Part Four: Database Testing

Weining Zhang

October 31, 2012

- Total Points: 100
- Total Weight: 6%
- Due: Friday, November 9, 2012, in class

Description

In this part, you will use Oracle to test and to manipulate your database. The purpose are two-folds. First, to design SQL queries that will eventually be included in the implementation of your application program. Second, to make sure the data in your database is sufficient for testing your application. Specifically, you need to do the following.

1. Learn SQL*PLUS

   You should try SQL*PLUS commands and SQL statements interactively, run SQL statements from a script file, and edit and run commands from the command buffer.

2. Test SQL Queries

   You should write and test at least 10 interesting, non-trivial SQL queries on your database. Ideally, the queries should include the ones you plan to use in your application program to implement services for users. Nevertheless, you should include at least one query in each of the following query types. Of course, some of your queries may be in more than one type.

   (a) Multiple table query;

   (b) Nested query (both correlated and not correlated);

   (c) Query using Union, Intersect, and minus;

(d) Query using exist, not exist, IN, NOT IN, ALL, etc.;

(e) Query with aggregate functions, group by, sort by, and having;

(f) Query that has a complex from clause; and

(g) Query using a view.

3. Database Update

   You should use SQL to make at least two updates of each of the following types:

   (a) Insert tuples to a table (both single tuple and by a query);

   (b) Delete tuples from a table (both single tuple and by a query);

   (c) Updates that changes existing tuples (both single tuple and by a query); and

   (d) Updates that cause a trigger to be fired.

4. Additional Schema Objects

   In addition to what you already have, you will add some new features to your database and test these features. More specifically, you need to do the following.

   (a) Design at least one view that accesses two or more base tables.

   (b) Design and test at least one trigger.

   (c) Design and perform some experiments to investigate view update. These experiments should try to perform various types of update on various type of views, and investigate if and how data can be updated through using views.

While you are doing this part, keep in mind the following:

- Test queries that can be used in your application program. Think how to retrieve data to provide services of your application program.

- Specify each query in different ways. It is well known that a query can be expressed in many ways in SQL. Pay attention to whether a result contains duplicate.

- Make sure that your interesting non-trivial queries get non-empty answers. Add more data to your database if necessary.

- Come up with each query in English first and then translate it into SQL.

What to Hand In

Hand in a well-formatted written report that includes the following items.

1. Revised report of Part Three

   This should be a fresh copy of Part Three incorporating all changes to all previous parts.

2. SQL Queries

   A list of your SQL queries. For each query, list the English version followed immediately by the SQL statements. These queries will be graded based on if they are interesting and if they are correct with respect to their English statements.

3. Views and Triggers

   A list of SQL statements that define your view and trigger.

4. Spool File

   A spool file that illustrates a successful execution of your queries, updates, views and triggers. Do not include answers that contain many pages of data. For updates, show the relevant data before and after the execution of your update commands.

5. View Investigation

   Include a two-page report describing your experiments with view update including the design of the experiments, your observations, and your conclusions. This report will be graded based on the design of experiments and on the quality of the technical writing.