CS 450 - Database Concepts

Spring 2012

Instructor: Dr. Jessica Lin

Project Assignment

General. Your project is to design and implement a database for a deal aggregator site similar to LivingSocial. At the back-end, the database manages all of its data in the Oracle database management system. At the front-end, you are required to implement a command-line interface to its users using Java and JDBC.

I will expect certain minimal functionality in your database. Minimal functionality includes a variety of queries and browsing capabilities over the current deals (e.g. the database user should be able to search by date, deal type, merchant, location, etc.), purchase history, and customers. The user should be able to update the database by inserting or deleting tuples.

The following is the list of "basic" information that the database should store. Note the organization of the list does not necessarily imply that this is how you should design the database. For example, for each customer, you have a list of items that need to be stored. It is up to you to decide how you should store and organize these items. Just because vouchers are listed under "Customers" doesn't mean that vouchers are part of the Customers entity set. You will need to come up with a design that makes sense, given the descriptions.

1. Customers. For each customer:
    a. Unique email address as login name
    b. Full name (first name, last name)
    c. Age
    d. Gender
    e. Credit card(s) on file (note the plural – each customer can store more than one card on file). One credit card should be designated as the default card.
    f. Home address
    g. Default location that the customer subscribes to. It does not have to be the same location as where the customer resides. This location should be one of the available locations offered by the site (see #2 below).
    h. Other locations that the customer is interested in (i.e. subscribes to). For example, my default location may be Washington DC, but I may be also interested in knowing deals that are in New York, NY. These locations should be part of the available locations offered by the site (see #2 below).
    i. Vouchers purchased. See #5 for more detail. Each voucher should have a status: current, expired, used, or refunded.
    j. Types of deals that the customer is interested in (e.g. food, health & fitness, adventure, service, etc). See 4(j) for more detail.

2. Locations – a list of cities that the deal aggregator site offers deals on. For an example, see http://www.livingsocial.com/cities/1165-arlington-alexandria/more_cities. In addition to city names, also record states (for the U.S.), countries, and continents.

3. Merchants. For each merchant:
    a. Unique merchant ID
    b. Merchant name
    c. Merchant headquarter – the address of the headquarter (the corporate office).
    d. Address(es) of deal location(s): There might be several branches that the deal applies to. For example, suppose there is a deal for the restaurant chain California Pizza Kitchen (CPK). CPK could decide whether the deal applies to only one location (e.g. Tyson's Corner), or several locations. The database needs to record the address(es) of all locations that the deal apply to. Note the location is not necessarily the same as the headquarter, though in some cases they might be the same (e.g. local B&B or attraction).

4. Deals. For each deal:
    a. Merchant that offers the deal
    b. Deal description
    c. Location(s) that the deal applies to. See 3(d).
    d. Original price
    e. Deal price
    f. Expiration date
    g. Quantity of the deal offered by the merchant – some merchants put a quota on the number of coupons they are willing to offer, but some may not.
    h. Sale start time – the time when the deal becomes available for purchasing
    i. Sale end time – the time when the sale is over (note this is different from deal *expiration* date)
    j. Type of deal – Each deal should belong to one of the following 9 categories: food & drink, service, automobile, health & fitness, children, fun & adventure, travel, goods, other.

5. Purchase History:
    a. Who purchased what
    b. Purchase date

c. Quantity of vouchers purchased by the customer
d. Each voucher should have a unique voucher number. For example, if a customer purchased 3 vouchers for a deal, then there should be three voucher numbers (one for each).
e. Who the voucher is for – a customer may buy a voucher for him/herself, or for someone else.
f. Location(s) that the voucher can be used at. See 4(c).

6. Reviews. Suppose customers that purchased deals can rate and review on the deal. The rating system is similar to that of eBay. More specifically, only customers that purchase a deal can rate on that deal. Indirectly, the customers are also rating on the merchant that offers the deal. Each review should contain:
    a. Reviewer ID
    b. Deal and merchant to be rated
    c. Rating, on the scale of 1 to 5, 1 being "poor" and 5 being "excellent"
    d. Review. Maximum number of characters is 500.

The descriptions above summarize the minimal information your database should contain. As mentioned, it does not necessarily imply the organization of the relations. You may re-arrange the attributes or add more attributes. The design decision is yours, but your design should be reasonably efficient, and well justified. You will be graded on both correctness and quality of the design.

You'll be graded for each phase. The project accounts for 20% of your total grade.

Phase 1: Conceptual Design. (4%) You need to submit a corresponding ER diagram according to the project description and the assumption you may have made. (See HW1 assignment for an example).

Phase 2: Schema Design. (4%) After your conceptual design is finished, you need to translate the ER diagram into relation schema. Submit a copy of your relation schema, the SQL script that you use to implement your database, and insert enough tuples in **each** table, e.g. 10 customers, 10 merchants, 10 deals (one for each merchant), 20 purchases, 20 reviews, etc. Also submit a copy of your original (with the TA's feedback) and the revised ER designs. Your grade for Phase 1 will be the average of your initial score from Phase 1, and the score for your revision. However, revision is not optional. It is **required** if you want to get full credit for Phase 2. The only exception is if you already receive 100% on Phase 1.

Phase 3: JDBC Implementation. (10%) Write a program in JDBC that will allow users to access and query your database. Command-line menu is acceptable. A list of queries will be posted later.

Putting it together. (2%) Prepare a final report. Your report should contain materials from all phases. Discuss and justify your design decisions, and challenges you encountered.

Timeline:

March 5: project released

March 21: Phase 1 (ER diagram) due

March 28: Phase 1 returned

April 4: Phase 2 (schema) due

(April 25): HW4 due

May 2: Phase 3 and final report due

May 9: Final