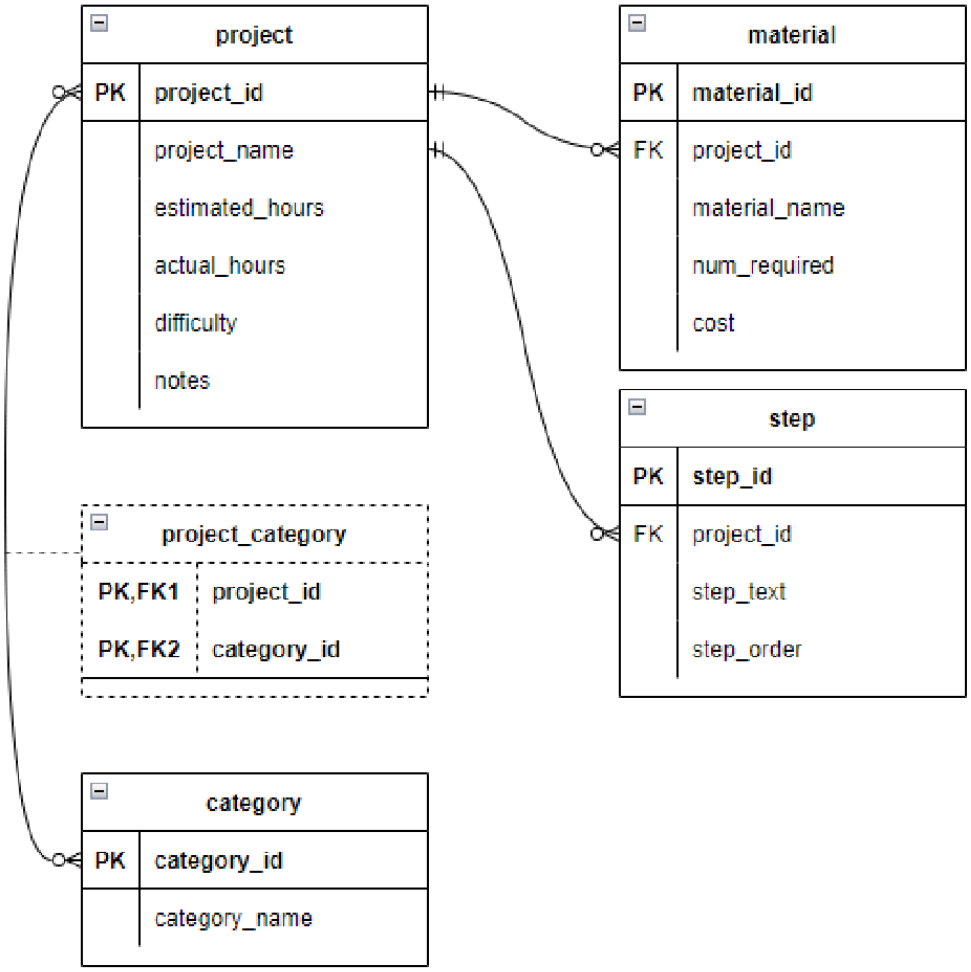
Background

The weekly exercises are designed to augment the video lessons. In the exercises, you will develop a menu-driven application in Java. This application will demonstrate how to perform CRUD (Create, Read, Update, and Delete) operations on a MySQL database.

You will be working in a Project schema (database) that contains do-it-yourself (DIY) projects. A DIY project contains project details, materials, steps, and categories. Below is a diagram of the tables and relationships in the Project schema. Don't worry at this point if you don't understand what the diagram is telling you. This will become clear soon. For now, just know that there are five tables in the Project schema: project, material, step, category, and project_category. This is what you will build in the



exercises.

There will be a final project in this (MySQL) part of the back-end course. These exercises will help prepare you for that.

Instructions

Points possible: 75

URL to GitHub Repository:https://github.com/bman354/Mysql-week1

URL to Public Link of your Video:https://youtu.be/PTe35SXdNGc

---

Instructions:

1. Follow the **Exercises** below to complete this assignment.
    - In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed.
    - Create a new repository on GitHub for this week's assignment and push your completed code to this dedicated repo, including your entire Maven Project Directory (e.g., mysql-java) and any .sql files that you create.
    - Include the screenshots into this Assignment Document indicated by: ☐ 📷
    - Create a video showcasing your work:
        - In this video: record and present your project verbally while showing the results of the working project.
        - Easy way to Create a video: Start a meeting in Zoom, share your screen, open Eclipse with the code and your Console window, start recording & record yourself describing and running the program showing the results.
        - Your video should be a maximum of 5 minutes.
        - Upload your video with a public link.
        - Easy way to Create a Public Video Link: Upload your video recording to YouTube with a public link.
2. In addition, please include the following in your Coding Assignment Document:
    - The requested screenshots, indicated by: ☐ 📷
    - **The URL for this week's GitHub repository.**
    - **https://github.com/bman354/Mysql-week1**
    - **The URL of the public link of your video.**
    - **https://youtu.be/PTe35SXdNGc**
3. Save the Coding Assignment Document as a .pdf and do the following:
    - Push the .pdf to the GitHub repo for this week.
    - Upload the .pdf to the LMS in your Coding Assignment Submission.

