

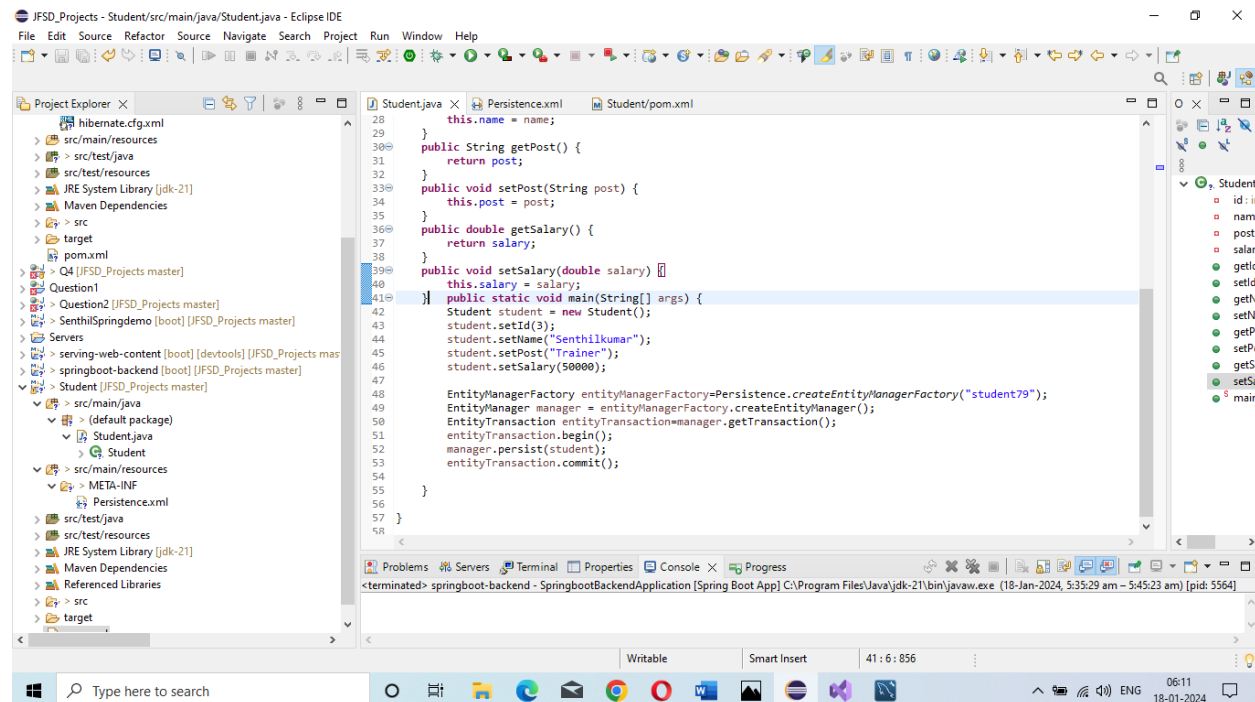
# Java Full Stack application with Hibernate CRUD operations

Dr.R.Senthilkumar, Government College of Engineering Erode.

Dt.4.1.2024

Email: [rsenthil\\_1976@yahoo.com](mailto:rsenthil_1976@yahoo.com), [rsenthil.optical@gmail.com](mailto:rsenthil.optical@gmail.com)

