# CIS 3308 Metadata Challenge

## *Overview:*

For this challenge, you will write two pages (outlined below) that perform some of the functions of report writer software. A "Report Writer" is software that connects to any standard type of database, extracts metadata, and allows users (and/or developers) to specify reports.

## *Background Information:*

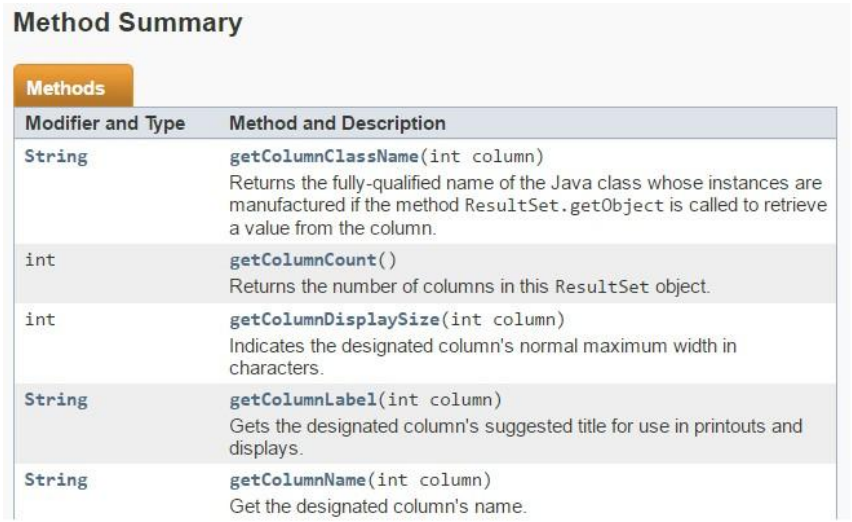
Databases contain two kinds of data:

- Metadata: specifies the database design (e.g., table names, column names, relationships, possibly more description about what should do into a particular column) and
- User data: data that is in the tables that are specified by Metadata.

As you know (from the "display data" lab), when we execute a sql select statement, we get a java.sql.ResultSet which (if you iterate using it's "next()" method) provides rows of columnar data. From a ResultSet, you can extract ResultSetMetaData that describes the column names, data types etc of all the columns that were selected.

Google "java resultset metadata javadocs" and you should come up with a link and information similar to this:

https://docs.oracle.com/javase/7/docs/api/java/sql/ResultSetMetaData.html

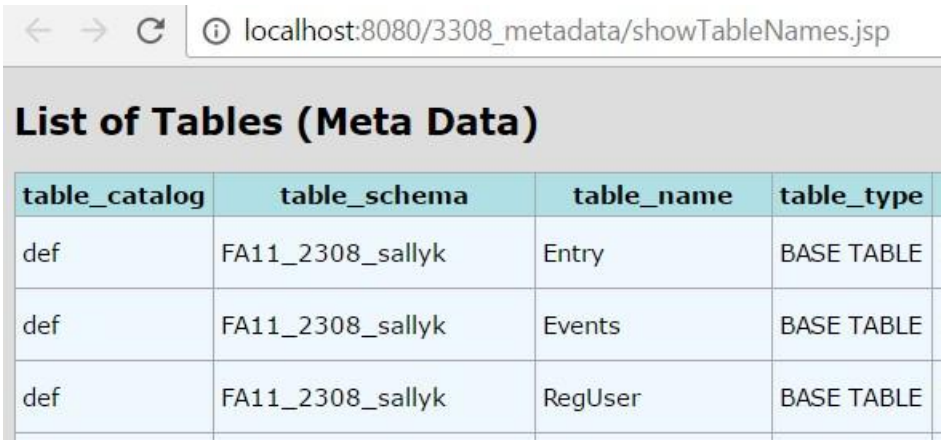




If you are not familiar with googling "javadocs", you should learn because this is a very important resource for java programmers.

*Sample Code:*

1. showTableNames.jsp: shows all the tables that are in a database schema. With this SQL, we can extract the metadata shown below.

```
String sql = "SELECT * FROM information_schema.tables WHERE table_schema = ?";
PreparedStatement st = dbc.getConn().prepareStatement(sql);
st.setString(1, dbc.getDbName( ));
```




**List of Tables (Meta Data)**

| table_catalog | table_schema | table_name | table_type |
|---|---|---|---|
| def | FA11_2308_sallyk | Entry | BASE TABLE |
| def | FA11_2308_sallyk | Events | BASE TABLE |
| def | FA11_2308_sallyk | RegUser | BASE TABLE |

