Homework 6 – Class Project Design: Normalization Phase

1. Using your corrected ERD, make sure you have no multi-valued attributes – if you do create a separate entity (or additional attributes) to handle this design flaw.
2. For any M:N relationships in your conceptual design, go through the process of creating the necessary linking tables.
3. If you have any relationships for which you want to add attributes, create the necessary entities.
4. Document the business rules (mandatory, optional, weak, strong, relationships, weak entities etc.) and table or process constraints you want to impose.
5. Verify that your model is in 1NF.
6. Test whether you have partial and transitive elements in each entities, show a diagram indicating such dependencies.
7. Resolve the partial dependencies and verify that the model is in 2NF.
8. Resolve the transitive dependencies and verify that the model is in 3NF.
9. Check whether you have any non-key to key dependencies that would mean you are not in the BCNF. You need not revise the model for this, just make sure you are aware of it.
10. Decide if you want to retain the current level of normalization – this is usually a practical decision based on how many joins it will require to answer a query.
    a. For each of your 5 scientific queries, what tables have the required data elements? Do you have to join through additional tables in order to bring this data together? If your query requires more than 4 joins you may want to de-normalize your model to speed up the result.
    b. Document any de-normalization decisions you have made.
    c. Document the final data model.
11. Define a data type and constraints for each attribute, using MySQL conventions (or your RDBMS of choice).
    a. Explicitly map from a sample of your actual data files to the attributes in your data model by giving an example of the sample name and type of data in that file to your database entity and column .
    b. Verify that for the queries you have written the necessary data exists.
12. Testing is VERY important. Write out a validation test plan for the data model.
    a. How are you going to verify that your database instance maps to the model you have created?
        i. If the model is not TOO complicated you can print out a schema and do a visual check of the plan and the instance)
    b. How are you going to verify that the data constraints are properly set?
        i. One way is to use an INSERT statement for each attribute and try to insert incorrect data types. Remember however that SQLite is not rigorous about this – the column can accept data in broad categories, the type is attached to the individual values.
13. Testing the data loading. What validation tests do you will run when verifying your data loading steps?
    a. A common step is to count rows in a target table before and after data loading updates.

b. Another method is to load synthetic data that has some deliberate errors and make sure that they are caught.
c. Yet another is to deliberately try to load in violation of a FK requirement.
d. Note: I do not want a generic answer to this that only mimics what I have stated here, you can use those steps but must express them in terms of a concrete plan for YOUR schema, with an explanation about what you are testing and what you expect.

Note: Since you will not yet have created the schema-preparation scripts you cannot carry out the validation steps yet – this is a plan. In my experience, if you do not have a plan you won't systematically test your product.

Note on what I expect to receive for this assignment

1. Turn in the before and after diagrams for steps 1, 2 and 3 together.
2. Summarize your business rules, then conclude by explaining why you think you have achieved 1NF.
3. Provide a diagram for step 4 that shows any partial and transitive dependencies.
4. Provide a diagram that shows your 3NF model.
5. Explain whether any tables show non-BCNF behavior and if they do but you are not going to resolve that explain why.
6. For each of your 5 scientific queries, explain which tables will be required in the solution. You may decide to modify your queries at this time.
   a. How many joins will each query take?
   b. Are you going to meet the requirements for the class? If not, modify those queries.
   c. Document the reason for any denormalization you are going to carry out.
   d. Provide a schema diagram of your final data model. This is going to be your working data model.
7. Document the data dictionary where you define the data types and constraints for the attributes in your tables. Provide a screen shot of your sample data file (just the top 2-3 rows) and explain where they will go in your database.
   a. Make sure there are no detectable exceptions to the constraints you have chosen.
8. Verify that for the 5 scientific questions the data elements have been mapped from a source file to an entity in a table by explaining which sample data file has the data.
9. Write a detailed model validation plan with steps you can implement. There should be at least one test per table with sub-tests for each attribute and a test of each relationship (remember order of operations and business rules).
10. Write a detailed data loading plan, with steps you can implement.