

INFO 1103 - Individual Project

Due date for the project: 11:59 p.m. December 2.

Goal

The goal of this project is to provide a realistic experience in the conceptual design, logical design, implementation, operation, and maintenance of a relational database and associated applications.

The project can go well beyond the minimal requirements outlined at the end. Such extensions are encouraged. I reserve the right to give extra credit for a project that goes "above and beyond".

The description given here of the enterprise you are modeling is vague and incomplete. This is by design - in real life, your "customers" are managers in the enterprise whose degree of computer literacy is not equal to yours. You will need to fill in the holes in this document to create a precise design and concrete implementation of the interfaces and applications using the database.

Enterprise description

The enterprise is a retailer, such as a music store, department store, discount store, supermarket, convenience store, etc.

Each of you will choose a specific retailer (either use a real one or a made-up one) as your model. To keep the project manageable, focus only on the retail sales activities.

Your retailer sells a large variety of products at a single large store. The store maintains an inventory of products and needs to decide when to reorder and in what quantity. Customers may identify themselves by joining your frequent-shopper program. Others may remain anonymous.

In order to keep it manageable, the database will track inventory at a single store (if you want to do a multi-store model that is fine). It will track customer purchases (by customer, where possible), sales history, inventory, etc. Various user interfaces and applications access the database to record sales, initiate reorders, process new orders that arrive, etc.

You may pick the enterprise that you will model. Any of these is acceptable: Canadian Superstore, Walmart, Canadian Tire, Sears, Costco, Sobeys, IGA, Tim Horton's, Home Depot.

Products: Products may come in a variety of sizes. Each product has its own UPC code (the bar code that is scanned at the checkout). Each product has a certain inventory at a particular point in time.

Vendors: Products are sold to stores by vendors. A vendor may sell many products.

Customers: Customers who join a frequent-shopper program provide some personal information based on what the enterprise requests. They may refuse to provide some information. Customers come into a store (or go online) to buy a market basket of goods.

Processes: Think about what processes you will want to model. Two important ones are inventory control and sales checkout.

Data Generation: For simplicity, perfectly realistic data is not required. However, you should strive for a good degree of realism in your data. Where appropriate, randomly generated data is acceptable (and a good way to avoid having lots of data entry to do). Note the comments on collaboration below and that sharing DATA (but not implementation artefacts such as code, schemas) with others is acceptable as long as appropriate credit is given to your source.

The process you should follow

- **Data Model** - construct an ER diagram representing the conceptual design of the database. Be sure to identify primary keys, relationship cardinalities, etc.
 - **Relational Model** - After creating an initial relational design from your E-R design, refine it based on the principles of relational design.
 - **Create the relations** in a MySQL database. Create constraints as appropriate.
 - **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 data and/or share data with fellow students.
 - **Build and test 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 clients (managers from the retail enterprise) may find of interest. These queries may provide further hints about database design, so think about them at the outset of your work on this project.
1. What are the 20 top-selling products at each store?
 2. What are the 20 top-selling products in each department?
 3. How many customers have a loyalty card?
 4. How often do customers visit the store?

5. What is the average purchase by a customer who does not live in the city where the store is?

Interfaces: There are several types of users who access the database and several applications that run on their own. The database administrator (you) may use SQL via MySQL's command line interface. Online customers need an elegant Web interface or in store application to order products. However, for this project, a command-line interface will suffice if you don't want to develop Web and/or GUI applications. (This is a database course - not a web apps or GUI course.)

Your system may generate reorders automatically using triggers. Or, you may have an application that runs periodically to scan the database looking for items to reorder. It would be nice if your database tracked reorders and their delivery (i.e. records the increase in inventories resulting from the arrival of a shipment.)

Each checkout register (including any virtual ones) runs an application that records the items in each basket, updates inventory, and gathers frequent-shopper data. These interfaces can be built as:

- Web applications using Java applets or a scripting language,
- A standalone Java application using Swing to create a GUI,
- Other GUI development tools you may know (but be sure they are platform independent, see note below), or
- A simple command-line interface (since this course is neither a Java course nor a GUI course). In fact, even if you are expert in GUI development, you may want to start with a simple command-line and then upgrade later. I highly recommend incremental and iterative development.

Logistics

Please discuss the various stages of your project with me. I am here to help you.

Deliverables

The final version of the project is to be turned in as a single zip file and emailed to me.

1. A short description of your database and what it does.
2. ER diagram - I will accept paper for the ER diagram since we are not covering drawing tools for these diagrams. The diagram needs to be clearly legible. At a minimum you must include all the entity and relationship sets.
3. Relational schema – you can print these out if you want.

4. A set of sample queries with their results demonstrating that the database works.
5. The code to implement the various interfaces. (By restricting your development to Java, your code should be platform independent. Fairly basic interfaces are acceptable, e.g. command line with a modest command set.) It is advisable to avoid platform-specific solutions (but if you have experience with another platform that you want to use, please come talk to me). Please submit your java code as .java files that I can compile and run. Please do not submit them embedded in a NetBeans project or any other IDE.
6. A README file in the top-level folder that explains what is where, etc. Include usage instructions for the interfaces.

You may create and run the project on your own computer if you wish. In fact this may be significantly easier to do.

Presentation

On December 3, each of you will individually present your project to me (not the class). We will review its design, functionality etc. I will assign a grade after I review your submitted materials. I will note grade you on your presentation (again, because this is not a presentation course).

Grading:

I will use the following approximate template for grading:

- Description and ER design: 20%
- Relational design, including constraints and indices: 20%
- Data creation: sufficient quantity, realism, sufficiently "interesting": 10%
- User interfaces, including proper features, proper updating of the database, functionality, etc.: 50%

I reserve the right to give extra credit for exceptional solutions to (parts of) the project.

Your project design and interface implementation is to be your own work. You may share data to load into your database. You may also share code that generates those data. Please credit your source in each such case.