COMP 310 **Database Theory and Design**

Project 1 & 2

Assigned: March 05 *Due: April 07/May 07*

Note 1: This project is heavily based off of a project given for a course at Lehigh University.

Note 2: This is really a single large project, but has been split into two separate pieces, project 1 and project 2. These pieces will be turned in and graded separately.

Abstract

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. A real project of this sort would require a substantial development team working for several months (or more). You will do this over about two months, at first alone and then in small groups. I have chosen individual projects at first because the goal of this project is for you to gain a personal appreciation of the depth and breadth of issues that go into the design of a database application, rather than to have you specialize in just one aspect (and rely on others for the rest).

The project can go well beyond the minimal requirements I outline at the end. I encourage such extensions. They could turn into a senior capstone project or other independent work.

The description given here of the enterprise you are modeling is necessarily somewhat vague and incomplete. This is by design – in real life, your "customers" are managers in the enterprise whose degree of computer literacy is, to put it kindly, variable. You will need to fill in the holes in this document in create a precise design and concrete implementation of the interfaces and applications using the database.

Important Dates

Date	Description
March 05 (Fri.)	Project released
March 10 (Wed.)	Enterprise chosen
March 19 (Fri.)	E-R diagram checkpoint meeting completed
March 26 (Wed.)	Relational model checkpoint meeting
April 07 (Wed.) midnight	Project 1 Due
April 09/12 (Fri./Mon.)	Project 1 Presentations
April 09 (Fri.)	Project 1 submissions released to all
April 16 (Fri.)	Applications checkpoint 1 meeting completed
April 23 (Fri.)	Applications checkpoint 2 meeting completed
May 05 (Wed.)	Project Presentations
May 07 (Fri.) midnight	Project 2 Due

Enterprise description

The "enterprise" is a retailer, such as a department store, discount store, supermarket, convenience store, etc. Each of you will choose a specific retailer (either use a real one as your model or a made-up one). To keep the project within bounds, we'll ignore issues of employees, corporate finance, etc., and focus on the retail sales activities. Your retailer sells a large variety of products at multiple stores. Not all products are at all stores. Pricing may be different at different stores. Each store has its own 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. Your retailer has a web site that accepts orders. From a database perspective it is just a special store that has no physical location and has no anonymous customers. The database tracks inventory at each store, customer purchases (by customer, where possible), sales history by store, 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. I'd like to see a wide variety chosen, so here is a list to serve as starting point to give you ideas, but other choices are welcome, indeed encouraged: Walmart, Target, J.C. Penny, Sears, Costco,BJ's, Best Buy, American Eagle, Nordstrom, Safeway, Aldi, Albertsons, Acme, HEB,Food Lion, Piggly Wiggly, Wegmans, Walgreens, Rite Aid, CVS, Longs, Superfresh, Carrefour, Tengelmann, Hankyu, Dillards, Wawa, Sheetz, Modells, Petsmart. I've included some non-US enterprises on this list to encourage you to consider them, or other non-US retailers.

- **products**: Products come in a variety of sizes or means of packaging. Each product has its own UPC code (the bar code that is scanned at the checkout).
- **brands**: A variety of products may be sold under the same brand (e.g. Pepsi and diet Pepsi). For such applications as reorder, specific products and sizes matter. For other applications, data may be aggregated by brand.
- **product types**: A particular type of product may be sold in a variety of sizes anda variety of brands. For example, cola is sold under such brands as Pepsi and Coke. Product types form an specialization/-generailzation hierarchy. For example cola is a type of soda, which is a type of beverage, which is a type of food. Some products fit into multiple categories. For example, baking soda is a cleaner, a food (since it is used for baking), and a drug (since it may be used an an antacid), but it is not a type of soda.
- **vendors**: Products are sold to stores by vendors. A vendor may sell many brands (e.g. Pepsico sells Pepsi, Tropicana, Aquafina, Gatorade, Lay's, Doritos, Quaker, and others).
- **stores**: Stores sell certain products, each of which has a certain inventory amount at any point in time. Stores have locations (addresses), hours at which they are open, etc.
- **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 ofgoods. Not only must this data be stored, but also the system must be able to handle multiple customers buying goods at the same time.

Project 1

Throughout the project, think very carefully about how the enterprise (and the database system) would actually operate in real life. Remember that the manager(s) who defined the specifications is not computer literate, so the specifications should not be viewed as necessarily being precise and complete.

Client Requirements and Deliverables

1. **E-R Model**: Create an E-R diagram representing the conceptual design of the database.
   - Be sure to identify primary keys, relationship cardinalities, etc.
   - This can be hand-written and then scanned as a PDF or drawn natively in a digital format. Microsoft Powerpoint works surprisely well for drawing diagrams, though some of you may have familiarity with another tool, which is fine.
2. **Relational Model**: Create a `.ddl` file that specifies the schema for your database.
   - After creating an initial realtional design from your E-R design, refine it based on the principles of relational design (chapter 7).

- It is likely (hopefully) that many of you produce a very detailed and extensive E-R design. In this case (at the E-R diagram checkpoint meeting, or later if this checkpoint finds problems with your design) we will meet and agree on using only a portion of the resulting relational design that will be implemented in the physical database. This will allow us to cut implementation and data-entry effort into something realistic for the course time frame.
- Create the relations in a relational DBMS of your choice. The best thing to do is to install one on your local machine. If you really want, we can set you up something on the department server. As long as I can easily install a free version of whatever you're using, it's ok. Good choices include MySQL, MariaDB, PostgreSQL, and Microsoft SQL Server, but there are other reasonable alternatives.
- Create indices and constraints as appropriate.
- If, as you refine your design, you discover flaws in the E-R diagram, go back and change it (even if I previously met with you and said your E-R diagram was fine). Your final E-R design must be consistent with your relational design.

