CSC 675 Introduction to Database Systems
Fall 2012, Prof. Murphy

CSC 675 Term Project Description: Backend Database for TaggedPhoto App

For this semester's term project option, you will be designing and implementing a backend database to support a simple TaggedPhoto Application. This is an example of relational database design for Internet/Mobile applications. Your test queries will be implemented in interactive SQL as well as JDBC and a scripting language, and will use the CSC 675 ER Design Tool to (partially) automate the design process.

Requirements for the TaggedPhoto App Backend Database

The TaggedPhoto App will be used to create digital 'TaggedPhoto' that can be viewed on any device with Internet access. TaggedPhotos are displayed as rectangular windows with a banner across the top stating 'From the Device of <user> on <date>', a photograph, the associated tags & tag values, and a row of buttons across the top to email the TaggedPhoto to another user, archive the TaggedPhoto to a remote server, change the tags and/or tag values, and delete the TaggedPhoto. Examples of tags might include the date, time, subject, GPS coordinates, etc. Note that the exact tags may be specified at run time (and all tag values are stored as simple Ascii text). ***Don't worry about the exact details of how the user interface is implemented, you will just be designing the back-end database to support this application!*** The backend database should include enough descriptive data for TaggedPhotos so that a photo and all tags can be accurately displayed on any device. When a TaggedPhoto is mailed to another user, a copy is added to that user's collection of TaggedPhotos; when a TaggedPhoto is deleted, only that user's copy is deleted.

You will first need to design an ER Diagram to describe all of the information stored in the backend database. Note that it must be possible to uniquely identify each object in the backend database; and it must be possible for the same individual to store many objects (of the same or different types) without any restrictions. The easiest way to support this is for your system to generate ID's (use select max (ID) and add one to generate a new unique ID when you insert the data). ***Discuss how you might implement some sort of digital rights management system for stored media content. Why is this necessary?***

Your system should support the following ten queries (some of which may require multiple SQL statements):

1. Create (enter) a new TaggedPhoto by entering all of the required metadata to completely describe the TaggedPhoto contents.
2. Delete a TaggedPhoto given its TaggedPhoto-ID.
3. Move a TaggedPhoto to the remote archive, given its TaggedPhoto-ID.
4. Change the value of a tag on a TaggedPhoto given its TaggedPhoto-ID.
5. Add a new tag & tag value to a TaggedPhoto given its TaggedPhoto-ID.
6. Given a TaggedPhoto-ID, retrieve all of the information needed to display that TaggedPhoto on a display.
7. Retrieve the TaggedPhoto-ID's of all TaggedPhotos ***owned*** by a specific user.
8. Retrieve the TaggedPhoto-ID's of all TaggedPhotos ***emailed*** by a specific user.
9. Retrieve the TaggedPhoto-ID's of all TaggedPhotos created by a specific user after a specific date.

10. Retrieve the TaggedPhoto-ID's of all TaggedPhotos with an associated tag & associated values (for example, GPS coordinates).

Implementation Notes

You will implement this project using the DB design methodology introduced in the homework assignments, and work with the CS 675 ER Database Design Tool to partially automate the design process. First, design an ER schema that captures the semantics of the application described above. Your schema should be described using an ER diagram, as well as a textual description (be sure to use complete sentences in describing the objects in your schema!) Your ER schema should include all necessary key and structural constraints, and the textual description should explain in English what each of these constraints signifies. You should also explain clearly which constraints are given explicitly in the requirements, and which are assumptions you are making about the application. Media objects can be represented either as character strings containing the name of the file containing the actual media object, or can be stored in the database using BLOB type attributes as an extra credit option. Enter your ER schema into the CS 675 ER Design Tool and generate the pseudo-English ER constraints, then re-write these constraints in clear and correct English that explains what the constraints mean **without using the term INSTANCE (i.e. explain the constraints purely in terms of the application semantics without using any ER terminology).**

Second, translate your ER schema into an equivalent relational schema that is as efficient as possible (i.e. which has the minimum number of tables, irrespective of NULL foreign keys) and into another equivalent relational schema that requires no NULL foreign key attributes (i.e. which is as portable as possible). Note: the 675 ER Design Tool does these translations automatically but it has many residual bugs; your writeup should explain your schema is correct (using the same detailed step by step explanation that you used in your homework solutions)!

Third, implement queries three through ten above using relational algebra, and all ten queries using SQL, and then test your SQL queries using interactive SQL Server 2008/2012. You should try to use the minimum number of SQL statements for each query, and to use temporary tables only if necessary. After you have debugged the interactive queries, write a JDBC program and a simple script to execute your queries (this program should prompt the user for any necessary information, then execute one of the ten queries on request).

Test your program using at least three examples of representative data. Be sure to design your test data so that you have illustrative output for each of the queries given above. You can use different test data for each of the different queries. Your documentation should include an explanation for each SQL query, a script and example output for the (interactive) SQL query and complete annotation, along with an example of the JDBC program output. Your test data can include the GuineaPig photos posted on the class web site (along with your own photos and descriptive metadata) for extra credit.

Extra Credit: Modify your design so that a ***single*** media object of any type (photo, audio clip, video clip, link to .pdf document) can be placed on the TaggedPhoto instead of just a photo.

Additional Extra Credit: Modify your design so that you can store a single media object of any type (photo, audio clip, video clip, link to .pdf document) as the value of a tag on an existing TaggedPhoto.

More Extra Credit: Implement your application as a Web App to allow the demonstration queries above to be executed remotely from a Web Browser. You can also implement your application using the OnBoard SQLite database to allow the demonstration queries above to be executed from within iOS/Android App code.

Other Extra Credit options are possible – email me your ideas for feedback & approval!

CSC 675 Introduction to Database Systems
Fall 2012, Prof. Murphy

Term Project Grading Criteria

_____(15) Introduction and Overview (Executive Summary)
_____(10) ER Diagram
_____(15) ER Schema—English text description (use complete sentences!)
_____(15) Description of ER key and structural constraints (in English, **without using the word INSTANCE)**

_____(20) Relational Schema with all Constraints shown and explanation of how these schema and constraints are derived using the mapping rules if NULL foreign key values are allowed.
_____(10) Relational Schema with all Constraints shown and explanation of how these schema and constraints are derived using the mapping rules if NULL foreign key values are **not** allowed.

_____(20) SQL Queries and Explanation
_____(15) Relational Algebra, Query Trees and Explanation
_____(10) Scripts (or Java programs) to create tables & populate with data
_____(25) Interactive SQL Server output for SQL queries (with complete annotation)
_____(10) Script to execute SQL queries (with comments & complete annotation)
_____(15) JDBC program to execute SQL queries (with comments & complete annotation)

_____(10) Summary, conclusions, recommendations and future work
_____(10) Overall appearance of report

Total: 200 points

Term projects are due on the last day of instruction, Friday December 14 before midnight. **NO LATE PROJECTS WILL BE ACCEPTED FOR CREDIT!! Late projects are ones handed in *after midnight* on Monday December 14.** Project grading sheets and scores will be emailed & paper summaries distributed at/after the final exam.