Lab 4: Software Design, Database Creation, Query Prototyping

Due date: Thursday, May 7, in-class.

This is a **team lab**. Each team submits one set of deliverables.

Lab Overview

This is the last project-related lab for the class. During this lab each team will do the following:

- Finalize the database design started in Lab 3.
- Create the database using data provided by the customer.
- Prepare the software design and the GUI prototypes for the key functionality of the software.
- Develop and debug prototypes for the SQL (or SQL/PL) queries to be used by the software you are developing.

The lab does not have specific stages, however, on **Tuesday, May 5**, we will receive a visit from Dave Cumberland. He will spend about 20 minutes talking to each team in turn. The specific instructions for what each team is expected to achieve during its time with the customer are provided below.

Software Specification

The main document driving your software specification and prototyping efforts is the **Use cases** document created by the customer. The first version of the document is available at this url:

http://users.csc.calpoly.edu/~dekhtyar/366-Spring2009/project/UseCases.pdf

The document is linked to from the course web page and the course wiki. An updated version of the document will be released shortly.

The document provides the list of use cases that the customer wants implemented in the software. The specific details of implementation of these use cases, whenever they go beyond the text of the document are left to individual teams. Some comments on the implementation strategies are given below.

Web-based and standalone application

The customer does not have a preference for the form in which the database application is delivered. Two forms have been discussed: a web-based tool and a standalone Java tool.

Standalone Java Tool. The initial intent of the project was to have each team design a standalone Java tool. For such a tool, Oracle DBMS is to be used as the back end, and JDBC connectivity — to pass information between the Java front end of the system and the Oracle back end. The system is expected to have graphical front end (use Swing or SWT, or any other Java GUI library) that is designed to accommodate all use cases and display all desired information.

Web-based Tool. As a challenge, and for some extra credit, teams may elect to implement the database application as a **web-based on-line tool**.

Note, that the **use cases document** is **independent of the implementation platform.** Therefore, functionally, there will be no difference between the on-line and the standalone versions of the tool. However, you will be using different tools for your development.

Each team working on an on-line application will receive a virtual machine (VM) on the CSL's `csc.calpoly.edu` domain. On this machine, each team (possibly with some assistance) will install the LAMP stack ("A" is for Apache web server, "M" is for MySQL DBMS and "P" is for PHP) for web development. If a team desires a different web application development environment, the team is responsible for installation and further support of it on the VM. (teams will be given root priveleges to their VMs).

Lab Assignment

System Components

Each team has to prepare the following system components as deliverables.

The database. The initial database design you proposed in Lab 3 will be returned to you with the instructor's comments on *Thursday, April 30.*

Each team shall prepare the final database design based on the newly available information and the instructor's comments. Instructor's comments, especially if/when they relate to incompleteness or outright incorrectness of the database design, need to be taken into account during the redesign. We will use the *Thursday, April 30 lab period* to discuss the results of Lab 3 and work with each team on finalizing the database design.

Bulk-loader. There is no use case associated with outright bulk loading of the database in the **Use Cases** document. However, there is a Manufacturer Updates Products in Bulk use case (which will be further clarified in the updated version of the document) use case.

To make your work on SQL queries efficient, you are asked to produce a special-purpose bulk loader for the customer data. The bulk loader can be written in any programming and/or scripting language and shall be a stand-alone program. (You can and **should** later incorporate portions of it in the code base of your system).

The bulk loaded shall load the data into an empty database in the following manner:

1. **First,** the bulk loader shall access the list of catalogs provided by the customer in the `catalogs.csv` file. From this list, the bulk loader shall extract the information about the Manufacturers and the Catalogs entity sets and their associations and populate appropriate database tables.

2. **Next,** for each catalog (and for each language in which the catalog is available), the bulk loader shall discover (see below) the names of the .csv files that store the products data. The bulk loader shall process these files to populate the database tables storing information about the products, product localizations (names, descriptions, prices), product options and their localizations.

3. **Finally,** the bulk loader shall create a number of customers, orders and carts in the database.

   Note, that orders and carts are not provided as part of the customer-supplied data, but their presence in the database is necessary to successfully compose, and then, validate a number of important SQL queries in order to satisfy some customer use cases.

   To help your development efforts, create a SQL script file `orders.sql`. Manually populate this file with all `INSERT` statements necessary to create the following:

   - **Five** customers.
   - **Five** placed orders for specific products from specific catalogs. Of these five orders:
     - **at least two orders** shall be from the same catalog;

  - **at least two orders** shall be from the same customer (it is ok to have customers with no orders in the database);
  - **at least one order** shall have more than one product ordered.
  - **at least two orders** shall be for products with options.
  - **at least one order** shall use a language other than English.
- **Two carts** not yet turned into orders. The carts should be for different catalogs and shall contain at least one item each.

Use specific items from specific catalogs in the customer-supplied data.

Your bulk loader shall execute the `orders.sql` script after all other data had been inserted.

Query Templates. Each team shall study the use cases provided by the customer, and, based on them, develop and debug SQL templates for each *major* type of query that needs to be used in the system. An SQL template is an SQL statement for a user-input-dependent query. Values of specific attrbiutes, which are expected to be provided by the user can be replaced with templates. The SQL templates can be presented as-is or wrapped into Java methods, PL/SQL procedures, or other mechanisms that allow parameter passing.