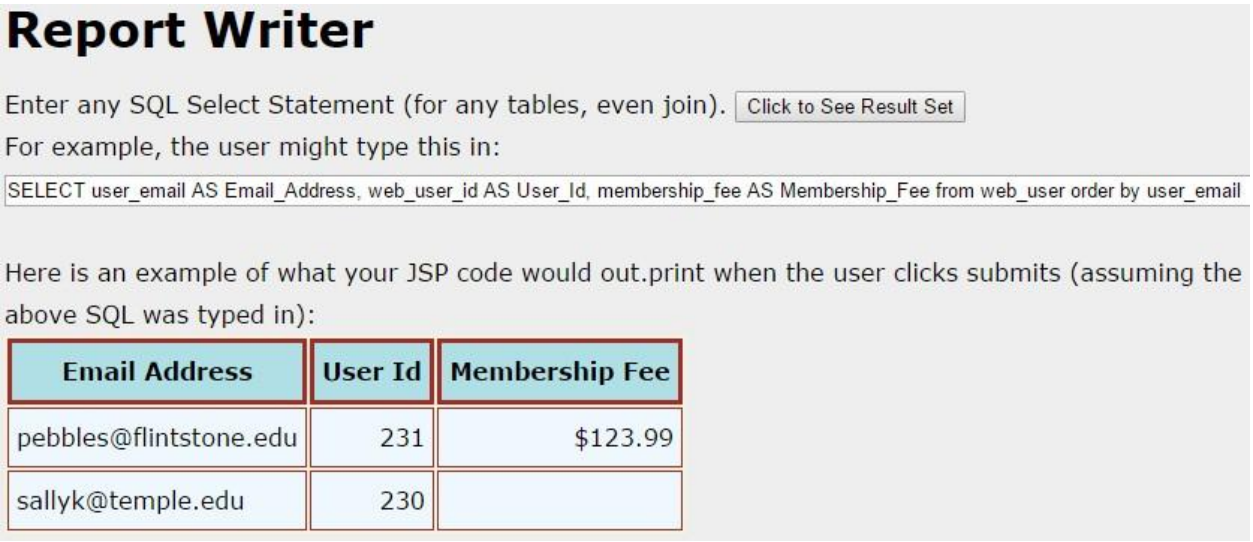
2. showMetaData.jsp: shows all the column information about a result set. For example, if the SQL was "SELECT * FROM web_user", we might extract this metadata from the result set:

| column name | type | display size | precision | scale | autoincrement |
|---|---|---|---|---|---|
| web_user_id | INT | 11 | 11 | 0 | true |
| user_email | VARCHAR | 45 | 45 | 0 | false |
| user_password | VARCHAR | 45 | 45 | 0 | false |
| birthday | DATE | 10 | 10 | 0 | false |
| membership_fee | DECIMAL | 10 | 8 | 2 | false |
| user_info | VARCHAR | 2147483647 | 2147483647 | 0 | false |
| user_role_id | INT | 11 | 11 | 0 | false |

*Requirements:*

- listFields.jsp: This page shall provide the user with a picklist (select tag) of all the tables that currently exist (that you designed) in your database. Use a SQL select statement similar to showTableNames.jsp (sample code) to extract the names of all the tables from your database schema. When the user selects one of the tables from the list (and clicks submit), they shall see metadata information for all the columns of that table. The output of this page shall look similar to the showMetaData.jsp page (from the sample code).

- report.jsp: This page shall accept a SQL select statement and (upon submit) execute the select statement and then show either (1) the formatted HTML table generated from the result set or (2) the error message provided by the database. All columns shall be formatted to the correct data type (e.g., formatting decimals as dollar amounts with commas, formatting integers with commas, dates displaying nicely as you would expect, numbers right aligned, etc). When the user enters the select statement, they should alias column names to have the values they would like to see displayed as column headings, in the <th> tags. The code shall replace any underscore character (in result set column names) with a space. For example:




If you were to View Source on the above page, you'd see something like this:

```
<tr>
        <td style='text-align:left;'>pebbles@flintstone.edu</td>
        <td style='text-align:right;'>231</td>
        <td style='text-align:right'>$123.99</td>
</tr>
```

You can use FormatUtils, a class (already provided in sample code for previous labs) to nicely check for nulls and format data appropriate to its type. You only need to implement these data types: int, varchar, date, and decimal.

*Submission:*

Implement both JSP pages and the associated java classes (using good programming practices as described in the "requirements for all labs and project" section of the 3308 labs page). Make a blog entry in your labs page that links to both pages from this challenge. Publish your web application and submit a zip file into the “metadata challenge” blackboard assignment. The code for your challenge shall be integrated with your web application project that includes all the work from all the labs (not be a separate NetBeans project).