3. **Populate Relations**: Create one or more `.sql` files that can be used to load the data into an empty schema. The files will mostly contain `insert into` statements.
   - Include enough sample data to write some test queries (below) that generate interesting and nontrivial results.
   - You may find it helpful to write a program to generate test data (in the form of `.sql` files. You can look up the details for random number generation or random choice from a list for Python or your favorite language, and use that for generation. I might, for example, create a list of beer brands and a list of beer types, and generate 100 products by first randomly choosing a brand from the list and then randomly choosing a type from the other list.
4. **Example 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 thosed that your clients (managers from the retail enterprise) may find of interest. They may provide further hints about database design, so think about them at the outset of your work on this proejct.
   - What are the 20 top-selling products at each store?
   - What are the 20 top-selling products in each state?
   - What are the 5 stores with the most sales so far this year?
   - In how many stores does Coke outsell Pepsi? (Or, a similar query for enterprises that don't sell soda)
   - What are the top 3 types of product that customers buy in addition to milk? (Or, a similar question for non-food enterprises.)

Submission

In addition to your E-R diagram, `.sql` and `.ddl` files, provide a report that describes:

- A thorough description of your enterprise.
- Some explanation of the design decisions you made, any simplifications/assumptions, and their consequences, both for your E-R diagram and your relational model.
- A description of your strategy for generating sample data.
- A description of your test queries, the corresponding SQL queries, and the expected output.
- Anything else you want to tell me about the project.

Once complete, put all your files in a zip file and email it to me.

Presentation

You will give a 15 minute presentation that should summarize your enterprise, relational model, and at least 1 interesting query.

Rough Rubric

Description	Points
E-R Design	15
Relational Design, including constraints and indices	20
Data creation: sufficient quantity, reasonable realism, sufficiently "interesting"	10
Checkpoint meetings	5
Presentation	10
Total	60

I reserve the right to give extra points for exceptional solutions to parts of the project.

Project 2

For project 2, you may (optionally) work in groups of 2 or 3 (no larger). You may choose to start with anyone's project 1 results or from a combination.

Client Requirements and Deliverables

1. Modifications to your project 1 starting point. You may find some things from project 1 need clarification or fixes. You may find that someone else's project 1 is better overall, but needs a change or two to do what you want. You should make a note of these changes, submitting a new version of whatever needs to change, along with a description of what changed and why.

2. **Interfaces/Application**: There are many types of users of your system. Each of them needs a different interface to access parts of the database. In other words, you need a variety of **applications**. In the below list I describe these applications as they might exist in the real world. However, since this is not a web development or UI design class, you can "simulate" those interfaces with a command line interface that still achieves the same goals. At a bare minimum, you must implement two of the following interfaces/applications:

    - An app for business analysts to use. These analysts run lots of OLAP queries concerning the performance of various products, stores, etc. This application should be specialized to your enterprise.
    - A web app for online customers to use.
    - An app for handling recordes automatically. This could entirely or partially be imlemented using triggers, or you may write a separate application that scans the database periodically to find items to reorder.
    - Vendors periodically need to access the database for reorder requests, which they fill by entering into the database a shipment, a delivery date, and the reorder purchase order or orders that are satisfied by the shipment.
    - An application that handles the arrival of a shipment, e.g., by increasing the inventories. (For simplicity, you may assume that shipments arrive at the time specified by the vendor for the shipment. This it the kind of simplification/assumption you are allowed to make for this project. Such assumptions make your life easier and the project feasible, though you must make sure to note and explain them.)
    - An app to run at each checkout register. This "POS" (point of sale) application records the items in each basket, updates inventory, and possibly gathers frequent-shopper data.

It's perfectly fine to think of and implement other applications, as well.

If you only implement command-line interfaces, you should plan to implement at least 3 of the applications for a reasonable grade, unless your command-line interfaces/applications are spectacular. Alternatively, you could start with 2 command-line interfaces, getting them into reasonable shape, and then develop one into a full-blown GUI or web application. You could, for example, use Java Swing or JavaFX to build an GUI application. Or you could use Flask or Django to build a Python-based web application. (I have experience with Flask and could help you with that if you go that route.)

3. **Concurrency**: The company stock will go down rapidly if the database cannot support more than one customer at a time making a purchase or if it cannot deal with more than one item being reordered. Be sure the DBMS transaction mechanism is providing the needed guarantees. By running the various queries and applications in separate sessions, you can simulate the real-life operation of your enterprise.

Submission

In addition to the necessary code, include a `README` file that describes how to set up and run your code.

Include a report that describes:

- whose project 1 you started from
- any changes you made to the project 1 components
- a description of your interfaces/applications
- the important design decisions you made for your applications, especially any related to accessing the database
- an explanation of how concurrency affects the system and its applications

Once completed, again zip up all your files and email them to me.

Presentation

You will again give a 15 minute presentation. You will need to briefly summarize your enterprise and project 1 starting point, then spend the bulk of the time discussing your applications and how they interact with the database.

Rough Rubric

Description	Points
User "interfaces", including proper features, proper updating of the database, etc.	35
Concurrent operation of interaces	10
Checkpoint meetings	5
Presentation	10
Total	60

I reserve the right to give extra points for exceptional solutions to parts of the project, especially for GUI or web application interfaces.