买”表之后自动将售出数加一。（When a new record is inserted into the Purchase table, the number of sales is automatically increased by one.）

**视图（view）**：

**CustomerInformation**：顾客信息

**Productinfo**：产品信息

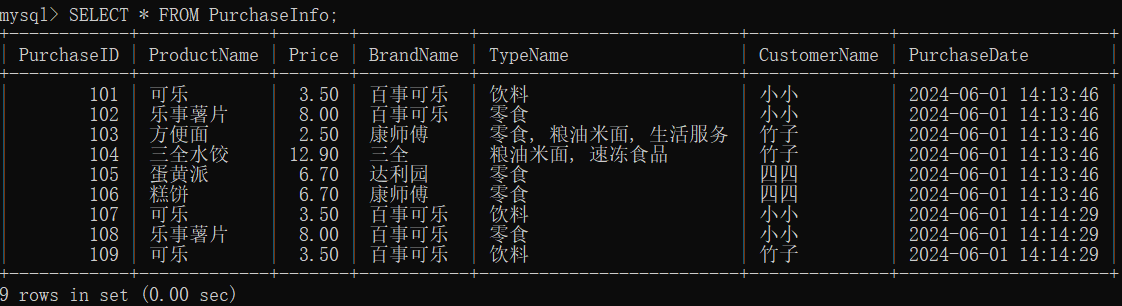
**PurchaseInfo**：购买记录

**Salesranking**：售量排名

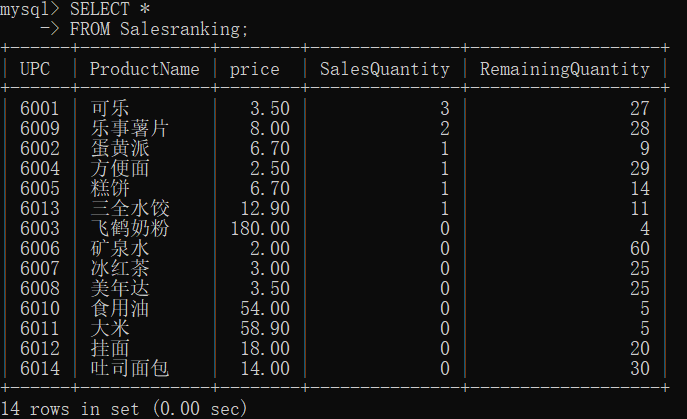
**BrandSales**：品牌销售额

**三**. 示例查询（Example query）。

SELECT \* FROM PurchaseInfo;



SELECT \* FROM Salesranking;



**四**. MySQL的命令行客户端软件（Command-line client software for MySQL）

在插入合适的信息后，用户可以通过视图可以查询顾客实时信息、商品信息、购买商品记录、商品销售量从高到低，品牌销售额排列，实现收银的销售记录。

After inserting the appropriate information, the user can query the real-time information of the customer, the product information, the purchase record, the sales volume of the product from high to low, and the brand sales volume through the view, so as to realize the sales record of the cashier.