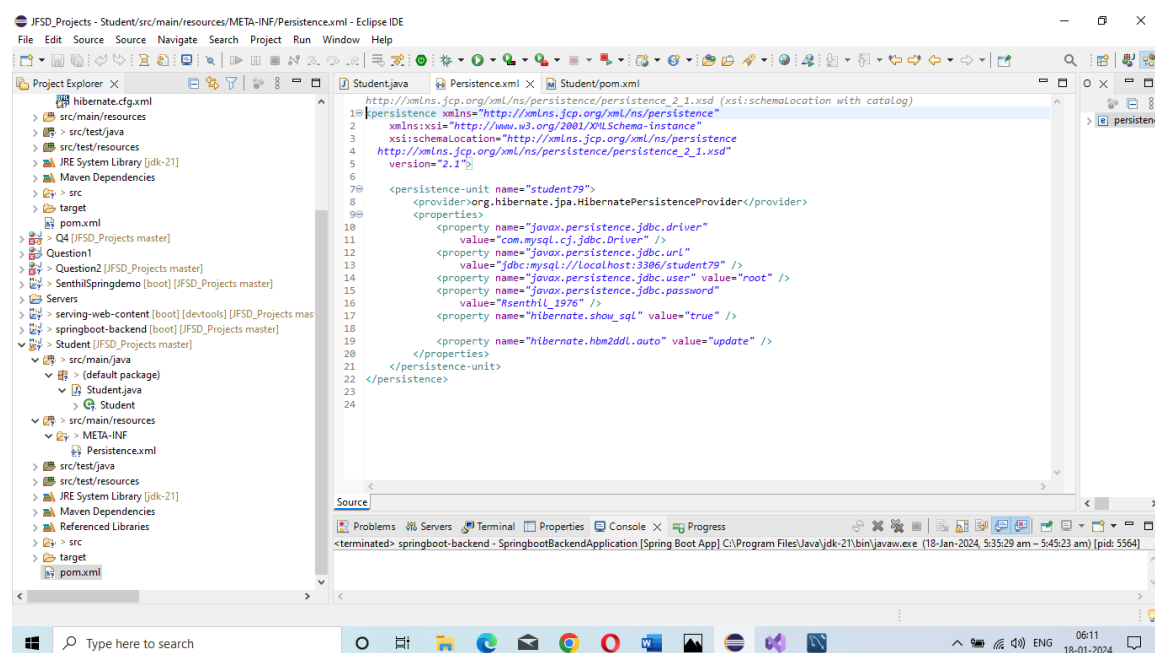
## Student.java



The screenshot shows the Eclipse IDE with the 'Student.java' file open. The code defines a 'Student' class with attributes 'name', 'post', and 'salary'. It includes methods for getters, setters, and a static 'main' method. The 'main' method creates a 'Student' object, sets its attributes, and uses an 'EntityManager' to persist the object. The 'Project Explorer' on the left shows the project structure, including 'src/main/java', 'src/main/resources', and 'META-INF'. The 'Console' at the bottom shows the application running successfully.

```
28 this.name = name;
29
30 public String getPost() {
31     return post;
32 }
33 public void setPost(String post) {
34     this.post = post;
35 }
36 public double getSalary() {
37     return salary;
38 }
39 public void setSalary(double salary) {
40     this.salary = salary;
41 }
42 public static void main(String[] args) {
43     Student student = new Student();
44     student.setId(3);
45     student.setName("Senthilkumar");
46     student.setPost("Trainer");
47     student.setSalary(50000);
48
49     EntityManagerFactory entityManagerFactory = Persistence.createEntityManagerFactory("student79");
50     EntityManager entityManager = entityManagerFactory.createEntityManager();
51     EntityTransaction entityTransaction = entityManager.getTransaction();
52     entityTransaction.begin();
53     manager.persist(student);
54     entityTransaction.commit();
55 }
56
57 }
```

