

Principles of Data Management

Phase 2

null'); DROP TABLE Students;

Andrew Hamrick

Daniyal Iqbal

Oscar Lowry

Qasim Ali

March 24, 2018

1 Design

Our database was designed with a Retail environment in mind. The choices for entities and relations were made by considering what retail employees and stores would likely need to check or keep track of. The retail environment envisioned focuses on electronic devices. Currently our design supports Computers and TVs but can easily be expanded to other product types. We decided to focus on computers and TVs as that is an area of expertise for us.

Our database centers around the product entity as that is largely seen as the most important component of a retail environment. The other entities were chosen to track the flow of money in and out of the stores, or to support the product. Support for the products includes keeping track of current inventory, acquiring more of the product, and categorizing the product into product types.

2 ER Diagram

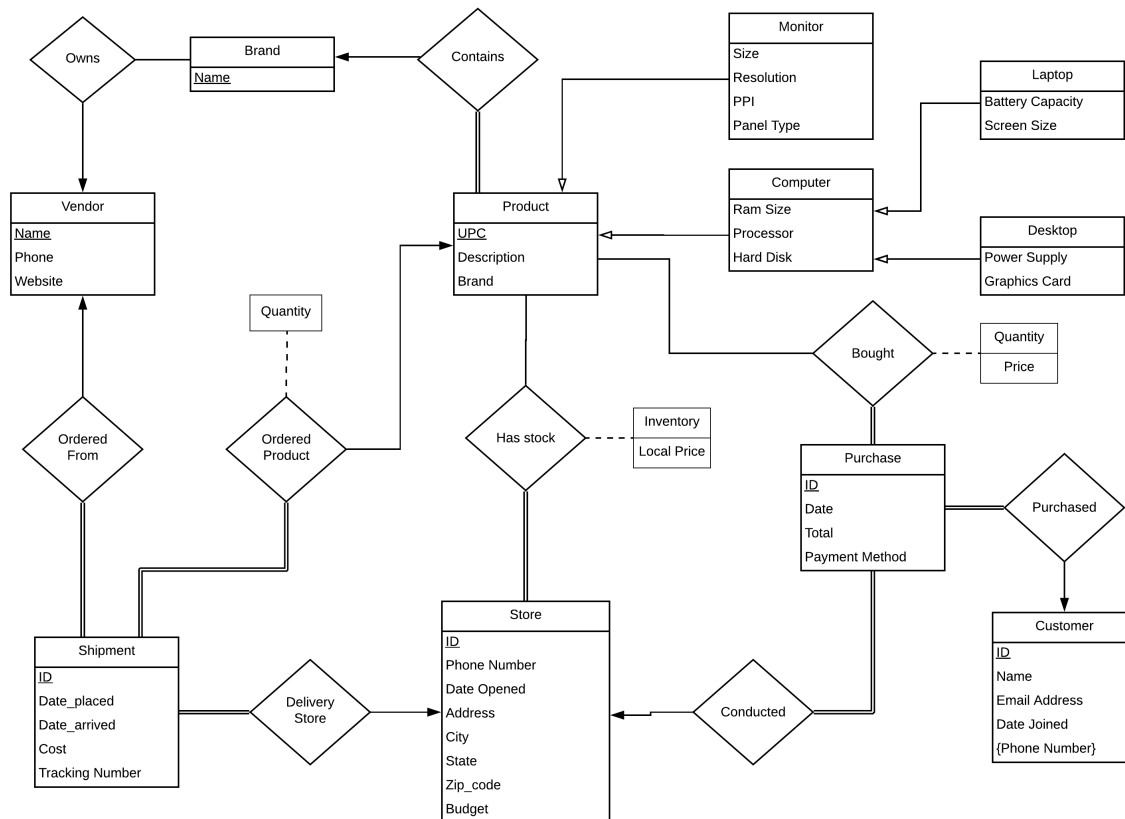


Figure 1: ER Diagram for Database

The Product entity is a major entity in our database. It has a relation with almost every entity. This makes sense for a retail database as retailers are generally focused on the products they sell. The major products sold at our establishment are Laptops, Desktops, Computers, and Monitors. Our product types are Screen and Computer as those make up the two major groups of our products. These subgroups inherit identical attributes from parent entities. TV obviously inherits from Screen. As the screen is a major component of a laptop, laptop also inherits from Screen. Desktop obviously inherits from Computer, however because a laptop is also a computer, it inherits from Computer in addition to Screen.

