

CSCI/CMPE 4333 Database Design and Implementation

Project Stage 3: Database Application Development

Instructor: Artem Chebotko

Assigned on: September 1, 2009

Due on: See the course webpage

Description: The third stage of the Project consists of the following tasks:

1. Develop a database application with GUI that can:
 - connect to the RDMBS (Oracle/MySQL); the application should request all information required for connection from a user, such as login, password, connection URL, etc.
 - create the database schema using the previously developed SQL scripts; the application should provide a button that will initiate the database schema generation process.
 - populate the database with sample tuples, such that the result of the queries described later is not empty; the application should provide a button that will initiate the database population process.
 - query the database; the application should take any user query, execute it, and return its result to the user.

Note that you can choose to develop (1) one application that can switch between both Oracle and MySQL or (2) two separate applications, one for Oracle and one for MySQL. You are also free to select a programming language and appropriate technologies for the task. For example, you may decide to develop a three tier application with a Web-based interface that deploys as a browser, Web server, and database server, or you may go with a two tier application (like DBExplorer). While there are no restrictions on programming languages, architectures, and technologies to use for this Project stage, the course webpage provides the DBExplorer application as a template. DBExplorer is a Java application with GUI and MySQL-JDBC support. It features the database connection dialog (see Fig. 1) and querying capabilities (see Fig. 2), which you are welcome to reuse.

2. Design six useful SQL queries for your database that use (1) union, (2) intersection, (3) difference, (4) division, (5) aggregation, and (6) at least three joins (inner or outer).

For example, for the banking enterprise, sample queries are as follows.

Query 1: Retrieve information about customers (e.g., SSN/ID and name) who have checking or savings accounts. [Hint: union]

Query 2: Retrieve information about customers who have both checking and savings accounts. [Hint: intersection]

Query 3: Retrieve information about customers who have both checking and savings accounts, but not a retirement account. [Hint: intersection and difference]

Query 4: Retrieve information about customers who have all types of accounts. [Hint: division]

Query 5: Retrieve information about each account (e.g., account number) and (next to it) number of recorded transactions for the account. [Hint: aggregation]

Query 6: Retrieve names of customers and their financial advisors. [Hint: inner joins between Customer, Account, Manages, and Advisor]

Note that Oracle's and MySQL's SQL dialects have some subtle differences. Thus, the same query may have different SQL representations for Oracle and MySQL (e.g., Oracle has the explicit construct `MINUS` for difference, but MySQL does not).

Implement these six queries in your application, such that the user can select one of them via GUI and further execute.

Submission: Submit the printed copy of your SQL scripts and database implementation report to the instructor or teaching assistant. You should also schedule a 10 minute demo session with the teaching assistant to demonstrate the functionality of your software (no grade will be assigned for the project without the software demonstration)

Grading: Tasks 1 and 2 receive 50 points each (100 total, if solved correctly). The best solution (one team) receives 5 extra points.

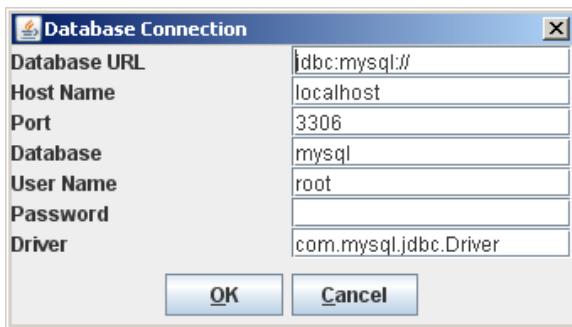


Fig. 1. Database connection dialog

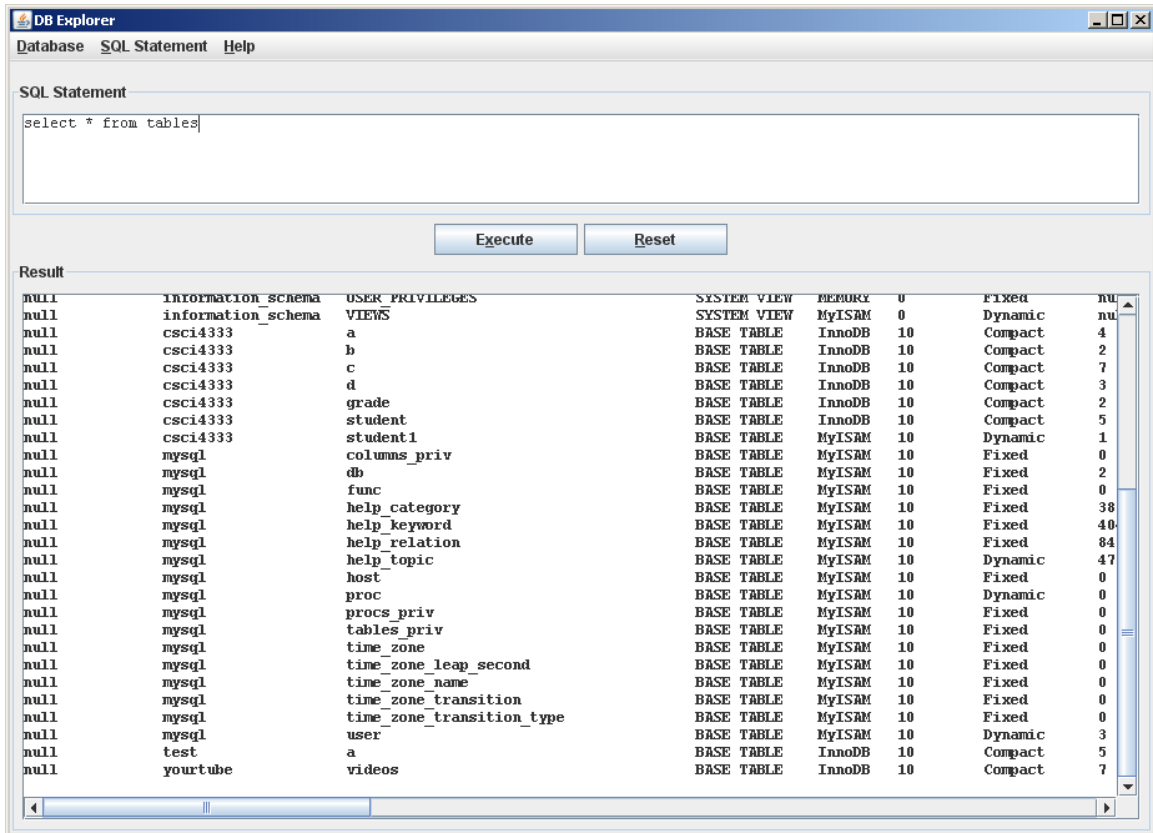


Fig. 2. DBExplorer GUI and querying capabilities