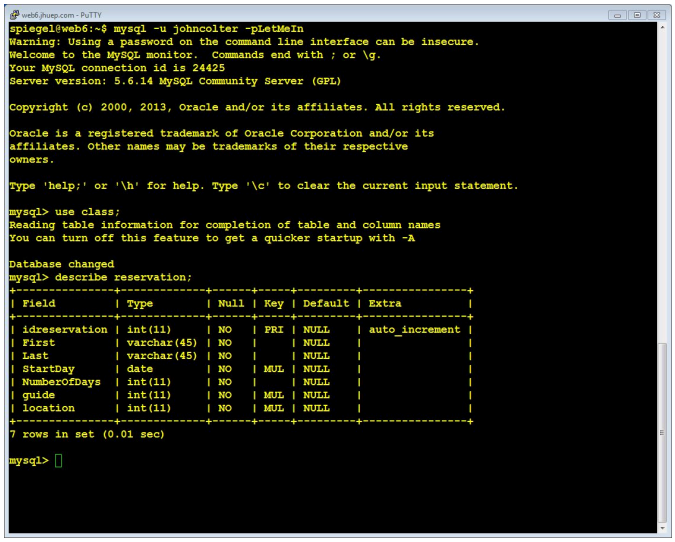
Homework for JDBC

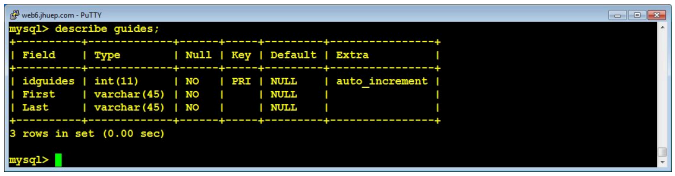
Assignment:

In this assignment we will deviate from the previous work of requesting quotes for tours and develop and MVC model (with new page/form) to query existing tour information from the Bryce Canyon Hiking Company database.

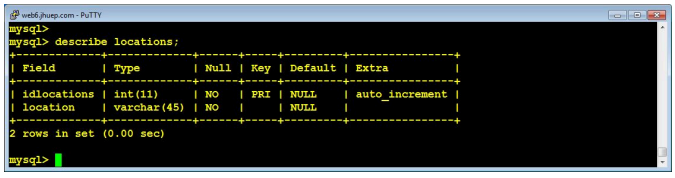
Bryce Canyon Hiking Company has stored all of its booking data on a MySQL database on web6.jhuep.com. They have configured the database to contain three separate tables. The first, reservation, contains specific tour reservation data with references to the guides hired for those tours as well as the locations of the tours. The following figure provides detail for the table:



The previous figure also contains the connection information for the database on web6.jhuep.com (not web3 where we will be doing our work). The username and password for read access to the database are in the figure as well. The second table, guides, contains the guides first and last names and is illustrated below:



The final table, locations, contains a list of the tour locations and is illustrated here:



BHC would like it's employees to be able to view this database from a web page. They would like you to create an MVC web application with a web page that lets users view all tours in the database that are booked on or after a given reference date (using a form).

You will need to create a JSP with a form that will accept a year, month and day, and then query the database. The data returned should be formatted into an html table and shown to the user. The displayed table should contain the starting date, the number of days, the location of the tour, the guides first and last names, and the first and last name on the reservation in chronological order. Note that these results are based on a query of the start day only, **there is no end day** to consider.

This project is distinct from our previous work of getting quotes. As such you will not have the same field contstraints. Specifically, your client should handle **ANY** date in a client request and return the appropriate response (which can consist of **NO RECORDs**).

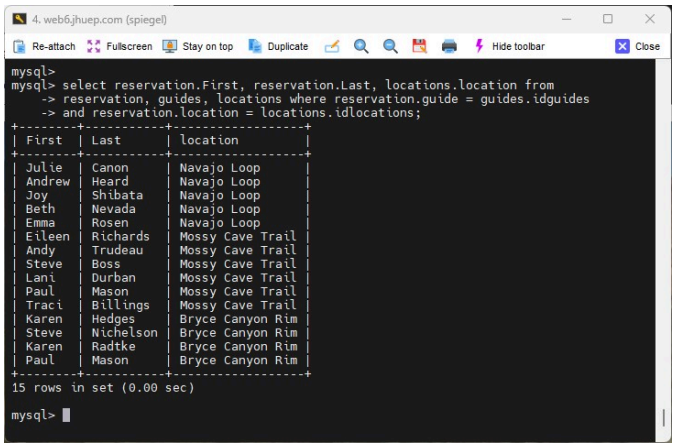
Oh, and of course you'll need to connect to the database. You can manually connect to the database if you are logged in to the class computer with the parameters as in the command:

```
/usr/bin/mysql -h web6.jhuep.com -u johncolter -pLetMeIn class
```

The gives you read-only access to the database.

To actually develop your application, you'll need to make a new database connection in your code.

While there are many ways to acquire data from the database and process it in Java, you are expected to get the data is a useable form that does not require much (if any) manipulation in Java. A database is optimized for acquisition and manipulation of data, so gettting data from each table and then combining it and sorting it with Java is extremely sub-optimal. You need to use a SQL query that get's all of the data at once and delivers it in a sorted fashion (aka use JOINS). Here is an example (not applicatable to the assingment) of a query that combines information from multiple tables:



As an MVC application, your work should follow responsibilities of MVC. JSP pages for display only (no business logic other than that necessary for collecting data from a user or from the server or for display), a servlet to appropriately route requests, Javabeans for sharing data between components, and other Java classes for server side logic (aka database logic).

Note that anything you share with a JSP should have no dependency on the database or JDBC classes. All of the database code should reside on the server, and the Javabean should be self-contained objects for sharing data only.

Test Cases

Here is a minimal set of test cases that you must handle correctly in your code. I will leave it to you to determine what the result should be (valid or invalid) for most of them. Depending upon your choice of input I will also be testing other forms of errant input as well, like missing fields or bad data (characters for numbers, etc). As I already said I will be testing through your HTML client as well as direct input to your servlet (avoiding your HTML).

Requested Date	Records In Result/Error
12-21-2010	15
8-1-2015	10
8-1-2018	6
6-1-2021	0
8-62-2017	Bad Date
Bad text in some or all fields	Indicator of field with bad text
Missing fields	General missing fields error

Submission:

Once you have completed your application, "build" it and upload the WAR file to the app server on the course computer. Make sure you properly "embed" your jdbc connector inside the web application or it will fail to connect to the database.

Upload to Canvas:

1. The URL of your web application in the submission comment on Canvas. You must not forget this!
2. An appropriately named zip file of your entire web application project

Grading:

Use the following breakdown as a GUIDELINE ONLY for general weighting of different parts of the assignment. Specific grading may not completely fall into the breakdown below:

1. Reasonable code design/Coding Guidelines/General Grading Guidelines 15%
2. Error Messages (Complete and Accurate) from both the server and client side 10%
3. Application runs from class server and passes all test cases laid out above 25%
4. MVC Architecture using Servlet, JSPs and Javabeans 25%
5. Use of JDBC including API and a reasonable SQL query for display data 25%