Information School
UNIVERSITY of WASHINGTON

INFO 330: DATABASE SYSTEMS AND DATA MODELING

Group Project

Phases 2-4: Logical Model, Database Implementation, and Query Design

Due Date: Thursday, December 7 (by end of day)

Purpose

The purpose of this project is to give you hands-on experience in database development. The process of database development goes through phases from conceptual design to implementation and maintenance of the database. The design phase can particularly be very challenging since it is less structured.

Specifically, the project will reinforce the following learning objectives:

- Evaluate information needs of the target organization, and effectively model the data using entity-relationship model and the relational model.
- Design and implement a physical model based on principles of relational database design, and organization's needs regarding data, data organization, and data storage.
- Develop working SQL statements for simple and intermediate queries to create and modify data and database objects to store, manipulate, and analyze company data.

Note

Phase 1 of the project was given to you in a separate document. This document provides guidelines for the subsequent phases of the project.

Phase 2: Logical Data Modeling

Your group should develop a logical data model, considering the following:

1. Each entity should contain a full set of attributes keeping the minimal data rule in mind (i.e., what is there is needed, and what is needed is there).
2. All the attributes should be atomic (i.e., all composite and multi-valued attributes should be resolved to be atomic).
3. Furthermore, all the many-to-many relationships should be resolved.
4. Appropriate primary keys (PKs) should be identified for each entity, and foreign keys (FKs) should be identified for entities that require them.
5. With the above requirements in place, draw an entity-relationship (ER) diagram for your logical model.

6. Identify the appropriate data type for each attribute and write the relational schema (**note**: please include the data type for each attribute in your relational schema).

Phase 3: Database Implementation

The first step of database implementation is physical design whereby you convert each of the relations in your relational schema into database tables using data definition language (DDL) of SQL.

1. Begin by giving your database an appropriate descriptive name based on your topic (e.g., project_db, customer_db, etc.).
2. Using SQL DDL, complete the following tasks:
3. Write a query to create your database.
4. Write queries to create the table structure for each table in your database. Make sure that you add the appropriate constraints in your table structure, such as primary key and foreign key constraints, not NULL, default values (if any), and check constraints (if any)
5. Create an Excel file for each table of your database and populate each table with relevant (appropriate) data. The main tables (those on the "1" side of the 1:M relationships) should have at least 15 records (rows) of data and other tables (those on the "M" side of the 1:M relationships) should have at least 45 records (rows) of data. To have true 1:M relationships, make sure that some records on the "1" side are associated with more than one record on the "M" side (e.g., a customer with at least 2 orders in the orders table would have their ID appear at least twice in the Order table).
6. Please save each of your Excel files as a CSV file.
   - Go to http://www.convertcsv.com/csv-to-sql.htm and import each of your CSV files to create DDL and DML scripts (A class example will be provided next week). Given that you need to insert many values into your database tables, writing queries manually is not efficient. This online tool makes the process efficient.
   - **Note**: the online CSV-to-SQL converter tool does not generate foreign key constraints. Please make sure that you add foreign key constraints to your DML scripts wherever necessary. However, I suggest that you add foreign key constraints after you import data into SQL server to avoid FK headaches. Obviously, for this to work out well, your data should have high integrity.
   - Combine the individual table scripts into ONE SQL dump file. Please save this sql dump file with a name that is reflective of your database project (e.g., if the database is for automata company, saving a file as "***automata_sql_dump.sql***" would be great).
7. Open your SQL dump file in either SQL Server Management Studio or Azure Data Studio and run (execute) it to create the database.

Phase 4: Retrieving Data from the Database

1. Think of relevant information your database could provide to support managerial decision making. (Class examples should serve as a source of useful insights as you develop relevant queries for your project context). Your queries should be complex enough to justify the need for the database (i.e., they should pull data from more than one table in the database).
2. **Create at least 5 queries per group member** from your database (those who create at least one additional query per group member will get bonus points! Note: ***Do not create queries that do not add much value just for the sake of getting bonus points. Both quality and quantity will be taken into account when grading your work***). **Also, if you just copy and paste queries from class examples or other sources, you are not going to earn high marks for the project, and depending on the severity of the problem, you could fail the project.**
3. **Note**: appropriate complex queries would include most of the following:
    - Integrate data from more than one database table (JOIN operations).
    - Use derived (computed) attributes.
    - Filtering of the data using WHERE and HAVING conditions.
    - Use aggregate functions (SUM, AVG, COUNT, MAX, MIN).
    - Use subqueries.
    - Use common table expressions (CTEs).

Besides the above common components, groups that create at least one query that leverage database objects (i.e., stored procedure, user-defined function, or trigger), or a transaction will get bonus points.

4. Your queries should be named appropriately (i.e., the name should reflect the type of information generated by the query).
5. Please save each of your queries as a SQL script.
6. Please incorporate screenshots of the results (outputs) of your queries into your project report. **For each query**, give a brief description of its purpose, summarize the results, and provide managerial implications/recommendations based on the results of your query. This narrative should be included in the report.

What to submit (Upload to Canvas)

1. An ER diagram (uploaded as a PDF file)
2. A PDF document containing phase 1, relational schema (no. 6 of phase 2), and screenshots of query results and discussion of the results (No. 6 of phase 4) – you can create this in a Word processor and save it as a PDF file type.
3. SQL dump script used to create the database
4. SQL scripts for the queries from phase 4 (please combine your queries into one script file).