

Project: Package delivery company

Course: Databases

Enterprise description:

The application is a package delivery company (similar to FedEx, UPS, DHL, the USPS, etc.). The company needs to keep track of packages shipped and their customers. To find out more about this application, think about any experiences you may have had shipping packages and receiving packages, and browse shippers' web sites. In our hypothetical company, the manager assigned to solicit database design proposals is not very computer literate and is unable to provide a very detailed specification.

Here are a few points to consider:

- There are different kinds of service possibly based upon the type of package (flat envelope, small box, larger boxes, etc.), the weight of the package, and the timeliness of delivery (overnight, second day, or longer).
- Some customers have a contract with the shipper and bill their shipments to an account number. They are billed monthly. Other customers are infrequent customers and pay with a credit card. Certain shipments are prepaid, as is might be the case of someone is returning something that was purchased by phone or Internet (e.g. returning clothes that don't fit, or returning malfunctioning electronics).
- For the most part, the shipping company does not care what is being shipped. However, there are cases where it matters. Some examples include
 - hazardous materials
 - international shipments, for which a customs declaration stating the contents and their value is needed
- The company needs to track packages from the time the customer drops it off (or it is picked up by the company) until the time it is delivered and signed for. Take a look at the online tracking offered by various shipping companies to get an idea of how this service works. If you are having something shipped to you, you'll find you can get every little detail of where the package is, where it has been, and to where it is currently headed. Beyond what the customer sees, the company itself needs to know on which truck or plane or warehouse the package is at any point in time.
- Tracking is not just an "in the present" issue. The company may want to look back in time and find out where the package was yesterday, for example. It may also want to look at data from the standpoint of a truck or warehouse.
- There are other aspects to the operation of the company besides package tracking such as the routing of trucks and planes, the assignment of staff to them, etc. For this assignment, we'll consider only the package handling and billing aspects of the database.

Data Generation:

For simplicity, I will not require realistic data. You can just create some names or get real ones from the car company web site.¹ There are many different vehicles, grouped into a variety of (possibly overlapping) categories. If you get realistic here, things get to be interesting. Thanks to "badge engineering" many vehicles are the same except for name.² That means they can be build in the

same plants from the same parts from the same suppliers. 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 wants turned in. They may provide further hints about database design, so think about them at the outset of the project
 - Assume truck 1721 is destroyed in a crash. Find all customers who had a package on that truck at the time of the crash. Find all recipients who had a package on that truck at the time of the crash. Find the last successful delivery by that truck prior to the crash.
 - Find the customer who has shipped the most packages in the past year.
 - Find the customer who has spent the most money on shipping in the past year.
 - Find the street with the most customers.
 - Find those packages that were not delivered within the promised time.
 - Generate the bill for each customer for the past month. Consider creating several types of bills.
 - – A simple bill: customer, address, and amount owed.
 - – A bill listing charges by type of service.
 - – An itemize billing listing each individual shipment and the charges for it.

Customers like their bills to be readable. While the client will accept a relational table coming from Oracle as the bill, it would be “nice” to have a good-looking bill.
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.