COP 5725 - Fall 2015

e-Commerce Website Database and Data Interface

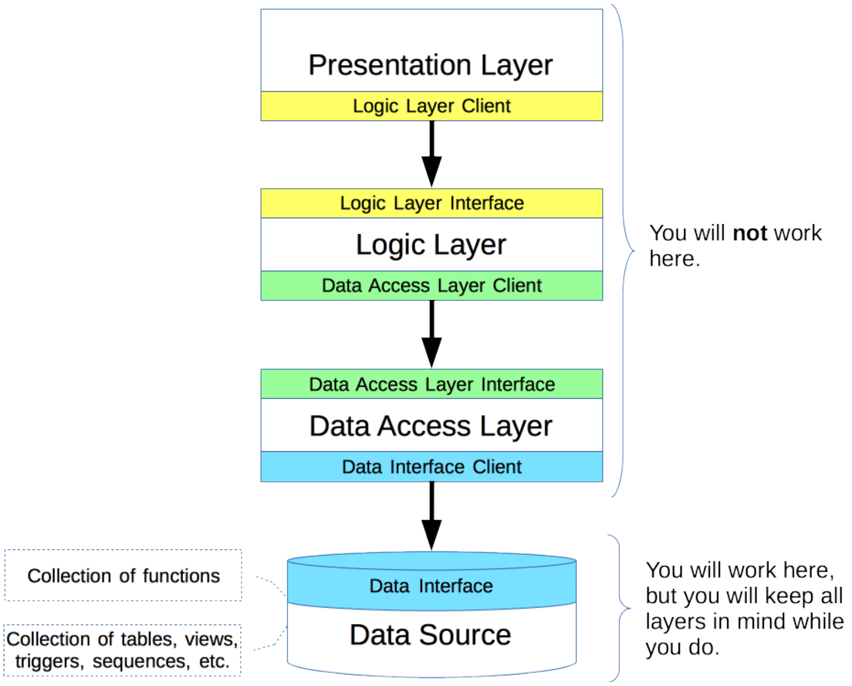
Term Project

Database

The database contains most of the information used by the web application. A database is a collection of related data in the form of schemas, tables, views, stored procedures, triggers, indexes, and other objects.

Data Interface

The data interface is the point of contact between the application data access layer and the database. Similar to the software package interfaces you already know, the data interface provides a pre-defined set of functions that will serve as the gateway to the data from the application's data access layer.


Presentation Layer
Logic Layer Client
Logic Layer Interface
Logic Layer
Data Access Layer Client
Data Access Layer Interface
Data Access Layer
Data Interface Client
You will **not** work here.
Collection of functions
Collection of tables, views, triggers, sequences, etc.
Data Interface
Data Source
You will work here, but you will keep all layers in mind while you do.

Universe of Discourse

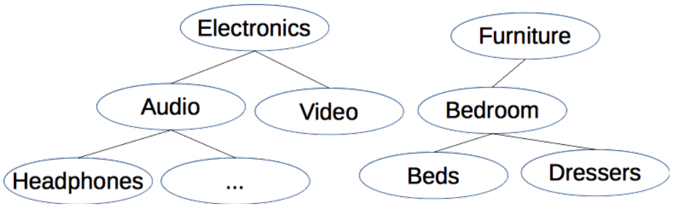
A database represents some aspect of the real word called the miniworld or the universe of discourse. You will design a database based on the description of the miniworld contained in this section. The subject matter of this project was selected to increase the likelihood of students being familiar with its main concepts and make it easier to find relevant information online to guide the design decisions; however, when explicit requirements given in this description contradict information you find from other sources regarding this subject matter, your solution must satisfy the requirements given here. **You must explicitly state and justify all assumptions that are not part of this description.**

Your task consists of the design of the database and data interface of an e-commerce website. As such, you will deal with customers, suppliers, products, product categories, carriers, orders, shipments, returns, exchanges, payments, and refunds.

You will need to store information on the full name of your customers and one or more addresses to be used for shipping and billing. Each customer may have one or more credit cards on file, each one associated with a billing address. You may not duplicate addresses in the database, although the same address may be used for multiple purposes (e.g. the billing address of the card used to pay for an order and the shipping address of the order). You must allow customers to specify their preferred shipping address and payment method. In order to communicate with your customers, you must store their e-mail address and have the option of adding their phone number. You will control access to customers' accounts with credentials consisting of username and password. The username will be the customer's e-mail address and the password will be stored encrypted in the database. You must allow customers to change their e-mail address (and hence, username) and password at any time. Each customer will have a non-negative store credit balance. This balance can be topped-up by the customer or increased by the company as a result of a refund.

The company running the e-commerce business does not manufacture any products. Instead, it purchases all products from suppliers. In order to expedite orders, the company maintains a stock of all the products it offers for sale. You need to keep a list of suppliers and store the company name, business address, sales representative's contact information, and discount percentage (i.e., a fixed percent discount applied to all purchases from this supplier). Each product must have at least one supplier. Each supplier in the database must supply at least one product. Each supplier may supply products at different prices.

