Project Description

Deliverable(s)

Word document with summary, diagrams, glossaries, SQL code and screenshots. The format should include a cover page with your name, the course section, and a title. Readability counts! Use clearly identifiable headings to break up the elements of your report. Make sure screenshots fit nicely on the page and the content can be read when viewed on a computer monitor.

Set up

You will design and implement a database to solve a data management problem of your choice. Applicable topics can include but are not limited to personal data tracking, commercial enterprise problems, web application backends, academic data management, or inventory tracking. Your instructor will work with you to refine your topic to ensure it covers all of the requirements for the course and is achievable in a reasonable amount of time. The intent is to provide you with an opportunity to work on something of interest to you that should have some real utility as a completed database. This project is broken up into 2 parts. This document describes those parts and there constituent elements.

The first part is the design specifications detailing the data to be tracked and how all of the elements work together. You'll also need to specify any business rules that dictate how the data are to be managed. As in any good development project, you'll also want to identify your stakeholders and detail what data they will need to access and maintain. Document at least five data questions your data must answer to be relevant. There will likely be more, but pick five important questions and thoroughly develop those.

The second part is the implementation of the design created in part 1. This will include the SQL statements to create the tables and columns to hold the data and any constraints that implement the business rules. Also included are representative statements for the basic Data Manipulation Language (DML) that implement the create, read, update, and delete statements (referred to as CRUD) used in maintaining your data. The final deliverable will be a complete document including the revised version of part 1 and all of the artifacts generated in part 2.

Part 1 – Due Week 4

Summary

Begin with a summary of your project. Carefully describe your project and its importance. It is here that you will introduce the reader to your project and set up the business case, high level rules about the data, its stakeholders, and expectations of the final outcome. Edit this summary whenever necessary throughout the project to reflect any changes in scope or content. You should have a brief working version of this summary by Week 2 and this text will initially serve as your project proposal, gradually gaining more detail as you progress through the process.

Conceptual Model

Fully dress your business rules and identify all of your stakeholders. Gather and provide unrefined representative examples of your data and provide an Entity-Relationship Diagram (ERD) with a glossary describing in readable text the entities, attributes, and relaitionships in your model.

Normalized Logical Model

Decompose your conceptual model into relations and normalize your relations. Provide a database diagram of your normalized relations. Be specific about how the logical model flows from the conceptual model and explain any additional attributes that arise (surrogate keys, etc). It is in this step that you will make choices about data types. Be clear as to what influenced these decisions. It may not be necessary to elaborate on each and every instance of the choice if you implement database-wide rules. For instance, if you choose to implement all primary keys as surrogate keys using the SQL Server identity property, or any date or time attribute will be implemented using the datetime data type, you do not need to provide that information at the table level. Simply state any such decision at the top of this section.

Part 2 – Due End of Course

Physical Database Design

Provide your Data Definition Language (DDL) code to build the database tables and any objects such as stored procedures, views, and functions. Make sure your SQL code has adequate comments throughout that describe what the code does and provide commentary on the object being implemented (what are the paramters, assumptions, and controls on a stored procedure, for example). Also include the necessary code to remove these objects. You should be able to repeatedly execute the code in this section without error.

Data Creation

Provide representative examples of how your data will be created. This should include approximately five to ten INSERT statements to show the commonly acceptable means of entering data using SQL. If a table has defaults, be sure that you have INSERT statements that show how to INSERT when a column value is provided as well as when one isn't and the server is responsible for setting a column to a default. Provide comments as necessary to show the intent of the statement. Once SQL stored procedures are learned, consider coding procedures that abstract these statements and provide control and safety when adding data.

Data Manipulation

Provide representative examples of how data are manipulated through updates and deletions. Whenever possible, be sure to provide comments that tie the statements to the business rules (for instance, if your business rule states you must mark a customer as "on credit hold" when their unpaid balance exceeds a credit limit, provide the UPDATE statement that implements this rule and comment on it as such. Once SQL stored procedures and functions are learned, consider coding procedures and functions that abstract these statements and provide control and safety when adding data. For instance, if you implement the rule about customer credit holds, consider coding a function that makes that decision and returns the result of it's analysis.

Answering Data Questions

Code and demonstrate through screenshots of results the SELECT statements that answer the data questions posed in the summary. Be sure to comment on these queries. Include the question being answered, any assumptions made by the data, and rationale for any decisions made. As with other

sections, be sure to correlate your code to business rules outlined in the conceptual model. Once views are learned, consider encapsulating these statements into views for greater portability and ease of use.

Implementation

Using the tool of your choice, build a basic front end that provides a user interface for maintaining and reporting on your data. You can use Access, web technologies such as PHP, or full applications as your skillset and availability allow. As this part will likely be completed near the end of the project, Microsoft Access is a good choice in this course as it provides many mechanisms for rapidly prototyping the interface. In addition to data entry and maintenance screens, provide report output (Access Reports, for example) for the queries that answer your data questions. Include in your final report a screenshot of each screen/form and report with data present in the fields.

Reflection

To conclude, provide a brief reflection on the process. Some questions to ponder:

- What assumptions did you have at the start of your project that changed by the end? Think in terms of both your own problem domain as well as your knowledge of the process.
- The next time you do this, what will be different?
- Regardless of whether you go through these steps again, how do you think it will inform your approach to data as an information professional?

Those are just priming questions to reflect on. If you have other thoughts and observations you would like to share, be sure to include them.

Summary (again)

Combine the work done in parts 1 and 2 into one document for the second deliverable. Take this opportunity to revise anything from part 1 that changed during your work on part2. Add details to your summary to include your implementation. Discuss how you answered the data questions, how you came to choose the user interface tool, and other such things that inform the rest of the report.

Tips!

Start early and keep pace throughout the course. It is often best to complete each section as soon as the concepts are covered. The labs are designed to provide you with templates for building the deliverables in this document, so try to do this project in tandem with those sections.

Don't work in a vacuum. Be sure to communicate any road blocks or questions to your instructor as soon as possible. Also, consider collaborating with your classmates on topics you may find difficult. You may be able to provide them with assistance on other elements.

Choose a problem domain that is interesting to you and can have lasting utility. This will work to improve your motivation to complete it.