An example of a SQL template query that retrieves the list of all employees who work in a given department is:

```
SELECT *
FROM Employees
WHERE department = <<DepartmentName>>;
```

(feel free to you your own notation for template values, as long as it a clear distinction between the template values and the actual values of attributes is made)

For each major query established, prepare the query template, and one or more actual examples that would return answers from the test database. Submit all templates and all queries in a single SQL script called `templates.sql`. Query templates should be commented out. Additionally, put comments specifying what each query template is designed to do.

For example, the portion of the `templates.sql` file containing the template about could look as follows:

```
--  Template #1: Select the list of employees from a given department.
--   Parameters: <<DepartmentName>>: name of the department.
--
--  Template:
--
--  SELECT *
```

```
--  FROM Employees
--  WHERE department = <<DepartmentName>>;
--
-- Examples:
--
SELECT *
FROM Employees
WHERE department = 'Toys';
```

Software Design

Each team shall select the type of the software system it will be building (standalone or web-based) and create a prototype of the front end of the system. The key goal of the prototype is to give the instructor and the customer the idea of how the human-computer interaction outlined in the **Use Cases document** (or implied by it) will be implemented in the system.

Each team can choose how to prototype the front-end of the system for this lab.

May 5 lab period: meet with the customer. Dave Cumberland will join us for the lab period on *Tuesday, May 5* to discuss the software design with each group and to answer any questions about other pending issues.

Each team shall have a draft of their software prototype ready for Dave to examine and comment on. Each team will have about 15–20 mins of Dave's time during the lab period to show the design and consult on other issues. As with **Q&A** session, Dave's comments need to be recorded and documented — eventually — on team's wiki page.

Finalizing design. Each team shall prepare the front-end design specification. The format of the specification is left to each team, but the following should be in the specification:

- For each use case, screenshots/storyboards of the GUI.
- For each use case, team's notes on the use case.
- For each use case, customer's comments and team's reaction (as applicable).

More about Bulk Loader

The **bulk loader** program/script you are asked to write for this assignment needs to satisfy the following two requirements in addition to the specifications above:

Database Connectivity. The bulk loader shall read the database connectivity information from the file `account.db` located in the same directory as the bulk loader executable.

`account.db` file will consist of two lines. First line will contain the loginId of the Oracle account on the `ora10g.csc.calpoly.edu` Oracle server which the bulk loader needs to use to connect to the database. The second line will contain the password for this account.

For example, if I want to create a database under Oracle account `user05` which has password `goPoly01`, `account.db` file will look as follows:

```
user05
goPoly01
```

Catalog Files. The bulk loader shall take as input a single file name. The file name will identify the file containing the information about catalogs that the bulk loader needs to load and the locations of the files for this catalog.

The input file will be in CSV (comma-separated values) format. Each line of the input file will describe which CSV files contain the information about products and options for a given catalog—language pair.

The format of the CSV file record is:

```
<Manufacturer>,<Catalog>,<Language>,<ProductFile>,<OptionsFile>
```

Here, the column names mean the following:

Column Name	Meaning
`Manufacturer`	Name of the manufacturer of the catalog
`Catalog`	Name of the catalog
`Language`	Lanugage of the catalog
`ProductFile`	CSV file with the product data
`OptionsFile`	CSV file with the product options data

Note, that the first three columns in this file are the same as the `MFG_NAME`, `CATALOG_NAME` and `LANGUAGE` columns from the `catalogs.csv` file.

The product file name and the options file name **will include the relative path to the file from the root of the `product_data` directory**.

A sample line from an input file, describing the `Ernie Ball` US catalog will look as follows:

```
"Ernie Ball","Ernie Ball","en_us",ernieball/ernie_ball_us_product_data.Ernie_Ball.USD.en_us.csv,ernieball/ernie_ball_us_option_data.Ernie_Ball.USD.en_us.csv
```

An input file `all-catalogs.csv` containing the information about all customer-supplied catalogs is available to you from the course web page:

http://users.csc.calpoly.edu/∼dekhtyar/366-Spring2009/project/all-catalogs.csv

Submission Instructions

This lab has both hardcopy and electronic deliverables. It is expected that all hardcopy deliverables are also available electronically on the team's project wiki page (as wiki pages or as attachments).

Submit as hardcopy:

1. Final database design. You can submit the printout of the **DB-setup.sql** file.
2. Changelog for the final database design.
3. **Original design submissions from Labs 2 and 3.**
4. Initial software design specification.
5. Query templates. Submit the printout of the `queries.sql` file.
6. `orders.sql` file.

Submit electronically using `handin`

1. The DB-setup.sql file with all `CREATE TABLE` statements for the group's database.
2. The DB-drop.sql file containing `DROP TABLE` statements for each table in your database.
3. The full code for the bulk loader program/script.
4. `README` file explaining how to run the bulk loader (and containing any other necessary explanations).
5. `orders.sql` file.
6. `queries.sql` file.

Put all the files into a single zip or gzipped tar archive `lab4.zip` (or `lab4.tar.gz`) and submit using the following command:

```
handin dekhtyar lab04-366 lab4.<ext>
```