The Tables for our database are:

- Product(UPC, Description, Brand);
- Laptop(UPC, Battery_Life, Screen_Size);
- Brand(Name, Vendor);
- Store(ID, Phone_Number, Date_Opened, Budget, Address, City, State, Zip_Code);
- Desktop(UPC, Power_Supply, Graphics_Card);
- Computer(UPC, Ram_Size, Processor, Hard_Disk);
- Monitor(UPC, Size, Resolution, PPI, Panel_Type);
- Customer(ID, Name, Email_Address, Date_Joined, Phone_Number_1, Phone_Number_2);
- Vendor(Name, Phone, Website);
- Shipment(ID, Date_placed, Date_arrived, Cost, Tracking_number, Vendor_name, UPC, Quantity, Store_ID);
- Purchase(ID, Date, Total, Payment_Method, Customer_ID, Store_ID);
- Bought_Products(Product_UPC, Purchase_ID, Quantity, Price);
- Stock(Store_ID, UPC, Inventory, Listed_Price);

2.1 Products

Product contains the attributes UPC and Description. UPC is a unique identifier to identify the product in the database. Description is a short description of the product for customer and employee benefit. All other attributes associated with a product specified by that products subtype. Our subtypes are Computer and Monitor. All products must also have a UPC that matches one in the product above it. For example, a Laptop must match a computer that must match a product.

2.1.1 Monitor

Monitor was chosen as a product type due to our focus on electronics. A large number of electronics these days feature a screen. In addition, the monitor is a major consideration in the purchase of an electronic device.

2.1.2 Computer

Computer is the other major section of the products represented by our database. Computer's attributes that are common among all types of computers are RAM size, the processor, and the Hard Drive size.

- Laptop
Laptop inherits from both Computer and Screen because it fits well in both categories. All important parts of a Laptop are covered in either Screen or Computer, except for Battery Life.
- Desktop
A desktop computer is just the tower. It does not include a keyboard, a mouse, or a monitor, therefore it does not inherit from Screen. Attributes that apply to desktops that don't apply to other computers (Laptops) are the Power Supply, and the Graphics Card.

2.2 Brand

Brand is an alternate way to group products together. The major aspects of Brand is the Name and the vendor it belongs to. The Brand names must not match the brand name of any other brand, and the vendor must pair with an existing vendor in the vendor table.

2.3 Vendor

A vendor represents a company that our stores purchase products from to sell to consumers. A vendor is represented by their company name, their phone number, and their website. A vendor is identified by their name as that is how they are identified.

2.4 Shipment

Shipment represents an shipment placed by an employee to resupply a stores stock. The shipment consists of a unique ID to identify it, the date it was placed, the date it arrived, the total cost of the shipment, the tracking number for the shipment, the vendor name, a UPC, and the quantity, and a store ID. The quantity and UPC are used to keep track of how many of each product was shipment. The vendor name is used to link the shipment with the vendor. The store id is used to keep track of the store the product is being delivered to.

2.5 Bought_Products

A table was needed to handle the relation between Purchase and Product as multiple of the same product could be purchased in one purchase. Price also needed to be tracked as prices of a particular product change at different stores.

2.6 Store

Our database may need to track information about multiple stores. A store is represented by a unique ID, the store's phone number, the date the store opened, a yearly budget for the store, and the store's address.

2.7 Stock

A table is needed to manage interactions between Stores and Products. A store will sell multiple products, and a product may be stored at multiple stores. Stores will have varying inventory of a certain product, and may also vary the products type. Stock is represented by a store's unique ID, the UPC of a product, how much of that product is currently in the store, and the price the store is selling the product for.

2.8 Customer

Our stores offer a customer loyalty program. This allows customers to be tied to the purchases they make. A customer is represented by a unique ID, their name, their email address, the date they joined the program, and their phone number(s).

2.9 Purchase

A purchase consists of a unique ID, the date it was placed, the payment total, the method of payment (e.g. Credit Card, Debit Card, Cash, etc.), a customer ID (e.g. phone number, etc.), and the store ID. A purchase can be tied to a customer for the loyalty program. A purchase will not be tied to a customer if the customer is not a part of the loyalty program. Due to the loyalty program it is likely a customer will be tied to multiple purchases, but a single purchase will only be tied to a single customer.

