

Fix the Bank's Database

Scenario

Your local bank is interested in a new, improved version of their software. The original software has been in use since 1960, was written in Cobol, and is due for an upgrade to an object-oriented solution. One major revision, which you have been put in charge of solving, involves speeding up teller transactions. The problem stems from the overuse of an SQL database. The database has a warm backup located in North Carolina, uses several MyISAM tables, and each transaction takes several seconds to complete. Since this isn't an assignment for a database class, modifying the database tables to use InnoDB to solve this problem is not an option.

Problem

Your task is to ensure that individual bank transactions are performed rapidly. The current software is being developed to interact directly with a *Database* object using the *executeSQL* command. At the start of the business day, the method *openDatabase* is called, and at the end of a business day, the method *closeDatabase* is called. The bank has pointed out that each SQL transaction (i.e., each *executeSQL* command) could be executed at the end of the day when *closeDatabase* is called, but this would involve a major rewrite of other parts of the software developed already. From the existing software's perspective, your solution must be easily exchanged with the current *Database* class.

Deliverables

1. Identify the design pattern you used to solve this problem, and the participants (i.e., the roles each class takes).
2. An implementation in a language of your choice.
3. A class diagram of your solution (including existing classes), so future developers can easily see how to work with your solution.