Database Design Term Project CS6360 Spring 2015

Project Description

Design, develop, and test a university library database. The project consists of four parts: conceptual database design (Phase I), logical database design (Phase II), Oracle relational database implementation (Phase III), and final report &demo (Phase IV).

1. The library has books and magazines.
2. Each book is identified the unique by ISBN (International Standard Book Number, string of length 13). The title, publisher, authors (may be more than one author), year of publication (later than 1900) are also recorded.
3. Each magazine is identified by ISSN (International Standard Serial Number, string of length 8). The title and publisher are recorded for each magazine.
4. A magazine may have several issues, each of which is identified by volume number, issue number (both are positive integers) and ISSN of the magazine. The publication date (later than 01/01/1900) is also recorded for each issue.
5. Publishers have unique name and website. We also record the address (city, state and country) of a publisher.
6. Materials in the library includes both books and magazine issues. The library has one or more copies of each material. Each copy is identified by its unique ID (the value is between 1000000 and 9999999), i.e. each ID identifies a unique item in the library. The purchase date and price are also recorded for each copy. A flag is also used to represent whether the copy is lost.
7. Readers of the library have unique ID. Names of readers are also recorded.

8. There are two types of readers: faculty and student. A reader must be either a faculty or a student, not both.
9. Faculties may reserve materials (materials not copies). For each reservation, we record the date of reservation and expiration date. When a reservation is closed (expired, fulfilled or canceled), we also record the close date.
10. When a reader borrows a copy from the library, a record is generated. Each record is identified by the combination of the reader, the copy and the checkout time. For each record, we need to store its checkout time, due date, return date (null value before return).
11. There may be some fines related to a checkout record. Each fine is identified by the checkout record and the reason of fine (must be chosen from “lost copy”, “damaged copy”, “late return”, “others”). One checkout record may have several fines with different reason (e.g. damaged copy + late return). The amount of fine, issue date and payment date are also recorded.

Project Questions

a) Can you think 2 more rules (other than the one explicitly described above) that are likely to be used in the system?

b) Is the ability to model super-class/subclass relationships likely to be important in such environment? Why or why not?

Project Phases

I. Draw an EER diagram to accurately represent this set of requirement. This will be your Conceptual Design. Clearly specify any assumption that you are making. You can use any tools (software) to draw the EER diagram. You don’t need to describe the value constraints of the attributions in the EER diagram. (20%) (Due on 2/23)

II. Design Logical model for the database. Perform the following steps. (20%) (Due on 3/23)

a. Convert your Conceptual model to a Logical model that can be implemented in a relational DBMS like Oracle. During this process you replace M-N relationships and multi-valued attributes with constructs that can be implemented in the relational DBMS. Draw EER diagram for the logical model after your modifications. Feel free to change your conceptual model (first delivery) if needed.

b. Document your design in Database Schema format like the one we discussed in the class. Briefly justify your choice of mapping rules.

III. Use a relational DBMS to implement the database. (45%) (Due on 4/27)

a. Normalize all of your tables to third normal form. Make any necessary changes to the EER. Explain why these changes needed to be made.

b. Draw a dependency diagram for each table.

c. Write SQL statements to create database, tables and all other structures. Primary keys and foreign keys must be defined appropriately. The quantity constraints of the relation between the entities, which should be described in EER diagram, are not required.

d. Populate your database to test the following views/queries. You don't need to submit your data in report though.

e. Use the Create View statement to create the following views:

    1. Current reservations: show currently open reservations with faculty name and material information.

    2. Past due copies: show copies that are past due with reader ID and name of their borrower.

    3. Unpaid fines: show currently unpaid fines, with fine information and reader name.

    4. Current stock: for each material in the library, show its information and number of copies (including copies checked out but excluding lost copies).

f. Answer the following Queries. Feel free to use any of the views that you

created in previous part.

1. Retrieve the number of current reservations of each faculty.
2. Retrieve the books that was borrowed by a particular reader, with reader information and book information.
3. For each material, retrieve the number of further checkouts allowed (# in stock - # checked out - # reserved).
4. Retrieve the reader name and total amount of unpaid fines for each reader with unpaid fine.
5. Retrieve the materials whose copies are all purchased 20 year ago.
6. Retrieve the information of the material whose average price of copies is highest.
7. Retrieve faculties that had lost copies in the past year.
8. Retrieve the top 10 most popular materials (by the number of total checkouts).
9. Retrieve the list of materials that is published in this year.
10. For materials with more than 10 copies, retrieve their number of checkouts in the last year.
11. Retrieve the most common reason of fine in this year.

IV. Document the final term project report and demo. (15%) (Due on 5/4)

a. Problem description (Copy it from project description).
b. Project questions (Answer questions listed in this project).
c. EER diagram with all assumptions.
d. Relation schema after normalization. All relations must be in 3NF. The relation schema should include primary keys as well as foreign keys (if any) for all relations.
e. All requested SQL statements.
f. Dependency diagram.

g. Demo.