Project: Electronics vendor
Course: Databases

Enterprise description:
The application is an electronics vendor that operates both a Web site and a chain of many physical stores. Examples include Best Buy and Circuit City. To find out more about this application, think about any experiences you may have had making purchases both online and in-store, and browse their Web sites. In our hypothetical company, it has been decided to redesign a major part of the database that underlies company operations. Unfortunately, the manager assigned to solicit database design proposals is not very computer literate and is unable to provide a very detailed specification at the technical level. Fortunately, you are able to do that.
Here are a few points to consider:

- There are many different products, grouped into a variety of (possibly overlapping) categories. Groupings can by type of product (cameras, phones, etc.), by manufacturer 1 (Sony, Apple, etc.), or by other means (for example, a Gateway PC might be packaged with a Sony monitor and an HP printer and marketed as a package).

- Some customers have a contract with the company and bill their purchases to an account number. They are billed monthly. Other customers are infrequent customers and pay with a credit or debit card. Card information may be stored for online customers, but not for in-store customers.

- Online sales must be sent to a shipper. The company needs to store the tracking number for the shipping company so it can respond to customer inquiries.

- Inventory must be accurate both in stores and in warehouses used to replenish stores and to ship to online customers. When inventory is low, a reorder should be sent to the manufacturer and listed in the database. When goods arrive, inventory should be updated and reorders marked as having been filled.

- Sales data are important for corporate planning. Marketers may want to look at sales data by time period, product, product grouping, season, region (for stores), etc.

Data Generation:
For simplicity, I will not require realistic data. You can just create some names or get real ones from the electronics vendors web site. There are many different devices, grouped into a variety of (possibly overlapping) categories. If you get realistic here, things get to be interesting. The ISA relationship will get heavy use here.

Tasks:

1. E-R Model
   - Construct an E-R diagram representing the conceptual design of the database.
   - Be sure to identify primary keys, relationship cardinalities, etc.

2. Relational Model
   - After creating an initial relational design from your E-R design, refine it based on the principles of relational design (Chapter 8).

- Create the relations in PostgreSQL database.
- Create indices and constraints as appropriate.
- If as you refine your design, you discover flaws in the E-R design, go back and change it (even if the earlier design passed the checkpoint.) Your final E-R design must be consistent with your relational design.
- Your relational design have to satisfy conditions of all normal forms (Chapter 7)

3. Populate Relations
   - Include enough data to make answers to your queries interesting and nontrivial for test purposes.
   - You may find it helpful to write a program to generate test data.

4. Queries: You should run a number of test queries to see that you have loaded your database in the way you intended. The queries listed below are those that your client (the manager from the package delivery company) wants turned in. They may provide further hints about database design, so think about them at the outset of the project.
   - Assume the package shipped by USPS with tracking number 123456 is reported to have been destroyed in an accident. Find the contact information for the customer. Also, find the contents of that shipment and create a new shipment of replacement items.
   - Find the customer who has bought the most (by price) in the past year.
   - Find the top 2 products by dollar-amount sold in the past year.
   - Find the top 2 products by unit sales in the past year.
   - Find those products that are out-of-stock at every store in California.
   - Find those packages that were not delivered within the promised time.
   - Generate the bill for each customer for the past month.

5. Everything should be in a single folder and uploaded to git repository.

6. Add a README file in the top-level folder that includes short description of the project and explains what is where, etc.