CS313: Lab Assignment 5

B Siddharth Prabhu

`200010003@iitdh.ac.in`

01 October 2022

Part A

In this part, we set up JDBC using Eclipse and try to run queries that make database calls to MySQL. Here, we mainly focus on linking the Java program to the MySQL Server. (JDBC is the API that allows Java programs to access database management systems such as MySQL.)

Note that: When running `Driver.java`, all the 3 Questions of Part A run in sequence. While Q1 and Q2 simply display outputs, Q3 requires user input for `semester`, `year` and `building`, after which the required output is displayed.

Submission for Part A: `Driver.java`

1 Running Java+JDBC Code

Using `Driver.java`, we shall interact with the MySQL 'university' database, that we created and modified in prior assignments.

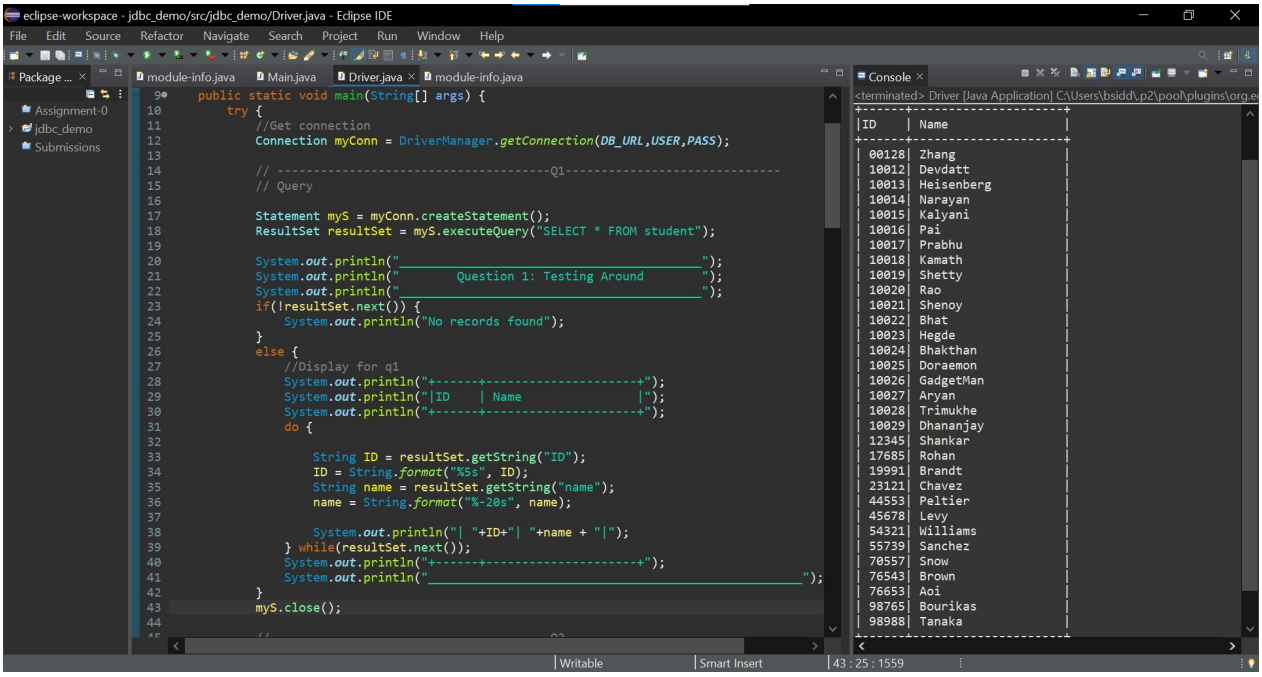
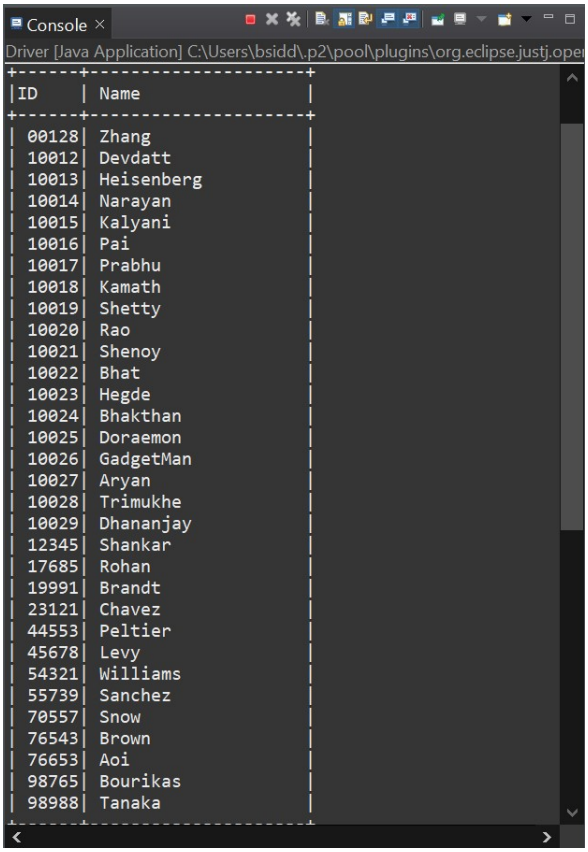


