

Homework Assignment 4

Java Database Connectivity

ECE 155B – Spring 2009
Demos – Friday, May 22

In this assignment you will modify your Party Planner Service System that supports Clients, Party Planners and Caterers so that all information at the Party Planner is stored in an SQL database, rather than an XML file.

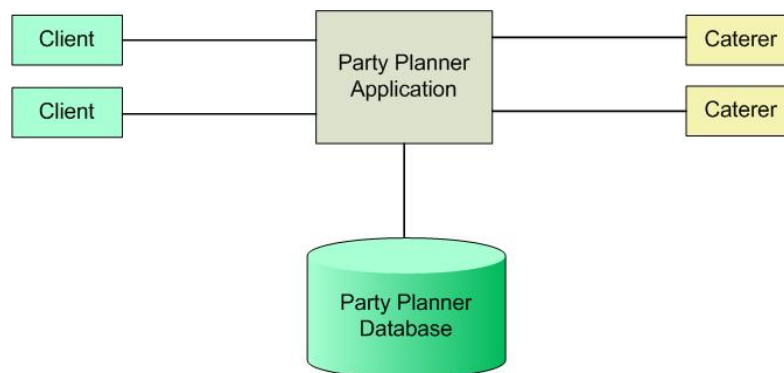
For this assignment, your Party Planner Service System will be implemented as a three-tier distributed application, using Java Database Connectivity (JDBC) and the MySQL database system. You will be provided MySQL accounts that you will use for this project.

Please contact the TA for a MySQL account, username and password for the MySQL account. Provide your desired username/password in your email to the TA.

Requirements

This assignment requires that you modify your Party Planner Service System to run as a three-tier distributed application with a database in the back-end (third) tier at the Party Planner. The Party Planner application in the middle tier will use the JDBC APIs to encapsulate the SQL commands to the MySQL database system in the third tier.

The Client in the first tier will have a Graphical User Interfaces (GUI) and provide the same functionality as in the previous project. Likewise, the Caterer application will be the same as in the previous project (unless you choose to use a database instead of an XML file at the Caterer). The Party Planner application will insert, update and retrieve data from the database in the third tier at the Party Planner.



Your Database at the Party Planner will consist of the following tables:

- o **Client** (Client profiles)
- o **Facility** (Facility profiles)
- o **Caterer** (Caterer profiles, network address and port number)
- o **Party** (relational table referencing Client, Facility and Caterer)

Optional [10 points]:

- o Provide a database for the Caterer with (at least) the following table:
 - **PartyPreparation**

Example Client, Facility, Caterer and Party table definitions.

Table Name:
client
Database:
miaodb
Comment:
InnoDB free: 10240 kB

Columns and Indices
Table Options
Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
clientid	VARCHAR(20)	☒	☐	☐ BINARY		
password	VARCHAR(45)	☒	☐	☐ BINARY		
interest	VARCHAR(45)	☒	☐	☐ BINARY		

Table Name:
facility
Database:
miaodb
Comment:
InnoDB free: 10240 kB

Columns and Indices
Table Options
Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
facilityid	VARCHAR(20)	☒	☐	☐ BINARY		
capacity	INTEGER	☒	☐	☒ UNSIGNED ☐ ZEROFILL		
description	VARCHAR(45)	☒	☐	☐ BINARY		

Table Name:
caterer
Database:
miaodb
Comment:
InnoDB free: 10240 kB

Columns and Indices
Table Options
Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
catererid	VARCHAR(20)	☒	☐	☐ BINARY		
ip	VARCHAR(45)	☒	☐	☐ BINARY		
port	INTEGER	☒	☐	☒ UNSIGNED ☐ ZEROFILL		

Table Name:
party
Database:
miaodb
Comment:
InnoDB free: 10240 kB; ("caterer

Columns and Indices
Table Options
Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
partyid	VARCHAR(10)	☒	☐	☐ BINARY		
host	VARCHAR(20)	☒	☐	☐ BINARY		
facility	VARCHAR(20)	☒	☐	☐ BINARY		
guests	INTEGER	☒	☐	☒ UNSIGNED ☐ ZEROFILL		
caterer	VARCHAR(20)	☒	☐	☐ BINARY		

You can use the auto increment feature for assigning IDs.

Your Party Planner application in the middle tier will communicate with the database in the third tier at the Party Planner to store and retrieve information.

To process MySQL commands you will need to provide a suitable driver for the Java SQL libraries. You will be provided the necessary JDBC driver. You can then use the Java SQL libraries by importing:

```
import java.sql.*;
```

Check out the template available at the class Web site and work with that file.

You might find it helpful to use the following applications to manipulate and query the MySQL database; links to these applications are also available at the class Web site.

- *MySQL Administrator*
- *MySQL Query Browser*

Design

In database applications, it is good practice to use a layered design. That is, have a class that is totally dedicated to querying and another class that deals with updates (with the help of the other class). For example:

PartyPlanner object:

boolean AddClient (Client)	{... Using PartyPlannerDB object, call methods}
boolean DelClient (Client)	{... Using PartyPlannerDB object, call methods}
boolean UpdateClient (Client)	{... Using PartyPlannerDB object, call methods}
Vector ListClients ()	{... Using PartyPlannerDB object, call methods}
Vector GetParties ()	{... Using PartyPlannerDB object, call methods}

PartyPlannerDB object:

```
boolean AddClient (Client)
boolean DelClient (Client)
boolean UpdateClient (Client)
Vector<Client> ListClients()
Vector<Party> GetParties()
```

Here the server does not need to know how queries are sent to the database; the implementation details are in the PartyPlannerDB class. When there is a change in the database system being used (for example, Oracle or DB2 instead of MySQL), the change will affect only the PartyPlannerDB class. Moreover, if you change the design for the server (for example, so that it uses Java Server Pages (JSPs) instead of a Java application), you can still use the PartyPlannerDB class. Again, the server does not need to know how to interact with the database, it can achieve the interactions with the help of the PartyPlannerDB class.

If you implement Project 4 like this, you will not have to do much coding related to the database in Project 5. For this reason, it is very important to understand and use a layered design.

On the class Web site for this project, you will find a template for the PartyPlannerDB class.

Report

Your report will count 20% of your grade for this project.

The report should be clear, well organized, and concise. It needs to include the following:

- 1) Database table structures
- 2) All MySQL commands [*Table Create, Insert, Delete, Update, Query*]
- 3) Details of how your implementation works
- 4) A diagram of the structure of your database, its tables, records and fields.

Submission (10%)

You will submit a compressed file (*tar*). You will need to type “*ant submission*” in order to create your submission file. If there is anything missing in the submission file, edit the *build.xml* file to include the missing information. Make sure you modify the *User* property in the *build.xml* file. The *ant* call will create submission file with the username you provide.

The compressed file should include a folder with all project content necessary to build/run the program (.java, README, Report). The projects are to be sent to ece155b@gmail.com with subject line “Project 04: FirstName LastName”.