Objectives

In these exercises, you will:

Use MySQL Workbench to create a schema and user.

Use MySQL Workbench to assign schema privileges to a user.

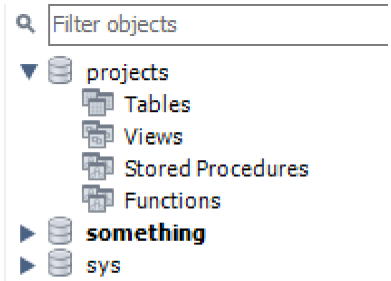
Create a Maven project in Eclipse.

Add the MySQL driver as a dependency in pom.xml (Maven's Project Object Model – POM).

Separate project concerns by creating packages.

Write Java code to connect to a MySQL database and schema.

My Screenshots and Solutions



Schema	Privileges
projects	ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, REFERENC

mysql-week1-assignment/pom.xml

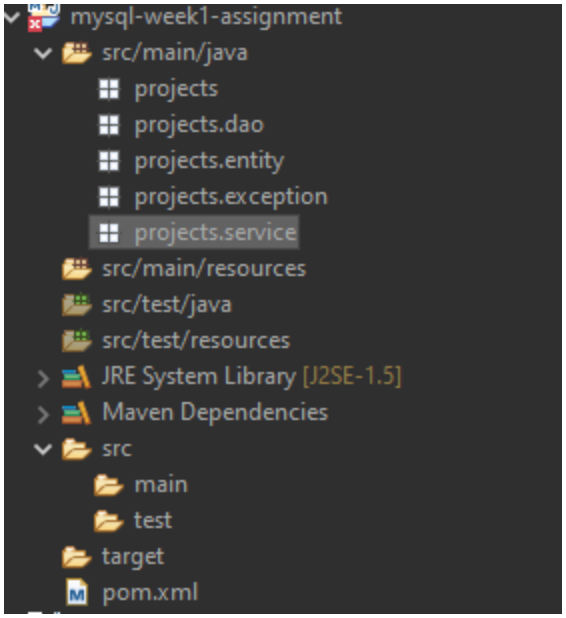
```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.promineotech</groupId>
  <artifactId>mysql-week1-assignment</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <properties>
    <java.version>17</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.30</version>
    </dependency>
  </dependencies>

  <build>
    <pluginManagement>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.10.1</version>
          <configuration>
            <source>${java.version}</source>
            <target>${java.version}</target>
          </configuration>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>


</project>
```


mysql-week1-assignment
src/main/java
projects
projects.dao
projects.entity
projects.exception
projects.service
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
main
test
target
pom.xml

DbException.java

```java
package projects.exception;

import java.lang.RuntimeException;

@SuppressWarnings("serial")
public class DbException extends RuntimeException {
    public DbException(String message) {
        super(message);
    }

    public DbException(Throwable cause) {
        super(cause);
    }

    public DbException(String message, Throwable cause) {
        super(message, cause);
    }
}
```

DbException.java DbConnection.java

```java
package projects.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import projects.exception.DbException;

public class DbConnection {
    final static String HOST = "localhost";
    final static String PASSWORD = "projects";
    final static int PORT = 3306;
    final static String SCHEMA = "projects";
    final static String USER = "projects";

    public static Connection getConnection() {
        String uri = SCHEMA + "://" + USER + "@" + HOST + ":" + PORT;

        try {
            System.out.println("Attempting to connect");
            Connection connection = DriverManager.getConnection(uri);
            System.out.println("Connection was successful at " + uri);
            return connection;
        } catch (SQLException exception) {
            System.out.println("Could not connect at " + uri);
            throw new DbException("Could not connect at " + uri);
        }
    }
}
```

DbException.java DbConnection.java ProjectsApp.java

```java
package projects;

import projects.dao.DbConnection;

public class ProjectsApp {

    public static void main(String[] args) {

        DbConnection.getConnection();

    }

}
```

Console

<terminated> ProjectsApp [Java Application] C:\Program Files\Java\jdk-17.0.3.1\bin\javaw.exe (Oct 22, 2022, 6:02:15 PM – 6:02:16 PM) [pid: 8496]

```
Attempting to connect
Connection was successful at jdbc:mysql://localhost:3306/projects?user=projects&password=projects
```