Figure 1: Picture of Setup + Output of Q1

For a closer look at the output, an image is available on the next page.

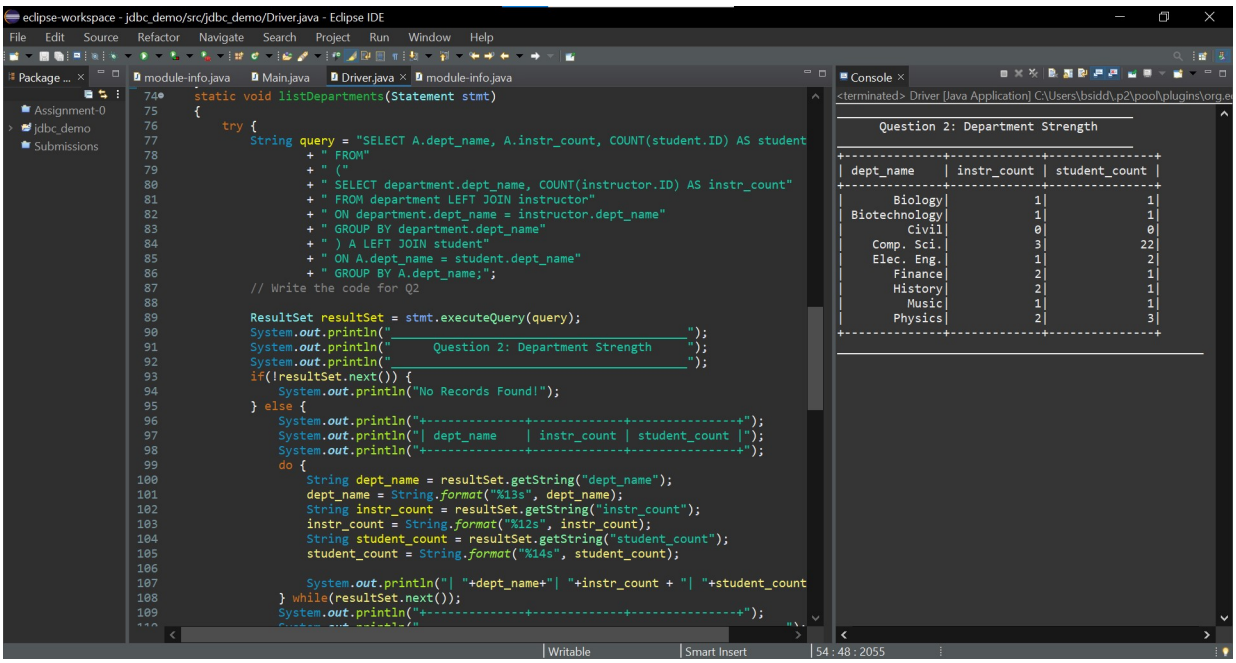


```
Console ×
Driver [Java Application] C:\Users\bsidd\.p2\pool\plugins\org.eclipse.justj.oper
+------+---------------------+
|ID    | Name                |
+------+---------------------+
| 00128| Zhang               |
| 10012| Devdatt             |
| 10013| Heisenberg          |
| 10014| Narayan             |
| 10015| Kalyani             |
| 10016| Pai                 |
| 10017| Prabhu              |
| 10018| Kamath              |
| 10019| Shetty              |
| 10020| Rao                 |
| 10021| Shenoy              |
| 10022| Bhat                |
| 10023| Hegde               |
| 10024| Bhakthan            |
| 10025| Doraemon            |
| 10026| GadgetMan           |
| 10027| Aryan               |
| 10028| Trimukhe            |
| 10029| Dhananjay           |
| 12345| Shankar             |
| 17685| Rohan               |
| 19991| Brandt              |
| 23121| Chavez              |
| 44553| Peltier             |
| 45678| Levy                |
| 54321| Williams            |
| 55739| Sanchez             |
| 70557| Snow                |
| 76543| Brown               |
| 76653| Aoi                 |
| 98765| Bourikas            |
| 98988| Tanaka              |
+------+---------------------+
```

