Assignment #8
Due Thursday December 7, 5:00 PM

Problems to Work and Hand In

The following two questions refer to relation `Serves(bar, beer, price)` from the "Beer Drinkers' Relational Database Schema."

1. (10 pts.) Write C code with embedded SQL2 (`EXEC SQL` statements) to:

   (a) Ask a user for the name of a beer.

   (b) Find and print the lowest price for that beer.

   (c) Find and print the names of all the bars that serve this beer at the lowest price.

   You may use comments in place of C code as appropriate, using the course reader as a model.

2. (10 pts.) Suppose that A is the owner of the `Serves` relation. The following sequence of `GRANT` and `REVOKE` actions take place.

Step	By	Action
1	A	`GRANT SELECT ON Serves TO` B `WITH GRANT OPTION`
2	A	`GRANT SELECT ON Serves TO` C
3	B	`GRANT SELECT ON Serves TO` D `WITH GRANT OPTION`
4	D	`GRANT SELECT ON Serves TO` B, C, E `WITH GRANT OPTION`
5	B	`REVOKE SELECT ON Serves FROM` D `CASCADE`
6	A	`REVOKE SELECT ON Serves FROM` C `CASCADE`

   Show the grant diagrams after steps 4, 5, and 6. Since all privileges in this problem are `SELECT ON Serves`, you do not have to show that fact explicitly in your diagrams.

Eighth and Final Step of Your Personal Database Application

Some of these problems involve Sybase programming and some do not. For those that require programming, it is fine to use the `isql` interface.

1. (15 pts.) Compose an `isql` input file that creates a transaction, using the Sybase-SQL statement `BEGIN TRANSACTION`, does a data modification, and then commits the transaction with SQL statement `COMMIT`. Issue a query that demonstrates the modification has been made to the database. Then begin another transaction and repeat the modification (on the original database). This time, end the transaction with the SQL statement `ROLLBACK`, which aborts the transaction and rolls back any changes to the database. Issue a query that demonstrates the change has not been made to the database. Hand in a script showing all of these statements and the results of the queries.

2. (10 pts.) First, please remind us of the schema for your PDA that you implemented in Sybase. Suppose multiple clients are operating on your PDA database. Give an example where problems with concurrent access can arise if transactions are not used. For each example, please specify:

(a) An initial set of data in one or more relations of your PDA (the initial "state").

(b) SQL commands issued by two clients concurrently (one or more commands per client).

(c) A final state of the relations in which the commands have been executed in an interleaved fashion so the resulting state is intuitively incorrect.

Your example will be graded on simplicity as well as correctness. Note that you do not have to write any C code for this problem, and you do not have to run your SQL statements.

3. (10 pts.) Now suppose that your multiple clients use transactions in order to alleviate the problems with concurrent access. Give one example where the final state of your PDA database depends on the order in which the transactions are executed. Please specify:

   (a) An initial state for one or more relations.

   (b) SQL transactions issued by two clients concurrently (one transaction per client).

   (c) Two different valid final states of the relations, depending on which transaction executed first.

   Again, simplicity counts as well as correctness, and no code needs to be run.

4. (10 pts.) Using your PDA, describe a simple scenario in which an application program (a C program running on your PDA using embedded SQL and transaction commands) might decide to abort a transaction based on user input. Again, no code is needed; just hand in a prose description of the scenario.

5. (15 pts.) If you have not already done so, modify your PDA schema to include all appropriate key and foreign key (referential integrity) constraints. There must be at least one of each type. If you do not see any such constraints in your database, you should probably consult course staff, and if they agree that there are none, you'll have to modify your schema or "pretend." Load your schema with either your small or large data set or a few hand-entered tuples.[1] Now select database modifications of the following five kinds to demonstrate the effect of violating these constraints.

   (a) An `INSERT` command creating a key violation

   (b) An `UPDATE` command creating a key violation

   (c) An `INSERT` command creating a referential integrity violation

   (d) A `DELETE` command creating a referential integrity violation

   (e) An `UPDATE` command creating a referential integrity violation

   Hand in your modified schema with the integrity constraints and a script showing your modification statements and their results.

6. (20 pts.) For your PDA, write one plausible assertion (using the SQL2 notation) that should be obeyed. Sybase 10.0 does not support SQL2 assertions, but you can implement the assertion by one or more triggers that fix up the problems caused by database modifications that violate your assertion. Implement these triggers and show a script in which one of them is triggered. Hand in both the code for your triggers and the script.

[1] Remember, however, that we are asked not to load large data sets with indexes on.