

COP4710 Project.

Points:

- Individual Projects – You should work on your own as you do with your homework.
- User Interface – Don't sweat this, it can just be a handful of queries with instructions.
- Required Features – Tables, instructions, I/O, reports.
- Project Deliverables – What you need to turn in for each step.
- Final Deliverables – Details for your final submission for a grade.
- Project Ideas – Sometimes this is the hardest part.

Individual Projects

Even though these are long instructions, this is a short project. Each student will create an individual database project. You can use any database product to implement your project. The steps to implement the tables and user interface are practical exercises (that means you have to actually make it work) and require you to use a database product (a DBMS).

User Interface

Building a user interface for an application is especially treacherous because they seem so deceptively easy but are very tricky to do. I would expect your interface to be the *very simplest of user interfaces*. Our textbook does not describe how to build a database *application*, and it's not of importance in a broad introductory class so we didn't cover them. Therefore, *I do not expect a polished user interface*.

Web interface

However, I know that some of you are adept in web applications. Chapter 7 of our textbook describes web services, but not how to build an application. Seeing one of the many ways to implement a web interface would be interesting. If you *already know how to do this*, and want to go this route, feel free to do so.

MS Access interface

A much easier way to build a project would be to use MS Access or some other application builder. Many small database applications use this platform. If you go this way, you must show me the queries that the user interface uses to communicate with the database. You might even write some VBA to support your project. *Do not use the built-in documentation feature of MS Access*. Unless Microsoft has updated this feature significantly, it's not useful in the least and just wastes paper.

“Manual Query Method” interface ← Nudge, nudge! Use this if you’re not an *expert* with user interfaces!

I would go this way unless I was already VERY adept at writing applications. The simplest way nearly always presents the fewest opportunities to lose points.

If the only system you have to work with is the MySQL database server on the CS server, and (like me) you don’t have the ability to put a fancy user interface on that, then don’t worry! Your user interface would be a printed (and organized) set of instructions on how to manually edit and run a set of queries that you’ve prepared for that purpose.

There are plenty of systems in the world that have a user interface that involves manually editing and running a handful of SQL queries. I’ll refer to this as the “*manual query method*” of building a user interface. Not only is the “manual query method” acceptable for your project, it’s the easiest, quickest, and *best* way to go. You get to show off your database skills without extraneous user interface noise getting in the way.

The Manual Query Method captures all of the database aspects of a database application. With any user interface, a SQL query supplies the lines to print a report. Any input screen builds a SQL query to get the data to and from the database. The “manual query method” accomplishes all the same database processing steps as would an application with a “proper” user interface.

What you would miss is the programming experience of putting together a SQL query in a string, sending the query to the database to execute, and having a result set returned. This is not the least bit important in this class.

Required Features

Your application must include these features:

1. Multiple tables that use some linking. Don’t make it so simple that there are no foreign keys.
2. Simple instructions on how to use your system.
3. A way to input data (insert, update, and delete queries).
4. A way to produce a report. Reports are frequently (but not always) intended to be printed, and are frequently (but not always) columnar. Include a query with a *GROUP BY* clause. *GROUP BY* and *HAVING* are very handy for making reports and can make the Manual Query Method very effective.

Project Submission Steps:

Students always want to let due dates slide. (I always dreaded a due date for a project!) I would like to see you submit *something*, even if it's a rough preliminary, by the due date. It won't hurt your grade so submit something rough that you can revise later. This will keep the ball rolling.

The idea behind these steps is that you won't wait until the last night to build your project. I want to make it easier for you by having you work on a schedule. Each submission adds pieces to your finished result.

You might feel funny that you're submitting the same material more than once. This is because the submissions don't ask you to submit different things each time, but they ask you to build up the pieces of your project and get them organized into your final result. You can mold your project into an "A" by acting on the feedback to your submissions.

Submission 1

1. Title of your project. This could be a sentence. (System to track automobile gas mileage, baseball card collection, etc.).
2. Outline of what your project does. Tell about input and output, short description of the user interface and data.
3. Description of what the user interface might look like. It's OK to completely change your mind later, but I want you to get started thinking about this. For the Manual Query Method, list the functions it will provide to the user.
4. ER diagram. This is the structure of your database.

Submission 2

1. Set of table schemas (table names and attributes with keys underlined. You can leave out the data types).
2. Explanation demonstrating that the tables are normalized.
3. DDL to create the tables.
4. Run the DDL to actually create the tables in a DBMS.
5. Add some sample data to your tables.
6. Firm description of your user interface.
7. Start implementing your user interface (work on your queries).
8. Submit a "How's it going" statement. Tell how you are doing with the project. You could submit some queries, screen shots, a verbal description, whether the data model you chose seems satisfactory, etc. Let me know how things are going, good or bad.

Submission 3

Turn in your Final Deliverables.

Final Deliverables

1. Title of your project.
2. Outline of what your project does.
3. ER Diagram.
4. Table schemas.
5. An explanation demonstrating that the tables are normalized.
6. DDL to create the tables.
7. List the queries for interacting with the database.
8. Instructions for using the system. For the Manual Query Method, this will make up your user interface.
9. Screen shots (printouts) of the user interface.
10. Include samples of reports.

These can be combined depending on how you organize your documentation.

If you are pleased with the way your project is turning out, you can share it with the class. If it's a web project, tell us how to get to it. If it is an MS Access project, you can post the mdb. If you have some SQL queries you are proud of, or a "manual query method" user interface that works well, you can post the queries and instructions. Even though an executable is not part of the required deliverables, I would like to see your work run. (If not, it won't affect your grade).

Project Ideas

Ideas for a project might be:

- An application to keep track of gasoline purchases and vehicle gas mileage.
- Record and report on home utility usage and expense.
- Savings account system for a bank.
- Statistics for baseball players.
- Library book check-out system.
- Hotel room reservations.
- Travel agent database.
- Software version tracking system.
- Etc... I'm sure that some of you already have something in mind.