

Project: Automobile company

Course: Databases

Enterprise description:

The application is an automobile company, such as General Motors, Ford, Toyota, or Volkswagen (or maybe a company from yesteryear like Studebaker, Hudson, Nash, or Packard). 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. The company needs to keep quite a bit of data, but we shall focus on the following aspects of corporate operations.

- **vehicles** Each vehicle as a vehicle identification number (VIN). Lots of stuff is encoded in real VINs (they are well described on Wikipedia), but you can just make them up if you want.
- **brands:** Each company may have several brands (for example, GM has Chevrolet, Pontiac, Buick, Cadillac, GMC, Saturn, Hummer, Saab, Daewoo, Holden, Vauxhall, and Opel and Volkswagen has Volkswagen, Audi, Lamborghini, Bentley, Bugatti, Skoda, and SEAT)
- **models:** Each brand offers several models (for example, Buick's models are the Enclave, LaCrosse, and Lucerne, and Mercury's models are the Mariner, Milan, Sable, and Grand Marquis). Each model may come in a variety of body styles (4-door, wagon, etc.)
- **options:** We'll stick to color, and maybe engine and transmission.
- **dealers and customers:** dealers buy vehicles from the manufacturer and sell them to customers. We'll keep track of sales by date, brand, model, and color; and also by dealer. This will allow us to use SQL's OLAP tools. Note that a dealer may not sell some of the car company's brands. Dealer's keep some cars in inventory. Some of course, are already sold, but the dealer still keeps track of that fact.
- **suppliers:** suppliers supply certain parts for certain models
- **company-owned manufacturing plants:** Some plants supply certain parts for certain models; others do final assembly of actual cars.
- **customers:** In reality, lots of demographic data are gathered. We'll stick to name, address, phone, gender, and annual income for individual buyers. The customer may also be a company (e.g. Hertz, Avis, or other companies that maintain corporate fleets, but we'll skip that).
- We'll skip data on corporate finance, pending bailouts, bankruptcy status etc. Not that these data are unimportant, but we need to keep the project within bounds.

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 (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

- Show sales trends for various brands over the past 3 years, by year, month, week. Then break these data out by gender of the buyer and then by income range.
- Suppose that it is found that transmissions made by supplier Getrag between two given dates are defective. Find the VIN of each car containing such a transmission and the customer to which it was sold. If your design allows, suppose the defective transmissions all come from only one of Getrag's plants.
- Find the top 2 brands by dollar-amount sold in the past year.
- Find the top 2 brands by unit sales in the past year.
- In what month(s) do convertibles sell best?
- Find those dealers who keep a vehicle in inventory for the longest average time

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.