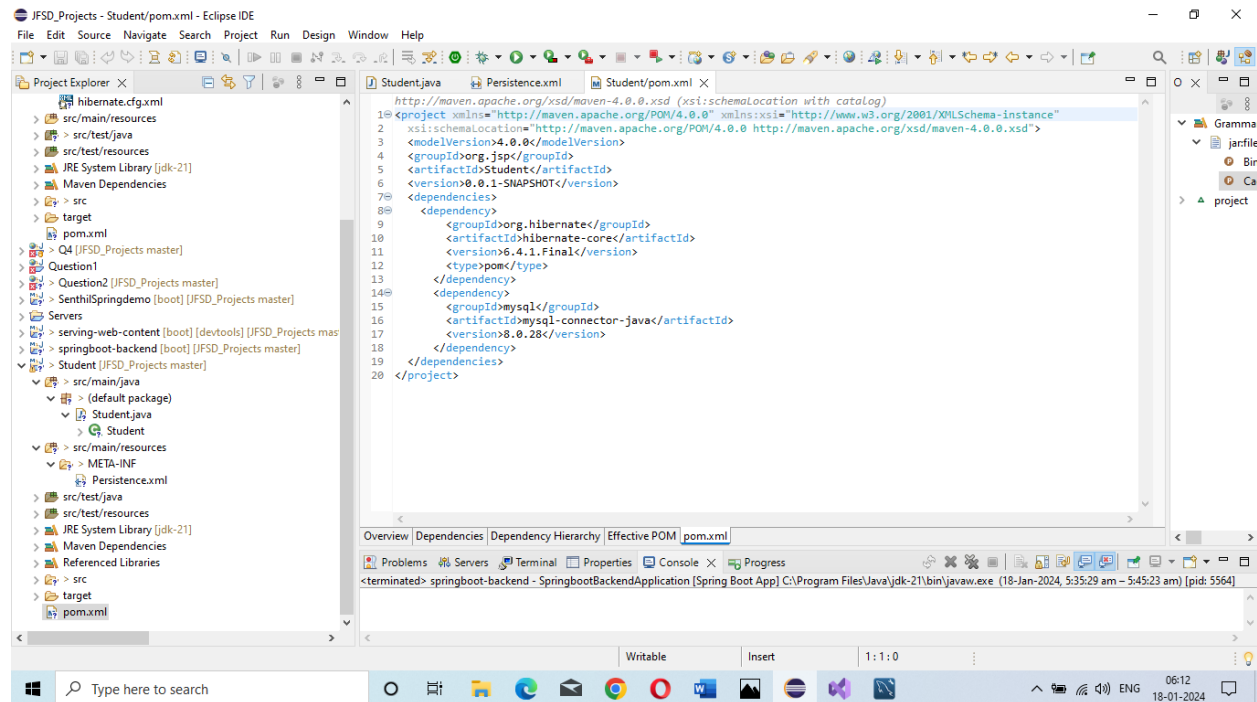
## Persistence.xml



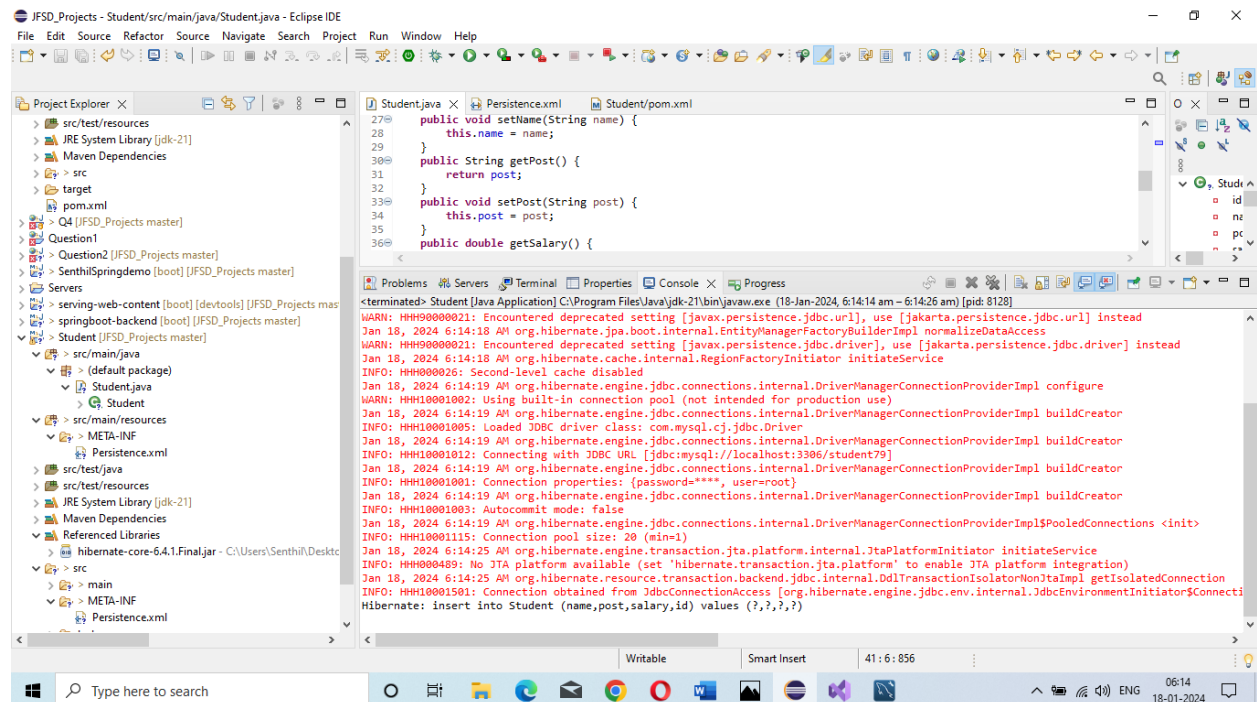
The screenshot shows the Eclipse IDE with the 'Persistence.xml' file open. The code defines a 'persistence' unit named 'student79' using 'org.hibernate.jpa.HibernatePersistenceProvider'. It includes properties for the database driver, URL, user, password, and hibernate settings. The 'Project Explorer' on the left shows the project structure, including 'src/main/resources', 'src/main/java', and 'META-INF'. The 'Console' at the bottom shows the application running successfully.

```
1 <?xml version="2.1" ?>
2 <persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
5     http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
6     version="2.1" ?>
7
8     <persistence-unit name="student79">
9         <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
10         <properties>
11             <property name="javax.persistence.jdbc.driver"
12                 value="com.mysql.cj.jdbc.Driver" />
13             <property name="javax.persistence.jdbc.url"
14                 value="jdbc:mysql://localhost:3306/student79" />
15             <property name="javax.persistence.jdbc.user" value="root" />
16             <property name="javax.persistence.jdbc.password"
17                 value="Rsenthil_1976" />
18             <property name="hibernate.show_sql" value="true" />
19             <property name="hibernate.hbm2ddl.auto" value="update" />
20         </properties>
21     </persistence-unit>
22 </persistence>
23
24
```

