

Prompts

1. What is JDBC? What does it stand for? How does it differ from or relate to MySQL?
2. Give an example of the primary JDBC Classes? What Classes do we use to do the following, and give an example of syntax to accomplish the following:
 - (a) Attach or Connect to a database
 - (b) Create a statement (to perform one of the CRUD operations on a database)
 - (c) Execute a query
 - (d) Process the data retrieved,
 - (e) Close the connection to the database
3. What are the two method calls to execute a query in JDBC, and when do you use each?

Purpose

- This Research Assignment is meant to go beyond the course curriculum and increase your understanding of relevant topics while exposing you to online resources you'll frequently use on the job.
- As developers, **research** is a constant part of our job.
- A common saying is that 90% of software development is "Googling". While that is an exaggeration, search engines are highly utilized tools in the software development role.
- The **Java Documentation** is a great place to start, and there are so many other great resources!

Instructions

- Choose at least **two (2)** of the above prompts.
- Research those prompts.
- Write at least one paragraph for **two (2)** of the above prompts
- Include URLs from where you found the information to cite your sources.
- Create a PDF, and Submit that to the LMS.

Do NOT copy and paste text from the internet or any other source!

- **Use** the information you find in your research as a starting point
- **Summarize the concepts, in your own words.**
- Plagiarism will result in a zero for the assignment as well as disciplinary actions.

This assignment is graded based on participation.

1. What is JDBC? What does it stand for? How does it differ from or relate to MySQL?

<https://www.geeksforgeeks.org/introduction-to-jdbc/>

<https://www.talend.com/resources/what-is-mysql/>

The acronym JDBC stands for, Java Database Connectivity, and it is a way to connect to a database. It is a way to access the database while performing a query via Java program or application. This has been showcased in our class via Dr. Rob's lessons, he demonstrates with the Recipe Schema how to access it from Java and how to write information to it via the application as well. This is different from MySQL because, MySQL is a relational database management system (RDBMS). It is a place to hold or store data and it is retrieved in a relational way by using the query language called SQL (Structured Query Language). While JDBC serves to connect to MySQL, and query data from it, JDBC does not store data.

2. What are the two method calls to execute a query in JDBC, and when do you use each? There are two commands or way to write a query via JVM using a JDBC, one method is called `executeQuery()` and the other is called `executeUpdate()`. For the former, we assume that a successful connection has been made without error, the confirmation of that connection then allows the successful run command of `executeQuery()` which then lead to the executable SQL statement one may have entered in. To update a row in the database, you can use the call of `executeUpdate()`, of course this is still dependent of a successful connection to the database. Below is an example of getting a connection to a database, and running the `executeUpdate()`; command after receiving and input from a user with a new project. The items being entered in are everything inside the try statement parentheses, these are the values that we are gathering.

```

39 public Project insertProject(Project project) {
40     //@formatter: off
41     String sql = ""
42         + "INSERT INTO " + PROJECT_TABLE + " "
43         + "(project_name, estimated_hours, actual_hours, difficulty, notes) "
44         + "VALUES"
45         + "(?, ?, ?, ?, ?)";
46     //@formatter: on
47
48     try(Connection conn = DbConnection.getConnection()){
49         startTransaction(conn);
50
51
52         try(PreparedStatement stmt = conn.prepareStatement(sql)){
53             setParameter(stmt, 1, project.getProjectName(), String.class);
54             setParameter(stmt, 2, project.getEstimatedHours(), BigDecimal.class);
55             setParameter(stmt, 3, project.getActualHours(), BigDecimal.class);
56             setParameter(stmt, 4, project.getDifficulty(), Integer.class);
57             setParameter(stmt, 5, project.getNotes(), String.class);
58
59             stmt.executeUpdate();
60
61             Integer projectId = getLastInsertId(conn, PROJECT_TABLE);
62             commitTransaction(conn);
63
64             project.setProjectId(projectId); //don't know what is going on here
65             return project;
66         }
67         catch(Exception e) {
68             rollbackTransaction(conn);
69             throw new DbException(e);
70         }
71     }
72     catch(SQLException e) {
73         throw new DbException(e);
74     }
75 }
76 }

```

3.