3 Application

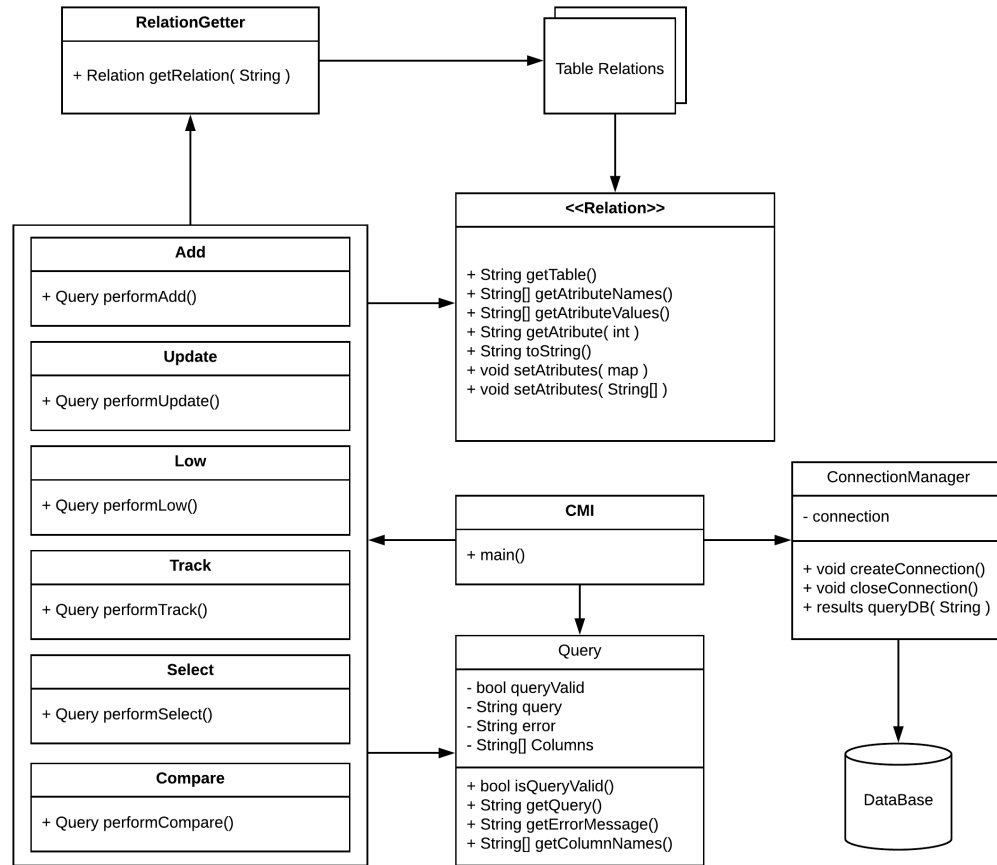


Figure 2: UML Diagram for Application

3.1 CMI

The CMI handles getting the next command from the user, calling the functions for the specified commands, initializing the connections, sending queries to the connection manager and dealing with query results.

3.2 ConnectionManager

This holds the connection to the database and is the only one to query the database.

3.3 Relations

Using a relation interface, we will implement a class for every table our project deals with. The relation interface will specify functions that will allow getting the values and columns without having to know what relation is being used. This allows for creation of generic queries usable with all possible relations.

3.4 Commands

Every command has its own class that generates code based on its specified function (see use cases). They will also get the specified values for the sql using the relation interface to select the requirements for its query. It returns a query or an error depending on the values given by the user.

3.5 RelationGetter

Returns a relation object of the type specified.

4 Test Data

Our test data was generated to fit the tables made from our ER diagram. The data was mostly generated through a random data generator (<https://www.mockaroo.com/>) and manual construction. Data that was independent of other data (such as stores and customers) was generated by the random data generator. The rest of the data was generated manually and/or through the random data generator to maintain consistency across different tables.

5 User Interface

Our user interface is a command line application using straight forward commands. The current commands are as follows.

- Add: customer, transaction, store, product, order, vendor, brand

Examples:

- Add customer name="John Smith" email=john@gmail.com phone=5855551234
- Add transaction customer=123 product "Sony TV 52" 2

- Update: store, stock, price, customer

Examples:

- Update store 123 phone 1235557890
- Update product "Dell Laptop 15" price=1000 store=123

- Lowest Inventory:
 - Product
 - Number of low inventory products to show
 - Sort field: inventory number, expected days until out of stock
 - Search field: stores

Examples:

- Low 50 products store=123
- Low 5 days store=all

- Top/Bot: Number of top or bottom records
 - Search term: products, product type, stores, brand, vendor
 - Sort field: revenue, total items sold
 - Criteria: product type, month/year, store

Examples:

- Top 5 stores items year=2017
- Bot 10 computers revenue store=12
- Top 1 brand items product=computer
- Top 5 products revenue

- Compare:
 - Comparison of: store, state
 - Comparison term: brands, products, vendors
 - Comparison field: revenue, items sold

Examples:

- Compare stores products laptops computers revenue
- Compare states brands Dell HP items

- Track: Show items commonly bought together
 - Number of records to show
 - Products, product type

Examples:

- Track 2 Desktop

6 Design Changes

A couple design changes were made since Phase 1. The changes are as follows:

- TV was removed
- Screen changed to Monitor
- Transaction was renamed to Purchase
- Order was renamed to Shipment

- Order/Shipment was changed to include a store
- Store's address was expanded to include City, State, and Zip

Several changes were made from Phase 1 to improve clarity and simplify the design. TV was removed to reduce the number of tables needed for our design. Laptop was separated from Screen for simplicity. Screen was then changed to Monitor and became a separate product. Transaction was renamed to Purchase because transaction is a keyword for SQL. Order was renamed to Shipment because ORDER is an SQL keyword. Shipment was extended to include a store so the design could work with multiple stores. Our design for our UI included the ability to query by state. In order for us to know a Store's state, Store's address needed to be expanded. Address is now just the street address and Store now includes City, State, and Zip Code.

6.1 TV and Monitor