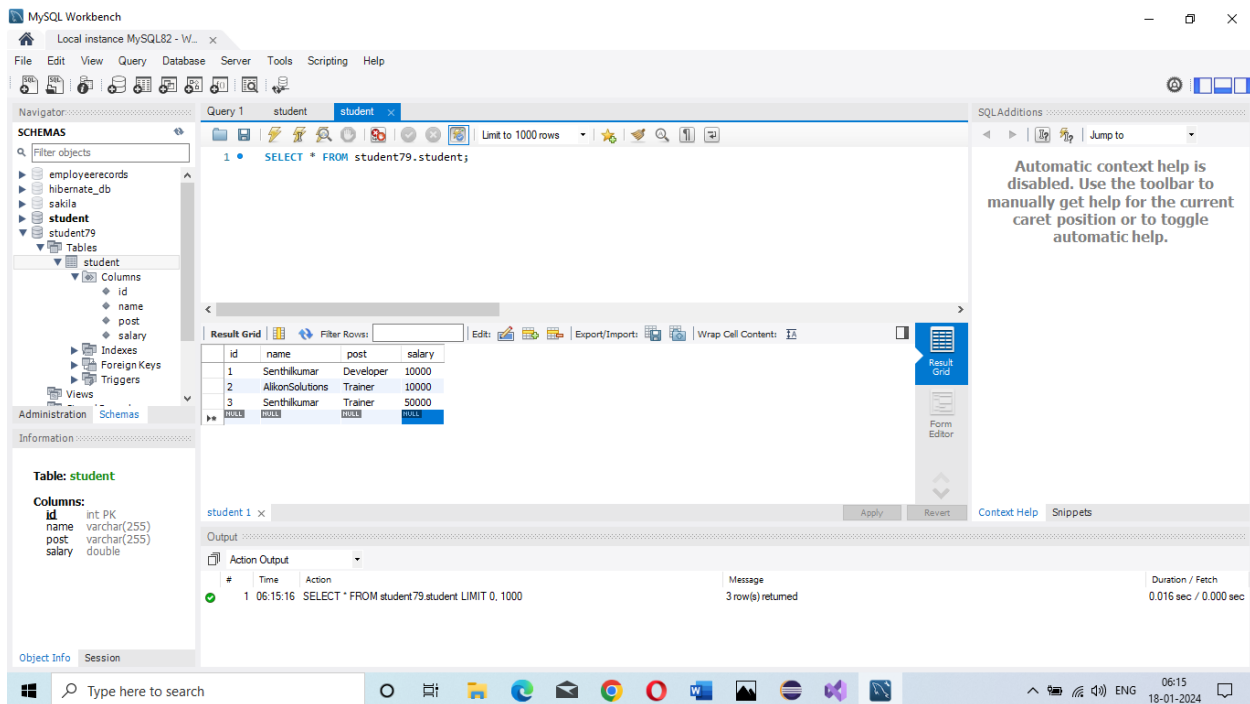
# Pom.xml



# Output



## MySQL Workbench: DB- student79



MySQL Workbench interface showing the 'student' table in the 'student79' database. The 'Result Grid' displays the output of a query: `SELECT * FROM student79.student;`

The table structure is as follows:

id	name	post	salary
1	Senthilkumar	Developer	10000
2	AlikonSolutions	Trainer	10000
3	Senthilkumar	Trainer	50000
NULL	NULL	NULL	NULL

The 'Information' panel shows the table structure:

Table: student

Columns:

- id: int PK
- name: varchar(255)
- post: varchar(255)
- salary: double

## Student79-Database

	id	name	post	salary
	1	Senthilkumar	Developer	10000
	2	AlikonSolutions	Trainer	10000
	3	Senthilkumar	Trainer	50000
	NULL	NULL	NULL	NULL