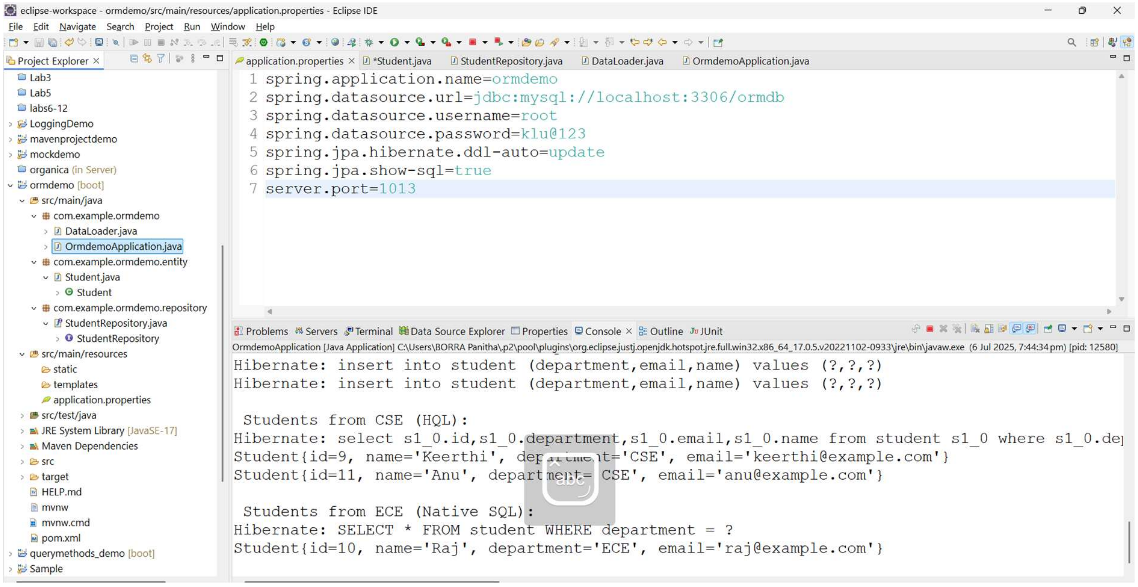
Demonstrate writing Hibernate Query Language and Native Query

```
package com.example.ormdemo.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "student")
public class Student {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Long id;

  private String name;
  private String department;

  @Column(unique = true)
  private String email;

  public Student() {}


  public Student(String name, String department, String email) {
    this.name = name;
    this.department = department;
    this.email = email;
  }
```

```java
public Long getId() {
   return id;
}

public String getName() {
   return name;
}

public String getDepartment() {
   return department;
}

public String getEmail() {
   return email;
}

public void setId(Long id) {
   this.id = id;
}

public void setName(String name) {
   this.name = name;
}

public void setDepartment(String department) {
   this.department = department;
```

```java
    }


    public void setEmail(String email) {
        this.email = email;
    }



    @Override
    public String toString() {
        return "Student{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", department='" + department + '\'' +
                ", email='" + email + '\'' +
                '}';
    }
}
package com.example.ormdemo.repository;

import com.example.ormdemo.entity.Student;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import java.util.List;

public interface StudentRepository extends JpaRepository<Student, Long> {



    @Query("SELECT s FROM Student s WHERE s.department = ?1")
    List<Student> findByDepartmentHQL(String department);



    @Query(value = "SELECT * FROM student WHERE department = ?1", nativeQuery = true)
    List<Student> findByDepartmentNative(String department);
}
package com.example.ormdemo;

import com.example.ormdemo.entity.Student;
import com.example.ormdemo.repository.StudentRepository;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

@Component
public class DataLoader implements CommandLineRunner {

    private final StudentRepository repo;

    public DataLoader(StudentRepository repo) {
        this.repo = repo;
    }

    @Override
    public void run(String… args) {

        repo.deleteAll();
        repo.save(new Student("Keerthi", "CSE", "keerthi@example.com"));

        repo.save(new Student("Raj", "ECE", "raj@example.com"));

        repo.save(new Student("Anu", "CSE", "anu@example.com"));



        System.out.println("\n Students from CSE (HQL):");

        repo.findByDepartmentHQL("CSE").forEach(System.out::println);



        System.out.println("\n Students from ECE (Native SQL):");

        repo.findByDepartmentNative("ECE").forEach(System.out::println);

    }

}
```

OUTPUT:



```
spring.application.name=ormdemo
spring.datasource.url=jdbc:mysql://localhost:3306/ormdb
spring.datasource.username=root
spring.datasource.password=klu@123
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
server.port=1013
```

OrmdemoApplication [Java Application] C:\Users\BORRA Panitha\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (6 Jul 2025, 7:44:34 pm) [pid: 12580]

```
Hibernate: insert into student (department,email,name) values (?,?,?)
Hibernate: insert into student (department,email,name) values (?,?,?)

 Students from CSE (HQL):
Hibernate: select s1_0.id,s1_0.department,s1_0.email,s1_0.name from student s1_0 where s1_0.de
Student{id=9, name='Keerthi', department='CSE', email='keerthi@example.com'}
Student{id=11, name='Anu', department='CSE', email='anu@example.com'}

 Students from ECE (Native SQL):
Hibernate: SELECT * FROM student WHERE department = ?
Student{id=10, name='Raj', department='ECE', email='raj@example.com'}
```