Figure 2: Output of Q1

2 Querying with JDBC

We want to list departments (in ascending order) and the total number of students and instructors they have.



```
eclipse-workspace - jdbc_demo/src/jdbc_demo/Driver.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package ...
Assignment-0
jdbc_demo
Submissions
module-info.java   Main.java   Driver.java ×   module-info.java

74    static void listDepartments(Statement stmt)
75    {
76        try {
77            String query = "SELECT A.dept_name, A.instr_count, COUNT(student.ID) AS student
78                    + " FROM"
79                    + " ("
80                    + " SELECT department.dept_name, COUNT(instructor.ID) AS instr_count"
81                    + " FROM department LEFT JOIN instructor"
82                    + " ON department.dept_name = instructor.dept_name"
83                    + " GROUP BY department.dept_name"
84                    + " ) A LEFT JOIN student"
85                    + " ON A.dept_name = student.dept_name"
86                    + " GROUP BY A.dept_name;";
87            // Write the code for Q2
88
89            ResultSet resultSet = stmt.executeQuery(query);
90            System.out.println("____________________________________________");
91            System.out.println("      Question 2: Department Strength       ");
92            System.out.println("____________________________________________");
93            if(!resultSet.next()) {
94                System.out.println("No Records Found!");
95            } else {
96                System.out.println("+--------------+------------+---------------+");
97                System.out.println("| dept_name    | instr_count | student_count |");
98                System.out.println("+--------------+------------+---------------+");
99                do {
100                   String dept_name = resultSet.getString("dept_name");
101                   dept_name = String.format("%13s", dept_name);
102                   String instr_count = resultSet.getString("instr_count");
103                   instr_count = String.format("%12s", instr_count);
104                   String student_count = resultSet.getString("student_count");
105                   student_count = String.format("%14s", student_count);
106
107                   System.out.println("| "+dept_name+"| "+instr_count + "| "+student_count
108               } while(resultSet.next());
109               System.out.println("+--------------+------------+---------------+");

Console ×
<terminated> Driver [Java Application] C:\Users\bsidd\.p2\pool\plugins\org.e
____________________________________________
      Question 2: Department Strength
____________________________________________
+--------------+------------+---------------+
| dept_name    | instr_count | student_count |
+--------------+------------+---------------+
|      Biology|           1|              1|
| Biotechnology|           1|              1|
|        Civil|           0|              0|
|   Comp. Sci.|           3|             22|
|   Elec. Eng.|           1|              2|
|     Finance|           2|              1|
|     History|           2|              1|
|       Music|           1|              1|
|     Physics|           2|              3|
+--------------+------------+---------------+
____________________________________________

Writable   Smart Insert   54 : 48 : 2055
```

Figure 3: Picture of Setup + Output of Q2

For a closer look at the output, an image is available on the next page.

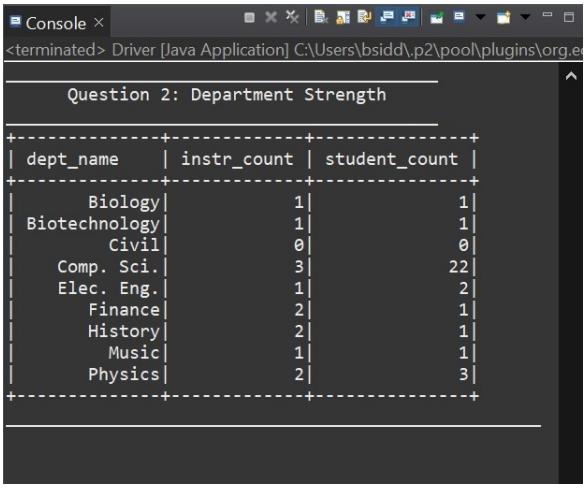


Figure 4: Output of Q2

3 Using PreparedStatement

We want to, for a given building, find classrooms (room_no) with capacity more than 30 and in which no courses are scheduled this year and semester.

