

Database Design Experiment: Using MySQL and PHP to Implement the Charity Management System

This project is to use MySQL and PHP to implement the Charity Management System application. In this case, you'll design a relational database for a company that keeps track of charities and their donors. After your database design is completed and correct, you will create database tables and populate them with data. Then you will produce some queries and some functions. The functions will allow the company to easily record donations to each charity. The queries will allow the management, donors, and charities to access important information. In addition, these queries will list charities rated by their program expenses, donors in a specific category of charity, top givers in a specific charity, frequency of giving by donors, and matching gifts for a specific charity. After all databases and tables are well done, an implement of Web interactive interface using PHP is needed which ensure users of your system can use your project conveniently.

Background

Charitable organizations are big business. Although most people who give to charities see it as a "one-time" donation, charities view a donation as the beginning of a lucrative relationship with the donor. Studies have shown that the three months after a charitable donation are critical in cementing the donor/charity relationship. Donors who give again within three months are more likely to contribute than those who give 12 months after the initial gift. Charities walk a fine line: They want to contact donors for additional gifts, but don't want to pester them. Also, mailings to prospective donors can be very expensive. Charities should use sophisticated statistical analysis to figure out who would be a good donor, like marketers use to figure out what customers will buy.

A college friend of yours, Dong Yunhao, is interested in charities both from the organizational side and the donors' side. He has started a small company called Charity Info to track charities' statistics and donors' statistics. His vision is to share charity statistics with potential donors and help them to select the best charity for their money and interests. Yunhao also will help charities in tracking donors' habits and analyze where they can best put their fundraising resources. He has already obtained a few accounts and realizes that the information must be organized in a database to make it useful to the charity or donor. You have been asked to design and implement a charity database because you know about database design.

You must keep several parameters in mind when designing the charity database. For example, you must keep information on the donors. All donors are assigned a unique ID number when they register to donate. Information about names, addresses, and telephone numbers is also recorded. To track charities, you must record information that includes the charity's name, address, and telephone number, along with statistics for program expenses, administrative and fundraising expenses, and recent revenue. Program expenses are the percentage of the charity's total budget that is spent on programs for its benefactors. Likewise, administrative expenses are the percentage of the budget spent on administering the charity and fundraising. If fundraising expenses are reduced by Charity Info's analysis, then more money will be available for needed programs.

Each charity also fits under a specific category, such as animals or environmentalism, that allows potential donors to narrow their search for a charity. Finally, all donations must be recorded.

Yunhao would like to have a way to record donations from each donor easily. After donor and charity information is in the database, he wants to be able to enter donation information quickly for repeat donors. You suggest using functions to expedite this chore. Eventually, this form can be migrated to the Web for self-service.

Yunhao also has several questions for the database. For example, donors call and ask about a charity's program expenses. Often their decision to give to an organization depends on whether most of the money goes to the needy or to administrative costs. Therefore, you need to create a query that lists the charities rated by their program expenses. Then, for marketing purposes, charities want to know who donates to specific categories of charities such as arts, environment, or human services. A query can answer such questions easily.

Within each charity, Yunhao wants to know the top donors. In addition, charities often would like to see the frequency of giving by donors and their average gifts, because this affects their soliciting of future gifts. Queries can answer both requests. Finally, for some charities, anonymous donors will match existing donations. This has already happened with a charity in Phoenix, Arizona, that deals with international hunger. An anonymous donor has agreed to match 50 cents for every dollar donated. A query can calculate the donations and total both the original and the matching donation.

1. Building the database (15 points)

In this part, please use the SQL statement to create a database with at least 6 tables to store different data to meet the basic standard and more table is also welcomed if necessary.

- For every charity, it has its own information including but not limited to ID, category, name, address, city, state, telephone, 2005 revenue and expenses. The expenses includes the three aspects of program admin and fundraising and you need to record the data of these three aspects separately.
- For every donors, you should record their ID, their last and first name, their address, city, state, zip and telephone.
- For every gifts, its donor, charity, date and amount is required.
- For the reason of security, a table which include operation logs is also required. You are suggested implement this table by triggers on other tables. This table describes who (the login name of a database user) has performed what operation (insert, delete, update) on which table (give the table name) and which tuple (as indicated by the value of the primary key of the tuple) at what time.
- You may select the appropriate data types for these fields and create the tables. Then you should insert some data (more than 20 in each table) to your tables. Note that you need to write out the data type of each field and the primary key and foreign key of the tables in the report you submit later.

The meanings of most of the tables and their attributes are clear. If they are not clear to you, please check the information in Background.

2. MySQL Implementation (55 points)

You need to create five queries and give a presentation as outlined in the Background section of this case.

1. (6 points) Create a query called Charities rated by Program Expenses. This query should list the Charity Name, Address, City, State, Zip, and Program Expenses. List the highest Program Expenses for the charity to the lowest.
2. (6 points) Create a query called Donors by Requested Category. List the Category, Donor Last and First Names, Street Address, City, State, Zip, and Telephone. Your query should prompt for a category and allow the user to enter the desired category.

3. (6 points) Create a query called Top Giver of Requested Charity. Display the Charity ID, Charity Name, Donor Last and First Names, and Total amount given. This query should prompt for a desired Charity ID. Keep in mind that some donors give more than once, but you want to display the total amount given. Also, your output needs to be sorted from top giver to lowest giver. Note the change to the column heading.
4. (6 points) Create a query called Frequency of Giving by Donor. List the Donor Last and First Names, the Number of Gifts, and the Average Gift. Sort alphabetically on Donor Last Name. Note the column headings' changes.
5. (6 points) Create a query called Matching Gifts that displays the Charity Name, Date, and Amount, and then calculate the Matched amount and the Total Donation. Recall that an anonymous donor is willing to match 50 cents on every dollar donated to the Food for the Hungry charity.
6. (8 points) Create two function to manage donors and charity. If the data(donors or charity) does not exist, it is inserted into the table, otherwise it is just updated.
7. (8 points) Add a tuple to the logs table automatically whenever any table is modified. You are suggested implement this table by triggers on other tables. This tuple describes who (the login name of a database user) has performed what operation (insert, delete, update) on which table (give the table name) and which tuple (as indicated by the value of the primary key of the tuple) at what time. You should also record both the updated value and the previous value.
8. (9 points) You need to make your code user friendly by designing and displaying appropriate messages for all exceptions. More specifically, when there exist some exceptions for the operations of the database, your program should report the problem clearly.

3. Interface (20 points)

Implement a Web interactive interface using PHP. Your interface program should utilize as many of your MySQL stored procedures/functions as possible.

4. Documentation (10 points)

Documentation consists of the following aspects:

1. Each procedure and function and every other object in both database and Web interface created for your project needs to be explained clearly regarding its structure objective and usage.
2. Your code needs to be well documented with in-line comments and perfect indentation.

5. Hand-ins, Demo and Grading

1. You should submit all your work as a technical report to Blackboard which includes following subjects **at least**: introduction and implement of your databases and Web interface, documentation, project running results etc. In addition, particular methods to setup your project and all source code is also required to submit as attachments so that the instructor can check your work easily.
2. The grading will be based on the quality of your code, the presentation, the report, the documentation and on how successful of your demo is.
3. **NEVER CHEAT!** Do not copy! Including but not limited to code, picture, technical report, and documentation. Proven copy will be marked as zero!