All products must have a name, description, picture, and price. Each product must be associated with exactly one category. You will design a way of storing trees of categories. Even though you should think of the categories hierarchy as trees, you will store them as a table in the database. Populate the table with a set of categories of your choice. Your chosen category hierarchy must be at least three levels deep. Products may be associated with any category, not just with the leaves of your trees. The following are two very basic trees:



Customers will add products to the shopping cart and you must store the contents of the cart so that it can be preserved across visits. Each order must be associated with exactly one debit/credit card. You can assume that cards are valid and have enough funds to cover the transaction. Store credit balances are automatically applied to payments when an item ships; only the outstanding balance after using up the store credit is charged to the customer's card. Since customers can update or delete cards and addresses on their accounts, you must store a copy of the payment and shipping information for each order so that the system can generate invoices in the future (you will not model invoices, though). It must be possible to determine the amount of store credit and card charge used to pay for an item. When a customer completes the purchase of the items in the shopping cart, those items must be cleared from the cart. The company inventory is updated only when an item is shipped; however, you must keep track of the items ordered but not yet shipped so that you don't allow any item to be oversold. Since product pricing may change at any time, you must store the price of each product at the time the order is placed so you know what price to charge when the item is shipped. Each product will have a low inventory threshold. When shipping an order brings the count of a product below the threshold, a restocking reminder must be generated by the database.

Restocking reminders are used by the company to place orders with suppliers. You will not model orders to suppliers beyond the details of restocking reminders. A restocking reminder must have information about the product version and the best supplier to order from. The system will determine the best supplier at the time of creation of the reminder by comparing the final cost of the item to the company taking into account the supplier's price for the item and the arranged discount, if any.

A customer may return any part of an order or the entire order within 30 days of placing the order. You must create and store each return request with its associated unique Return Merchandize Authorization code and status (initiated, denied, completed). We will use the simplifying assumption that all accepted return items can be re-sold immediately. When a return is complete, you need to issue a refund (to the customer's store credit balance), record the refund transaction, and update the store's inventory.

A customer may also exchange an item for one of equal or less price (you are not required to model or check restrictions on eligible items for exchanges beyond their price). If the new item costs less than the original item, you must refund the difference to the customer's store credit and store the transaction for future reference. You must update the store inventory accordingly.

Once an order is shipped, it can no longer be modified for any reason. Returns and exchanges are related to orders, but are recorded separately. Customers can change an order before it is shipped and this does not constitute an exchange. Changes to unshipped orders are reflected directly in the original order. Given the possibility of changes, you may not charge the customer's account or update the store inventory until the order ships. We will use the simplifying assumption that packages arrive immediately after shipping.

The e-commerce company hires different carriers to deliver orders to customers. Not all carriers ship all kinds of products to all destinations. You must at least model the following services offered by carriers: (a) international/domestic (all US territories) shipping, (b) hazardous/non-hazardous materials. All items in an order must be shipped to a single address; however, the order may be shipped in multiple packages using multiple carriers if it contains multiple items. Some products may require separate shipment due to their size or weight. Each product must indicate whether it contains hazardous materials and whether it can be shipped with other products. You must store all shipments associated with each order and include information about the carrier and ship date. When an order is fulfilled in multiple shipments, all shipments must be marked as being "partial." In your design, it must be possible to find out which shipment contained any item of any order.

Deliverables

ER Diagram: Turn in as a PDF document. State and explain all assumptions. Use the notation from the class notes for the ER diagram.
Due date: Sep. 29, 2015 (TUE)

Important Note

As deliverables are related, you will be given the opportunity to improve your previous work so that your future work does not carry over mistakes. In this regard, when a deliverable is due, you will be given the opportunity to turn in the previous deliverable if you revised it based on the feedback you received when it was graded. The due deliverable will then be graded considering its consistency with the revised deliverable, not with the original one. Note that this is optional and that it always applies to *the* previous deliverable only, not to all previous deliverables. If you choose to take advantage of this opportunity to improve your previous work, your grade should improve since your future deliverables should get better over the course of the semester.

Submission

Submit your PDF / sql file with the format **Deliverable#_YourName_Fall2015** as the filename and upload it to Moodle system.

For example, for first deliverable, you should name your file as Deliverable1_LebronJames_Fall2015.pdf

pgAdmin Instruction

- You have to first download the software called pgAdmin from http://www.pgadmin.org/
- Choose the latest version from the link PostgreSQL mirror network and download it on your computer.
- Double click the execution file (msi or dmg).
- Keep all the default setup and finish the installation process

Log in to pgAdmin Server

- ### On-Campus Instruction

If you are connecting to the servers using FIU IP, please start by doing the following steps,

1. Open your pgAdmin
2. Click on the top left most button called "Add a connection to a server"
3. Click on the link on top called "Properties"
4. Fill up the information as follows:

   Name: COP5725

   Host: cop5725-postgresql.cs.fiu.edu

   Maintenance DB: postgresql

   Username: fall15_FIUAccount (For instance, fall15_hha001)

   Password: Your pantherID (For instance, 1234567)

- ### Off-Campus Instruction

If you are connecting to the servers using IP outside the campus, please start by doing the following steps,

1. Open your pgAdmin
2. Click on the top left most button called "Add a connection to a server"
3. Click on the link on top called "SSH Tunnel"
4. Check the box called "Use SSH Tunneling" and choose password for Authentication.
5. Fill up the information as follows:

   Tunnel Host: ocelot.aul.fiu.edu

   Username: Your FIU Account (For instance, hha001)

   Password: first name initial, pantherid, last name initial (For instance, h1234567h)

Once the above-mentioned steps are done, please follow the same steps mentioned in the On-Campus Instruction.