Input taken from user: semester, year, building

This is structured as such because "this year" and "this semester" don't make sense in the context of university database. If it really does mean Fall 2022, the same can be entered as input during execution.

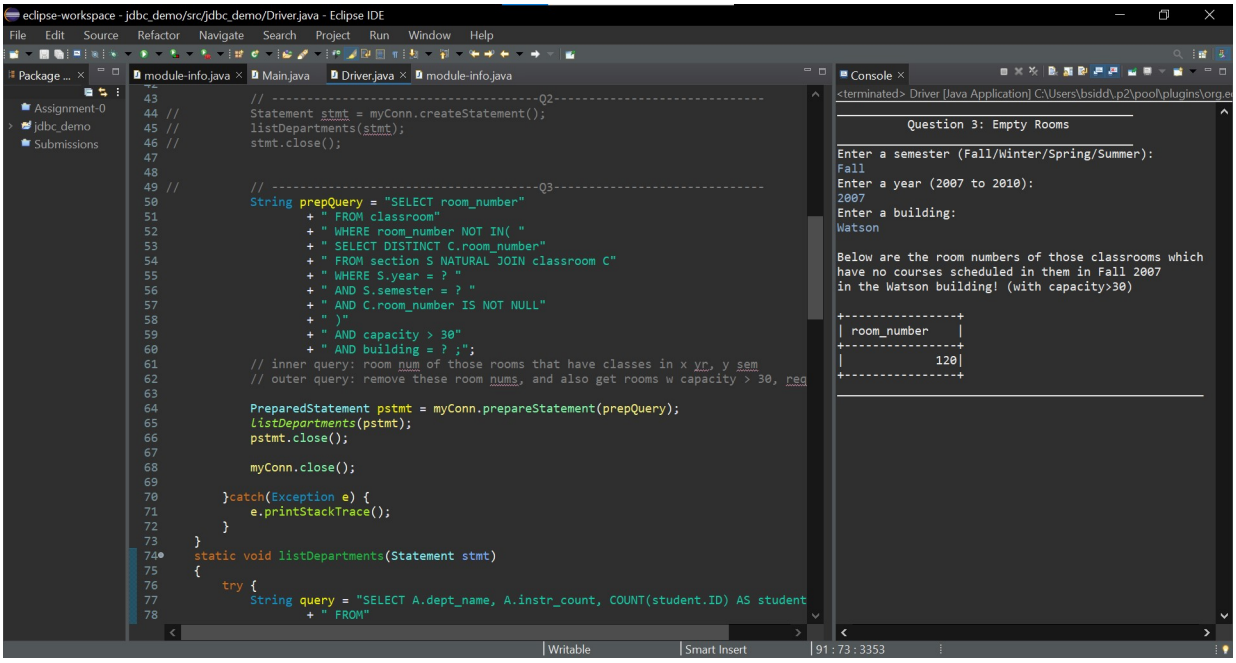


Figure 5: Picture of Setup + Output of Q3

For a closer look at the output, an image is available on the next page.

```
Console ×
<terminated> Driver [Java Application] C:\Users\bsidd\.p2\pool\plugins\org.e
_____________________________________________
          Question 3: Empty Rooms
_____________________________________________
Enter a semester (Fall/Winter/Spring/Summer):
Fall
Enter a year (2007 to 2010):
2007
Enter a building:
Watson

Below are the room numbers of those classrooms which
have no courses scheduled in them in Fall 2007
in the Watson building! (with capacity>30)

+----------------+
| room_number    |
+----------------+
|             120|
+----------------+
_____________________________________________
```

Figure 6: Output of Q3

```
Console ×
<terminated> Driver [Java Application] C:\Users\bsidd\.p2\pool\plugins\org.e
_____________________________________________
          Question 3: Empty Rooms
_____________________________________________
Enter a semester (Fall/Winter/Spring/Summer):
Fall
Enter a year (2007 to 2010):
2009
Enter a building:
Packard

Below are the room numbers of those classrooms which
have no courses scheduled in them in Fall 2009
in the Packard building! (with capacity>30)


No records found!!
```

Figure 7: Another Output of Q3

Part B : next page ⇒

Part B

In this part, we use J2EE to set up a simple Web Application. The Web App takes Advisor ID as input, and produces the department name of that particular advisor as output. This is like Part A, but with an end-user web interface.

1 Advisor Servlet

We would like to design a new html page to take advisor id as input, and write a servlet to display the department to which the advisor belongs using the Java and J2EE program. The final output should contain advisor id and department name. **Submission for Part B:** `index.html` , `AdvisorServlet.java`

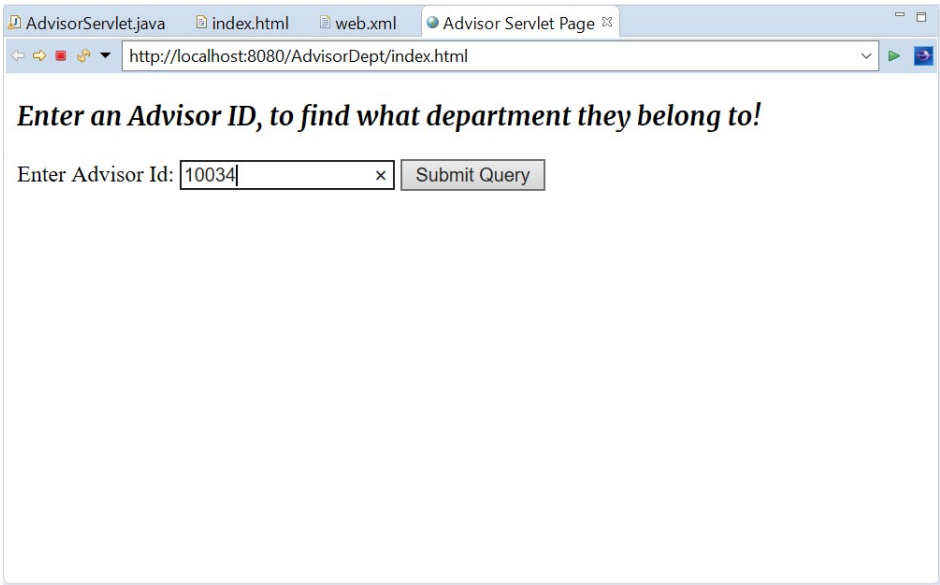


Figure 8: Screenshot of `index.html`

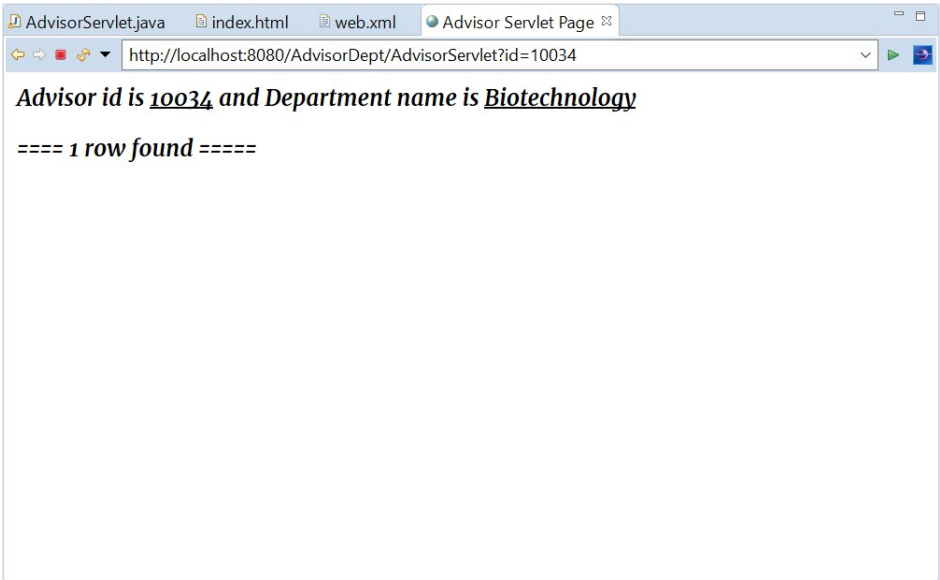


Figure 9: Screenshot of final output

Note: `web.xml` file should contain the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
<servlet>
<servlet-name>AdvisorServlet</servlet-name>
<servlet-class>AdvisorServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AdvisorServlet</servlet-name>
<url-pattern>/AdvisorServlet</url-pattern>
</servlet-mapping>
</web-app>
```